



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Leon Cesar Carmona

**Desenvolvimento e Implantação de Solução *IoT* em Equipamento Legado:
Máquina Injetora de Alumínio de Rotores**

Florianópolis
2022

Leon Cesar Carmona

**Desenvolvimento e Implantação de Solução *IoT* em Equipamento Legado:
Máquina Injetora de Alumínio de Rotores**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Carlos Barros Montez, Dr.

Supervisor: Fabricio Lux, Eng.

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Carmona, Leon Cesar
Desenvolvimento e Implantação de Solução IoT em
Equipamento Legado : Máquina Injetora de Alumínio de
Rotores / Leon Cesar Carmona ; orientador, Carlos Barros
Montez, coorientador, Fabricio Lux, 2022.
128 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia de Controle e Automação,
Florianópolis, 2022.

Inclui referências.

1. Engenharia de Controle e Automação. 2. IoT. 3.
Internet das Coisas. 4. Equipamento Legado. 5. Injeção de
Alumínio em Rotores. I. Barros Montez, Carlos. II. Lux,
Fabricio. III. Universidade Federal de Santa Catarina.
Graduação em Engenharia de Controle e Automação. IV. Título.

Leon Cesar Carmona

**Desenvolvimento e Implantação de Solução *IoT* em Equipamento Legado:
Máquina Injetora de Alumínio de Rotores**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 24 de Março de 2022.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Carlos Barros Montez, Dr.
Orientador
UFSC/CTC/DAS

Fabricio Lux, Eng.
Supervisor
WEG S.A.

Prof. Rômulo Silva de Oliveira, Dr.
Avaliador
UFSC/CTC/DAS

Prof. Eduardo Camponogara, Dr.
Presidente da Banca
UFSC/CTC/DAS

Este trabalho é dedicado aos meus colegas de classe e
aos meus queridos pais.

AGRADECIMENTOS

Dedico este espaço para agradecer a todas as pessoas que, de alguma forma, contribuíram para a realização deste trabalho.

*"The land of opportunity.
The golden chance for me.
My future looks so bright.
Now I think I've seen the light."
(Dave Ellefson, Dave Mustaine, 1990)*

RESUMO

Este documento tem como finalidade descrever o trabalho realizado como Projeto de Fim de Curso, entre os meses de Março a Dezembro de 2021 na empresa *WEG*. O trabalho consiste no desenvolvimento e implementação de um Sistema *IoT* para uma máquina Injetora de Alumínio de Rotores, considerada como um equipamento legado. O objetivo deste trabalho é promover a digitalização das variáveis do processo de Injeção de Alumínio de Rotores, por meio da utilização das tecnologias inerentes à Internet das Coisas, além da utilização de diversos conceitos relativos a redes industriais. No decorrer deste documento são detalhadas todas as etapas realizadas, desde a concepção da arquitetura do Sistema *IoT*, passando pela instrumentação e configuração da arquitetura, bem como a programação da lógica que irá gerenciar a aplicação, até finalmente na visualização dos dados. Também são descritas algumas funcionalidades desenvolvidas para o sistema, além de alguns *cases* em que a aplicação foi utilizada no decorrer do período em questão.

Palavras-chave: Internet das Coisas. IoT. Injeção de Alumínio em Rotores. Equipamento Legado.

ABSTRACT

This document has as defined the work carried out as the End of Course Project, between the months of March and December 2021 at the company WEG. The work consists in the development and implementation of an IoT System for an Aluminum Rotor Injection Machine, considered as a legacy equipment. The objective of this work is to scan the variables of the rotor aluminum injection process, through the use of technologies inherent to the Internet of Things, in addition to the use of various concepts related to industrial networks. Throughout the document, all steps of this procedure will take place, from the conception of the IoT System architecture, through the instrumentation and configuration of the architecture, as well as the programming of the logic that will manage an application, until finally in the data visualization. Some functions developed for the system will also be explained, as well as some cases where the application was applied during the period in question.

Keywords: Internet of Things. IoT. Aluminium Die Casting. Legacy Equipment.

LISTA DE FIGURAS

Figura 1 – Sequência de Etapas da Metodologia.	26
Figura 2 – <i>WEG</i>	27
Figura 3 – Arquitetura de Solução <i>IoT</i> com <i>WEGnology</i>	30
Figura 4 – Protocolos de Segurança do <i>WEGnology</i>	31
Figura 5 – Parte Inferior do Molde de Injeção de Alumínio em Rotores.	33
Figura 6 – Molde de Injeção de Alumínio em Rotores.	34
Figura 7 – Pacote de Lâminas com Pino de Alinhamento.	34
Figura 8 – Rotores Injetados.	35
Figura 9 – Massalotes.	35
Figura 10 – Dispositivo WCD-ED300.	36
Figura 11 – Arquitetura do Sistema <i>IoT</i>	37
Figura 12 – Arquitetura Ilustrada do Sistema <i>IoT</i>	38
Figura 13 – Tela Inicial IHM.	39
Figura 14 – <i>Toolbar</i> de Navegação do <i>Easybuilder</i>	39
Figura 15 – Tela Principal da IHM.	40
Figura 16 – Janela de Configuração do Endereço de Memória para Leitura da VP.	41
Figura 17 – Planilha de Mapeamento de VPs.	42
Figura 18 – Sensor de Vazão e Temperatura SV5204.	43
Figura 19 – Esquemático de Conexões do Sensor.	43
Figura 20 – Sensores de Vazão/Temperatura Instalados na Injetor.	44
Figura 21 – <i>PLC Allen-Bradley SLC500</i>	45
Figura 22 – Módulo de Expansão Analógico do <i>PLC</i>	45
Figura 23 – <i>Ladder</i> para Conversão dos Sinais dos Sensores.	46
Figura 24 – IHM <i>WEG MT8150XE</i>	48
Figura 25 – Inclusão de Nova Macro.	49
Figura 26 – Criação de Nova Macro.	49
Figura 27 – Adicionar Nova Macro.	49
Figura 28 – Ambiente de programação de Macro.	50
Figura 29 – Declaração de Variáveis na Macro.	51
Figura 30 – Declaração de Variáveis na Macro.	51
Figura 31 – Recuperação de Variáveis do CLP.	52
Figura 32 – Alocação de Variáveis na Memória da IHM.	53
Figura 33 – Planilha de Mapeamento de VPs.	54
Figura 34 – Frequência de Execução da Macro.	55
Figura 35 – Dispositivo WCD-ED300.	55
Figura 36 – Conectores Frontais do ED300.	56
Figura 37 – Conectores Laterais do ED300.	56

Figura 38 – Conectores Traseiros do ED300.	56
Figura 39 – Tela de <i>Login</i> da Interface do ED300.	58
Figura 40 – Página de <i>Status</i> do ED300.	58
Figura 41 – Painel de Configuração de <i>Wi-Fi</i>	59
Figura 42 – Botão de Salvamento de Configurações.	59
Figura 43 – Integração <i>Cloud</i>	60
Figura 44 – Gerando uma <i>API Token</i>	60
Figura 45 – Adicionando uma <i>API Token</i>	61
Figura 46 – Configuração da <i>API Token</i>	61
Figura 47 – <i>Download da API Token</i>	62
Figura 48 – Aba de <i>Access Key</i>	62
Figura 49 – Botão <i>Add Access Key</i>	63
Figura 50 – Configuração da <i>Access Key</i>	63
Figura 51 – Gerando a <i>Access Key</i>	64
Figura 52 – Lista de Dispositivos no <i>WEGnology</i>	64
Figura 53 – Tela Inicial da Interface da IHM.	65
Figura 54 – Criação de Servidor Modbus.	66
Figura 55 – Configuração do Servidor Modbus.	66
Figura 56 – Lista de <i>Devices</i> do Sistema.	67
Figura 57 – Arquitetura de Rede com Dispositivo Edge.	68
Figura 58 – Criação do <i>Edge Workflow</i>	68
Figura 59 – Parâmetros do novo <i>Workflow</i>	69
Figura 60 – Ambiente de Desenvolvimento do <i>Workflow</i>	69
Figura 61 – Bloco “ <i>Timer</i> ”.	70
Figura 62 – Bloco <i>Modbus: Read</i>	71
Figura 63 – Configuração do Bloco <i>Modbus: Read</i>	72
Figura 64 – Configuração do Bloco <i>Device: State</i>	73
Figura 65 – Configuração de Atributos do <i>Device</i>	73
Figura 66 – Atributos do <i>Device</i>	74
Figura 67 – Configuração de Atributos do <i>Device</i>	74
Figura 68 – Localização das VPs no <i>Payload</i>	75
Figura 69 – Localizando os Endereços dos Atributos no <i>Payload</i>	75
Figura 70 – Realizando o <i>Deploy</i> no <i>Edge Device</i>	76
Figura 71 – Exibindo o <i>Live Look View</i> do <i>Workflow</i>	77
Figura 72 – Endereço do Atributo no <i>Payload</i>	77
Figura 73 – Copiando o Endereço do Atributo no <i>Payload</i>	78
Figura 74 – Copiando o Endereço do Atributo no <i>Payload</i>	78
Figura 75 – Variável “ <i>TemperaturaForno</i> ”.	79
Figura 76 – Configuração do bloco <i>Math</i>	80

Figura 77 – Atualização do <i>Payload Path</i> da VP no bloco <i>Device: State</i>	80
Figura 78 – Atualização do <i>Payload Path</i> da VP no bloco <i>Device: State</i>	81
Figura 79 – <i>Payload</i> Completo do Processo.	82
Figura 80 – Inclusão de Novo <i>Dashboard</i> no <i>WEGnology</i>	83
Figura 81 – Inclusão de Novo <i>Dashboard</i> no <i>WEGnology</i>	84
Figura 82 – Configuração do Bloco de Vazão no <i>Dashboard</i>	84
Figura 83 – Seleção do Dispositivo do Bloco.	85
Figura 84 – Configuração do Estilo do Bloco.	85
Figura 85 – Visualização do Bloco no <i>Dashboard</i>	86
Figura 86 – Inclusão do Bloco de Temperatura no <i>Dashboard</i>	87
Figura 87 – Modelo Final do <i>Dashboard</i>	87
Figura 88 – <i>CLP CLIC-02</i> com Módulo de Expansão (ilustrativo)	89
Figura 89 – Seleção do modelo do <i>CLIC</i>	90
Figura 90 – Configuração <i>RS485</i>	90
Figura 91 – Conexão do <i>CLIC-02 Edit</i> com o <i>CLP</i>	91
Figura 92 – Seleção de Porta <i>COM</i>	91
Figura 93 – Módulo de Expansão Analógico do <i>CLIC-02</i>	92
Figura 94 – <i>Jumper</i> das Entradas de Corrente e Tensão do <i>CLIC-02</i>	92
Figura 95 – Configuração Serial do <i>Gateway ED300</i>	93
Figura 96 – Pares de Resistores para Conversão de Sinais de Corrente.	94
Figura 97 – Conversão de Sinais de Corrente para Sinais de Tensão.	95
Figura 98 – Sinais de Corrente Convertidos para Sinais de Tensão.	95
Figura 99 – Pinagem da Porta <i>RS485</i> do <i>Gateway ED300</i>	96
Figura 100–Inclusão do Bloco <i>Modbus</i> para Comunicação <i>Serial RTU</i>	97
Figura 101–Configuração do Bloco <i>Modbus</i> para Comunicação <i>Serial RTU</i>	97
Figura 102– <i>JavaScript</i> para Interpolação.	98
Figura 103– <i>JavaScript</i> para Interpolação.	99
Figura 104–Arquitetura do Sistema no CT-i13.	100
Figura 105–Arquitetura Ilustrada do Sistema - CT-i13 <i>IoT</i>	100
Figura 106–Criação de um <i>Slug</i> para a Aplicação <i>IoT</i>	102
Figura 107–Criação de um <i>Endpoint</i> para a Aplicação <i>IoT</i>	102
Figura 108–Funcionalidade do <i>Endpoint</i>	103
Figura 109– <i>Endpoint Method and Route</i>	103
Figura 110– <i>Endpoint Page</i>	104
Figura 111– <i>Endpoint Workflow</i>	104
Figura 112–Configuração de Acesso no <i>Endpoint</i>	105
Figura 113–Verificação das Configurações do <i>Endpoint</i>	105
Figura 114– <i>Endpoint</i> Criado.	106
Figura 115–Relacionando <i>Endpoint</i> com <i>Dashboard</i> do CT.	106

Figura 116–Criação da Interface de <i>Log-In</i> da Página <i>Web</i>	107
Figura 117–Página de <i>Log-In</i>	108
Figura 118–Criação da Interface da Página <i>Home</i> da Página <i>Web</i>	108
Figura 119–Página <i>Home</i>	109
Figura 120–Forno Dosador da Injetora de Alumínio.	110
Figura 121– <i>Workflow</i> do Sistema de Alerta.	111
Figura 122–Modelo de <i>SMS</i> do Alerta.	112
Figura 123–Sistema de <i>Reboot</i> Automático do <i>ED300</i>	113
Figura 124– <i>API</i> para Verificação de Feriados.	114
Figura 125–Função de Excedente de Alumínio no Massalote.	115
Figura 126–Função de Excedente de Alumínio no Massalote.	116
Figura 127– <i>Workflow</i> do Sistema <i>IoT</i> do CT-i07.	117
Figura 128–Etapas <i>Proof of Concept</i>	118
Figura 129–Queda de Temperatura do Forno Dosador.	120
Figura 130–Alertas de Temperatura do Forno Dosador.	120
Figura 131–Pistão.	121
Figura 132–Pistão (vista 2).	121
Figura 133–Câmara do Pistão.	122
Figura 134–Câmara do Pistão (vista 2).	122
Figura 135–Comportamento das Variáveis de Vazão e Temperatura.	123

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Program Interface</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
IHM	Interface Homem-Máquina
IIoT	<i>Industrial Internet of Things</i>
IoT	<i>Internet of Things</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
PoC	<i>Proof of Concept</i>
TLS	<i>Transport Layer Security</i>

SUMÁRIO

1	INTRODUÇÃO	17
1.1	INTRODUÇÃO À PROBLEMÁTICA	18
1.2	JUSTIFICATIVA	19
1.3	OBJETIVOS	19
1.4	RESULTADOS OBTIDOS	21
1.5	ESTRUTURA DO DOCUMENTO	21
2	METODOLOGIA	22
2.1	PESQUISA INICIAL SOBRE INDÚSTRIA 4.0 E <i>IOT</i>	22
2.2	INTRODUÇÃO À PLATAFORMA <i>IOT</i> DA WEG - <i>WEGNOLOGY</i>	22
2.3	IDENTIFICAÇÃO DAS VARIÁVEIS DE PROCESSO - INJEÇÃO DE ALUMÍNIO	22
2.4	SENSORIZAÇÃO DA MÁQUINA	23
2.5	PROGRAMAÇÃO DA IHM DA INJETORA DE ALUMÍNIO	23
2.6	CONEXÃO DO DISPOSITIVO <i>GATEWAY</i> COM A IHM DA INJETORA	23
2.7	DESENVOLVIMENTO DA LÓGICA DO <i>GATEWAY</i>	24
2.8	TRATAMENTO DAS VARIÁVEIS DE PROCESSO VIA <i>WEGNOLOGY</i>	24
2.9	CRIAÇÃO DE <i>DASHBOARD</i> PARA VISUALIZAÇÃO DOS DADOS	24
2.10	EXPANSÃO DO SISTEMA <i>IOT</i>	24
2.11	CRIAÇÃO DE <i>WEBSITE</i> PARA CENTRALIZAÇÃO DOS DADOS	25
2.12	FLUXOGRAMA DA METODOLOGIA	25
3	DESCRIÇÃO DA EMPRESA	27
4	FUNDAMENTAÇÃO TEÓRICA	29
4.1	INDÚSTRIA 4.0	29
4.2	<i>WEGNOLOGY</i>	30
4.2.1	<i>Introdução à plataforma de IoT da WEG</i>	30
4.2.2	<i>Visão Geral de Segurança do WEGnology</i>	31
4.3	O PROCESSO DE INJEÇÃO DE ALUMÍNIO DE ROTORES	32
5	DESENVOLVIMENTO	36
5.1	DA CONCEPÇÃO DA ARQUITETURA DO SISTEMA <i>IOT</i>	36
5.2	IDENTIFICAÇÃO DAS VARIÁVEIS DE PROCESSO - INJEÇÃO DE ALUMÍNIO	38
5.2.1	Mapeamento de Variáveis de Processo via <i>Easybuilder</i>	38
5.2.2	Variáveis de Temperatura e Vazão do Líquido Refrigerante	42
5.3	SENSORIZAÇÃO DA MÁQUINA	42
5.4	CONEXÃO DOS SINAIS DOS SENSORES COM O <i>CLP</i> DA INJETORA	44
5.4.1	Do <i>CLP</i> Integrado à Máquina Injetora de Alumínio	44

5.4.2	Conexão dos Sinais dos Sensores e Implementação da Lógica <i>Ladder</i>	45
5.5	PROGRAMAÇÃO DA IHM DA INJETORA DE ALUMÍNIO	47
5.5.1	Da “Ponte” com a IHM	47
5.5.2	Programação da Macro para Leitura e Gravação das Variáveis de Processo	47
5.5.2.1	Declaração de Variáveis	50
5.5.2.2	Recuperação de Variáveis	52
5.5.2.3	Escrita de Variáveis na Memória da IHM	53
5.5.2.4	Frequência de Execução da Macro	54
5.6	CONFIGURAÇÃO DO DISPOSITIVO <i>GATEWAY</i>	55
5.6.1	Acesso à Interface de Comunicação do ED300	56
5.6.2	Conexão do ED300 com uma rede <i>Wi-Fi</i>	59
5.6.3	Integração do ED300 com o <i>WEGnology</i>	60
5.7	CONEXÃO DO DISPOSITIVO <i>GATEWAY</i> COM A IHM DA INJETORA	65
5.8	DESENVOLVIMENTO DA LÓGICA DO <i>GATEWAY</i>	67
5.9	TRATAMENTO DAS VARIÁVEIS DE PROCESSO VIA <i>WEGNOLOGY</i>	79
5.10	CRIAÇÃO DE <i>DASHBOARD</i> PARA VISUALIZAÇÃO DOS DADOS .	82
5.10.1	Básico sobre <i>Dashboards</i> no <i>WEGnology</i>	82
5.10.2	Criação do <i>Dashboard</i> para o CT-i07	83
5.11	EXPANSÃO DO SISTEMA <i>IOT</i> PARA O CT-I13	88
5.11.1	Das Peculiaridades do Novo Centro de Trabalho - CT-i13	88
5.11.2	Da Inclusão do <i>CLP CLIC-02</i> à Arquitetura do Sistema	89
5.11.2.1	Da Configuração do <i>CLP CLIC-02</i> com Módulo de Expansão Analógico	89
5.11.2.2	Da Configuração do <i>Gateway ED300</i> para Comunicação Serial <i>RS485</i>	93
5.11.2.3	Da Conversão dos Sinais de Corrente para Tensão no Módulo Principal do <i>CLP CLIC-02</i>	94
5.11.2.4	Da Conexão <i>CLIC-02</i> com o Dispositivo <i>Gateway ED300</i>	96
5.11.2.5	Adaptação do <i>Workflow</i> via <i>WEGnology</i> para Leitura <i>Modbus RTU</i> .	96
5.11.2.6	Interpolação das Variáveis dos Sensores via Bloco <i>Function</i>	98
5.11.2.7	Arquitetura do Sistema <i>IoT</i> para o CT-i13	100
5.12	CRIAÇÃO DE <i>WEBSITE</i> PARA CENTRALIZAÇÃO DOS DADOS . .	101
5.12.1	Definição de <i>Experience</i> no <i>WEGnology</i>	101
5.12.2	Da Criação de um <i>Slug</i> para a <i>Aplicação IoT</i>	101
5.12.3	Da Criação dos <i>Endpoints</i>	102
5.12.4	Da Criação do <i>Website</i>	107
5.13	NOVAS FUNCIONALIDADES E INDICADORES DO SISTEMA <i>IOT</i> .	109
5.13.1	Implementação de Sistema de Alertas para os Fornos Dosadores das Injetoras de Alumínio	109

5.13.2	Sistema de <i>Reboot</i> Automático para os dispositivos <i>ED300</i> . . .	112
5.13.3	API para Identificação de Feriados	113
5.13.4	Criação de Indicador de Excedente de Alumínio no Massalote . .	114
6	ANÁLISE DE RESULTADOS	118
6.1	<i>PROOF OF CONCEPT (POC)</i>	118
6.2	OBJETIVOS PREVISTOS E RESULTADOS OBTIDOS	118
6.3	CASES DE APLICAÇÃO DO SISTEMA <i>IOT</i>	119
6.3.1	Forno Dosador - Falha na Manutenção da Temperatura do Alumí- nio	119
6.3.2	Engripamento de Pistão da Injetora de Alumínio	121
7	CONCLUSÃO	125
	REFERÊNCIAS	126

1 INTRODUÇÃO

De acordo com um estudo publicado pelo Fórum Econômico Mundial, em parceria com a empresa de consultoria industrial americana *A.T. Kearney*, denominado *“Readiness for the Future of Production Report 2018”*, em termos de perspectivas futuras relacionadas à evolução dos sistemas de produção, o Brasil figura em um grupo intitulado de *“Nascent Countries”*, ou “Países Nascentes”. Este grupo é caracterizado pelo baixo nível de preparação para o futuro da produção, decorrente do baixo desempenho nos dois principais componentes da avaliação realizada no estudo, sendo eles: *“Structure of Production”*, ou “Estrutura da Produção”, que tem relação com a atual linha de base de produção do país, e *“Drivers of Production”*, ou “Motores da Produção”, que está relacionado a alguns elementos habilitadores que propiciam que o país consiga capitalizar a 4ª Revolução Industrial e transformar seus sistemas de produção (WEF, 2018).

Num estudo anterior, também realizado em conjunto pelo Fórum Econômico Mundial com a *A.T. Kearney*, chamado *“Shaping the Future of Production: Four Contrasting Perspectives in 2030”* (WEF, 2017), são propostos quatro cenários para o futuro da produção até o ano de 2030, sendo eles:

- Disruptivo - em que as tecnologias habilitadoras da Indústria 4.0 finalmente eclodiram a um nível completamente revolucionário, convergindo para uma nova modelagem do mundo em seus aspectos mais fundamentais;
- Dissuadido - onde um grande ciberataque compromete grandes segmentos de infraestrutura crítica e interrompe processos de produção, gerando um significativo aumento na incerteza com relação à vulnerabilidade de sistemas ao redor do mundo, causando uma redução na produção;
- Danificado, sugere que questões como populismo e protecionismo resultariam em uma significativa desigualdade no cenário produtivo global, e que o distanciamento das economias e sistemas de produção promoveriam uma nova depressão econômica mundial;
- Deteriorado, uma mudança repentina e generalizada por parte de governos de economias desenvolvidas e em desenvolvimento, no sentido de priorizar questões ambientais devido ao alto grau de deterioração do meio ambiente e condições climáticas.

Embora este estudo não tenha por objetivo advogar ou opinar sobre a probabilidade ou realidade de cada um dos cenários propostos, é fato de que as partes mais interessadas, como governos, empresas e a sociedade civil em geral devem, em seus respectivos escopos de atuação, se preparar para as possibilidades que estão por vir

em termos de evolução tecnológica, bem como suas implicações no diversos âmbitos relacionados ao tema.

Com este documento sendo escrito no pandêmico ano de 2021, temos nove anos pela frente para identificar e compreender o que será, de fato, do futuro da indústria e como isso afetará a sociedade em que viveremos no ano de 2030.

1.1 INTRODUÇÃO À PROBLEMÁTICA

Num primeiro momento, a expressão “equipamento legado” pode gerar um sentimento de algo que já está em estágio avançado em termos de obsolescência ou até mesmo de completo desuso. No entanto, este entendimento não faz jus à realidade da grande maioria das indústrias, mesmo falando em níveis globais.

Ainda que estejamos em tempos onde muito se fala em “Indústria 4.0” e suas tecnologias habilitadoras, como *Internet of Things (IoT)*, *Big Data*, Sistemas Ciberfísicos, entre outras, um ponto muito importante que não recebe tanto destaque, é o fato de que os fabricantes já possuem a maior parte dos componentes e equipamentos que deverão impulsionar o estado atual da indústria para um futuro tecnológico. Os dados que já estão no chão de fábrica e que são gerados pelos então chamados de “equipamento legado” são, de fato, um dos ativos mais importantes do processo produtivo, uma vez que representam a propriedade intelectual relacionada aos métodos de produção de seus produtos e serviços.

Como publicado na Forbes em 2019, Sundblad argumenta que obter os dados não é apenas sobre transformar o fabricante em digital, mas sim sobre como o fabricante pode resolver seus maiores problemas. Nesta mesma linha, alguns fatores contribuem para o aumento da complexidade em uma linha de produção, como máquinas de diferentes fabricantes e protocolos de comunicação, e que acabam por criar problemas como visibilidade operacional defasada, manutenção reativa, paradas de máquina não planejadas, falta de flexibilidade e um longo “*time-to-value*”, ou o tempo até que se reconheça o valor de algo (SUNDBLAD, 2018).

O maior desafio, entretanto, tem relação com o propósito intrínseco dos chamados “equipamentos legado”, que é o de executar uma função específica, e não a de captar e analisar os dados que são gerados durante sua operação. Sundblad coloca que, devido a isso, gasta-se algo em torno de 30% à 70% do tempo apenas procurando informações, muitas vezes provenientes de diferentes sistemas e de diferentes fabricantes. Desta forma, profissionais de alto valor, como engenheiros, trabalham em atividades de baixo valor agregado, fazendo com que o solucionamento de problemas, atividade de alto valor, acabe ficando em segundo plano.

O acesso aos dados provenientes de equipamentos legado podem contribuir para solucionar este problema, como indica um estudo realizado pela *McKinsey*, empresa de consultoria empresarial americana, que revela as principais oportunidades

relacionadas à implementação do conceito de Indústria 4.0 e que poderão ser capitalizadas pelas empresas (MCKINSEY GLOBAL INSTITUTE, 2015), conforme a lista resumida a seguir:

- 3-5% no aumento de produtividade;
- 30-50% de redução no tempo de máquina parada;
- 20-50% de redução no custo de manutenção de estoque;
- 85% de aumento na acurácia de previsão;
- 5-20% de redução no “*time-to-market*”;
- 10-40% de redução nos custos de manutenção.

De maneira geral, ter acesso a dados provenientes de equipamentos legado proporciona que fabricantes possam resolver problemas existentes de forma mais rápida, além de problemas que, de outra forma, não seriam possíveis de serem resolvidos.

1.2 JUSTIFICATIVA

A *WEG*, apesar de ser destaque em níveis nacionais e internacionais em termos de tecnologia, é uma empresa que iniciou suas operações na década de 1960 e, conseqüentemente, adquiriu boa parte de seu maquinário produtivo atualmente em atividade ao longo de sua história. Isso faz com que exista uma ampla variedade de equipamentos de diferentes épocas e fabricantes, que não foram projetados para se comunicar uns com os outros, muito menos via *internet*, mas sim para executarem funções bastante específicas.

Desta forma, estes equipamentos podem ser considerados como equipamentos legado e, portanto, alvos potenciais de um projeto de digitalização, trazendo o conceito de *IoT*, ou melhor, de *Industrial Internet of Things (IIoT)*, para o seu chão de fábrica e, conseqüentemente, obtendo os retornos relativos à implementação desta tecnologia.

A justificativa citada acima se aplica para diversas unidades fabris da *WEG* e seus respectivos equipamentos, sendo que neste trabalho, a fábrica em questão é a de Injeção de Alumínio, mais especificamente, as máquinas injetoras de alumínio em rotores, que serão o objeto alvo do trabalho realizado neste documento.

1.3 OBJETIVOS

O objetivo geral deste trabalho é desenvolver e implementar uma solução *IoT* para algumas máquinas injetoras de alumínio da fábrica de Injeção de Alumínio, com

a finalidade de trazer vida aos dados gerados por estes equipamentos legado, possibilitando então que estes dados sejam posteriormente tratados e analisados com o intuito de contribuir para a geração de valor e tomada de decisão no processo. Além de auxiliar na identificação de problemas e contribuir para a prevenção de problemas em potencial.

Para que fosse possível a concepção deste objetivo geral, foi necessária a definição e execução de diversos objetivos específicos que estão listados de forma resumida a seguir:

- Identificação das variáveis de processo;
- Sensorização da máquina;
- Programação da IHM da máquina;
- Conexão de dispositivo *gateway* com a IHM da injetora;
- Desenvolvimento da lógica do *gateway*;
- Tratamento das variáveis de processo via WEGnology;
- Criação de *dashboard* para visualização dos dados;
- Expansão do sistema *IoT*;
- Criação de *website* para centralização dos dados.

Nesta etapa é importante destacar que, inicialmente, o planejamento era para desenvolver e implementar a solução *IoT* em apenas uma máquina da fábrica de Injeção de Alumínio, servindo como aprendizado e eventualmente piloto para as demais máquinas, em caso de validação do projeto.

Devido ao fato de que o projeto se mostrou bastante válido e com diversos ganhos potenciais para o processo, foi realizada a expansão do trabalho para outra máquina da mesma fábrica, com algumas modificações na arquitetura do sistema devido a certas peculiaridades dos equipamentos envolvidos. Esta expansão do projeto será devidamente detalhada no decorrer do capítulo 5 (desenvolvimento).

Além desta primeira expansão, até o momento da escrita deste trabalho, foi aprovado um orçamento para a expansão do sistema *IoT* para mais 4 centros de trabalho na fábrica de Injeção de Alumínio, desta forma, totalizando 6 máquinas que já receberam ou que ainda irão receber a implementação do projeto, reforçando, mais uma vez, a percepção de que o sistema foi validado e que já está trazendo retorno significativo ao processo em questão.

1.4 RESULTADOS OBTIDOS

Nesta seção convém indicar que o projeto atingiu, de forma bastante satisfatória, por meio da execução dos objetivos específicos, o objetivo geral, ou seja, o desenvolvimento e implementação de uma solução *IoT* que promovesse a digitalização das variáveis pertinentes ao processo de Injeção de Alumínio em Rotores, além da concepção de indicadores de processo, passíveis de serem visualizados via *website* desenvolvido com o intuito de centralizar o acesso à informação.

De forma resumida, os principais resultados obtidos foram:

- Desenvolvimento e implementação da solução *IoT* nas máquinas injetoras de alumínio;
- Criação de *website* para acesso às informações de processo;
- *Know-how* sobre implementação de solução *IoT* em equipamentos legado.

1.5 ESTRUTURA DO DOCUMENTO

Esta seção tem como finalidade descrever a forma que este trabalho está estruturado, ou seja, sobre como foi feito o encadeamento dos próximos capítulos, a fim de facilitar ao leitor a navegação pelo documento.

No Capítulo 2 é descrita a metodologia de trabalho, ou seja, os procedimentos e ferramentas adotadas em cada fase do projeto para o atingimento dos objetivos especificados na Seção 1.3.

O Capítulo 3 é dedicado à uma breve descrição da empresa.

No Capítulo 4 é apresentada uma fundamentação teórica com os principais temas que servirão para um melhor entendimento do trabalho realizado.

No Capítulo 5 é realizada toda a descrição do desenvolvimento do trabalho, passando por cada etapa do processo.

No Capítulo 6 é feita uma análise de resultados obtidos, além da descrição de alguns *cases* de uso do Sistema *IoT* desenvolvido.

No Capítulo 7 é feita a conclusão do trabalho. Ela compreende um resumo do que foi feito, a solução proposta e os principais resultados atingidos.

2 METODOLOGIA

Este capítulo tem como finalidade apresentar a metodologia aplicada na realização deste trabalho. Irá abordar, ainda que de forma resumida, como foi o planejamento e execução de cada uma das etapas do projeto até que se atingisse a realização dos objetivos finais.

Algumas etapas consideradas mais rápidas foram omitidas neste capítulo por fins de organização, contudo, serão devidamente detalhadas no decorrer do capítulo 5, de desenvolvimento, bem como as etapas descritas a seguir.

2.1 PESQUISA INICIAL SOBRE INDÚSTRIA 4.0 E IOT

Num primeiro momento, após o repasse do que seria escopo geral do projeto pela chefia da seção, foi realizado um estudo inicial relacionado à temática da Indústria 4.0 e, mais especificamente, sobre Internet das Coisas, ou *IoT*. Os materiais utilizados neste primeiro contato com o tema foram provenientes de um acervo digital interno que contém diversos artigos, revistas, documentos, entre outros, em sua maioria, elaborados pela Academia Nacional de Ciências e Engenharia alemã, conhecida como Acatech, além de diversos outros materiais encontrados via pesquisa na *internet*.

2.2 INTRODUÇÃO À PLATAFORMA IOT DA WEG - WEGNOLOGY

Num segundo momento, após o primeiro contato com a literatura relacionada à Indústria 4.0, o trabalho consistiu na introdução à plataforma de *IoT* da WEG, conhecida como *WEGnology*. A ferramenta possui um tutorial inicial para novos usuários poderem se orientar e dar primeiros passos dentro da plataforma, por meio do desenvolvimento de um sistema que busca e exibe informações referentes ao clima, como temperatura, umidade, pressão, entre outras, de uma determinada localização, como uma cidade, por exemplo. Maiores informações a respeito da plataforma de *IoT* da WEG serão apresentadas nos capítulos 4 e 5, de fundamentação e desenvolvimento, respectivamente.

2.3 IDENTIFICAÇÃO DAS VARIÁVEIS DE PROCESSO - INJEÇÃO DE ALUMÍNIO

Nesta etapa do desenvolvimento, é realizado um levantamento de algumas variáveis relacionadas ao processo de Injeção de Alumínio em Rotores, por meio na análise do *software* presente na IHM da máquina. A IHM da injetora possui comunicação com o *CLP* da mesma, tendo acesso às variáveis armazenadas em sua memória. Estas variáveis servirão para validar a comunicação que será desenvolvida no decorrer do projeto, uma vez que será possível comparar os valores recebidos pelo *WEGnology* com os valores mostrados na IHM da máquina.

É importante destacar que, no momento do início da realização deste projeto, parte deste levantamento de variáveis de processo relativo à Injeção de Alumínio em Rotores já havia sido feito anteriormente por outros colaboradores que trabalhavam em outro projeto, fazendo com que fossem aproveitadas as informações já obtidas e complementadas com outras variáveis provenientes da minha própria pesquisa no *software* da IHM da injetora. Maiores detalhes a respeito da lista de variáveis utilizadas neste projeto serão apresentados no decorrer do Capítulo 5, de desenvolvimento.

2.4 SENSORIZAÇÃO DA MÁQUINA

A fim de se obter informações relativas à temperatura e vazão do líquido refrigerante da injetora, foram instalados 4 sensores para tais medições nas mangueiras responsáveis por direcionar o líquido para o pistão, matriz, câmara e placa.

Cabe ressaltar que a instalação em si dos sensores ocorreu antes da concepção e início deste trabalho, com a finalidade de possibilitar o monitoramento local destas variáveis de processo, uma vez que os sensores possuem um *display* que exibe os valores medidos, além da visualização por meio de uma tela que foi implementada no *software* daquela IHM em especial. Estes dados, apesar de serem visualizáveis, não eram armazenados em nenhum lugar, e acabavam se perdendo a cada nova leitura (que ocorria praticamente em tempo real), impossibilitando uma análise no tempo.

Com a realização deste trabalho, foi possível aproveitar a existência prévia destes sensores e realizar a conexão dos mesmos com o CLP da Injetora em questão (CT-i07), possibilitando, conseqüentemente, seu aproveitamento como variáveis de processo no projeto que viria a ser desenvolvido naquele centro de trabalho.

2.5 PROGRAMAÇÃO DA IHM DA INJETORA DE ALUMÍNIO

Nesta etapa do trabalho é realizado o desenvolvimento de um código (macro) que será implementado ao programa da Interface Homem-Máquina (IHM) da Injetora de Alumínio e que, por sua vez, será responsável por solicitar ao CLP as informações referentes às variáveis de processo em suas respectivas memórias, numa determinada taxa de atualização, para que então estes dados sejam salvos em endereços de memória da própria IHM. Estes dados armazenados na IHM da máquina serão, posteriormente, solicitados pelo dispositivo *gateway* que fará parte da arquitetura do sistema.

2.6 CONEXÃO DO DISPOSITIVO GATEWAY COM A IHM DA INJETORA

O dispositivo *gateway* incorporado à arquitetura do sistema *IoT* exercerá, a grosso modo, a função de solicitar à IHM da máquina as informações relativas às

variáveis de processo que estão armazenadas em sua memória e, na sequência, enviá-las à nuvem, onde ficarão acessíveis através da plataforma *WEGnology*.

Para que o *gateway* pudesse se comunicar com a IHM da Injetora, foi necessário realizar a configuração de um novo servidor Modbus TCP/IP no programa da IHM, configuração esta feita através do *software* de criação de interfaces para IHMs *EasyBuilder* e, posteriormente realizado o *download* do novo programa para a unidade da máquina.

2.7 DESENVOLVIMENTO DA LÓGICA DO GATEWAY

O *software* presente no dispositivo *gateway* é desenvolvido através da plataforma *WEGnology* e, uma vez implementado no aparelho, o mesmo é executado localmente, sem a necessidade de conexão com a nuvem.

2.8 TRATAMENTO DAS VARIÁVEIS DE PROCESSO VIA WEGNOLOGY

Ainda dentro do ambiente de desenvolvimento da plataforma *IoT* da WEG, *WEGnology*, será feito o tratamento de todos os dados provenientes do processo de Injeção de Alumínio, que chegam brutos, por meio de lógica desenvolvida e incorporada à lógica do dispositivo *gateway*.

Aqui cabe destacar que o tratamento dos dados não necessariamente precisa ser realizado de forma local, ou seja, já na lógica presente no dispositivo *gateway*, podendo ser realizado apenas na nuvem para posterior utilização. No entanto, neste trabalho, toda a lógica responsável por tratar as variáveis de processo foi implementada juntamente à lógica do dispositivo, sendo que algumas outras funcionalidades foram desenvolvidas e implementadas apenas na nuvem.

2.9 CRIAÇÃO DE DASHBOARD PARA VISUALIZAÇÃO DOS DADOS

Uma vez que todos os dados referentes às variáveis de processo foram devidamente coletados e tratados dentro da plataforma *WEGnology*, se fez necessário exibi-los por meio de um *dashboard*, a fim de proporcionar uma visualização mais simplificada e intuitiva ao usuário final.

O desenvolvimento deste *dashboard* também foi realizado dentro do *WEGnology*, onde existe a possibilidade de criação de modelos com base em componentes de visualização já disponíveis, apenas pendentes de configuração e customização.

2.10 EXPANSÃO DO SISTEMA IOT

Graças à boa aceitação do projeto no primeiro centro de trabalho (CT-i07), foi feita a expansão do sistema *IoT* para o CT-i13, devido a ausência de entradas

analógicas livres no CLP da Injetora de Alumínio, houve a necessidade de se pensar em uma alternativa para que os sinais dos sensores de temperatura e vazão pudessem ser lidos. Aqui é onde o CLP da WEG, modelo CLIC-02, é incorporado à arquitetura do sistema neste caso específico, possibilitando então a leitura dos sinais provenientes dos sensores.

Devido a esta alteração na arquitetura do sistema *IoT* para a captação de sinais, houve a necessidade de implementar uma lógica adicional responsável por efetuar a leitura e o tratamento destes dados provenientes do CLIC-02, para que então as informações dos sensores pudessem ser exibidas juntamente com o restante dos dados provenientes da IHM da máquina.

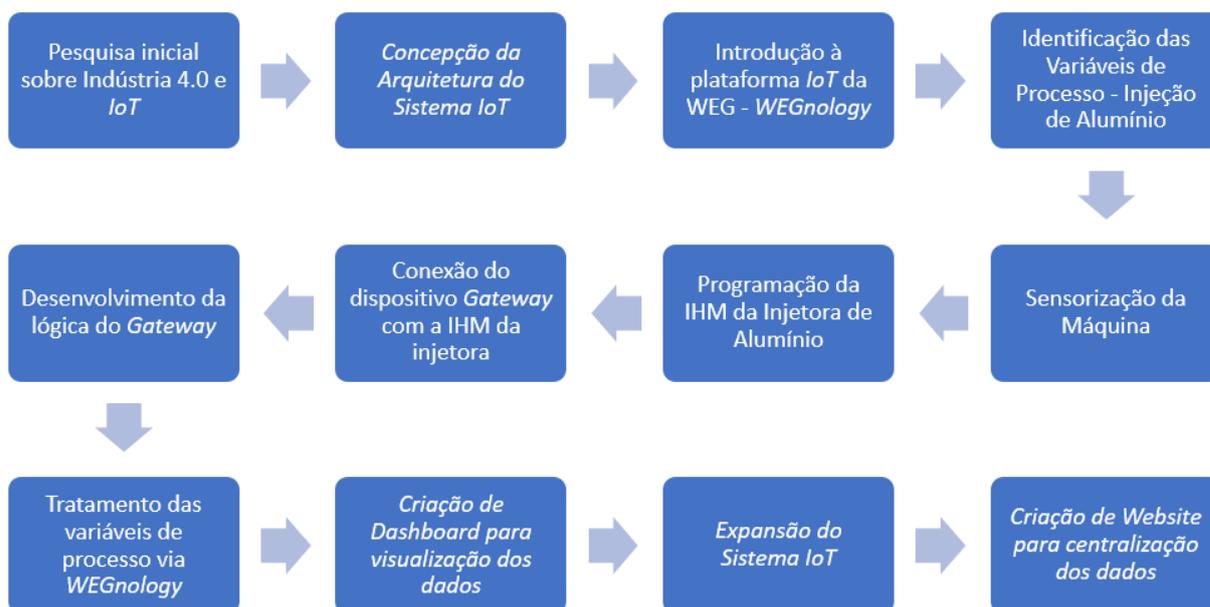
2.11 CRIAÇÃO DE *WEBSITE* PARA CENTRALIZAÇÃO DOS DADOS

Como já citado no capítulo introdutório, o sistema de *IoT* foi expandido para outro centro de trabalho (CT-i13), desta forma, foi identificada a necessidade de se criar um ambiente onde fosse possível de se acessar os *Dashboards* de todas as máquinas em que o projeto fosse implementado. Assim, a solução encontrada foi a criação de uma página *web* para que então o usuário pudesse ter à sua disposição as informações de todos os centros de trabalho.

2.12 FLUXOGRAMA DA METODOLOGIA

A Figura 1 a seguir, representa a sequência das etapas, a fim de facilitar o entendimento do ordem dos fatos em que se deu a realização deste projeto:

Figura 1 – Sequência de Etapas da Metodologia.



Fonte – Elaborado pelo autor (2021).

3 DESCRIÇÃO DA EMPRESA

A *WEG* é uma empresa multinacional brasileira que atua no setor de bens de capital nos mercados relacionados a geração, distribuição, transmissão, conversão de energia e automação. A empresa organiza o seu portfólio em cinco unidades principais: Motores, Energia, Transmissão e Distribuição, Automação e Tintas e Vernizes. Com fábricas em 12 países, filiais em 36, a empresa tem mais de 33,3 mil colaboradores (3600 engenheiros) e alcança mais de 1 milhão de m² construídos. A companhia foi fundada em 1961 por Werner Ricardo Voigt, Eggon João da Silva e Geraldo Werninghaus na cidade Jaraguá do Sul, no estado de Santa Catarina.

Figura 2 – *WEG*.



Fonte – *Website WEG* (2021).

Com uma ampla gama de produtos e serviços consolidados, a empresa desenvolve soluções para os seguintes setores de mercado: eficiência energética, energia solar, geração com resíduos sólidos, mineração, açúcar e etanol, construção civil, siderurgia, papel e celulose, óleo e gás, naval, mobilidade elétrica, entre outros.

Com um portfólio de mais de 1200 linhas de produtos e produzindo mais de 16 milhões de motores anualmente, a *WEG* é considerada um dos maiores fabricantes de equipamentos eletroeletrônicos do mundo. Possuindo as estratégias de diversificação e verticalização, e com qualidade reconhecida mundialmente, no ano de 2020 a empresa alcançou o faturamento líquido de R\$17,47 bilhões de reais (44% proveniente de vendas no Brasil), com um lucro líquido de R\$ 2,34 bilhões, valor aproximadamente 45% maior do que o lucro obtido no ano de 2019.

Com esses resultados, a empresa conquistou o 1º lugar da premiação “As Melhores Empresas da Bolsa” na categoria Bens Industriais, criado pela *InfoMoney* em parceria com a *Economática* e o *Ibmec*. A companhia foi eleita pelo sexto ano consecutivo como a 8ª empresa mais inovadora do Brasil pelo Prêmio “Inovação Brasil” concedido pelo Valor Econômico às empresas mais inovadoras do país.

Com uma cultura baseada nos seus fundadores e em toda a bagagem de conhecimentos adquiridos até o momento, a *WEG* possui como missão ter um crescimento contínuo e sustentável, mantendo a simplicidade. Como visão a empresa almeja

ser referência global em máquinas elétricas, com ampla linha de produtos, provendo soluções eficientes e completas. Para isso, a empresa possui os seguintes valores: Companhia Humana, Trabalho em Equipe, Eficiência, Flexibilidade, Inovação e Liderança.

4 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como finalidade proporcionar embasamento teórico, relacionado a processos, conceitos e tecnologias que foram necessárias e fizeram parte do desenvolvimento do projeto, contribuindo para o melhor entendimento de todas as etapas envolvidas e que serão detalhadas no capítulo subsequente, de desenvolvimento.

4.1 INDÚSTRIA 4.0

A evolução tecnológica promovida principalmente pela necessidade de atender às crescentes demandas industriais culminaram na 4ª Revolução Industrial e, em 2011, o termo “Indústria 4.0” foi utilizado na Alemanha para descrever a integração generalizada da informação e tecnologias de comunicação dentro de um contexto da produção industrial (SCHUH *et al.*, 2020), e nasce de um esforço por parte do governo alemão de implementar um novo conceito de política baseada em estratégias envolvendo o uso de alta tecnologia (ROBLEK; MEŠKO; KRAPEŽ, 2016).

O objetivo da Indústria 4.0 é promover a integração do ambiente de manufatura através do emprego das chamadas “Tecnologias Habilitadoras”, como Computação em Nuvem (*Cloud Computing*), Internet das Coisas (*IoT*), Sistemas Ciber-Físicos (*Cyber-physical Systems*) e *Big Data Analytics* (LEE; BAGHERI; KAO, 2015), com o intuito de estabelecer a comunicação permanente via *internet* entre máquinas e potencializar a competitividade (COOPER; JAMES, 2009). Desta forma, a Indústria 4.0 visa transformar as cadeias de produção industriais e modelos de negócio associados com o emprego de sistemas inteligentes de produção.

De acordo com a literatura, existem diversas tecnologias que são consideradas como habilitadoras da Indústria 4.0, sendo que as principais (MARTINELLI; MINA; MOGGI, 2019) são citadas a seguir, juntamente com uma breve descrição de cada uma:

- *Internet das Coisas (IoT)* - Uma rede de dispositivos físicos com tecnologia embarcada capazes de se comunicar ou interagir entre si ou com o ambiente externo (WEF, 2015);
- *Big Data* - Permite a coleta e ampla avaliação de dados provenientes de diversas fontes distintas, com o intuito de auxiliar no processo de tomada de decisão em tempo real (BAHRIN *et al.*, 2016);
- *Computação em Nuvem* - Consiste em alocar a capacidade de armazenamento e processamento de um computador dentro de um ambiente na *internet* afim de proporcionar serviços computacionais aos usuários (ZHANG *et al.*, 2014);

- *Sistemas Ciber Físicos* - Objetos físicos e diferentes plataformas de *software* são integrados, permitindo a interação entre os componentes de uma determinada arquitetura, além da troca de informações relevantes entre eles (KUMAR; ZINDANI; DAVIM, 2019);
- *Inteligência Artificial* - Habilidade de uma máquina de performar funções cognitivas, normalmente associadas à natureza humana (BUGHIN *et al.*, 2017);
- *Manufatura Aditiva* - Também conhecida por Impressão 3D, fabrica componentes e peças por camadas a partir de um arquivo digital (CAMACHO *et al.*, 2018). Pode ser usada para fabricar um produto com alto grau de customização de forma rápida e barata (HALEEM; JAVAID, 2019).

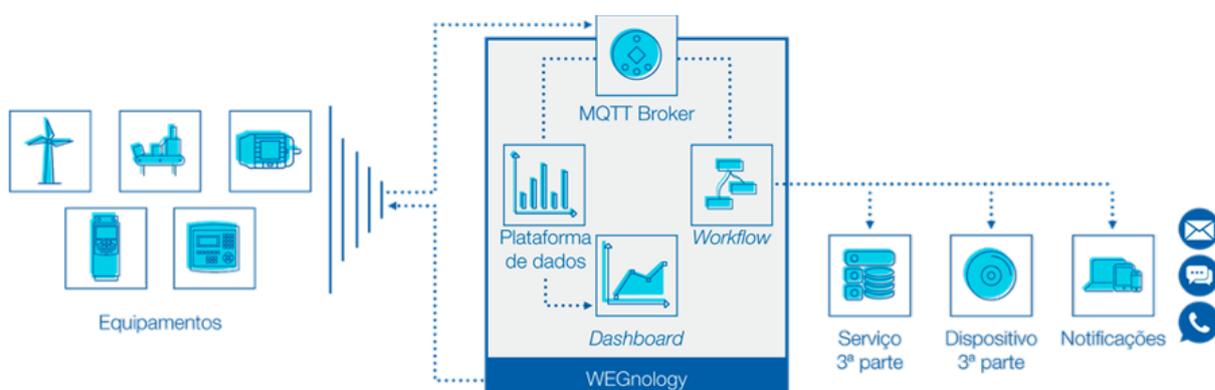
4.2 WEGNOLOGY

4.2.1 Introdução à plataforma de IoT da WEG

O *WEGnology* é a plataforma de *IoT* da WEG baseada em *cloud computing*, que possibilita ao usuário o desenvolvimento de aplicações que podem ser escaladas a diversos dispositivos, proporcionando a criação soluções conectadas. Nela, é possível coletar, agregar e visualizar dados de uma forma intuitiva, possibilitando o entendimento de grandes quantidades de informação provenientes do chão de fábrica, além de outras fontes não industriais.

A Figura 3 ilustra a posição do *WEGnology* dentro de uma arquitetura generalista de aplicação *IoT*.

Figura 3 – Arquitetura de Solução *IoT* com *WEGnology*.



Fonte – Documentação do *WEGnology* (2021).

A programação das aplicações dispõe de funcionalidade “*drag and drop*”, que basicamente se refere à programação por meio de blocos, que simplifica o desenvolvimento da lógica e agiliza eventuais alterações que venham a ocorrer com o tempo.

Ainda é possível o desenvolvimento de *dashboards* para a visualização dos dados extraídos do chão de fábrica, auxiliando no processo de identificação de falhas e tomada de decisão.

No decorrer do Capítulo 5, de desenvolvimento, serão abordadas algumas funcionalidades da plataforma em maior nível de detalhe.

4.2.2 Visão Geral de Segurança do WEGnology

Do ponto de vista da segurança do Sistema *IoT*, o *WEGnology* utiliza mecanismos de encriptação de padrão industrial para proteger os dados durante o trânsito e em repouso. A Figura 4 apresenta um panorama geral dos protocolos de segurança adotados por uma aplicação *IoT* construída com o *WEGnology*.

Figura 4 – Protocolos de Segurança do *WEGnology*.



Fonte – Documentação do *WEGnology* (2021).

A comunicação entre os dispositivos conectados e com a plataforma do *WEGnology* acontece de forma criptografada usando o padrão conhecido como *Transport Layer Security (TLS)*, projetado para fornecer segurança na comunicação em uma rede de dispositivos, adotado por diversos tipos de dispositivos embarcados, linguagens de programação e sistemas operacionais, além de comumente utilizado por navegadores para criptografar dados *Hyper Text Transfer Protocol Secure (HTTPS)*.

Dispositivos que se comunicam com a plataforma por *Message Queuing Telemetry Transport (MQTT)* utilizam chaves e senhas de segurança, que serão utilizadas como nomes de usuário e senhas, respectivamente, pelo protocolo de comunicação.

Além disso, cabe ressaltar que, este projeto em questão, uma eventual tentativa de ataque à plataforma do *WEGnology* não teria qualquer potencial prejudicial ao processo de injeção de alumínio na fábrica, uma vez que o Sistema *IoT* não foi desenvolvido para atuar diretamente na máquina.

4.3 O PROCESSO DE INJEÇÃO DE ALUMÍNIO DE ROTORES

O processo de injeção de alumínio de rotores é uma das diversas etapas da cadeia de fabricação de motores elétricos na WEG. As máquinas responsáveis por executar este processo são as injetoras de alumínio. Nesta seção, é apresentado um explicativo do processo em questão para que haja uma melhor contextualização da função executada por estas máquinas, escolhidas como alvo deste processo de digitalização.

Inicialmente, é necessário preparar uma das principais matérias primas envolvidas neste processo, que é o alumínio em forma líquida. Para isso, lingotes de alumínio (sólido) são depositados em fornos de fusão por indução e são aquecidos a temperaturas que ficam entre 700° C e 920° C, aproximadamente.

Uma vez que o alumínio já se encontra no estado líquido, uma empilhadeira faz a coleta do material utilizando uma panela de transporte, para que então o alumínio possa ser direcionado até o centro de trabalho que irá receber o abastecimento. No centro de trabalho, a panela de alumínio é elevada até uma determinada altura e então é tombada para realizar o despejo do material, que ocorre através de uma tampa localizada na parte superior do forno dosador atrelado à injetora de alumínio daquele centro de trabalho. Como se trata de um processo que envolve riscos significativos à vida humana, no momento do abastecimento, o trabalho na injetora é paralisado e é feito o isolamento dos arredores do centro de trabalho, ficando apenas o operador da empilhadeira e seu ajudante no local, minimizando, desta forma, eventuais danos colaterais em caso de acidente.

O forno dosador é responsável por receber o alumínio fundido e conservar a temperatura do material em um nível especificado (aproximadamente 750° C) para a utilização no processo de injeção. Ele também é responsável por abastecer a injetora de alumínio com a quantidade ideal de material para a realização de cada injeção, com base na especificação do rotor à ser injetado. Seu funcionamento se dá a partir de resistências, que se aquecem conforme são energizadas.

A injetora de alumínio em questão se trata de uma THT 100 de câmara fria vertical com câmara centralizada fixa, que é capaz de gerar uma força de disparo (ação do pistão) de 100 toneladas. Sua alimentação de alumínio é feita pelo forno dosador.

No processo de injeção do rotor, o metal líquido é desejado dentro da câmara de injeção e, quando o pistão é acionado, este desce pressionando o alumínio contra o pistão que, até então, mantinha o orifício de acesso à cavidade do molde fechado. Com a atuação do pistão, o contra pistão desce, liberando o orifício e permitindo a passagem do alumínio para a cavidade do molde. Uma vez concluída a injeção e o processo de solidificação do alumínio injetado, o contra pistão retorna à sua posição inicial, fazendo com que o canal de alimentação seja rompido e ejetando lateralmente

o material remanescente da câmara, conhecido como massalote (9) ou biscoito.

A Figura 5 ilustra a vista interna da parte inferior do molde¹, componente responsável por acomodar o núcleo do rotor (pacote de lâminas) que será injetado.

Figura 5 – Parte Inferior do Molde de Injeção de Alumínio em Rotores.



Fonte – Elaborado pelo Autor (2021).

A Figura 6 representa o molde completo (parte superior e inferior) utilizado no processo de injeção de alumínio de rotores.

¹ O molde em questão possui 2 cavidades, ou seja, 2 rotores são injetados a cada disparo do pistão. Existem moldes de apenas 1 cavidade e até de 4 cavidades.

Figura 6 – Molde de Injeção de Alumínio em Rotores.



Fonte – Elaborado pelo Autor (2021).

A Figura 7 ilustra o pacote de lâminas que será injetado na operação.

Figura 7 – Pacote de Lâminas com Pino de Alinhamento.



Fonte – Elaborado pelo Autor (2021).

A Figura 8 ilustra o produto final do processo de injeção de alumínio de rotores (rotores injetados).

Figura 8 – Rotores Injetados.



Fonte – Elaborado pelo Autor (2021).

A Figura 9 ilustra o massalote (biscoito) que sobra do processo de Injeção de Alumínio de Rotores. Este componente é destinado ao processo de fundição para o reaproveitamento no processo de injeção.

Figura 9 – Massalotes.



Fonte – Elaborado pelo Autor (2021).

5 DESENVOLVIMENTO

Este capítulo tem como objetivo discorrer de forma mais detalhada sobre todas as etapas envolvidas no processo de concepção do Sistema de *IoT* para as máquinas Injetoras de Alumínio, previamente caracterizadas como "Equipamento Legado".

5.1 DA CONCEPÇÃO DA ARQUITETURA DO SISTEMA *IoT*

Nesta seção é descrito o processo de concepção da arquitetura utilizada pelo Sistema *IoT* desenvolvido para digitalizar o processo de Injeção de Alumínio neste trabalho.

Primeiro, os sensores para medição da vazão e da temperatura do líquido refrigerante de alguns componentes chave no processo de injeção de alumínio já haviam sido instalados na máquina (CT-i07) num esforço inicial de se obter algum tipo de dado a respeito do processo. No entanto, os dados eram limitados à visualização direta no *display* dos sensores, o que inviabilizava o acompanhamento das variáveis ao longo do tempo e de forma remota.

Boa parte das variáveis do processo de injeção de alumínio já estavam mapeadas devido a necessidade de serem exibidas na *IHM* da máquina, no entanto, com as limitações locais e temporais descritas para os dados dos sensores.

Então, eis que é identificado no portfólio de produtos da *WEG* um dispositivo que viria a se tornar a peça central na arquitetura do sistema *IoT*, o *Gateway ED300* (Figura 10), um importante *hardware* que permite o fluxo de dados entre diversas redes de comunicação. O *Gateway ED300* é um dispositivo para aplicação *IoT* que possui a capacidade de processamento "Edge", ou seja, de forma local, com a finalidade de conectar equipamentos industriais à internet para a implementação de soluções digitais.

Figura 10 – Dispositivo WCD-ED300.



Fonte – Website WEG - Catálogo de produtos (2021).

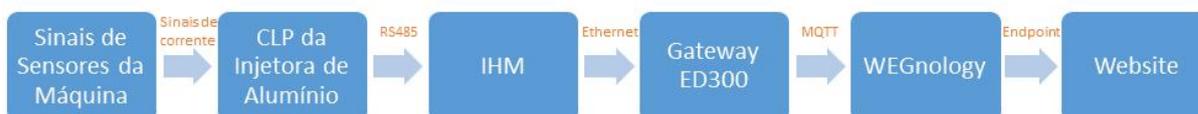
Além dos dispositivos físicos citados até aqui, a WEG também dispunha de uma poderosa plataforma *IoT* baseada em *Cloud Computing*, desenvolvida especialmente para criação de soluções conectadas aos usuários, que naturalmente foi incorporada à arquitetura do sistema, uma vez que era possível a conexão do dispositivo *Gateway ED300* com o *Broker* do *WEGnology*.

A plataforma de *IoT* da WEG dispõe ainda da possibilidade de criação de interfaces para a utilização do usuário final, com elementos gráficos que possibilitam a fácil visualização dos dados coletados no chão de fábrica, auxiliando no monitoramento e tomada de decisão com relação ao processo.

Ou seja, percebe-se que a WEG dispõe de todos os elementos, sejam eles *hardwares* ou *softwares*, para a implementação de um sistema *IoT* capaz de digitalizar o processo realizado por um Equipamento Legado.

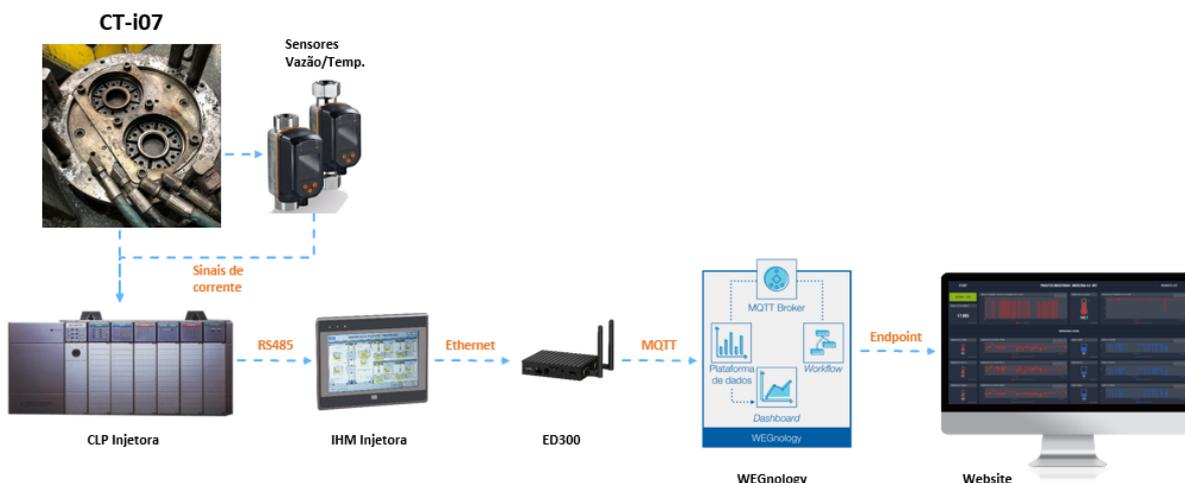
A Figura 11 exibe a arquitetura inicialmente concebida do Sistema de *IoT* para o primeiro centro de trabalho onde ela foi implementada, o CT-i07 da fábrica de Injeção de Alumínio de Rotores.

Figura 11 – Arquitetura do Sistema *IoT*.



Fonte – Elaborado pelo autor (2021).

Utilizando imagens dos componentes da arquitetura, foi construída a Figura 12.

Figura 12 – Arquitetura Ilustrada do Sistema *IoT*.

Fonte – Elaborado pelo autor (2021).

5.2 IDENTIFICAÇÃO DAS VARIÁVEIS DE PROCESSO - INJEÇÃO DE ALUMÍNIO

Nesta seção é feito o detalhamento do processo de identificação e mapeamento das variáveis de processo referentes ao processo de Injeção de Alumínio em Rotores, função executada pela máquina onde o sistema *IoT* foi implementado.

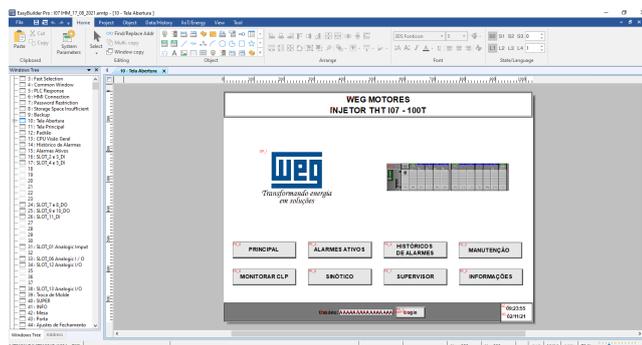
Aqui vale ressaltar que a IHM da máquina já possuía uma interface desenvolvida para auxiliar o processo realizado no centro de trabalho, com informações de diversas variáveis relacionados ao processo de injeção de alumínio em rotores. Para que estas informações pudessem ser exibidas na tela, a IHM realiza a constante leitura de diversos endereços de memória do CLP da injetora.

5.2.1 Mapeamento de Variáveis de Processo via *Easybuilder*

O processo de identificação e mapeamento das variáveis relacionadas ao processo de injeção de alumínio foi realizado através da análise do *software* desenvolvido para auxiliar no processo, utilizado pelo operador da máquina para fazer o apontamento e visualização de algumas informações no decorrer da operação.

O *software* utilizado para desenvolver a interface da IHM foi o *Easybuilder*, sendo que a tela inicial de exibição é mostrada na Figura 13.

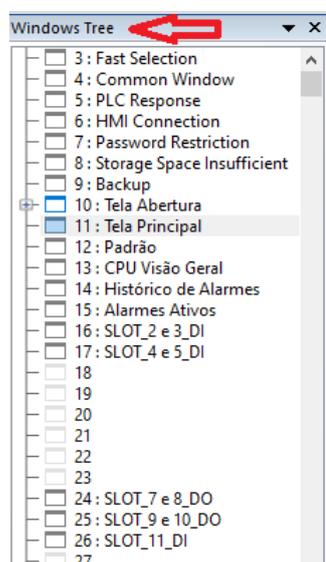
Figura 13 – Tela Inicial IHM.



Fonte – Software Easybuilder (2021).

No arquivo referente à interface da IHM, é possível realizar a navegação entre as diversas janelas do programa, através da “toolbar”¹ na lateral esquerda da ferramenta, conforme indicado na Figura 14.

Figura 14 – Toolbar de Navegação do Easybuilder.



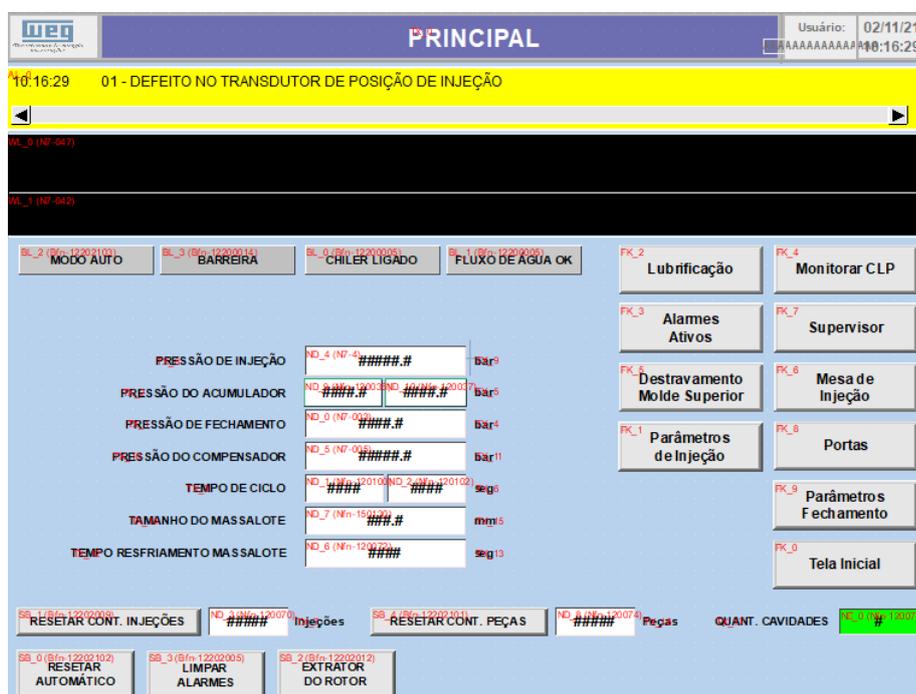
Fonte – Software Easybuilder (2021).

Utilizaremos algumas das variáveis de processo exibidas na janela da “Tela Principal” da IHM para exemplificar o processo de identificação e mapeamento, que irá se repetir para as demais variáveis em outras janelas. Um arquivo do tipo “.xlsx” foi utilizado para auxiliar no processo de mapeamento.

¹ Caso a *toolbar* não esteja sendo exibida, o usuário poderá habilitá-la acessando a guia “View”, na parte superior da tela, e marcando a caixa *toolbar* “Windows Tree”.

Assim sendo, na vista da “Tela Principal” da interface (Figura 15), temos algumas variáveis sendo mostradas ao operador, sendo elas “Pressão de Injeção”, “Pressão do Acumulador” (valor “setado” e valor real), “Pressão de Fechamento”, “Pressão do Compensador”, “Tempo de Ciclo”, “Tamanho do Massalote”, “Tempo de Resfriamento do Massalote”, além de um contador de injeções, um contador de peças e um campo que indica a quantidade de cavidades presentes no molde de injeção que está sendo utilizado.

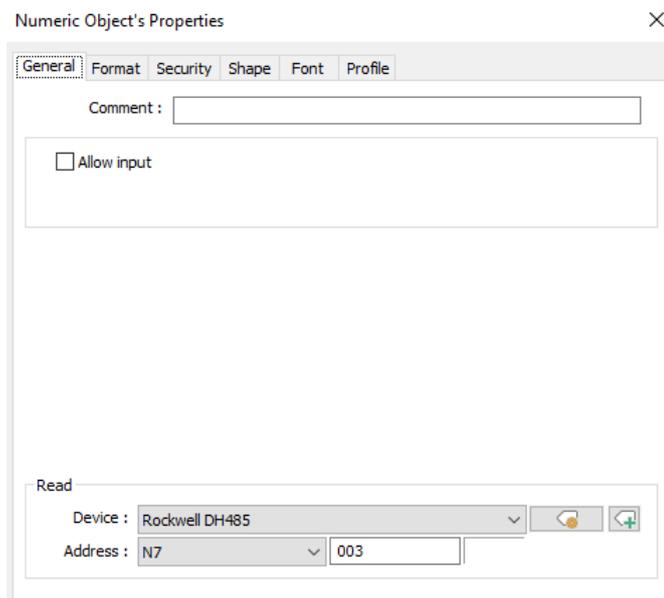
Figura 15 – Tela Principal da IHM.



Fonte – Software Easybuilder (2021).

Dando um duplo clique no campo relativo ao indicador de “Pressão de Fechamento”, uma janela com a informação do endereço de memória do CLP se abrirá, conforme a Figura 16.

Figura 16 – Janela de Configuração do Endereço de Memória para Leitura da VP.



Fonte – Software Easybuilder (2021).

Na aba Geral (“*General*”), é possível localizar o bloco “*Read*”, onde estão descritas informações como o nome do dispositivo (“*Device*”) e o endereço de memória daquela variável, composto por “*device_type*” e “*address_offset*”, que serão melhor explicados no decorrer da seção 5.5.2.2.

No caso da variável de “Pressão de Fechamento”, conforme visto na Figura 16, o nome do dispositivo é “*Rockwell DH485*” e o endereço de memória é o “*N7 003*” (*device_type* e *address_offset*, respectivamente). Estas informações serão incluídas na planilha afim de auxiliar no mapeamento.

O procedimento descrito acima foi realizado para diversas variáveis de processo, fazendo com que o detalhamento de cada uma delas fosse repetitivo, não agregando ao conteúdo deste trabalho.

Desta forma, a planilha utilizada para auxiliar no mapeamento das variáveis foi construída conforme a Figura 17.

Figura 17 – Planilha de Mapeamento de VPs.

4	Nº	Descrição	PLC	
5	1	Pressão Fechamento Pinça (bar)	N7	3
6	2	Pressão do Acumulador (bar)	Nfn	120037
7	3	Pressão do Compensador (bar)	N7	5
8	4	Pressão Injeção (bar)	N7	4
9	5	Velocidade de Injeção (%)	Nfn	120104
10	6	Fluxo de Saída Pistão (l/min)	Nfn	150006
11	7	Temperatura de Saída Pistão (°C)	Nfn	150007
12	8	Fluxo de Saída Câmara (l/min)	Nfn	150004
13	9	Temperatura de Saída Câmara (°C)	Nfn	150005
14	10	Fluxo de Saída Placa (l/min)	Nfn	150000
15	11	Temperatura de Saída Placa (°C)	Nfn	150001
16	12	Fluxo de Saída Matriz (l/min)	Nfn	150002
17	13	Temperatura de Saída Matriz (°C)	Nfn	150003
18	14	Tempo de Resfriamento Injeção (s)	Nfn	120072
19	15	Tempo de Abertura Válvula Injeção (s) (set)	Nfn	120084
20	16	Tempo de Abertura Válvula Injeção (s) (real)	Nfn	120078
21	17	Contador de Tiros	Nfn	120070
22	18	Contador de Peças	Nfn	120074
23	19	Número da Ferramenta	Nfn	150100

Fonte – Elaborado pelo autor (2021).

Aqui vale ressaltar que, durante o desenvolvimento do projeto, outras variáveis de processo passaram a ser consideradas e, portanto, houve a necessidade de fazer o processo de mapeamento conforme descrito anteriormente. Algumas destas novas variáveis incorporadas à aplicação serão detalhadas no decorrer deste capítulo.

5.2.2 Variáveis de Temperatura e Vazão do Líquido Refrigerante

Também foi levantada a necessidade de se monitorar as temperaturas e vazões de saída do líquido refrigerante de alguns componentes chave do processo de injeção de alumínio, como o pistão, câmara, matriz e placa.

Para tal, foi necessária a instalação de sensores medidores de vazão e temperatura junto às mangueiras responsáveis por canalizar o fluxo do líquido refrigerante para o interior dos componentes da máquina. Este processo será descrito no decorrer das seções que seguem neste capítulo.

5.3 SENSORIZAÇÃO DA MÁQUINA

Conforme proposto na seção anterior, viu-se a possibilidade de, além das variáveis de processo que já foram mapeadas via *software Easybuilder*, coletar também informações a respeito da vazão e da temperatura de saída do líquido refrigerante do processo de injeção, mais especificamente, de alguns componentes da máquina injetora, como o pistão, a câmara, a matriz e a placa.

Desta forma, foi feita a instalação, por parte da equipe de manutenção do departamento de Injeção de Alumínio, de sensores capazes de medir a vazão e a temperatura do fluido do processo.

O modelo de sensor, escolhido e adquirido pela equipe de manutenção por questões de familiaridade prévia, é o SV5204 da empresa “*IFM Electronic*”, ilustrado na Figura 18.

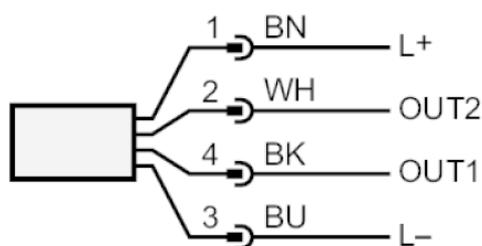
Figura 18 – Sensor de Vazão e Temperatura SV5204.



Fonte – Website do fabricante (2021).

O sensor possui um intervalo de medição de 2 à 40 litros por minuto para a vazão e de -10°C à 90°C para a temperatura. Para transmitir estes dados, o modelo dispõe de um conector do tipo M12 com 4 contatos, sendo 2 deles para a sua própria alimentação (L+ e L-) e outros 2 referentes às suas saídas analógicas (4 à 20mA), conforme ilustrado na Figura 19.

Figura 19 – Esquemático de Conexões do Sensor.



Fonte – Manual do Fabricante do Dispositivo (2021).

A saída “Out 1” possui o fio preto, e transmite a informação relativa à temperatura do líquido em questão, enquanto que, a saída “Out 2”, de fio branco, transmite a informação do fluxo volumétrico, ou seja, a vazão.

Uma vez instalados na máquina, os sensores ficaram dispostos conforme ilustra a Figura 20.

Figura 20 – Sensores de Vazão/Temperatura Instalados na Injetor.



Fonte – Elaborado pelo autor (2021).

Na seção a seguir é descrito o processo de conexão dos sensores com o CLP da injetora, além do desenvolvimento da lógica, via linguagem *Ladder*, para conversão dos sinais de corrente para valores de vazão e temperatura.

5.4 CONEXÃO DOS SINAIS DOS SENSORES COM O CLP DA INJETORA

Esta seção detalha o processo de conexão dos sinais dos sensores, instalados conforme descrito na seção anterior, ao CLP da máquina Injetora de Alumínio.

5.4.1 Do CLP Integrado à Máquina Injetora de Alumínio

Inicialmente, cabe aqui especificar o modelo de CLP utilizado pela máquina, que se trata de um *Allen-Bradley* da família *SLC 500* (Figura 21), lançado no ano de 1991. O modelo dispõe de 1 porta no padrão *RS-485* que é utilizada para estabelecer

a comunicação com a IHM da Injetora, que por sua vez, auxilia na operação realizada na máquina.

Figura 21 – PLC Allen-Bradley SLC500.



Fonte – Website do fabricante (2021).

5.4.2 Conexão dos Sinais dos Sensores e Implementação da Lógica *Ladder*

Como descrito na Seção 5.3, os sensores instalados para monitoramento de vazão e temperatura do líquido refrigerante possuem 2 sinais de saída de corrente cada, de 4mA à 20mA, que foram conectados às entradas analógicas de um dos módulos de expansão analógica disponível no CLP, conforme ilustrado na Figura 22.

Figura 22 – Módulo de Expansão Analógico do PLC.

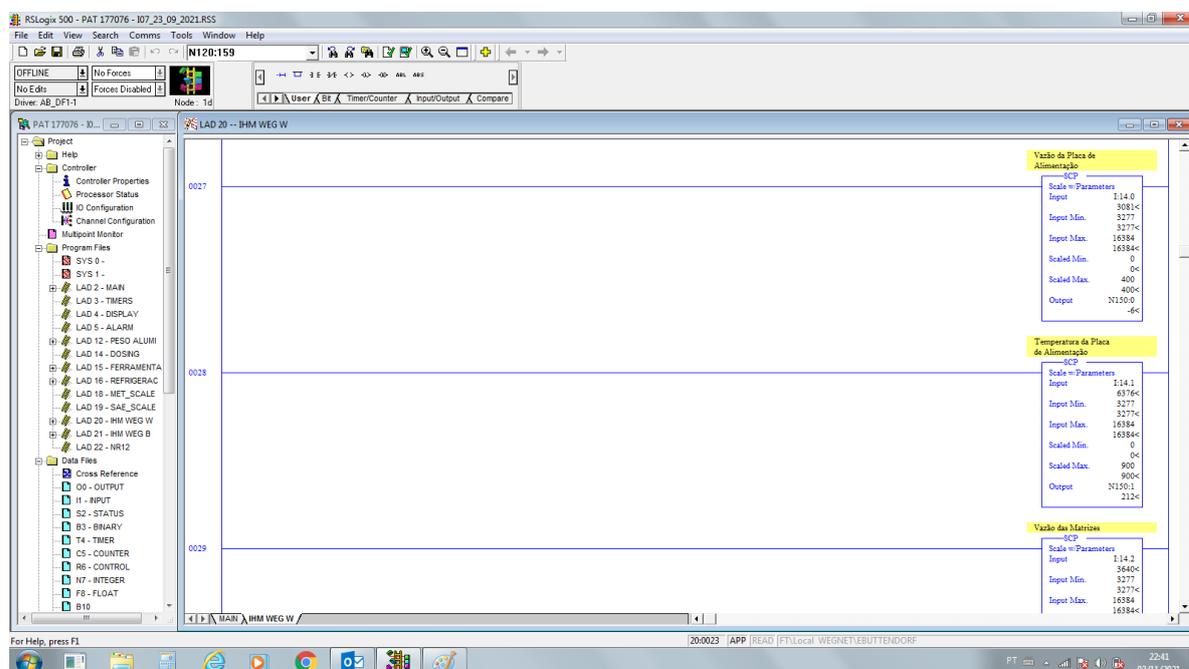


Fonte – Elaborado pelo autor (2021).

O software de automação chamado *RSLogix 500* é utilizado para programar os CLPs das famílias Allen-Bradley SLC 500 e Micrologix, da fabricante Rockwell

Automation, através da linguagem de programação *Ladder*. A Figura 23² ilustra a implementação da lógica responsável por tratar os sinais de corrente para vazão e temperatura provenientes dos sensores instalados na máquina injetora de alumínio.

Figura 23 – *Ladder* para Conversão dos Sinais dos Sensores.



Fonte – Elaborado pelo autor (2021).

O bloco lógico utilizado para interpolar os sinais dos sensores é o *SCP* (*Scale with Parameters*), que possui os seguintes parâmetros que devem ser configurados:

- *Input* - se trata do endereço de memória da entrada que está recebendo o sinal do sensor. No caso da Figura 23, o bloco para vazão foi configurado com *Input* “I.14.0” e o bloco para temperatura com o *Input* em “I.14.1”, que representa as entradas analógicas do módulo do *CLP*;
- *Input Min* - se trata do menor valor decimal recebido, que neste caso, é de “3277”;
- *Input Max* - se trata do maior valor decimal recebido, que neste caso, é de “16384”³;
- *Scaled Min* - onde deve ser especificado o menor valor da faixa de valores que se deseja efetuar o *scale*, que neste caso é de “0” litros por minuto;

² A figura em questão exibe a lógica para tratamento das variáveis do sensor de fluxo instalado na saída da Placa de Alimentação da máquina. A mesma lógica foi implementada para os demais sensores e respectivas variáveis.

³ De acordo com o manual do módulo de entradas analógicas do *CLP SLC500* utilizado na máquina, a representação decimal para valores de corrente na faixa de 4 à 20mA é de 3,277 à 16,384, respectivamente.

- *Scaled Max* - aqui deve ser especificado o maior valor da faixa de valores do *scale*, que neste caso é de “400”⁴;
- *Output* - onde deverá ser informado o endereço de memória em que o valor ajustado (“scaled”) será armazenado. No exemplo, a informação de vazão ficará armazenada no endereço “N150:0” e a informação de temperatura no endereço “N150:1”.

5.5 PROGRAMAÇÃO DA IHM DA INJETORA DE ALUMÍNIO

Nesta seção é descrito como foi feita programação da IHM (Interface Homem-Máquina) para recuperar as informações referentes às variáveis de processo em seus respectivos endereços de memória no CLP da injetora de alumínio.

5.5.1 Da “Ponte” com a IHM

Brevemente, também vale destacar o motivo pelo qual a conexão do CLP não pode ser feita de forma direta com o dispositivo *Gateway ED300*, que se deu pelo fato de que o CLP da máquina não dispunha de saída de comunicação sobressalente compatível com o ED300, desta forma, sendo necessária a “ponte” com a IHM. A única porta de comunicação RS485 do CLP já estava sendo usada para conectar com a IHM, que é indispensável para o processo.

5.5.2 Programação da Macro para Leitura e Gravação das Variáveis de Processo

A IHM utilizada no processo é a de modelo WEG MT8150XE (Figura 24), originalmente fabricada pela empresa *Weintek*. O modelo dispõe da possibilidade de criação de Macros, que proporciona funcionalidades adicionais que a aplicação possa vir a precisar.

⁴ Aqui cabe ressaltar que o sensor mede, de fato, um fluxo de vazão até 40 litros por minuto, no entanto, no bloco lógico utilizado não é possível parametrizar com informação decimal, portanto, neste caso, o valor deve ser multiplicado por 10 para remover a vírgula e, posteriormente, dividido por 10 para se obter o valor correto.

Figura 24 – IHM WEG MT8150XE.



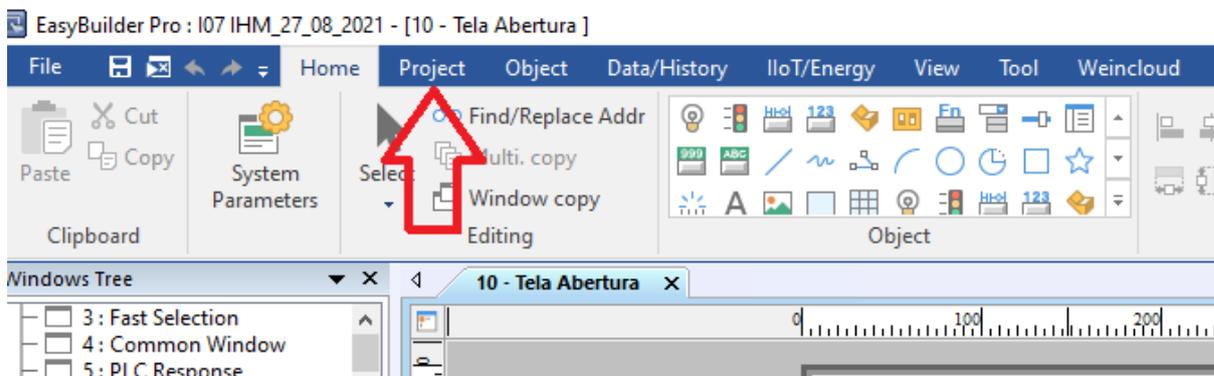
Fonte – Website WEG - Catálogo de produtos (2021).

Uma macro pode ser definida como uma sequência de comandos automatizados que serão executados de acordo com um determinado tempo de execução. No caso da aplicação desenvolvida neste trabalho, as macros serão responsáveis por ler as variáveis de processo em seus respectivos endereços de memória no CLP e salvá-las em endereços de memória da própria IHM, possibilitando, posteriormente, a leitura destas variáveis pelo dispositivo ED300.

Uma vez que as variáveis de processo já foram devidamente mapeadas conforme o processo descrito na Seção 5.2, estas devem ser constantemente “recuperadas” e salvas na memória da IHM. Para isto, a seguir será descrita a lógica por trás deste processo.

Dentro do *software* de programação *Easybuilder*, é necessário acessar a guia de Projeto (“*Project*”) da ferramenta (Figura 25).

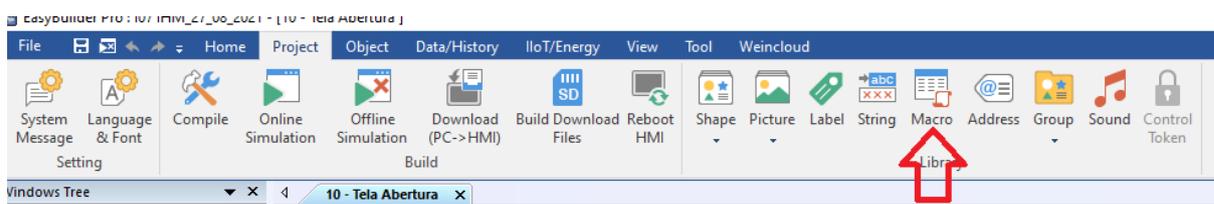
Figura 25 – Inclusão de Nova Macro.



Fonte – Software Easybuilder (2021).

Em seguida, selecionar o ícone de Macro, conforme a Figura 26.

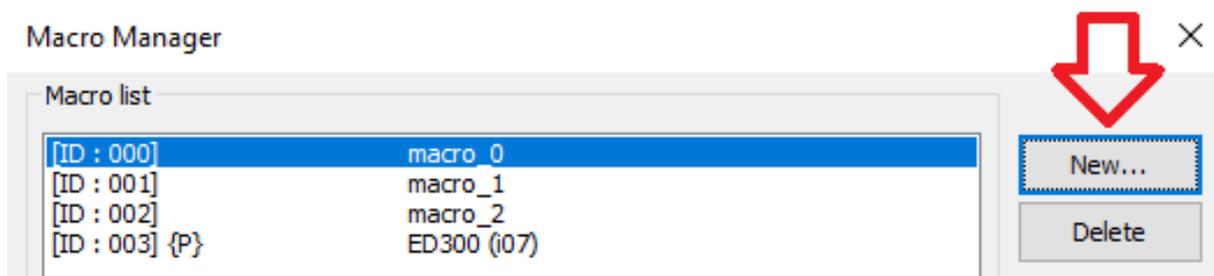
Figura 26 – Criação de Nova Macro.



Fonte – Software Easybuilder (2021).

Na sequência, é necessário que uma nova macro seja criada, através do botão “New...” (Figura 27).

Figura 27 – Adicionar Nova Macro.

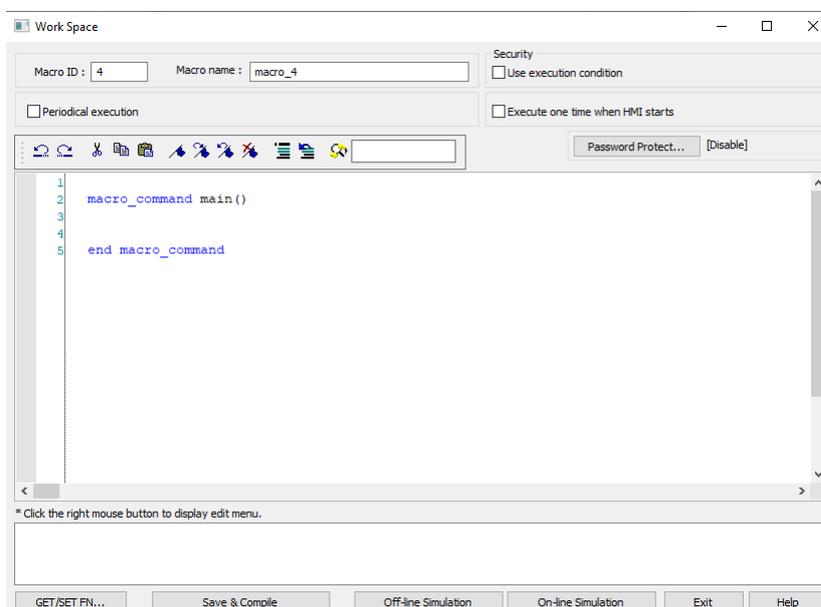


Fonte – Software Easybuilder (2021).

Desta forma, será aberto o ambiente de programação de uma nova macro, onde serão escritos os comandos a serem realizados pela IHM de forma paralela ao seu

programa principal (Figura 28).

Figura 28 – Ambiente de programação de Macro.



Fonte – *Software Easybuilder* (2021).

5.5.2.1 Declaração de Variáveis

Inicialmente, é necessário fazer a declaração das variáveis de processo que serão lidas e escritas pela macro. Neste código, as variáveis foram declaradas como sendo do tipo “short”, que pode armazenar 16 *bits* (*word*) dentro de um intervalo entre -32767 e +32767 (figuras 29 e 30).

Figura 29 – Declaração de Variáveis na Macro.

```
77 // Variáveis inteiras
78
79 //Pressão Fechamento Pinça
80 short i1
81 //Pressão do acumulador
82 short i2
83 //Pressão do compensador
84 short i3
85 //Pressão Injeção
86 short i4
87 //Velocidade de Injeção
88 short i5
89 //Fluxo de saída pistão
90 short i6
91 //Temperatura de saída pistão
92 short i7
93 //Fluxo de saída câmara
94 short i8
95 //Temperatura de saída câmara
96 short i9
97 //Fluxo de saída placa
98 short i10
99 //Temperatura de saída placa
100 short i11
101 //Fluxo de saída matriz
102 short i12
103 //Temperatura de saída matriz
104 short i13
105 //Tempo de Resfriamento (solidificação)
106 short i14
107
108 //Tempo de Abertura de Válvula Injeção (set)
109 short i15
110 //Tempo de Abertura de Válvula Injeção (real)
111 short i16
112 //Contador de Tiros
113 short i17
114 //Contador de Peças
115 short i18
```

Fonte – *Software Easybuilder* (2021).

Figura 30 – Declaração de Variáveis na Macro.

```
115 short i18
116 //Número da Ferramenta
117 short i19
118 //Massalote - Tolerância Mínima (mm)
119 short i20
120 //Massalote - Tolerância Máxima (mm)
121 short i21
122 //Massalote - Tamanho Real (mm)
123 short i22
124 //Massalote - Altura (mm)
125 short i23
126
127
128
129
130 //Tempo para Fechar Porta
131 short i83
132 //Tempo para Abrir Porta
133 short i88
134
135 //Temperatura do Forno - Kaizen
136 short i99
```

Fonte – *Software Easybuilder* (2021).

5.5.2.2 Recuperação de Variáveis

Uma vez que as variáveis já foram declaradas, o próximo passo é realizar a recuperação da mesmas, ou seja, a leitura das VPs em seus respectivos endereços de memória no CLP. A leitura de variáveis é feita através da seguinte sintaxe: `GetData(read_data, device_name, device_type, address_offset, data_count)`, onde o campo `read_data` será definido como a variável declarada para aquela respectiva VP a ser lida, o campo `device_name` é o nome do dispositivo que “possui” as informações das VPs, e que neste caso será o CLP `Rockwell DH485` (o nome do dispositivo deve ser colocado entre aspas duplas), o campo `device_type` é o tipo de endereço de memória em que informação será lida, que irá ser diferente conforme for a variável. O campo `adress_offset` é o endereço propriamente dito, dentro daquele `device_type`. Finalmente, o campo `data_count` será a quantidade de valores lidos, uma vez que o comando `GetData` em questão também pode ser usado para leitura de `arrays`. Como neste caso será usado para realizar a leitura de apenas um endereço, receberá valor 1.

Escrevendo o comando `GetData` para todas as variáveis de processo mapeadas anteriormente, o código ficará conforme a Figura 31.

Figura 31 – Recuperação de Variáveis do CLP.

```
138 // Recupera variaveis
139
140 GetData(i1, "Rockwell DH485", N7, 3, 1)
141 GetData(i2, "Rockwell DH485", Nfn, 120037, 1)
142 GetData(i3, "Rockwell DH485", N7, 5, 1)
143 GetData(i4, "Rockwell DH485", N7, 4, 1)
144
145 GetData(i5, "Rockwell DH485", Nfn, 120104, 1)
146
147 GetData(i6, "Rockwell DH485", Nfn, 150006, 1)
148 GetData(i7, "Rockwell DH485", Nfn, 150007, 1)
149 GetData(i8, "Rockwell DH485", Nfn, 150004, 1)
150 GetData(i9, "Rockwell DH485", Nfn, 150005, 1)
151 GetData(i10, "Rockwell DH485", Nfn, 150000, 1)
152 GetData(i11, "Rockwell DH485", Nfn, 150001, 1)
153 GetData(i12, "Rockwell DH485", Nfn, 150002, 1)
154 GetData(i13, "Rockwell DH485", Nfn, 150003, 1)
155 GetData(i14, "Rockwell DH485", Nfn, 120072, 1)
156
157 GetData(i15, "Rockwell DH485", Nfn, 120084, 1)
158 GetData(i16, "Rockwell DH485", Nfn, 120078, 1)
159 GetData(i17, "Rockwell DH485", Nfn, 120070, 1)
160 GetData(i18, "Rockwell DH485", Nfn, 120074, 1)
161 GetData(i19, "Rockwell DH485", Nfn, 150100, 1)
162 GetData(i20, "Rockwell DH485", Nfn, 150110, 1)
163 GetData(i21, "Rockwell DH485", Nfn, 150120, 1)
164 GetData(i22, "Rockwell DH485", Nfn, 150130, 1)
165 GetData(i23, "Rockwell DH485", Nfn, 120116, 1)
166
167 GetData(i83, "Rockwell DH485", Nfn, 120183, 1)
168 GetData(i88, "Rockwell DH485", Nfn, 120188, 1)
169
170 //Kaizen - Forno
171 GetData(i99, "Rockwell DH485", Nfn, 120159, 1)
```

Fonte – Software Easybuilder (2021).

5.5.2.3 Escrita de Variáveis na Memória da IHM

Após o processo de recuperação de variáveis de processo em suas respectivas memórias do CLP da máquina, o próximo passo a ser realizado é a escrita das variáveis recuperadas em endereços de memória da própria IHM.

O comando de escrita de variáveis na memória da IHM é realizado conforme a sintaxe a seguir: `SetData(send_data, device_name, device_type, address_offset, data_count)`. A lógica é parecida com a do comando `GetData`, sendo que no campo “`send_data`” deverá ser informada a variável em questão à ser escrita, o campo “`device_name`” se trata do nome do dispositivo onde será realizada a escrita da variável, que no caso será “Local HMI” (o nome do dispositivo deve ser colocado entre aspas duplas), na sequência, no campo “`device_type`” será o tipo de endereço de memória em que a variável será salva, o campo “`address_offset`” é o endereço propriamente dito da variável na memória da IHM, e por fim, novamente, “`data_count`” tem relação com a quantidade de valores a serem salvos, que neste caso será 1.

Uma vez escrito os comandos “`SetData`” para todas as variáveis de processo, o código na macro ficará conforme a Figura 32.

Figura 32 – Alocação de Variáveis na Memória da IHM.

```
173 //Escreve variaveis
174
175 SetData(i1, "Local HMI", LW, 500, 1)
176 SetData(i2, "Local HMI", LW, 502, 1)
177 SetData(i3, "Local HMI", LW, 504, 1)
178 SetData(i4, "Local HMI", LW, 506, 1)
179 SetData(i5, "Local HMI", LW, 508, 1)
180 SetData(i6, "Local HMI", LW, 510, 1)
181 SetData(i7, "Local HMI", LW, 512, 1)
182 SetData(i8, "Local HMI", LW, 514, 1)
183 SetData(i9, "Local HMI", LW, 516, 1)
184 SetData(i10, "Local HMI", LW, 518, 1)
185 SetData(i11, "Local HMI", LW, 520, 1)
186 SetData(i12, "Local HMI", LW, 522, 1)
187 SetData(i13, "Local HMI", LW, 524, 1)
188 SetData(i14, "Local HMI", LW, 526, 1)
189 SetData(i15, "Local HMI", LW, 528, 1)
190 SetData(i16, "Local HMI", LW, 530, 1)
191 SetData(i17, "Local HMI", LW, 532, 1)
192 SetData(i18, "Local HMI", LW, 534, 1)
193 SetData(i19, "Local HMI", LW, 536, 1)
194 SetData(i20, "Local HMI", LW, 538, 1)
195 SetData(i21, "Local HMI", LW, 540, 1)
196 SetData(i22, "Local HMI", LW, 542, 1)
197 SetData(i23, "Local HMI", LW, 544, 1)
198
199 SetData(i83, "Local HMI", LW, 583, 1)
200 SetData(i88, "Local HMI", LW, 588, 1)
201
202 //Kaizen Forno
203 SetData(i99, "Local HMI", LW, 599, 1)
204
205
206 end macro_command
```

Fonte – Software Easybuilder (2021).

Na planilha de mapeamento, foi feita a complementação quanto aos endere-

ços de memória utilizados na IHM para armazenamento das variáveis de processo, conforme a Figura 33⁵.

Figura 33 – Planilha de Mapeamento de VPs.

N°	Descrição	Endereço			IHM
		PLC	Macro		
1	Pressão Fechamento Pinça (bar)	N7	3	i1	LW500
2	Pressão do Acumulador (bar)	Nfn	120037	i2	LW502
3	Pressão do Compensador (bar)	N7	5	i3	LW504
4	Pressão Injeção (bar)	N7	4	i4	LW506
5	Velocidade de Injeção (%)	Nfn	120104	i5	LW508
6	Fluxo de Saída Pistão (l/min)	Nfn	150006	i6	LW510
7	Temperatura de Saída Pistão (°C)	Nfn	150007	i7	LW512
8	Fluxo de Saída Câmara (l/min)	Nfn	150004	i8	LW514
9	Temperatura de Saída Câmara (°C)	Nfn	150005	i9	LW516
10	Fluxo de Saída Placa (l/min)	Nfn	150000	i10	LW518
11	Temperatura de Saída Placa (°C)	Nfn	150001	i11	LW520
12	Fluxo de Saída Matriz (l/min)	Nfn	150002	i12	LW522
13	Temperatura de Saída Matriz (°C)	Nfn	150003	i13	LW524
14	Tempo de Resfriamento Injeção (s)	Nfn	120072	i14	LW526
15	Tempo de Abertura Válvula Injeção (s) (set)	Nfn	120084	i15	LW528
16	Tempo de Abertura Válvula Injeção (s) (real)	Nfn	120078	i16	LW530
17	Contador de Tiros	Nfn	120070	i17	LW532
18	Contador de Peças	Nfn	120074	i18	LW534
19	Número da Ferramenta	Nfn	150100	i19	LW536
20	Massalote - Tolerância Mínima (mm)	Nfn	150110	i20	LW538
21	Massalote - Tolerância Máxima (mm)	Nfn	150120	i21	LW540
22	Massalote - Tamanho Real (mm)	Nfn	150130	i22	LW542
23	Massalote - Altura (mm)	Nfn	120116	i23	LW544

Fonte – Elaborado pelo autor (2021).

Com isto, conclui-se a sequência de comandos automatizados, ou seja, a Macro que será responsável por recuperar e, em seguida, salvar as diversas variáveis de processo da memória do CLP para a memória da IHM da injetora de alumínio.

5.5.2.4 Frequência de Execução da Macro

Um ponto fundamental que ainda deve ser considerado é com relação à frequência de execução da macro. Por configuração padrão, conforme pode ser observado na Figura 28, existe uma “checkbox” com o nome “Periodical Execution”, que tem a ver justamente com a frequência de execução daquela macro, ou seja, caso permaneça desmarcada, a macro irá ser executada uma única vez.

Do ponto de vista do projeto desenvolvido neste trabalho, é necessário que a macro seja constantemente executada numa frequência relativamente elevada, afim de garantir a atualização de todas as variáveis de processo no decorrer do tempo. Portanto, é imprescindível a marcação desta opção e da configuração de uma taxa de execução compatível com a necessidade do sistema.

⁵ Algumas variáveis foram incluídas num momento posterior ao mapeamento inicial básico de VPs, portanto não foram citadas anteriormente.

Neste projeto, a frequência de execução da macro foi configurada para 1 execução a cada 200ms, conforme ilustra a Figura 34, garantindo uma taxa de atualização satisfatória dos dados.

Figura 34 – Frequência de Execução da Macro.

Macro ID : 3	Macro name : ED300 (j07)
<input checked="" type="checkbox"/> Periodical execution	Time interval (0 ~ 864000) : 2 x 100ms

Fonte – *Software Easybuilder* (2021).

5.6 CONFIGURAÇÃO DO DISPOSITIVO GATEWAY

Nesta seção é feito o detalhamento do processo de configuração do dispositivo *Gateway ED300*, de marca WEG, mostrado na Figura 35. Este dispositivo é responsável pela recuperação de todas as variáveis de processo armazenadas na memória da IHM injetora e, posteriormente, enviar estes dados à nuvem para que possam ser acessados via plataforma *WEGnology*.

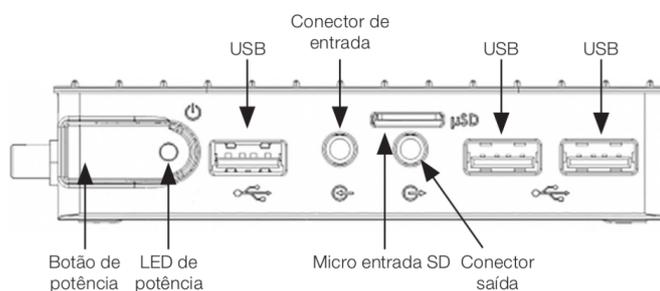
Figura 35 – Dispositivo WCD-ED300.



Fonte – *Website WEG - Catálogo de produtos* (2021).

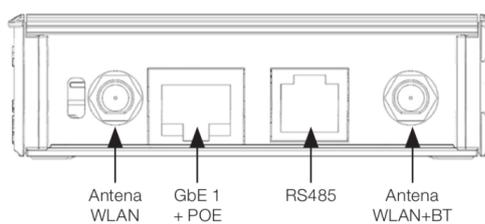
O dispositivo possui diversos conectores de entrada para variados protocolos de comunicação, conforme ilustrado nas figuras 36, 37 e 38.

Figura 36 – Conectores Frontais do ED300.



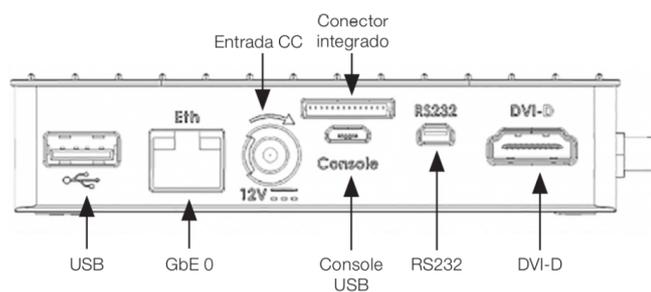
Fonte – Manual do ED300 (2021).

Figura 37 – Conectores Laterais do ED300.



Fonte – Manual do ED300 (2021).

Figura 38 – Conectores Traseiros do ED300.



Fonte – Manual do ED300 (2021).

5.6.1 Acesso à Interface de Comunicação do ED300

O dispositivo ED300 possui uma interface (servidor *weblocal*) de comunicação para que seja possível realizar diversas configurações, como a conexão com uma rede

de *internet*, integração com o *WEGnology*, configuração dos parâmetros de comunicação RS485, entre outras.

Existem duas formas de acessar esta interface de comunicação, a primeira delas é via conexão *Ethernet*, conforme os passos descritos a seguir:

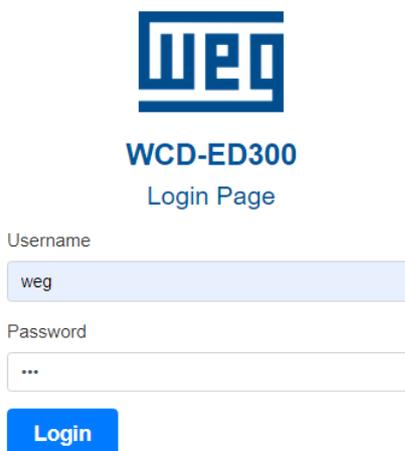
1. Realizar a conexão do ED300 e um computador via cabo *Ethernet* através da porta "GbE 0"(Figura 38);
2. Por definição de fábrica, o IP do dispositivo é 192.168.0.10, fazendo com que o computador deva ser configurado com um IP estático dentro da mesma rede (192.168.0), ou seja, entre 192.168.0.0 e 192.168.0.24, excluído o IP já utilizado pelo ED300;
3. Utilizando um navegador *web*, deve-se digitar o IP do ED300 no campo de URL e teclar *enter*.

A segunda forma de acessar a interface de configuração é via *Wi-Fi*, fazendo uso do *Access Point* criado automaticamente pelo dispositivo no momento em que ele é ligado:

1. A rede será criada com o nome "**WCD-ED300-**", seguido de parte do "*Mac Address*" da porta "GbE 0"⁶. Exemplo, **MAC: 00:01:c0:2a:fe:b4**, a rede será criada com o nome "**WCD-ED300-2a:fe:b4**".
2. A senha da rede será o próprio *MAC*, porém sem os sinais de dois pontos e substituindo as letras maiúsculas por minúsculas, portanto, conforme o exemplo, resultaria em "**0001c02afeb4**".
3. O acesso da interface pelo navegador deverá ser feito através do IP padrão "**10.10.10.1**".

Realizado o acesso pelo navegador (*Ethernet* ou *Wi-Fi*), será apresentada a tela de *login* da interface, conforme a Figura 39. Por padrão, tanto usuário e senha estão definidos como "**weg**".

⁶ O *MAC Address* citado neste passo coincide com o número de série do dispositivo ED300, que pode ser localizado na etiqueta colada na parte inferior do aparelho.

Figura 39 – Tela de *Login* da Interface do ED300.

WCD-ED300
Login Page

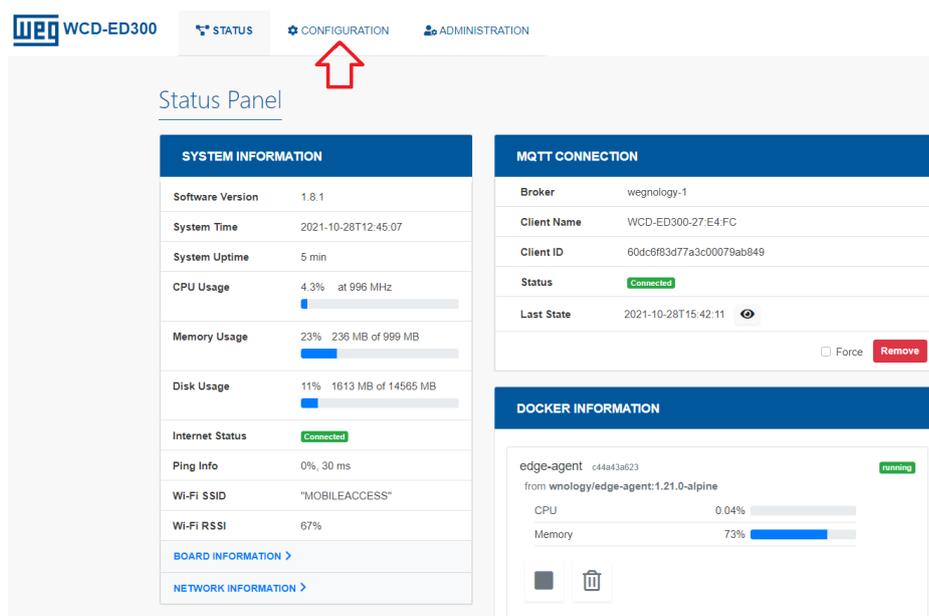
Username
weg

Password
...

Login

Fonte – Tela de *login* do ED300 (2021).

Uma vez logado, o usuário se encontrará na Página de Estado (“*Status Panel*”, figura), página inicial da interface de comunicação do dispositivo, e deverá seguir para a guia de Configuração (“*Configuration*”) para seguir com o processo de configuração do ED300, conforme indicação da Figura 40.

Figura 40 – Página de *Status* do ED300.

WCD-ED300 STATUS CONFIGURATION ADMINISTRATION

Status Panel

SYSTEM INFORMATION	
Software Version	1.8.1
System Time	2021-10-28T12:45:07
System Uptime	5 min
CPU Usage	4.3% at 996 MHz
Memory Usage	23% 236 MB of 999 MB
Disk Usage	11% 1613 MB of 14565 MB
Internet Status	Connected
Ping Info	0%, 30 ms
Wi-Fi SSID	"MOBILEACCESS"
Wi-Fi RSSI	67%
BOARD INFORMATION >	
NETWORK INFORMATION >	

MQTT CONNECTION	
Broker	wegnology-1
Client Name	WCD-ED300-27-E4:FC
Client ID	60dc6f83d77a3c00079ab849
Status	Connected
Last State	2021-10-28T15:42:11
<input type="checkbox"/> Force <input type="button" value="Remove"/>	

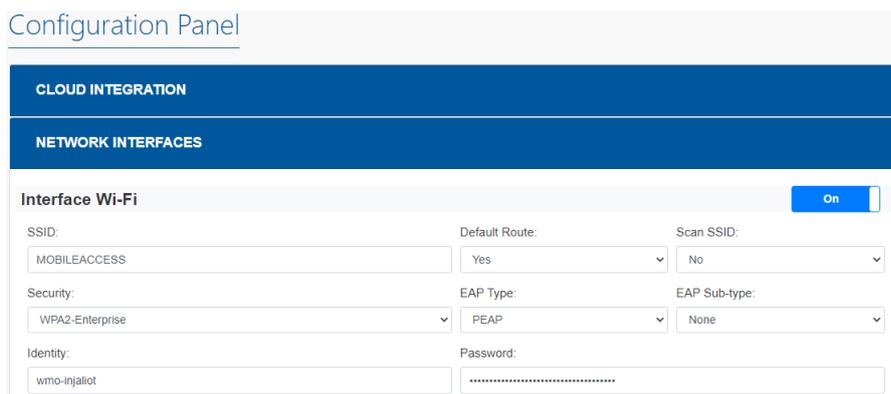
DOCKER INFORMATION	
edge-agent c44a43a623	Running
from wnology/edge-agent:1.21.0-alpine	
CPU	0.04%
Memory	73%
<input type="checkbox"/> <input type="checkbox"/>	

Fonte – Servidor local de configuração do ED300 (2021).

5.6.2 Conexão do ED300 com uma rede Wi-Fi

Na guia de Configuração, o próximo passo será configurar a rede de acesso à internet que será utilizada pelo dispositivo, através do painel de Interfaces de Rede, ou “Network Interfaces”, conforme a Figura 41.

Figura 41 – Painel de Configuração de Wi-Fi.



The image shows a web interface titled "Configuration Panel". It has a blue header with "CLOUD INTEGRATION" and "NETWORK INTERFACES" tabs. The "NETWORK INTERFACES" tab is active, showing the "Interface Wi-Fi" configuration. The interface includes a toggle switch set to "On". The configuration fields are as follows:

Field	Value
SSID	MOBILEACCESS
Default Route	Yes
Scan SSID	No
Security	WPA2-Enterprise
EAP Type	PEAP
EAP Sub-type	None
Identity	wmo-injalot
Password

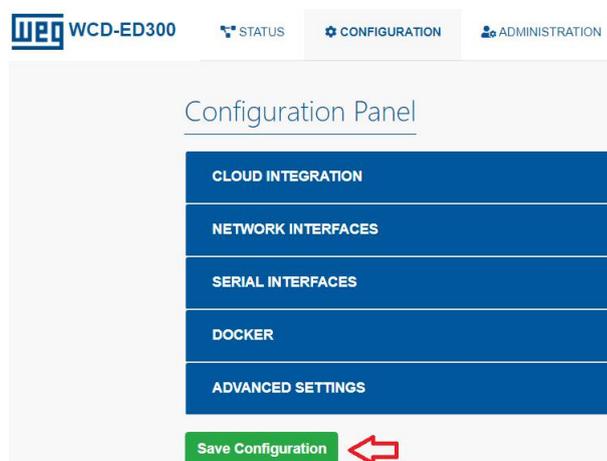
Fonte – Servidor local de configuração do ED300 (2021).

No campo “SSID” deverá ser incluído o nome da rede. No exemplo anterior, a rede possui o nome “MOBILEACCESS”.

Nos campos “Identity” e “Password” deverão ser informados o usuário e senha da rede, respectivamente. No exemplo, o nome de usuário da rede é “wmo-injalot”.

É recomendado realizar o salvamento das configurações após cada alteração, através do botão “Save Configuration” localizado no final da página (Figura 42).

Figura 42 – Botão de Salvamento de Configurações.



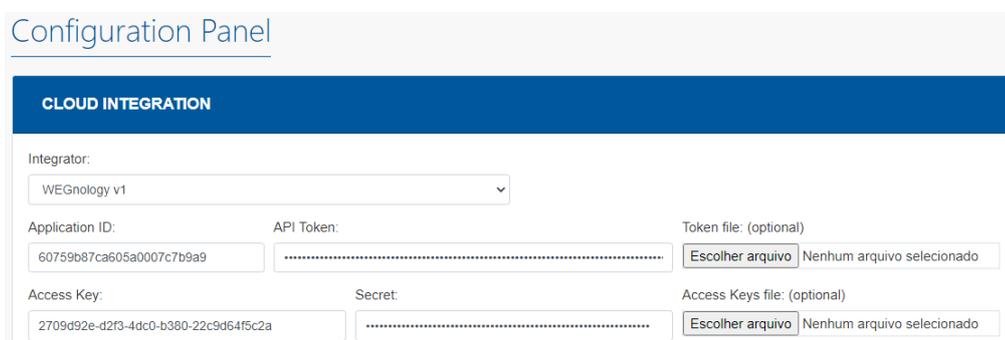
Fonte – Servidor local de configuração do ED300 (2021).

Após realizar o salvamento, o ED300 será reiniciado automaticamente e o “*status*” da conexão com a rede poderá ser observado no painel “*System Information*”, na página inicial da interface de comunicação, conforme a Figura 40.

5.6.3 Integração do ED300 com o *WEGnology*

Para realizar a integração do dispositivo ED300 com o *WEGnology*, o usuário deverá acessar novamente a guia de Configurações do dispositivo e configurar a aba denominada “*Cloud Integration*”, conforme mostrado na Figura 43.

Figura 43 – Integração *Cloud*.



The image shows a web interface titled "Configuration Panel" with a sub-section "CLOUD INTEGRATION". It contains several input fields and buttons:

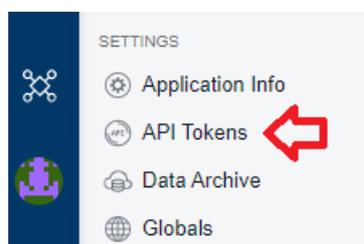
- Integrator:** A dropdown menu with "WEGnology v1" selected.
- Application ID:** A text input field containing "60759b87ca605a0007c7b9a9".
- API Token:** A text input field with a masked value ".....".
- Token file: (optional):** A button labeled "Escolher arquivo" and the text "Nenhum arquivo selecionado".
- Access Key:** A text input field containing "2709d92e-d2f3-4dc0-b380-22c9d64f5c2a".
- Secret:** A text input field with a masked value ".....".
- Access Keys file: (optional):** A button labeled "Escolher arquivo" and the text "Nenhum arquivo selecionado".

Fonte – Servidor local de configuração do ED300 (2021).

Para integração com a plataforma do *WEGnology*, é necessário selecionar o Integrador “*WEGnology v1*”. As informações de “*Application ID*”, “*API Token*”, “*Access Key*” e “*Secret*” serão geradas dentro do *WEGnology*.

Para gerar a “*Application ID*” e seu respectivo “*API Token*”, o usuário deverá acessar o ambiente de “*API Tokens*” na guia de “*Settings*” do *WEGnology* (Figura 44).

Figura 44 – Gerando uma *API Token*.



Fonte – *WEGnology* (2021).

Em seguida, é necessário clicar no botão “*Add Application Token*”, no canto superior direito da tela (Figura 45).

Figura 45 – Adicionando uma API Token.



Fonte – WEGnology (2021).

Finalmente, como último passo, é necessário configurar os parâmetros do novo *token API*, como seu nome, data de expiração (caso houver), e o tipo de permissão. Na realização do projeto, a data de expiração foi definida como “*Never Expires*”, ou seja, nunca irá expirar. A permissão concedida ao *token* foi total, através da opção “*All Permissions*”, que possibilita que o *token* tenha capacidade de ler e escrever informações nas aplicações em que for empregado. Após configurado, deve-se clicar no botão “*Create Application Token*” (Figura 46).

Figura 46 – Configuração da API Token.

Name

Description

The following values cannot be changed after token creation. If you need to make edits, you will have to create a new application token.

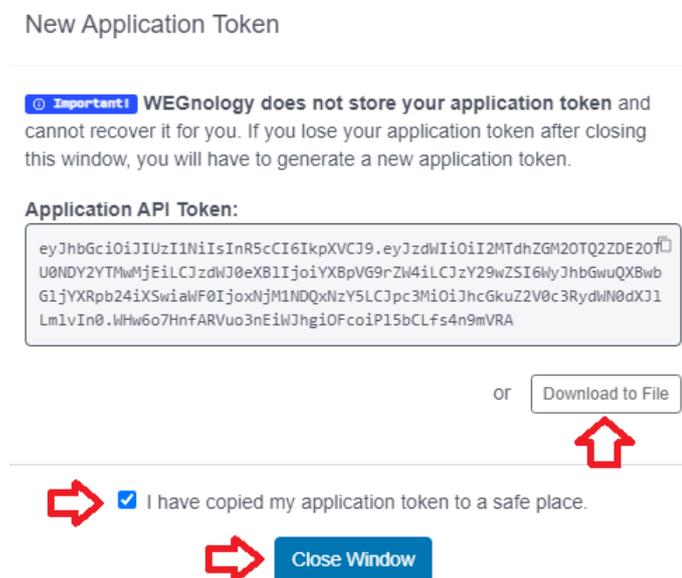
Token Expiration Date
 Never expires
 Expires at ...

Token Permissions
 All permissions
This token has read/write access to all application resources (including the application itself) as well as any additional resources that are added in the future.
 Read only
This token may read any application resource (including the application itself and device state data) as well as any additional resources that are added in the future.
 CLI developer permissions
This token provides read/write access to all application resources managed by the WEGnology CLI. Such a token can be provided to a web developer who has been tasked with building your application experiences.
 Custom ...
This token only has access to the resources and actions selected below. Resources and actions added in the future cannot be accessed using this token.

Fonte – WEGnology (2021).

Uma janela irá aparecer onde o usuário deverá realizar o *download* do arquivo contendo as informações relativas ao “API Token”, conforme ilustrado na Figura 47.

Figura 47 – Download da API Token.

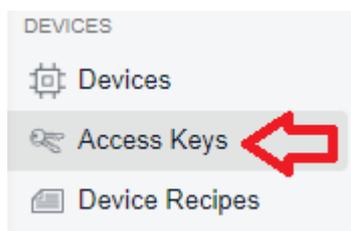


Fonte – WEGnology (2021).

O arquivo baixado terá o formato “.txt” e conterá as informações de “Application ID” e “Application Token”, que posteriormente deverão ser introduzidas na interface de comunicação do dispositivo ED300.

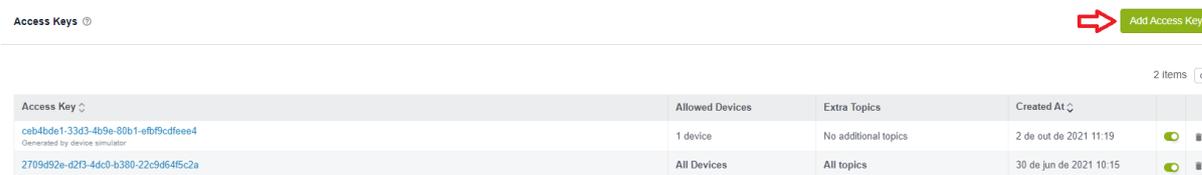
O processo para gerar a “Access Key” e o “Secret” é semelhante ao que foi visto acima. O usuário deverá acessar a aba “Access Keys” na guia “Devices” do WEGnology.

Figura 48 – Aba de Access Key.



Fonte – WEGnology (2021).

Após selecionar o botão “Add Access Key”, irá aparecer a tela ilustrada na Figura 49.

Figura 49 – Botão *Add Access Key*.

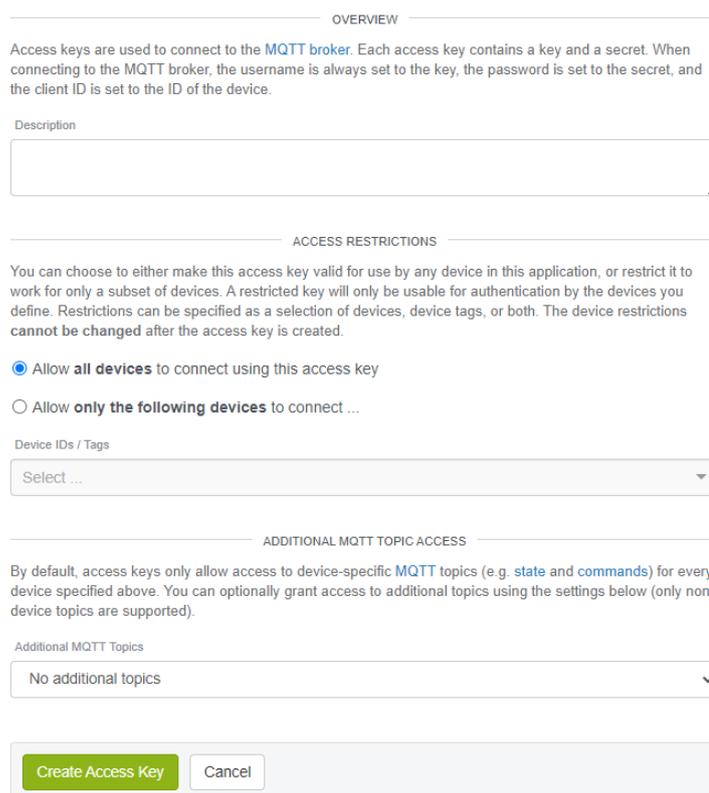
Access Keys ⊙ ➡ Add Access Key

2 items ▾

Access Key ▾	Allowed Devices	Extra Topics	Created At ▾	
ceb4bde1-33d3-4b9e-80b1-efb9cdfcee4 <small>Generated by device simulator</small>	1 device	No additional topics	2 de out de 2021 11:19	🟢 ▮
2709d92e-d2f3-4dc0-b380-22c5d645c2a	All Devices	All topics	30 de jun de 2021 10:15	🟢 ▮

Fonte – *WEGnology* (2021).

Neste caso, não será necessário efetuar alguma configuração específica da *Access Key* para a aplicação desenvolvida. O usuário poderá apenas clicar no botão *Create Access Key* no final da página, conforme ilustrado na Figura 50.

Figura 50 – Configuração da *Access Key*.

OVERVIEW

Access keys are used to connect to the **MQTT broker**. Each access key contains a key and a secret. When connecting to the MQTT broker, the username is always set to the key, the password is set to the secret, and the client ID is set to the ID of the device.

Description

ACCESS RESTRICTIONS

You can choose to either make this access key valid for use by any device in this application, or restrict it to work for only a subset of devices. A restricted key will only be usable for authentication by the devices you define. Restrictions can be specified as a selection of devices, device tags, or both. The device restrictions **cannot be changed** after the access key is created.

Allow **all devices** to connect using this access key

Allow **only the following devices** to connect ...

Device IDs / Tags

Select ... ▾

ADDITIONAL MQTT TOPIC ACCESS

By default, access keys only allow access to device-specific **MQTT topics** (e.g. **state** and **commands**) for every device specified above. You can optionally grant access to additional topics using the settings below (only non-device topics are supported).

Additional MQTT Topics

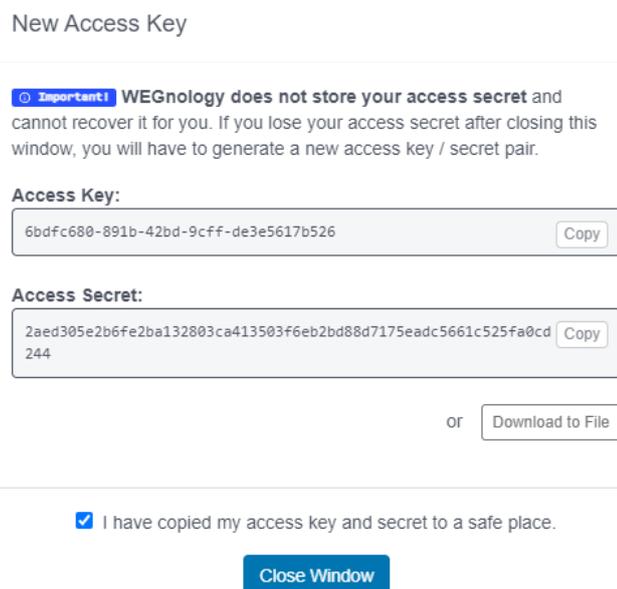
No additional topics ▾

Create Access KeyCancel

Fonte – *WEGnology* (2021).

Uma janela irá aparecer com as informações da *Access Key* e do *Access Secret*, o usuário poderá copiar e colar os códigos diretamente na interface de comunicação do ED300 ou poderá baixar o arquivo “.txt” com as mesmas informações (Figura 51).

Figura 51 – Gerando a Access Key.



Fonte – WEGnology (2021).

Uma vez que o usuário tenha gerado o “API Token” e a “Access Key”, ele deverá incluir as informações nos respectivos campos conforme visto na Figura 43. Por fim, deverá salvar as novas configurações através do botão de salvamento no final da página⁷.

Caso a integração entre o dispositivo e o WEGnology seja feita corretamente, o dispositivo deverá aparecer automaticamente na lista de dispositivos dentro da plataforma (Figura 52).

Figura 52 – Lista de Dispositivos no WEGnology.

Connection Status	Name	Device Class	Last Updated
Disconnected since 4 de out de 2021 14:45:46	Gateway - Node-RED ID: 612a45ec3995f5000780bec1	Gateway	2 de out de 2021 13:08
Disconnected since 28 de out de 2021 14:36:08	WCD-ED300-27-E4FC (4.0) ID: 60dc983d77a3c00079ab849	Edge Compute	10 de ago de 2021 08:08
Connected since 27 de out de 2021 19:14:56	WCD-ED300-2A-FE-B4 (07) ID: 60dc968ba3c900069d9bae	Edge Compute	13 de out de 2021 09:31
Connected since 28 de out de 2021 10:56:15	WCD-ED300-2A-FF-F2 (13) ID: 606dbad1d77a3c00079ade82	Edge Compute	1 de out de 2021 14:00

Fonte – WEGnology (2021).

⁷ As figuras possuem chaves com valores diferentes entre si pois foram elaboradas em momentos diferentes. Para o devido funcionamento da integração, as chaves geradas pelo usuário deverão ser as mesmas utilizadas na configuração da interface do ED300.

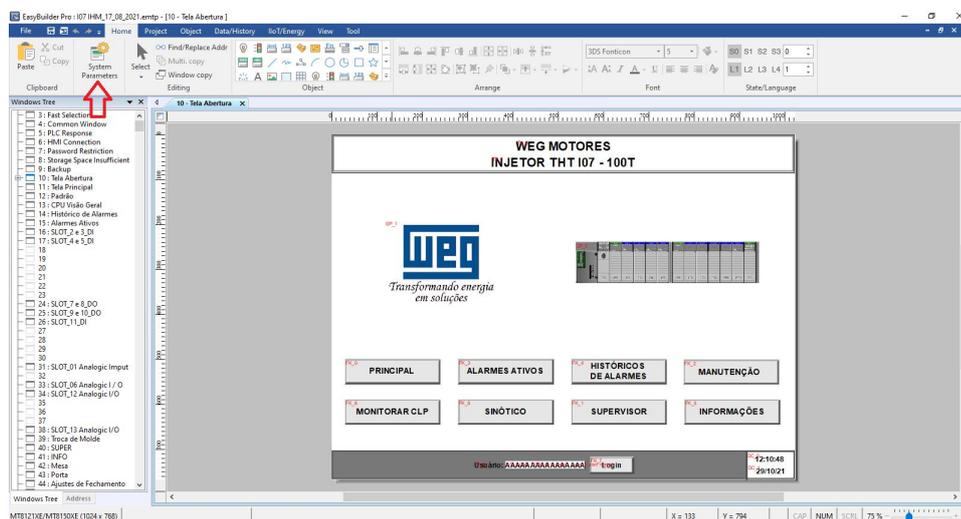
5.7 CONEXÃO DO DISPOSITIVO GATEWAY COM A IHM DA INJETORA

Nesta seção é exposto o procedimento utilizado para realizar a comunicação do dispositivo *ED300* com a IHM da máquina injetora de alumínio, através da configuração de um servidor *Modbus* com comunicação *Ethernet* via protocolo *TCP/IP* no *software* da IHM.

A configuração do servidor *Modbus* no *software* da IHM é realizada através do programa de criação de interfaces gráficas *Easybuilder*, e executando os passos descritos a seguir.

O arquivo do *Easybuilder* referente ao *software* presente na IHM da injetora do CT-i07 possui a seguinte tela inicial (Figura 53).

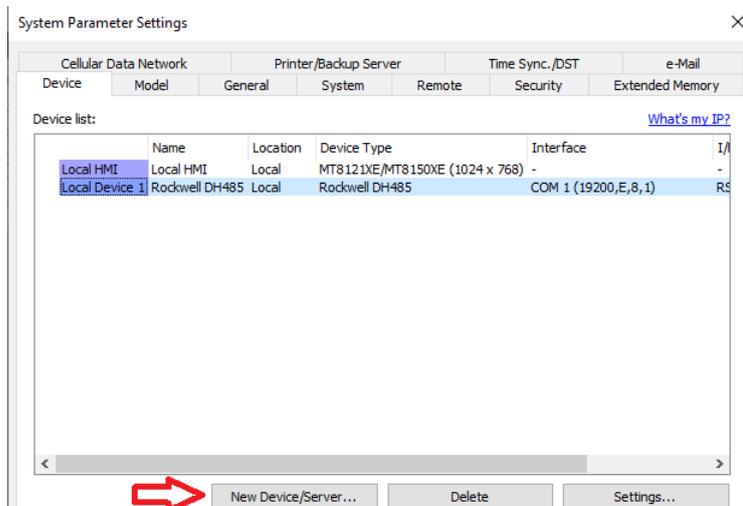
Figura 53 – Tela Inicial da Interface da IHM.



Fonte – Software *Easybuilder* (2021).

Selecionando o botão “*System Parameters*”, conforme indicado na Figura 53, deve ser feito a inclusão de um novo servidor através do botão “*New Device/Server*”, conforme Figura 54.

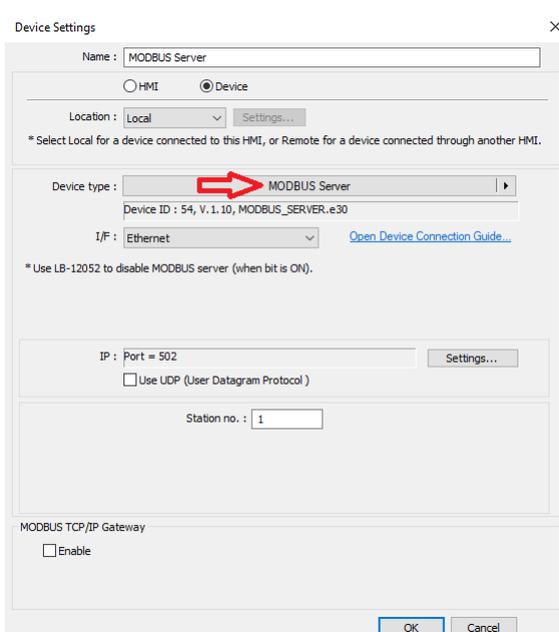
Figura 54 – Criação de Servidor Modbus.



Fonte – Software Easybuilder (2021).

Na próxima tela, deve ser selecionado o tipo de dispositivo/servidor (“Device type” como “Modbus Server”. É possível dar um nome ao novo servidor, no caso do exemplo, foi dado o nome “MODBUS Server”. As demais configurações serão mantidas como padrão e ao final deve-se pressionar o botão “OK” para salvar (Figura 55).

Figura 55 – Configuração do Servidor Modbus.



Fonte – Software Easybuilder (2021).

Após devidamente configurado, o novo servidor deverá aparece na lista de

dispositivos (Figura 56).

Figura 56 – Lista de *Devices* do Sistema.

Device list: [What's my IP?](#)

	Name	Location	Device Type	Interface	I/F Pr
Local HMI	Local HMI	Local	MT8121XE/MT8150XE (1024 x 768)	-	-
Local Device 1	Rockwell DH485	Local	Rockwell DH485	COM 1 (19200,E,8,1)	RS485
Local Server	MODBUS Server	Local	MODBUS Server	Ethernet (IP=Local, Port=502)	TCP/IP

Fonte – *Software Easybuilder* (2021).

Desta forma, o *software* da IHM estará configurado para receber a conexão do dispositivo *Gateway ED300*.

A conexão física entre a IHM da injetora e o dispositivo *Gateway ED300* é realizada por cabo *Ethernet*, através da porta *Ethernet* vista na figura 38 com a porta *Ethernet* disponível na parte traseira da IHM modelo MT8150XE.

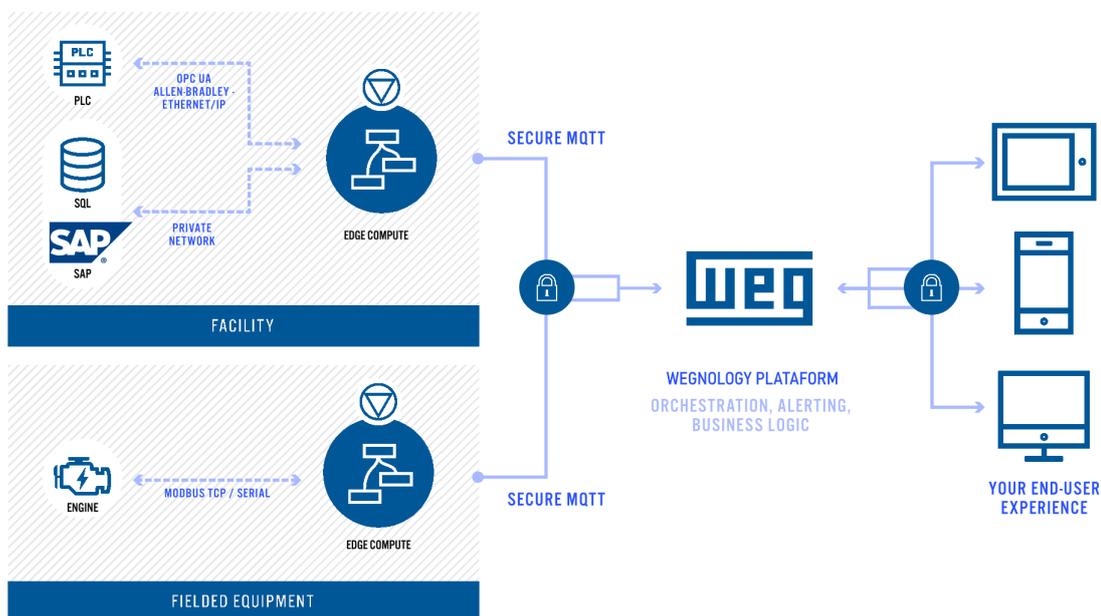
5.8 DESENVOLVIMENTO DA LÓGICA DO GATEWAY

Nesta seção é detalhado como foi feito o desenvolvimento da lógica responsável por solicitar as informações das variáveis de processo que foram alocadas na memória da IHM da injetora, conforme visto da Seção 5.5. A lógica implantada irá rodar de forma local do dispositivo *Gateway ED300*, enviando, numa determinada frequência, os dados para a nuvem via protocolo *MQTT* no formato de *payloads*.

Inicialmente, é importante considerar quanto ao tipo de *workflow* que deve ser utilizado afim de que este possa ser implantando de forma local num “*dispositivo edge*”. Neste caso, o tipo de *workflow* utilizado foi o “*Edge Workflow*”, que tem como característica o fato de que são executados localmente no dispositivo através do “*WEGnology Edge Agent*” ou “*WEGnology Edge Compute*”, que basicamente permite a implantação de *workflows* de forma local em dispositivos conectados às aplicações no *WEGnology*.

A Figura 57 ilustra um exemplo de arquitetura de rede envolvendo alguns dispositivos e ferramentas utilizadas na indústria, conectados a um dispositivo (podendo ser um ED300) responsável pelo gerenciamento e processamento de dados (“*edge computing*”), que irá se comunicar via protocolo *MQTT* com a plataforma do *WEGnology* e, por último, com os dados podendo ser exibidos dentro de uma aplicação para usuário final (“*end-user experience*”).

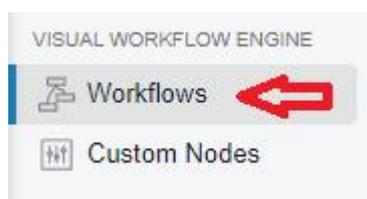
Figura 57 – Arquitetura de Rede com Dispositivo Edge.



Fonte – Documentação do WEGnology (2021).

Com o tipo definido, para criar um *edge workflow* dentro do WEGnology é necessário acessar a guia “*Visual Workflow Engine*”, na aba “*Workflows*”, conforme a Figura 58.

Figura 58 – Criação do *Edge Workflow*.



Fonte – WEGnology (2021).

Na janela que se abrirá, deve-se clicar no botão “*Add Workflow*” na parte superior direita da tela, abrindo a página para configuração de alguns parâmetros do novo *workflow*, conforme ilustrado na Figura 59.

Figura 59 – Parâmetros do novo *Workflow*.

CREATE WORKFLOW

Workflows provide the business logic behind all parts of your IoT solution. This can include processing data, generating notifications, interfacing with your local equipment, or even backing your entire custom user experience.

Workflow Name
i07

Workflow Type
Edge

Description

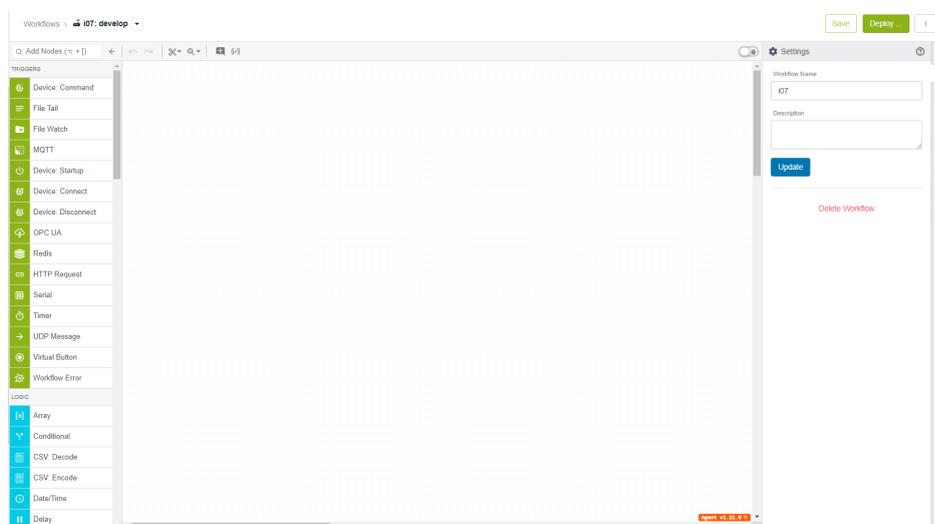
Minimum Agent Version (View Changelog)
Agent v1.21.0

Create Workflow Cancel

Fonte – WEGnology (2021).

No exemplo, foi atribuído o nome de “i07” ao *Workflow*, fazendo referência ao centro de trabalho da máquina injetora de alumínio onde o sistema *IoT* foi inicialmente implantado. Além no nome, também foi configurado o tipo do *workflow*, sendo do tipo “*edge*”. Por último, a versão do agente deve ser a mesma disponível do dispositivo ED-300, conforme pode ser observado na Figura 40, no painel “*Docker Information*”. Após configurado, deve-se clicar no botão “*Create Workflow*”.

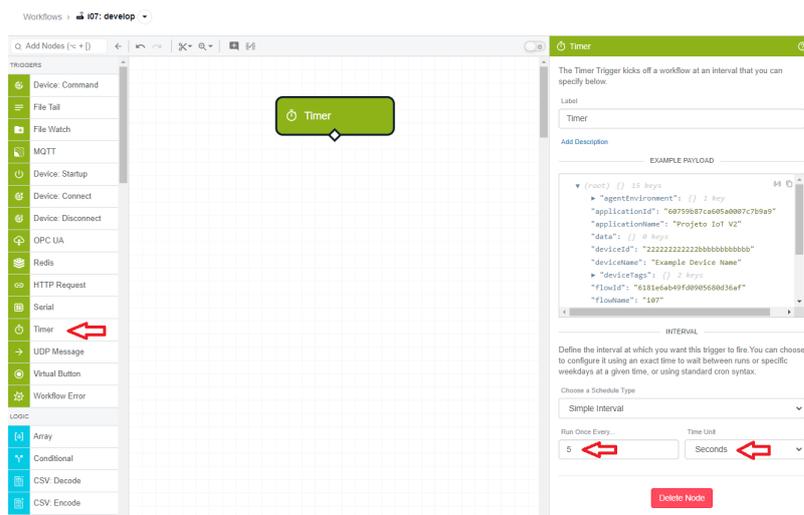
Uma vez criado o “*workflow*”, o usuário é direcionado ao ambiente de desenvolvimento referente à aquela aplicação, conforme ilustra a Figura 60.

Figura 60 – Ambiente de Desenvolvimento do *Workflow*.

Fonte – WEGnology (2021).

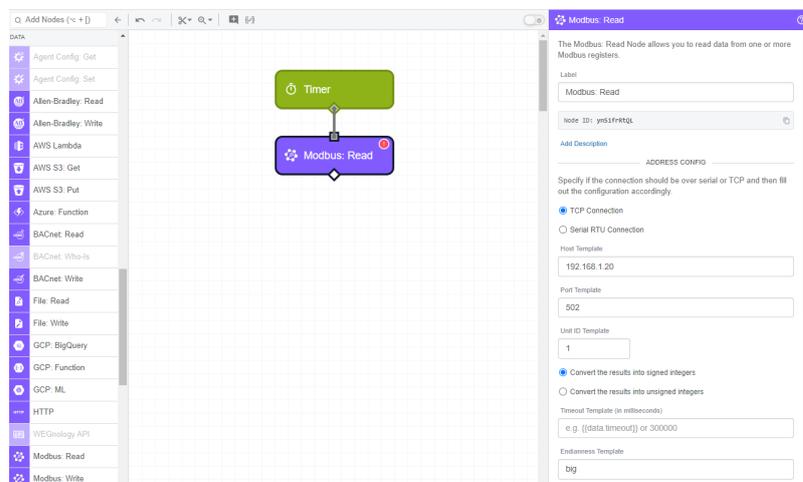
Inicialmente, deve ser colocado um bloco responsável por disparar o *workflow*, que neste caso será usado o bloco “*Timer*”, localizado dentro do subgrupo “*Triggers*” no painel de blocos à esquerda da tela. Uma vez adicionado, o bloco *Timer* deverá ser configurado quanto à taxa de execução, que neste caso foi definida como sendo de 1 disparo a cada 5 segundos (Figura 61).

Figura 61 – Bloco “*Timer*”.



Fonte – WEGnology (2021).

Depois, é necessário realizar a inclusão do bloco que será responsável pela leitura das variáveis de processo em seus respectivos endereços de memória na IHM da injetora de alumínio. O bloco em questão é o “*Modbus: Read*”, localizado no subgrupo “*Data*” do painel de blocos (Figura 62).

Figura 62 – Bloco *Modbus: Read*.

Fonte – WEGnology (2021).

No painel de configuração do bloco, à direita da tela, é necessário selecionar o tipo de conexão entre os dispositivos envolvidos (ED300 e IHM) que, neste caso, conforme descrito anteriormente na Seção 5.7, se dá através de uma conexão *Ethernet* e protocolo TCP/IP, desta forma, deve ser selecionado o tipo “*TCP Connection*”.

No campo “*Host Template*” deverá ser informado o IP do dispositivo “alvo” da comunicação, ou seja, da IHM. Por definição padrão, o IP da IHM utilizada no processo é “192.168.1.20”. Os campos “*Port Template*” e “*Unit ID Template*” devem ser preenchidos com os mesmos valores utilizados na configuração servidor *Modbus* da IHM, conforme visto na seção 5.7 e, mais especificamente, na Figura 55, no caso, “502” e “1”, respectivamente.

Seguindo com a configuração do bloco *Modbus: Read*, é necessário informar quanto às instruções para a realização da leitura de variáveis, como tipo de registro (“*Register Type*”), endereço de memória (“*Address Template*”), da quantidade de variáveis naquele endereço (“*Length Template*”) e finalmente atribuir um nome para a variável a ser lida (“*Result Key*”) (Figura 63).

Figura 63 – Configuração do Bloco *Modbus: Read*.

Fonte – WEGnology (2021).

Usando como exemplo a primeira variável de processo listada no mapeamento de VPs da Seção 5.2, ou seja, a variável “Pressão de Fechamento da Pinça” (Figura 17), o registro será do tipo “*Holding Registers*” (*Modbus FC = 03*), o endereço de memória será “500”, o campo “*Length Template*” será preenchido com valor “1” e nome da variável será “*PressaoFechamentoPinca*”, no campo “*Result Key*”.

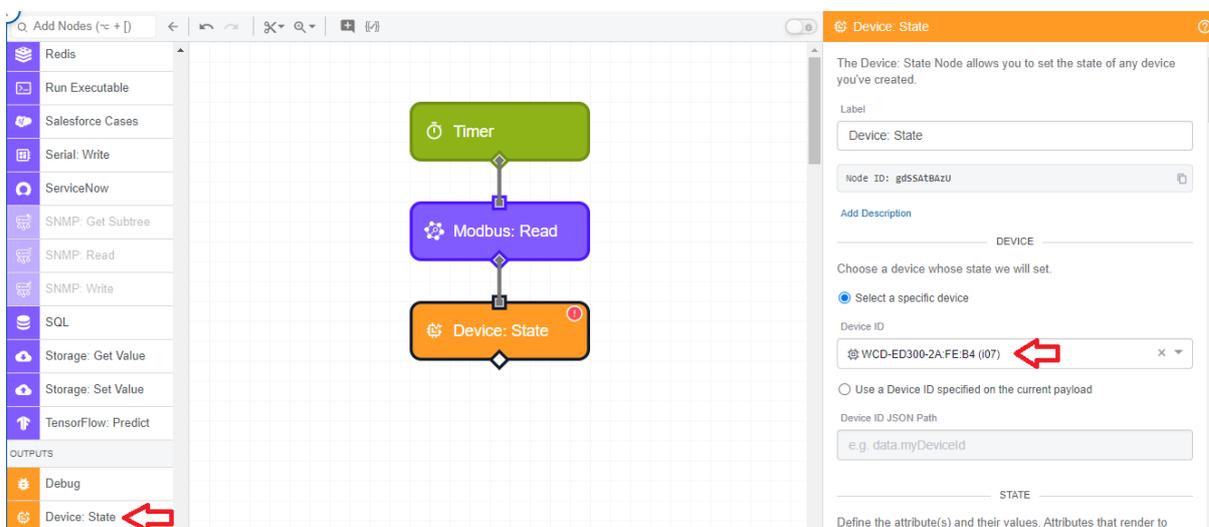
O processo descrito acima deve ser repetido para todas as variáveis de processo mapeadas na Seção 5.2, uma vez que todas as VPs deverão ser lidas.

Por último, é necessário definir um caminho dentro do *payload* onde ficarão registrados os resultados da leitura realizada pelo bloco *Modbus: Read*. Neste caso, o campo “*Result Path*” foi preenchido com o caminho “*results.modbusData*”.

O próximo bloco a ser inserido será o “*Device: State*”, que possibilita que o *workflow* atualize o estado de um dispositivo. Existem duas configurações que devem ser realizadas no bloco em questão, sendo a primeira delas a seleção do dispositivo em que o estado será atualizado, e a segunda configuração é com relação aos atributos e seus respectivos valores que serão atualizados com a execução do bloco.

No caso da aplicação em questão, deve ser selecionado o dispositivo ED300 previamente integrado à plataforma, conforme o procedimento visto na Seção 5.6.3 (Figura 64).

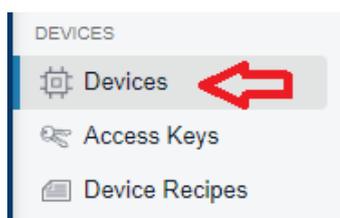
Figura 64 – Configuração do Bloco *Device: State*.



Fonte – WEGnology (2021).

Na sequência, deve ser feita a configuração dos atributos e seus respectivos valores para atualização de estado. Para configurar os atributos do bloco *Device: State*, estes devem ser primeiro criados dentro das configurações do dispositivo dentro do *WEGnology*. Para realizar este procedimento, deve-se acessar a guia de “*Devices*” no painel lateral esquerdo da plataforma *WEGnology*, conforme ilustra a Figura 65.

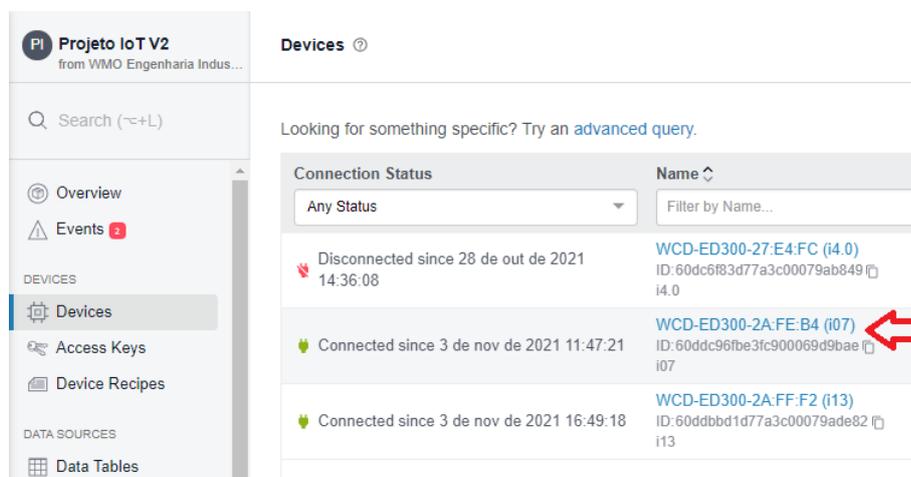
Figura 65 – Configuração de Atributos do *Device*.



Fonte – WEGnology (2021).

Uma lista com todos os dispositivos será exibida e deve ser selecionado o dispositivo em questão. No caso da aplicação descrita nesta etapa, o dispositivo em questão é o “WCD-ED300-2A:FE:B4 (i07)”, conforme indicado na Figura 66.

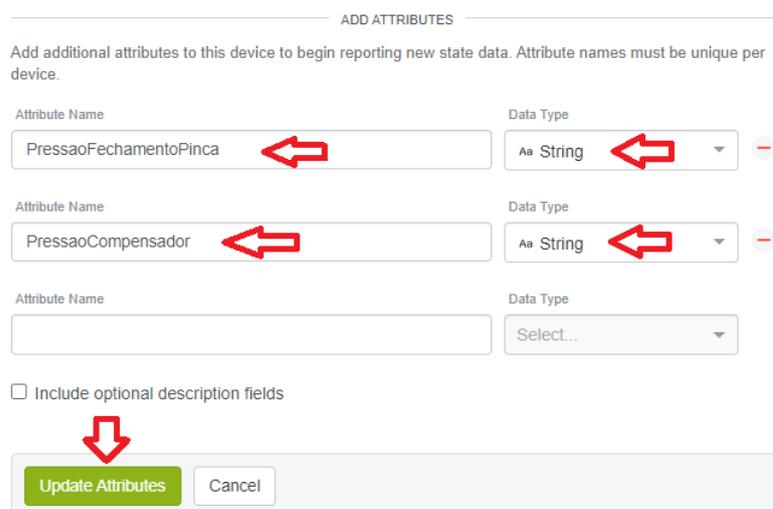
Figura 66 – Atributos do *Device*.



Fonte – WEGnology (2021).

Uma vez dentro do ambiente de configuração, na guia “Attributes” são listados todos os atributos relacionados à aquele dispositivo, além da possibilidade da inclusão de novos atributos. Deverão ser adicionados como novos atributos todas as variáveis de processo mapeadas na Seção 5.2, atribuindo um nome e selecionando o tipo de dado referente ao atributo. Nesta aplicação, os dados foram configurados como sendo do tipo “String”. Ao concluir a inclusão de todos atributos, deve ser feito a atualização das configurações através do botão “Update Attributes” (Figura 67).

Figura 67 – Configuração de Atributos do *Device*.

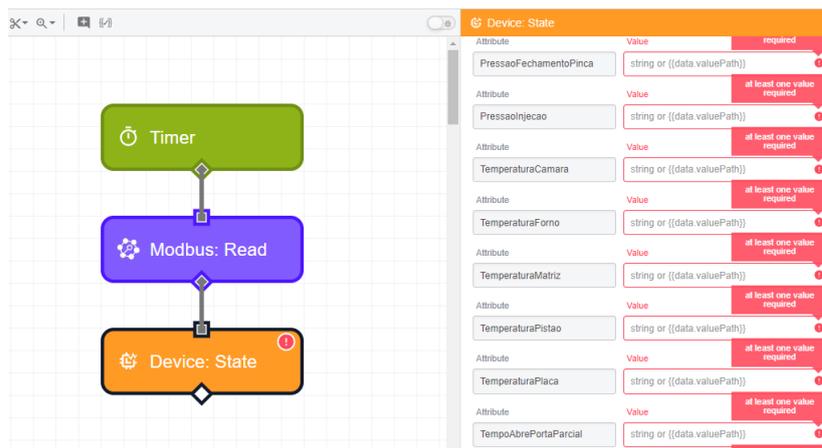


Fonte – WEGnology (2021).

Agora com os atributos (variáveis de processo) devidamente incluídos nas ca-

racterísticas do dispositivo, os mesmos deverão aparecer no contexto do bloco *Device: State*, ainda pendente das informações relativas aos valores que serão atribuídos aos atributos. O fato de ainda não possuímos a informação da localização das variáveis de processo no *payload* impossibilita o salvamento e a implantação do *workflow* no *Edge Device*, conforme ilustra a Figura 68.

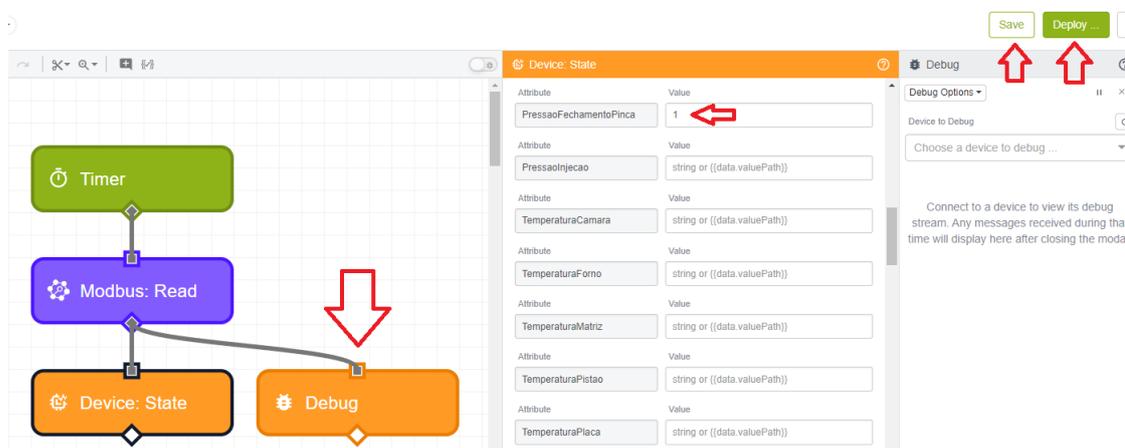
Figura 68 – Localização das VPs no *Payload*.



Fonte – WEGnology (2021).

Para contornar momentaneamente esta condição, atribuímos um valor fixo (por exemplo, “1”) a qualquer uma das variáveis, apenas para possibilitar o salvamento. Ao mesmo tempo, conectamos ao *workflow* um bloco “*Debug*” após o bloco *Modbus: Read* para localizar o endereço dos atributos dentro do *payload* do *workflow*, conforme demonstrado na Figura 69.

Figura 69 – Localizando os Endereços dos Atributos no *Payload*.



Fonte – WEGnology (2021).

Em seguida, deve ser feito o salvamento do *workflow* e então o “Deploy”. Uma janela para a configuração de algumas características se abrirá, onde deve ser feita a seleção do dispositivo no qual a versão do *workflow* será implantada. Neste caso, o dispositivo integrado deverá ser selecionado e então é necessário clicar no botão “Deploy Version” (Figura 70).

Figura 70 – Realizando o *Deploy* no *Edge Device*.

Deploy New Version

Deploying an edge workflow schedules a version of this workflow to be sent to one or more devices running the edge agent. If a selected device is connected now, it will immediately receive the update. Otherwise, the device will receive the new version when it connects again.

CREATE VERSION

Workflows must be versioned before being deployed to devices. Please enter the version details below to create a new version for this deployment.

Version Name

2021-11-04T00-38-15

Notes (optional)

SELECT DEVICES

Choose the devices, or families of devices, that will be scheduled to receive this workflow version.

Deploy To ...

WCD-ED300-2A:FE:B4 (i07)

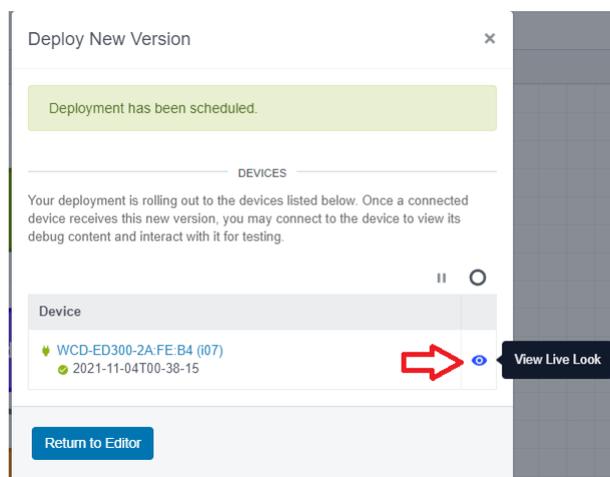
Preview 1 Device

Deploy Version Cancel

Fonte – WEGnology (2021).

Uma vez que a versão desenvolvida do *workflow* foi implantada no dispositivo, é possível visualizar o sistema em funcionamento, selecionando o ícone de “View Live Look”, conforme indicado na Figura 71.

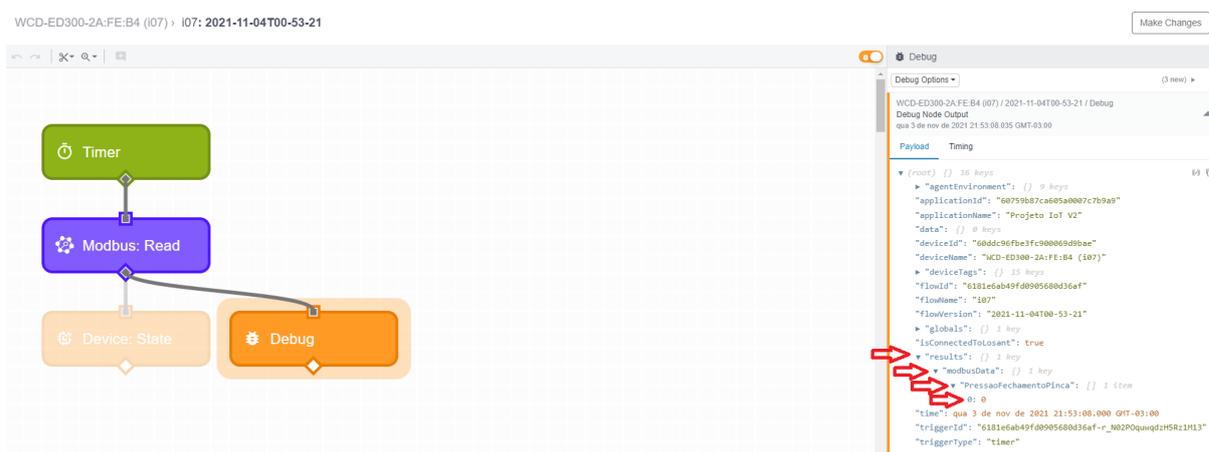
Figura 71 – Exibindo o *Live Look View* do *Workflow*.



Fonte – WEGnology (2021).

Neste ambiente é possível a visualizar a representação da execução do *workflow* implantando no dispositivo, além da visualização do *payload* através do bloco *Debug*. Também é possível visualizar o endereço dentro do *payload* da variável “*PressaoFechamentoPinca*”, conforme ilustrado na Figura 72.

Figura 72 – Endereço do Atributo no *Payload*.



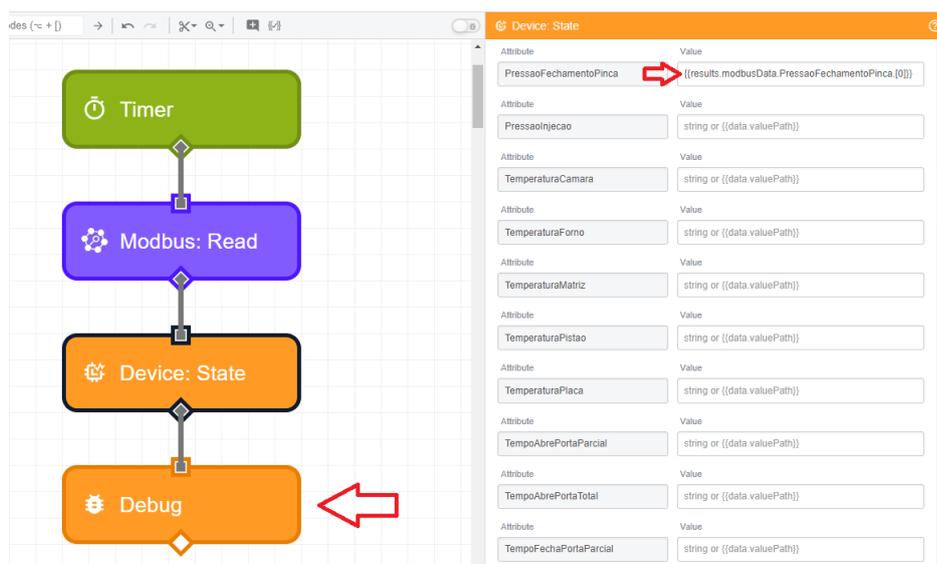
Fonte – WEGnology (2021).

Uma dica é clicar com o botão direito do *mouse* em cima do endereço no *Payload* para copiá-lo de forma mais prática, conforme Figura 73.

Figura 73 – Copiando o Endereço do Atributo no *Payload*.

Fonte – WEGnology (2021).

Uma vez que o endereço do atributo foi copiado, este deve ser colado entre duplas chaves⁸ (“{{Value}}”) no campo “Value” do atributo “PressaoFechamentoPinca” no bloco *Device: State*. Além disso, o bloco *Debug* deve ser posicionado após do bloco *Device: State* no *workflow*, conforme Figura 74.

Figura 74 – Copiando o Endereço do Atributo no *Payload*.

Fonte – WEGnology (2021).

Desta forma, o *workflow* deverá atribuir à variável “PressaoFechamentoPinca” o valor presente no caminho “{{results.modbusData.TemperaturaForno.[0]}}”. O procedimento descrito nesta seção deverá ser igualmente realizado para todas as variáveis de processo, afim de se incorporar seus respectivos valores a cada atualização de estado do dispositivo, ou seja, a cada execução do *workflow*.

Para fins de exemplificar a forma com que o dado chega até aqui, o procedimento descrito acima foi feito para a variável de processo “TemperaturaForno”, uma

⁸ As duplas chaves indicam “String Template”, utilizado para definir que o valor em questão se trata de um endereço de *payload*.

vez que esta variável deverá sempre trazer um valor diferente de zero.

Assim, a execução do *workflow* resultará no seguinte *payload*, onde o valor “7478” foi atribuído à variável “TemperaturaForno”, conforme Figura 75.

Figura 75 – Variável “TemperaturaForno”.

```
▼ "results": {} 1 key
  ▼ "modbusData": {} 1 key
    ▼ "TemperaturaForno": [] 1 item
      0: 7478 ←
```

Fonte – WEGnology (2021).

Apesar do valor “7478” estar correto, ele não está devidamente formatado, ou seja, a informação referente à casa decimal do número ainda precisa ser considerada. Na seção a seguir será abordado o tratamento das variáveis de processo recebidas via *payload*.

5.9 TRATAMENTO DAS VARIÁVEIS DE PROCESSO VIA WEGNOLOGY

Nesta seção é detalhado o processo de tratamento das variáveis que chegam via *payload*, que na maioria das vezes, ainda não representa o valor correto para aquele atributo. Usaremos o exemplo descrito na seção anterior com a variável “TemperaturaForno”, que foi passada com o valor “7478” no momento daquela leitura, porém ainda não representa o valor correto da temperatura do forno.

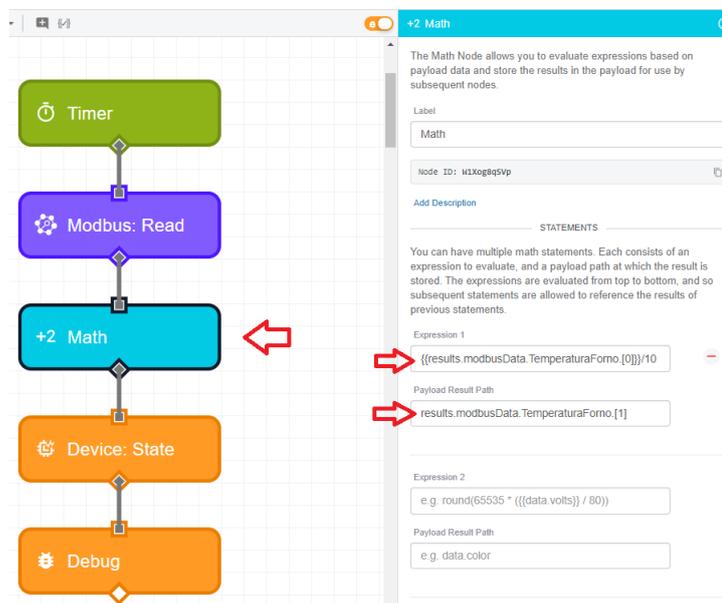
A informação da temperatura é multiplicada por 10 pelo CLP para manter a informação decimal do valor, sendo necessário então fazer a divisão por 10 para obter o valor correto. Essa operação é padrão para todas as variáveis que possuem alguma casa decimal, sendo necessário analisar cada caso. Variáveis que não possuem casas decimais, como por exemplo, “NumeroFerramenta”, não será necessário realizar qualquer operação.

Para realizar o tratamento de variáveis usaremos o bloco lógico “Math”⁹, que será responsável por efetuar a operação de divisão por 10 do valor encontrado no caminho da variável em questão.

O bloco *Math* deve ser configurado com a expressão matemática (*Expression*) que deverá ser realizada com algum determinado valor ou caminho de *payload* (*payload path*, e com um endereço no *payload* (*Payload Result Path*) onde o resultado da operação será armazenado. Seguindo com o exemplo da variável de temperatura do forno, a configuração do bloco *Math* se daria a seguinte forma, ilustrada na Figura 76.

⁹ Também é possível realizar operações via bloco “Function” ou até mesmo direto no campo de valor do atributo no bloco “Device: State”.

Figura 76 – Configuração do bloco *Math*.

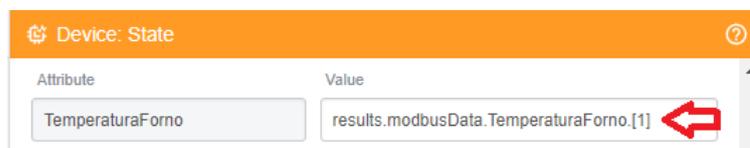


Fonte – WEGnology (2021).

O resultado da operação foi colocado em um endereço distinto para que possamos visualizar os dois valores de forma simultânea no *payload*. Neste caso, o novo endereço será “`{{results.modbusData.TemperaturaForno.[1]}}`”

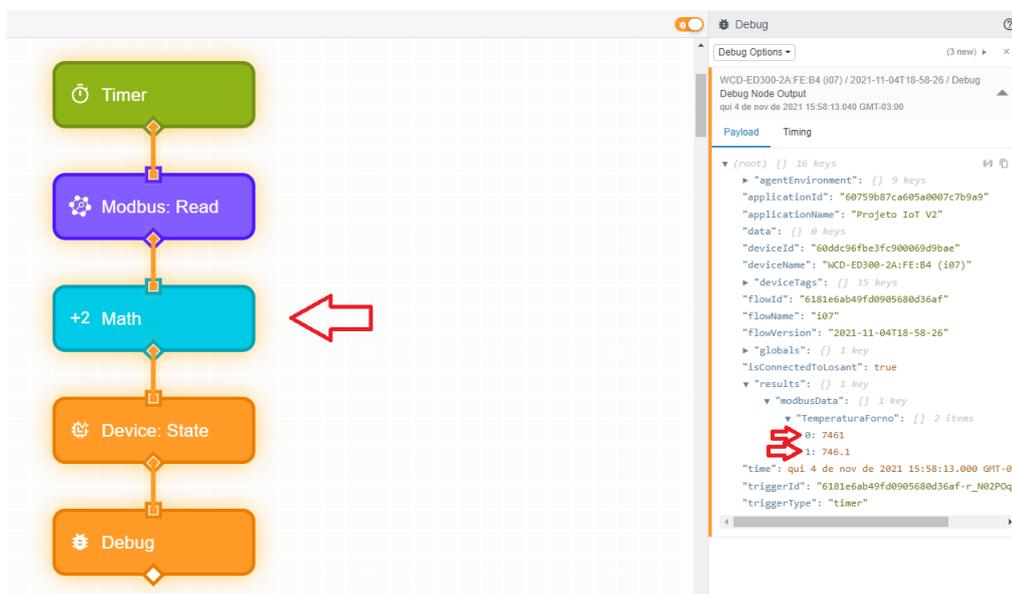
Como o valor após a operação é o correto para aquela variável, o endereço deve ser atualizado do campo do bloco *Device: State*, conforme ilustra a Figura 77.

Figura 77 – Atualização do *Payload Path* da VP no bloco *Device: State*.



Fonte – WEGnology (2021).

Deve ser realizado o salvamento e novo *deployment* do *workflow* para o *edge device*. Assim, a cada execução a lógica, o novo *payload* trará tanto a informação original referente à variável “TemperaturaForno” quanto o valor do bloco *Math*, que representa o valor correto (Figura 78).

Figura 78 – Atualização do *Payload Path* da VP no bloco *Device: State*.

Fonte – WEGnology (2021).

O bloco *Math* permite que várias expressões sejam incluídas, possibilitando que todas as variáveis de processo sejam tratadas diretamente por ele. Desta forma, é necessário realizar o mesmo procedimento descrito nesta seção para todos os atributos do bloco *Device: State*. A Figura 79 ilustra um *payload* com a informação de diversas variáveis de processo já tratadas.

Figura 79 – *Payload* Completo do Processo.

```
Debug
Debug Options (23 new)
  "results": {} 1 key
    "modbusData": {} 30 keys
      "ContadorPeca": 4
      "ContadorTiro": 65
      "NumeroFerramenta": 18025
      "PressaoAcumulador": 150.8
      "PressaoCompensador": 0.2
      "PressaoFechamentoPinca": 0
      "PressaoInjecao": 0.9
      "TemperaturaCamara": 20.1
      "TemperaturaForno": 746.5
      "TemperaturaMatriz": 28.1
      "TemperaturaPistao": 29.7
      "TemperaturaPlaca": 21.5
      "TempoAbrePortaParcial": 1.29
      "TempoAbrePortaTotal": 1.74
      "TempoFechaPortaParcial": 1.69
      "TempoFechaPortaTotal": 0
      "TempoInjecao": 2.21
      "TempoResfriamentoInjecao": 0
      "TempoValvulaInjecaoReal": 0
      "TempoValvulaInjecaoSet": 8
      "VazaoCamara": 12.5
      "VazaoMatriz": 9
      "VazaoPistao": 14.5
      "VazaoPlaca": 9.1
      "VelocidadeInjecao": 10
      "massaloteAltura": 0
      "massaloteMax": 15.5
      "massaloteMin": 11.5
      "massaloteReal": 14
      "massaloteTotal": 532.3
    "time": qui 4 de nov de 2021 16:24:11.000 GMT
```

Fonte – WEGnology (2021).

5.10 CRIAÇÃO DE *DASHBOARD* PARA VISUALIZAÇÃO DOS DADOS

Nesta seção é detalhado o processo de criação de um *Dashboard* para visualização das informações relacionadas ao processo de Injeção de Alumínio de Rotores vistas até agora.

5.10.1 Básico sobre *Dashboards* no WEGnology

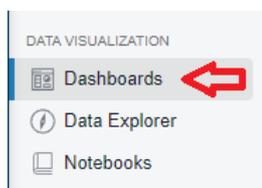
Dashboards proporcionam uma visualização mais fácil e intuitiva de dados relevantes de um determinado processo, sendo úteis para o monitoramento de mudanças de comportamento e padrões nos dados com o passar do tempo, e até mesmo na detecção de anomalias.

No *WEGnology*, o processo de criação de *Dashboards* é facilitado pela existência de blocos “embutidos” à ferramenta, seguindo o mesmo formato dos blocos lógicos utilizados nos *workflows*, sendo necessário apenas realizar algum tipo de configuração. Isso faz com que não seja necessário o conhecimento e trabalho envolvido no processo de desenvolvimento de uma interface *WEB*, o que envolveria diversas linguagens de programação para tal.

5.10.2 Criação do *Dashboard* para o CT-i07

Para realizar a criação de um *Dashboard* no *WEGnology*, o usuário deve acessar a aba de “*Dashboards*” na guia de “*Data Visualization*”, no painel de navegação da plataforma localizado na lateral esquerda da tela, conforme ilustrado na Figura 80.

Figura 80 – Inclusão de Novo *Dashboard* no *WEGnology*



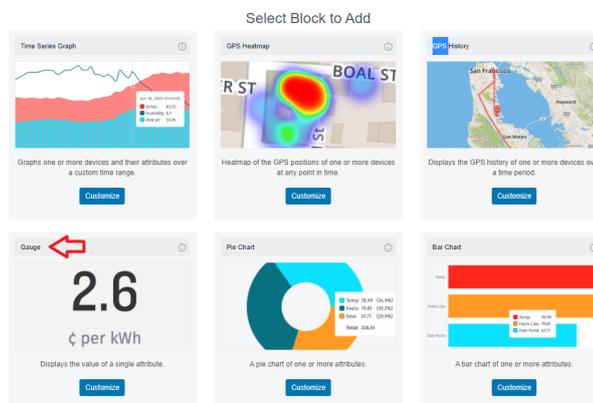
Fonte – *WEGnology* (2021).

Deve-se então clicar no botão “*Add Dashboard*”, localizado na parte superior direita da tela. dar um nome ao novo *Dashboard*, e clicar no botão “*Create Dashboard*”. Como ainda não há blocos adicionados ao *Dashboard*, o usuário será direcionado automaticamente para a página de seleção de blocos.

Neste documento será descrito o processo de configuração do bloco para visualização da informação da vazão e da temperatura da placa de alimentação, o processo é semelhante para os demais blocos e informações a serem exibidas no *dashboard*.

Para a variável de vazão, foi selecionado o bloco do tipo “*Gauge*”, que exibe a informação de um único atributo, seja na forma numérica ou uma representação visual em um mostrador (Figura 81).

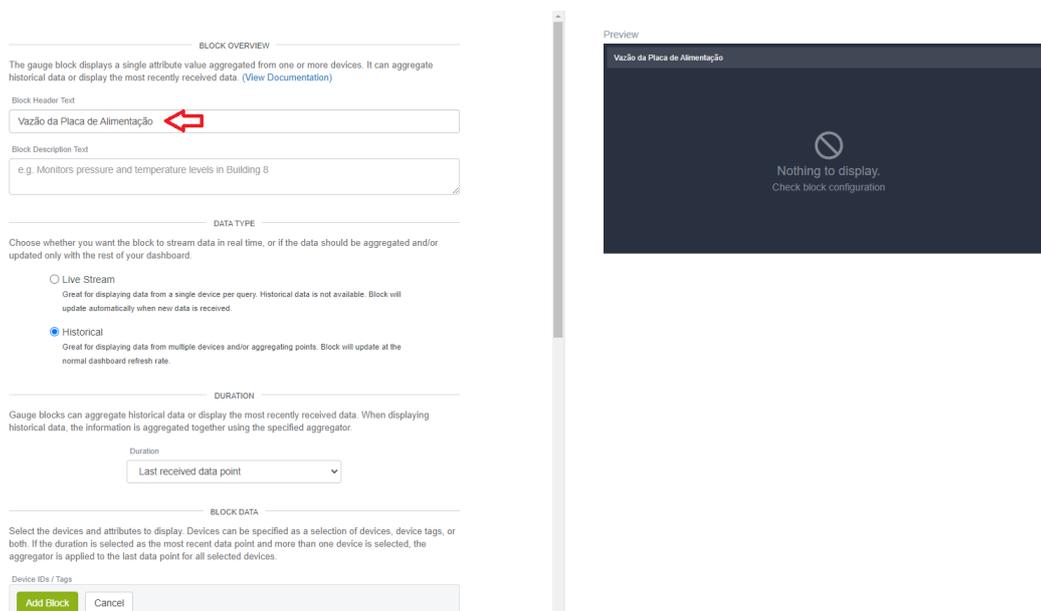
Figura 81 – Inclusão de Novo Dashboard no WEGnology.



Fonte – WEGnology (2021).

Nas configurações do bloco, no campo “Block Header Text” foi incluída uma descrição para o bloco, neste caso, “Vazão da Placa de Alimentação”. É possível acompanhar a construção do bloco conforme são incluídas as configurações por meio de uma imagem prévia (“Preview”), mostrada na parte direita da tela, conforme ilustrado na Figura 82.

Figura 82 – Configuração do Bloco de Vazão no Dashboard.

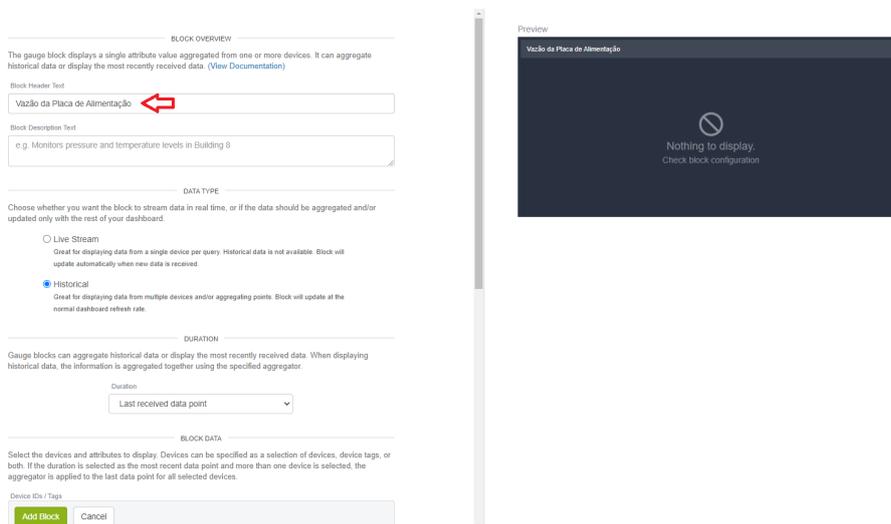


Fonte – WEGnology (2021).

Seguindo com a configuração, no painel “Block Data”, deve ser selecionado o dispositivo e atributo que deverá ser exibido neste bloco. Neste exemplo, o dispositivo

em questão é o *ED300* instalado na injetora do CT-i07, o atributo é o “VazaoPlaca” e o tipo de agregação como “*Last*”, que mostra o último valor atualizado no *device* (Figura 83).

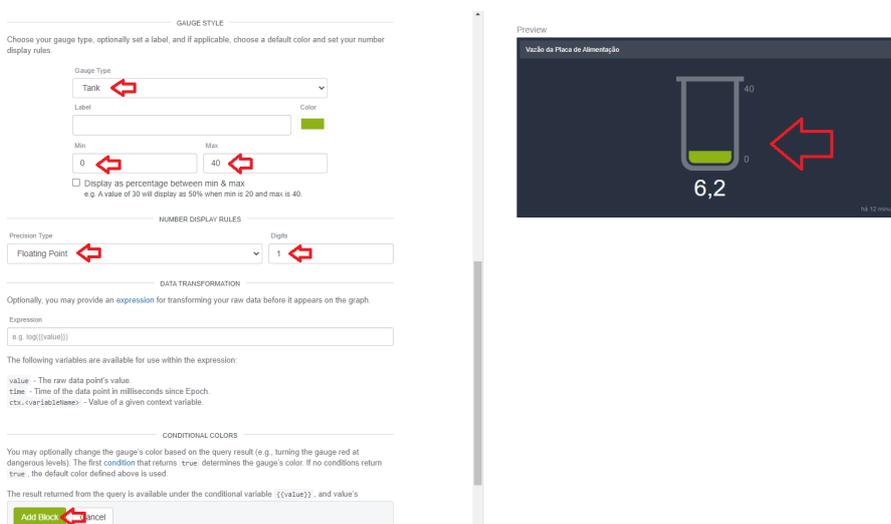
Figura 83 – Seleção do Dispositivo do Bloco.



Fonte – WEGnology (2021).

Nas configurações relacionadas ao estilo visual do bloco (“Gauge Style”), foi escolhido um elemento do tipo “Tank” para representar o fluxo de vazão da placa de alimentação da injetora, com intervalos de exibição de 0 a 40, conforme o intervalo de medição do sensor (Figura 84).

Figura 84 – Configuração do Estilo do Bloco.

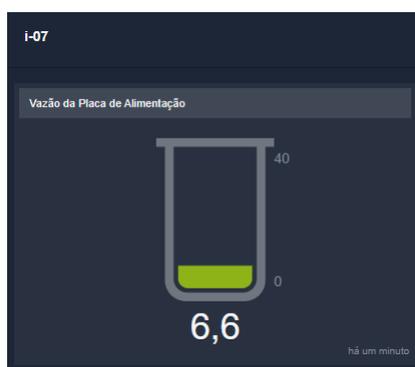


Fonte – WEGnology (2021).

Quanto às regras de exibição dos valores (“Number Display Rules”), o tipo de precisão foi definido como “Floating Point” e o número de dígitos igual a “1”. Isso significa que será exibido sempre um número com previsão de uma casa decimal. Quanto as demais possibilidades de configuração, não se fizeram necessárias para este bloco, logo, o bloco já pode ser adicionado ao *Dashboard*.

Uma vez adicionado ao *Dashboard*, o bloco deverá ser exibido conforme ilustra a Figura 85.

Figura 85 – Visualização do Bloco no *Dashboard*.

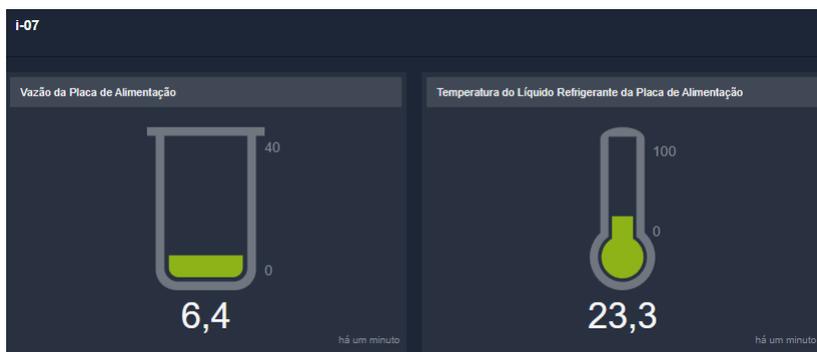


Fonte – WEGnology (2021).

O processo para inclusão do bloco para visualização da variável de temperatura do líquido refrigerante que passa pela placa de alimentação da máquina injetora de alumínio é bastante semelhante ao que foi descrito para a Vazão, utilizando do mesmo tipo de bloco (*Gauge*), selecionando o atributo “TemperaturaPlaca” e o tipo de *Gauge* como “*Thermometer*”, para uma representação mais intuitiva da variável de temperatura.

Após a inclusão do novo bloco, o *Dashboard* ficou conforme ilustrado na Figura 86.

Figura 86 – Inclusão do Bloco de Temperatura no *Dashboard*.

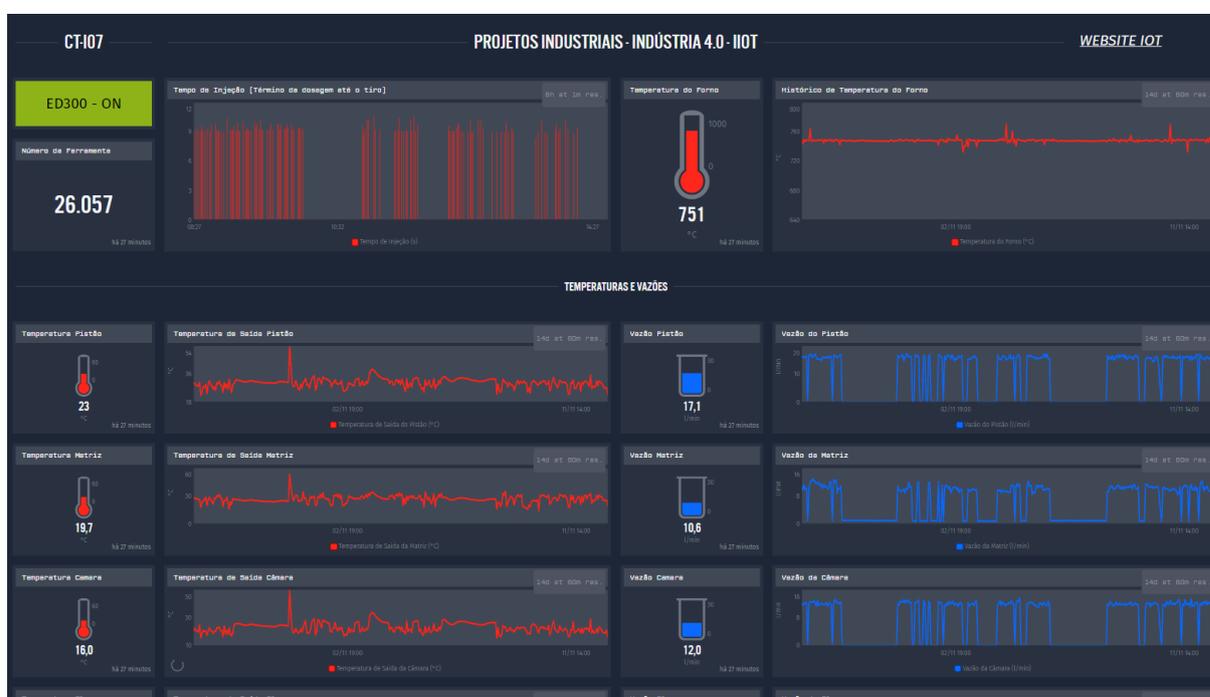


Fonte – WEGnology (2021).

O processo de inclusão dos blocos para visualização de todas as demais variáveis do processo de Injeção de Alumínio seguiram um passo a passo semelhante ao que foi descrito acima, portanto, não serão detalhados individualmente.

A Figura 87 ilustra parte do modelo do *Dashboard* até o momento da elaboração deste documento.

Figura 87 – Modelo Final do *Dashboard*.



Fonte – WEGnology (2021).

5.11 EXPANSÃO DO SISTEMA *IoT* PARA O CT-I13

Nesta seção é detalhado o trabalho de expansão realizado para o CT-i13, localizado ao lado do CT-i07, do sistema *IoT*. A possibilidade desta expansão se deu pelo fato de que o trabalho realizado anteriormente foi bem recebido pelos chefes e gerente do departamento de Injeção de Alumínio.

A lógica desenvolvida para o dispositivo *Gateway ED300* do CT-i07 foi aproveitada neste novo centro. No entanto, conforme é demonstrado a seguir, algumas adaptações tiveram de ser realizadas para atender às particularidades da nova arquitetura do sistema.

Aqui vale constatar que, para o novo centro, foi utilizado um novo dispositivo *Gateway ED300*, desta forma, todo o procedimento de configuração descrito até agora precisou ser realizado também no novo aparelho, conforme descrito nas seções deste capítulo.

5.11.1 Das Peculiaridades do Novo Centro de Trabalho - CT-i13

Após autorizada a primeira expansão do Sistema *IoT* desenvolvido e implementado no CT-i07 para o CT-i13, constatou-se uma limitação por parte do *CLP* da Injetora daquele novo centro quanto à disponibilidade de entradas analógicas, que não existiam, impossibilitando a conexão dos sinais provenientes dos sensores de vazão e temperatura. O modelo do *CLP* do CT-i13 é o mesmo descrito anteriormente para o CT-i07, na Seção 5.4.

Desta forma, foi necessário repensar na arquitetura do sistema para que estes sinais não fossem perdidos, uma vez que são importantes e não deveriam ser desconsiderados. A solução encontrada foi a inclusão de um segundo *CLP*, o CLIC-02, de marca *WEG*, para que os sinais dos sensores pudessem ser conectados neste dispositivo (Figura 88).

Figura 88 – CLP CLIC-02 com Módulo de Expansão (ilustrativo)



Fonte – Website da WEG (2021).

5.11.2 Da Inclusão do CLP CLIC-02 à Arquitetura do Sistema

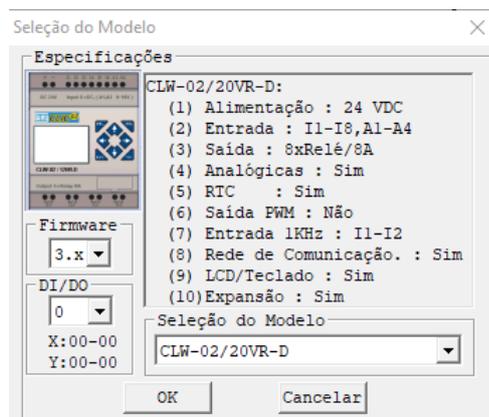
O módulo principal do CLP CLIC-02 adquirido para incorporação na arquitetura dispunha de 4 entradas analógicas para sinal de tensão, desta forma, não seria compatível com os sinais de corrente entregues pelos sensores, além do fato de que, no total, seriam 8 sinais de corrente.

Assim, foi necessário converter os sinais de corrente emitidos pelos sensores para valores de tensão, para que então pudessem ser lidos pelo CLIC02. Além disso, também foi necessário realizar a compra de um módulo de expansão analógico para que todos os 8 sinais dos sensores fossem considerados.

5.11.2.1 Da Configuração do CLP CLIC-02 com Módulo de Expansão Analógico

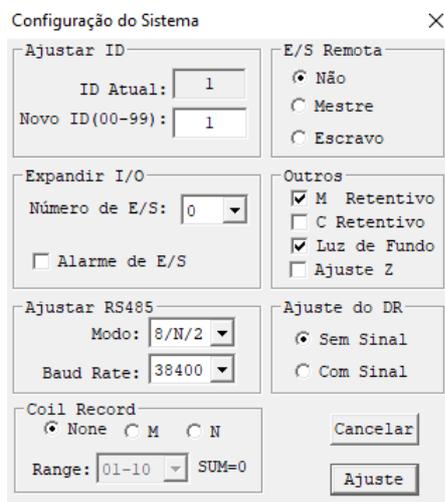
O CLP CLIC-02 possui um software de configuração e programação chamado CLIC-02 Edit, onde iremos detalhar as primeiras configurações do dispositivo.

Inicialmente, já com o software CLIC-02 Edit em execução, deve-se selecionar no programa o modelo do CLP utilizado, que neste caso é o “20VR-D”, conforme ilustrado na Figura 89.

Figura 89 – Seleção do modelo do *CLIC*.

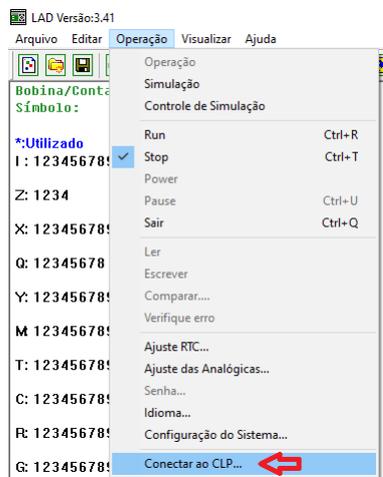
Fonte – Elaborado pelo autor (2021).

Depois, deve ser feita a configuração quanto à comunicação *RS485*, que neste caso foi configurada com 8 *bits*, sem paridade (“N”), 2 *bits* de parada e *baud rate* de 38400, conforme Figura 90.

Figura 90 – Configuração *RS485*.

Fonte – Elaborado pelo autor (2021).

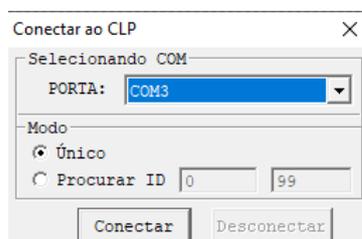
Inicialmente, já com o *software CLIC-02 Edit* em execução, deve-se realizar a conexão do programa com o *CLP*, através da guia “Operação” e “Conectar ao CLP”, conforme ilustrado na Figura 91.

Figura 91 – Conexão do *CLIC-02 Edit* com o *CLP*.

Fonte – Elaborado pelo autor (2021).

Na sequência, deve ser selecionada a “Porta COM” utilizada para conexão do *CLP CLIC* com o computador (Figura 92).

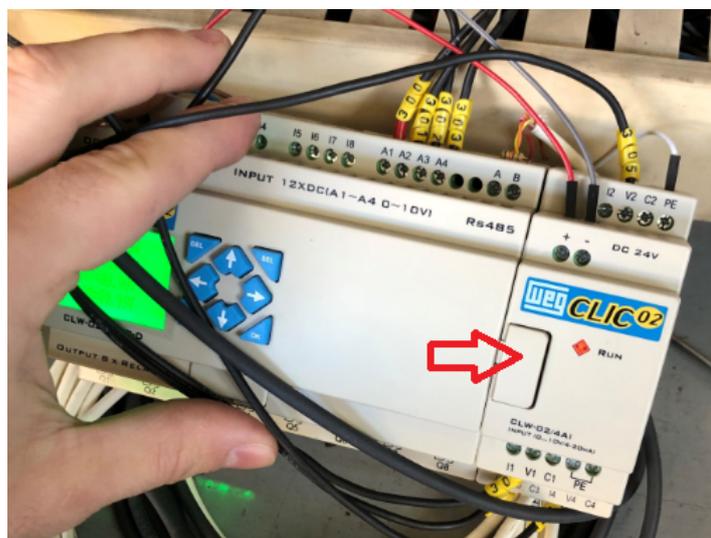
Figura 92 – Seleção de Porta COM.



Fonte – Elaborado pelo autor (2021).

Com a necessidade de 4 entradas analógicas adicionais para captar todos os sinais emitidos pelos sensores, foi necessária a compra de um módulo de expansão analógico para conexão no módulo principal do *CLIC-02*. Neste caso, a *WEG* dispunha de um módulo de expansão para sinais de corrente, desta forma, não foi necessário a utilização de resistores para conversão do sinal, uma vez que o sinal de corrente já é convertido internamente pelo módulo de expansão (Figura 93).

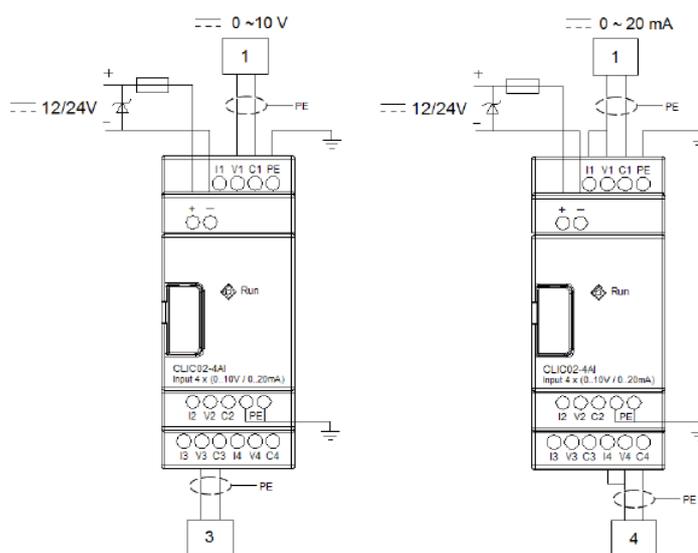
Figura 93 – Módulo de Expansão Analógico do CLIC-02.



Fonte – Elaborado pelo autor (2021).

O módulo de expansão analógica do *CLIC-02* possibilita a conexão de sinais de tensão e de corrente, sendo que no caso de sinais de corrente, é necessário realizar o “jumpeamento” das entradas de tensão e corrente, conforme ilustrado na Figura 94. A imagem da esquerda exibe a conexão para sinais de tensão no módulo de expansão analógico, enquanto que a figura da direita exibe a conexão “jumpeada” para sinais de corrente.

Figura 94 – *Jumper* das Entradas de Corrente e Tensão do *CLIC-02*.



Fonte – Manual do *CLIC-02* (2021).

Desta forma, os 8 sinais de corrente emitidos pelos sensores são captados pelo *CLP CLIC-02*, possibilitando que seja realizada o tratamento dos valores de tensão para interpolação e obtenção dos valores em termos de vazão e temperatura.

Com os passos descritos acima, o *CLP CLIC-02* deverá estar apto para uso juntamente com o módulo de expansão previamente mencionado.

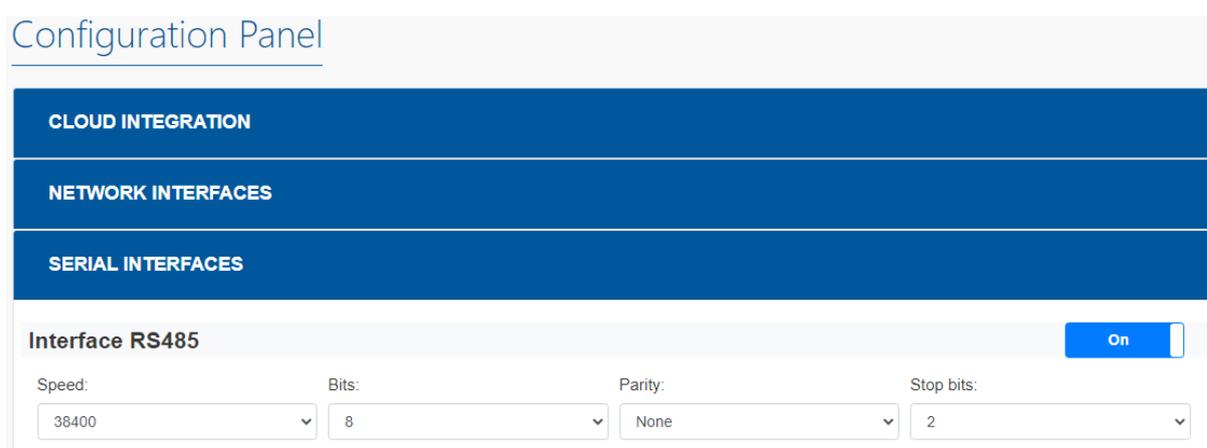
Foi determinado que a interpolação dos valores de corrente/tensão para valores de vazão e temperatura não seria realizada via *Ladder* no *CLIC*, optando-se por realizar os cálculos dentro do *workflow* executado localmente no dispositivo *ED300*, para que eventuais alterações nas fórmulas de interpolação pudessem ser realizadas remotamente. Caso tivesse sido optado por interpolar os valores no *CLP*, qualquer alteração teria de ser realizada no local onde ficaria o dispositivo, o que demandaria tempo de deslocamento e repetitividade no processo em caso de vários *CLPs*.

Detalhes com relação ao processo de interpolação via bloco de função no *workflow* do *ED300* serão descritos na sequência desta seção.

5.11.2.2 Da Configuração do *Gateway ED300* para Comunicação Serial *RS485*

Para realizar a comunicação entre o *CLP CLIC-02* e o dispositivo *Gateway ED300* foi necessário acessar o ambiente de configuração local do dispositivo, conforme descrito na seção 5.6 e incluir as configurações para comunicação serial *RS485* iguais às realizada na etapa anterior de configuração do *CLIC*, conforme ilustrado na Figura 95.

Figura 95 – Configuração Serial do *Gateway ED300*.



Fonte – Manual do *CLIC-02* (2021).

5.11.2.3 Da Conversão dos Sinais de Corrente para Tensão no Módulo Principal do *CLP CLIC-02*

Conforme citado anteriormente, o módulo principal do *CLP CLIC-02*, modelo 20VR-D, dispõe de 4 entradas analógicas para sinais de tensão, fazendo com que os sinais de corrente emitidos pelos sensores de vazão e temperatura sejam incompatíveis com as entradas do dispositivo.

Desta forma, foi necessário converter os sinais de corrente para valores de tensão, e a solução encontrada foi a utilização de pares de resistores de 1000 Ohms em paralelo, que equivalem à uma resistência de 500 Ohms, ligados a um ponto comum negativo da alimentação do *CLP* e aos sinais de corrente dos sensores no ponto das entradas analógicas. A Figura 96 ilustra a solução encontrada.

Figura 96 – Pares de Resistores para Conversão de Sinais de Corrente.



Fonte – Elaborado pelo autor (2021).

O cabo azul ligado à uma das extremidades dos resistores é a ligação com o ponto comum negativo da alimentação do *CLP*.

A Figura 97 ilustra a obtenção do sinal de tensão nos pontos de entradas analógicas A1, A2, A3 e A4 do *CLIC-02*, por meio da ligação entre os sinais de corrente e as extremidades dos resistores.

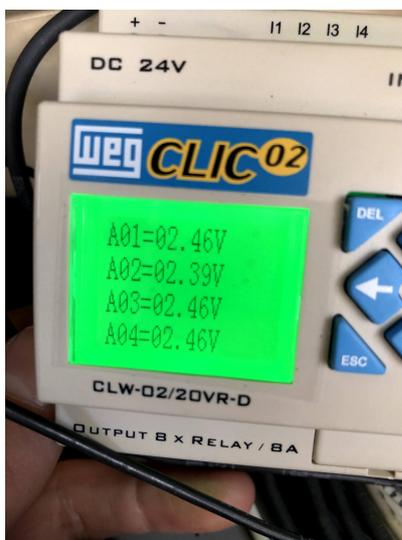
Figura 97 – Conversão de Sinais de Corrente para Sinais de Tensão.



Fonte – Elaborado pelo autor (2021).

Desta forma, é possível verificar no *Display* do *CLIC-02* os valores de tensão dos sinais convertidos na etapa anterior, conforme ilustra a Figura 98.

Figura 98 – Sinais de Corrente Convertidos para Sinais de Tensão.



Fonte – Elaborado pelo autor (2021).

5.11.2.4 Da Conexão *CLIC-02* com o Dispositivo *Gateway ED300*

A fim de transmitir os valores de tensão referentes às vazões e temperaturas dos sensores da máquina para a nuvem, foi necessário estabelecer a conexão do *CLIC-02* com o dispositivo *Gateway ED300*. O *CLIC-02* dispõe de entradas “A” e “B” para comunicação *RS485* e o dispositivo *ED300* dispõe da porta *RS485* conforme visto na figura 37, com pinagem descrita na Figura 99, para comunicação através do protocolo *Modbus RTU*.

Figura 99 – Pinagem da Porta *RS485* do *Gateway ED300*.

Pino	Sinal
1	RS-485 negativo
2	GND
3	RS-485 positivo
4	RS-485 positivo
5	GND
6	RS-485 negativo

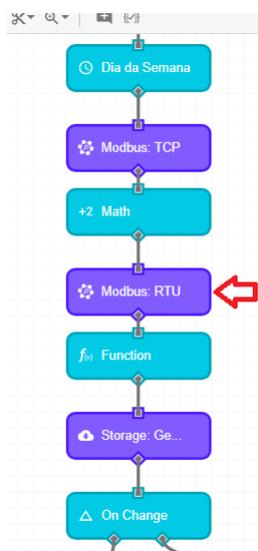
Fonte – Manual do *WEG Drive Motor Scan* (2021).

O meio utilizado para realizar a conexão foi um cabo *WEG* para conexão do *ED300* com uma IHM, por meio dos conectores *RJ11* (*ED300*) e *DB9* (IHM). Porém, como não iremos conectar a IHM, foi cortado o conector *DB9* do cabo e utilizados apenas os fios Laranja (A) e Amarelo (B), que foram ligados às portas A e B de comunicação *RS485* do *CLIC-02*.

É possível visualizar a conexão dos fios Laranja (A) e Amarelo (B) nas figuras 96 e 97 mostradas anteriormente. Também foi adicionado um resistor de 120 Ohms entre as entradas A e B conforme indicado no manual do “*CLIC-02*”.

5.11.2.5 Adaptação do *Workflow* via *WEGnology* para Leitura *Modbus RTU*

Para que a lógica disponível no *ED300* do CT-i13 pudesse efetuar a leitura das variáveis provenientes da comunicação serial entre o dispositivo e o *CLP CLIC*, foi necessário incluir um segundo bloco *Modbus: Read*, agora para o tipo de comunicação “RTU Serial” (Figura 100).

Figura 100 – Inclusão do Bloco *Modbus* para Comunicação *Serial RTU*.

Fonte – Elaborado pelo autor (2021).

Na Figura 101 é demonstrada a configuração do bloco *Modbus: Read* para comunicação serial em questão.

Figura 101 – Configuração do Bloco *Modbus* para Comunicação *Serial RTU*.

Modbus: Read

The Modbus: Read Node allows you to read data from one or more Modbus registers.

Label
Modbus: RTU

Node ID: 2\UTRBCoZF

Add Description
ADDRESS CONFIG

Specify if the connection should be over serial or TCP and then fill out the configuration accordingly.

TCP Connection

Serial RTU Connection

Path Template
/dev/ttyMXC6

Baud Rate Template
38400

Parity Template
none

Data Bits Template
8

Stop Bits Template
2

Unit ID Template
1

Fonte – Elaborado pelo autor (2021).

Os resultados da leitura “Modbus RTU” são armazenados no caminho “results.modbusRTUData”, e os resultados do bloco “Modbus TCP” foram direcionados

ao caminho “results.modbusTCPData” por questões de organização.

O campo “*Path Template*” é preenchido com o caminho da porta “RS485” do dispositivo *Gateway ED300*, sendo “/dev/ttymx6”. As demais configurações seguem a mesma lógica das configurações anteriormente descritas para os dispositivos *CLIC-02* e *Gateway ED300* com relação à comunicação *RS485*.

5.11.2.6 Interpolação das Variáveis dos Sensores via Bloco *Function*

Para realizar a interpolação e o tratamento das variáveis enviadas pelo *CLIC-02* ao dispositivo *ED300*, foi utilizado o bloco “*Function*” do *WEGnology*, que permite que o usuário escreva um código em *JavaScript* para manipular o *payload* do *workflow*.

O código desenvolvido para interpolar os valores de tensão para valores de vazão e temperatura é exibido na Figura 102.

Figura 102 – *JavaScript* para Interpolação.

Function (JavaScript)

```
1 // Principal
2 let measures = payload.results.modbusRTUData;
3 let i = 1;
4
5 for (let item in measures){
6   if (item != "Run" && i <= 4){
7     measures[item][1] = ((40*(measures[item][0]/100)-78)/7.93);
8     measures[item][1] = (measures[item][1]).toFixed(2);
9     if (measures[item][1] <= 0.05) {
10      |   measures[item][1] = 0;
11    }
12    i++;
13  }
14  else if (item != "Run" && 5 <= i <= 8){
15    measures[item][1] = ((measures[item][0]-135.8)/3.88);
16    measures[item][1] = (measures[item][1]).toFixed(2);
17    i++;
18  }
19 }
20
21 payload.results.modbusRTUData = measures;
```

Fonte – Elaborado pelo autor (2021).

Resumidamente, o código aloca o *payload* numa variável chamada “*measures*”, depois percorre os itens da variável, que é do tipo “*array*”, de 1 a 4, e aplica a seguinte fórmula de interpolação para determinar os valores vazão:

$$\frac{40 * V - 78}{7.93} \quad (1)$$

onde “V” é o valor de tensão recebido via *payload*. Existe ainda uma pequena tolerância para um valor de tensão = 0.05V, onde o mesmo é definido como 0V.

Para os itens de 5 a 8, referentes aos valores de temperatura dos sensores, a fórmula de interpolação é descrita a seguir:

$$\frac{V - 135.8}{3.88} \quad (2)$$

onde “V” é o valor de tensão recebido via *payload*.

Os parâmetros de interpolação foram obtidos através da análise dos intervalos de medição dos sensores, conforme observados na Seção 5.3 deste capítulo, com seus respectivos sinais de saída.

Na Figura 103 é possível visualizar¹⁰ as variáveis de vazão e temperatura já tratadas no *payload*.

Figura 103 – *JavaScript* para Interpolação.

```

Debug
Debug Options (20 new)
  results: {} 2 keys
    modbusRTUData: {} 9 keys
      A01: [] 2 items
        0: 194
        1: 0
      A02: [] 2 items
        0: 999
        1: "40.55"
      A03: [] 2 items
        0: 0
        1: 0
      A04: [] 2 items
        0: 0
        1: 0
      A05: [] 2 items
        0: 248
        1: "28.92"
      A06: [] 2 items
        0: 68
        1: "-17.47"
      A07: [] 2 items
        0: 0
        1: "-35.00"
      A08: [] 2 items
        0: 0
        1: "-35.00"

```

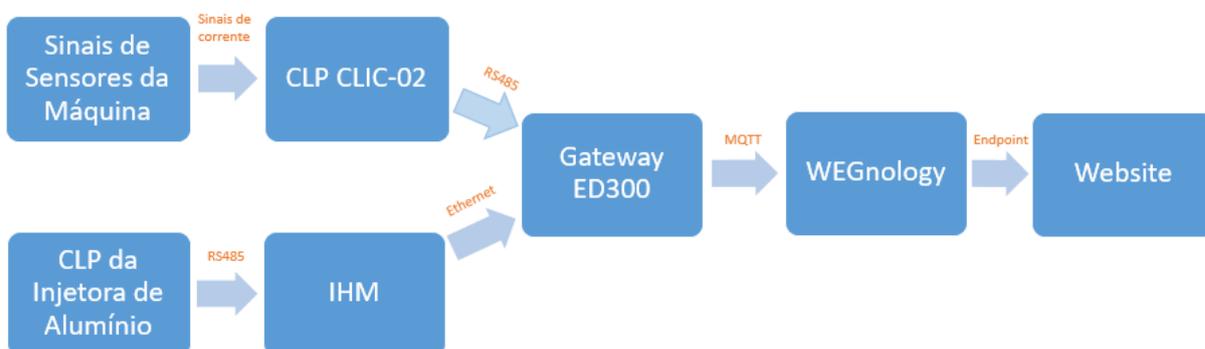
Fonte – Elaborado pelo autor (2021).

¹⁰ No momento da captura do *payload* em questão, 3 dos 4 sensores do CT-i13 se encontravam danificados, portanto, os valores visualizados nas variáveis A02, A03, A04, A06, A07 e A08 devem ser desconsiderados. O valor de vazão observado na variável A01 está correto, onde no campo “[0]” é exibido o valor em Volts sem a precisão das casas decimais (portanto, 1,94V), e no campo “[1]” é o valor interpolado igual à 0 litros por minuto, indicando que no momento da captura do *payload*, não havia fluxo de vazão passando no sensor. A variável A05 representa a temperatura daquele sensor, onde “248” indica 2,48 Volts e com valor interpolado de 28,92 °C.

5.11.2.7 Arquitetura do Sistema IoT para o CT-i13

A arquitetura do sistema de IoT para o CT-i13 é representada na Figura 104.

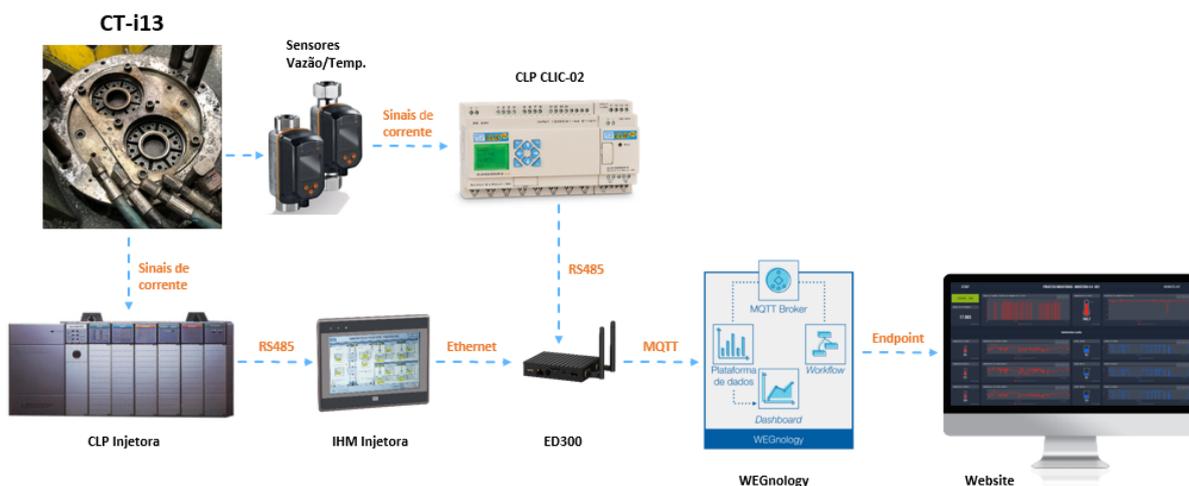
Figura 104 – Arquitetura do Sistema no CT-i13.



Fonte – Elaborado pelo autor (2021).

Utilizando imagens dos componentes da arquitetura, foi construída a Figura 105.

Figura 105 – Arquitetura Ilustrada do Sistema - CT-i13 IoT.



Fonte – Elaborado pelo autor (2021).

Após a finalização da primeira expansão do sistema IoT para o CT-i13, o *Dashboard* desenvolvido para o CT-i07 foi clonado e configurado para exibir as informações referentes ao novo centro de trabalho.

Na seção a seguir iremos abordar sobre o desenvolvimento de um domínio na rede (*Experience Slugs*) para que usuários possam acessar os *Dashboards* desenvolvidos para os dois (e eventuais outros) centros de trabalho.

5.12 CRIAÇÃO DE WEBSITE PARA CENTRALIZAÇÃO DOS DADOS

Com a expansão do Sistema de *IoT* para um segundo centro de trabalho, constatou-se a necessidade de centralizar o acesso dos *Dashboards* num mesmo lugar. Para tal, nesta seção, será detalhado o processo de criação e desenvolvimento de uma interface *front-end* dentro da plataforma do *WEGnology*.

5.12.1 Definição de *Experience* no *WEGnology*

Experiences são um meio de de construir uma interface (*website*) funcional que permite com que usuários interajam com as aplicações desenvolvidas na plataforma, como por exemplo, o sistema *IoT* descrito neste trabalho.

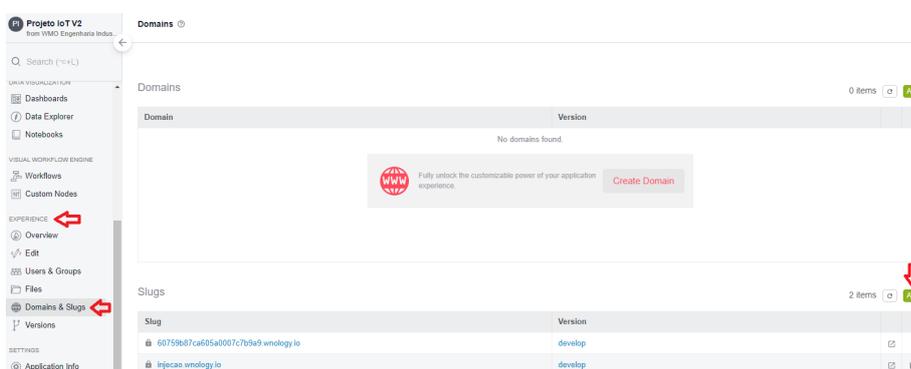
O funcionamento de uma *Experience* se dá através do conjunto de três componentes, são eles:

1. *Endpoint Request* - são as rotas de um endereço *web* que direcionam para algum ponto específico;
2. *Trigger Workflow* - a lógica que será executada com o acesso ao *Endpoint* anterior;
3. *Render Page* - a execução do *workflow* poderá retornar um dado puro ou redirecionar o usuário à uma *Experience Page* a ser renderizada como *HTML*, ou até mesmo uma página de *Dashboard* para renderizar um *Dashboard* do *WEGnology*.

5.12.2 Da Criação de um *Slug* para a Aplicação *IoT*

O primeiro passo é realizar a definição do *Experience Slug* que será utilizado para acessar o conteúdo da aplicação desenvolvida. Desta forma, acessando a aba de “*Domains & Slugs*” na guia “*Experience*” dentro do *WEGnology*, conforme ilustrado na Figura 106, deve ser selecionado o botão “*Add*” para adicionar o novo *Slug*.

Figura 106 – Criação de um Slug para a Aplicação IoT.



Fonte – Elaborado pelo autor (2021).

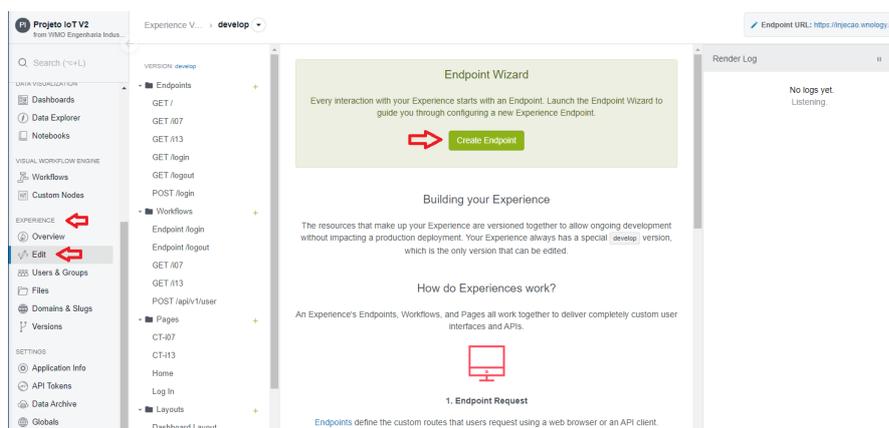
Para esta aplicação, foi criado o *Experience Slug* “*injecao.wnology.io*”, conforme exibido na lista de *Slugs* da Figura 106.

5.12.3 Da Criação dos Endpoints

O próximo passo é a criação dos *Endpoints* que irão direcionar para os *Dashboards* desenvolvidos até aqui. O *WEGnology* possui um “*Wizard*” para auxiliar na criação dos *Endpoints* para as aplicações, conforme será descrito na sequência.

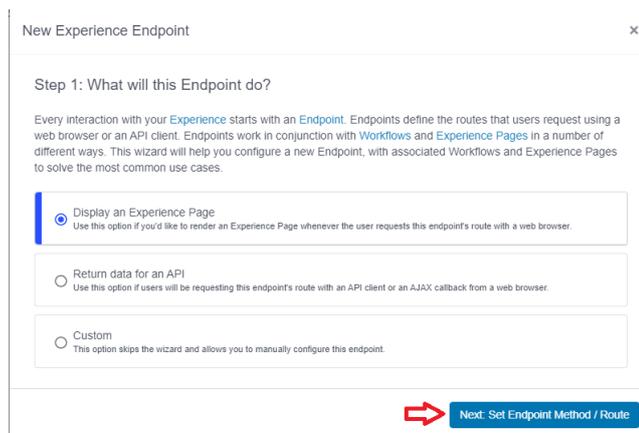
Na aba “*Edit*” da guia de “*Experience*” deve ser selecionado o botão “*Create Endpoint*”, conforme ilustrado na Figura 107.

Figura 107 – Criação de um Endpoint para a Aplicação IoT.



Fonte – Elaborado pelo autor (2021).

Deve ser definida a funcionalidade do novo *Endpoint*, que neste caso será exibir uma página de *Experience*, conforme ilustrado na Figura 108.

Figura 108 – Funcionalidade do *Endpoint*.

New Experience Endpoint

Step 1: What will this Endpoint do?

Every interaction with your Experience starts with an Endpoint. Endpoints define the routes that users request using a web browser or an API client. Endpoints work in conjunction with Workflows and Experience Pages in a number of different ways. This wizard will help you configure a new Endpoint, with associated Workflows and Experience Pages to solve the most common use cases.

Display an Experience Page
Use this option if you'd like to render an Experience Page whenever the user requests this endpoint's route with a web browser.

Return data for an API
Use this option if users will be requesting this endpoint's route with an API client or an AJAX callback from a web browser.

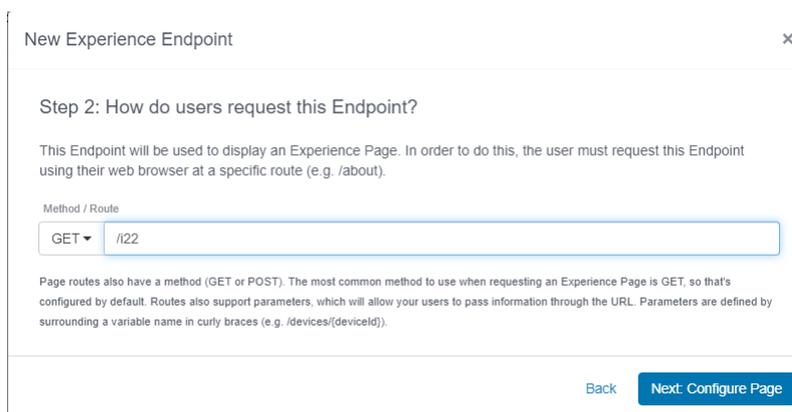
Custom
This option skips the wizard and allows you to manually configure this endpoint.

Next: Set Endpoint Method / Route

Fonte – Elaborado pelo autor (2021).

A título de exemplificar o processo descrito nesta etapa, utilizaremos o CT-i22 como exemplo para criação do *Endpoint* a seguir, sendo que para os CTs i07 e i13 foram feitos da mesma forma.

Assim, deve-se indicar o método e a rota para que os usuários possam solicitar o novo *Endpoint*, que neste caso terá o método do tipo “Get” e a rota “/i22” (Figura 109).

Figura 109 – *Endpoint Method and Route*.

New Experience Endpoint

Step 2: How do users request this Endpoint?

This Endpoint will be used to display an Experience Page. In order to do this, the user must request this Endpoint using their web browser at a specific route (e.g. /about).

Method / Route

GET /i22

Page routes also have a method (GET or POST). The most common method to use when requesting an Experience Page is GET, so that's configured by default. Routes also support parameters, which will allow your users to pass information through the URL. Parameters are defined by surrounding a variable name in curly braces (e.g. /devices/{deviceId}).

Back Next: Configure Page

Fonte – Elaborado pelo autor (2021).

O próximo passo é definir o nome página e como ela será exibida através daquele *Endpoint*, que neste caso, será “Dashboard Layout” (Figura 110).

Figura 110 – *Endpoint Page*.

New Experience Endpoint ×

Step 3: Which Experience Page will be displayed?

Now that you have a route configured that the user's web browser can request, you need to choose which Experience Page will be displayed for them. You can choose an existing page, or create a new one.

Create a new Experience Page ...
A new page will automatically be created and connected to this Endpoint.

New Page Name

Page Layout
 × ▾

Use an existing Experience Page ...
If you have an existing page you'd like to display, select it below.

Experience Page
 ▾

[Back](#) [Next: Configure Workflow](#)

Fonte – Elaborado pelo autor (2021).

Na sequência, o *Wizard* sugere a criação de um *workflow* que será responsável por direcionar o usuário que acessou o *Endpoint* para um ponto específico dentro do *Experience* (Figura 111).

Figura 111 – *Endpoint Workflow*.

New Experience Endpoint ×

Step 4: Does this page require a workflow to query any data?

If this page requires any dynamic data, either from [Devices](#), [Data Tables](#), [Events](#), or any other source, it will need a workflow to query that information. The page will then have access to that data while it's being rendered for the user. Pages that do not require any dynamic information (i.e. static pages) do not require a workflow and can be directly displayed to the user.

Create a new Workflow ...
A new Workflow will automatically be created and connected to this Endpoint.

New Workflow Name

No Workflow is required; this is a static Experience Page.
This Endpoint will directly display your Experience Page without using a Workflow to query any data first.

Use an existing Workflow.
Select this option if you already have a workflow you'd like to use. You will be required to modify the existing workflow to utilize this new endpoint.

[Back](#) [Next: Configure Access](#)

Fonte – Elaborado pelo autor (2021).

Após esta configuração, o *Wizard* irá solicitar configuração com relação ao acesso à página, que neste caso, será pública, portanto, foi selecionada a opção “All

Public Users” (Figura 112).

Figura 112 – Configuração de Acesso no *Endpoint*.

The screenshot shows a web interface for configuring a new experience endpoint. The title is 'New Experience Endpoint' with a close button (x). The current step is 'Step 5: Who is allowed to view this Page?'. Below the title, there is explanatory text: 'Access to this page can automatically be restricted based on whether or not the user is authenticated. You can also further restrict access for authenticated users based on which Experience Group(s) they are a member of.' There are three radio button options: 1) 'Any authenticated user.' with subtext 'Any user that is logged in can view this page, regardless of which groups they are a member of.' 2) 'Only users who are members of the following groups ...' with subtext 'The user must be logged in and a member of one of the groups specified below.' This option includes a 'Groups' dropdown menu with 'Select...' as the current selection. 3) 'All public users / custom authentication.' which is selected. Its subtext reads: 'Anyone can access this page regardless of whether or not they are logged in. This option also allows you to implement custom authentication using a workflow.' At the bottom right, there are two buttons: 'Back' and 'Next: Review Choices'.

Fonte – Elaborado pelo autor (2021).

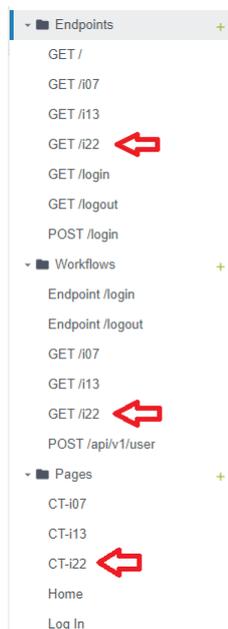
Por último, é exibido um resumo das configurações para fim de verificação e, após confirmação dos dados, concluí-se a criação do novo *Endpoint* através do botão “*Create Endpoint Resources*” (Figura 113).

Figura 113 – Verificação das Configurações do *Endpoint*.

The screenshot shows the final step of the configuration process, 'Step 6: Is everything correct?'. The title is 'New Experience Endpoint' with a close button (x). Below the title, there is explanatory text: 'Review the summary below and confirm your choices are correct. Once you click "Create Endpoint" below, all resources will be created and a followup screen with links will be provided.' A summary box contains the following information: **Endpoint Request:** GET /i22; **Page Info:** Create a new Experience Page ... Name: "GET /i22" Layout: "Dashboard Layout"; **Workflow Configuration:** Create a new Workflow ... Name: "GET /i22"; **Endpoint Access:** All public users / custom authentication. At the bottom right, there are two buttons: 'Back' and 'Create Endpoint Resources'.

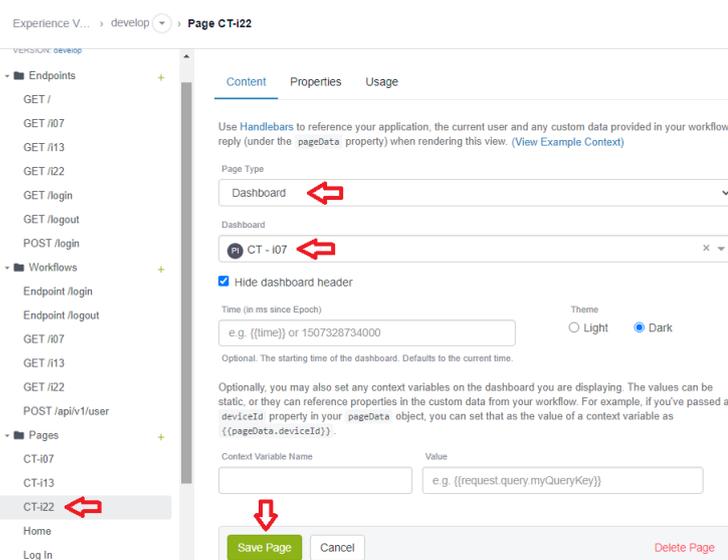
Fonte – Elaborado pelo autor (2021).

Assim, serão criados *Endpoint*, *Workflow* e *Page* para o CT em questão (Figura 114).

Figura 114 – *Endpoint Criado.*

Fonte – Elaborado pelo autor (2021).

A partir daqui usaremos os recursos criados para o CT-i07 para seguir com o detalhamento. O último passo é relacionar aquele *Endpoint* à página do *Dashboard* desenvolvido para o CT em questão, conforme ilustrado na Figura 115.

Figura 115 – Relacionando *Endpoint* com *Dashboard* do CT.

Fonte – Elaborado pelo autor (2021).

Desta forma, qualquer usuário poderá acessar o *Endpoint* do CT-i07 através da

URL “<https://injecao.wnology.io/i07>”. O mesmo processo foi realizado para o CT-i13, resultando no *Endpoint* da URL “<https://injecao.wnology.io/i13>”.

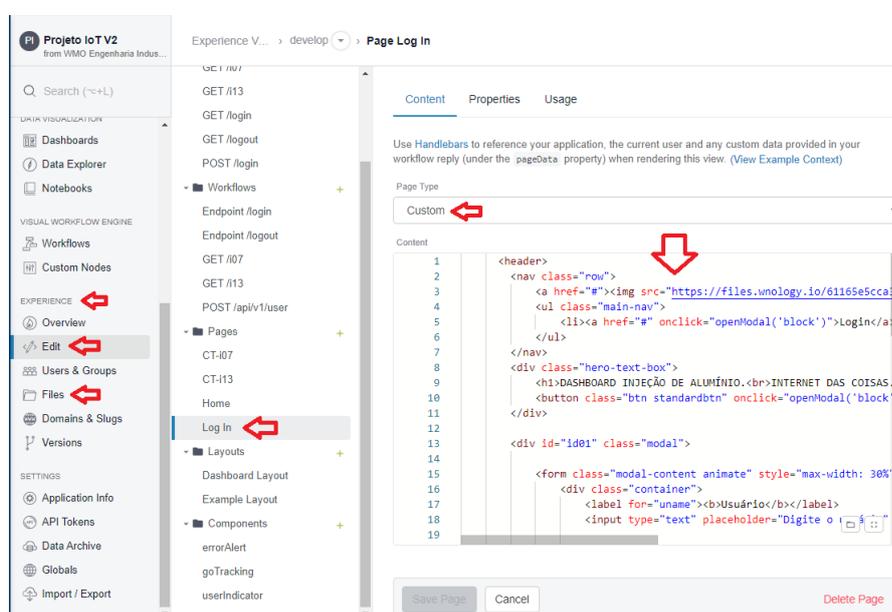
5.12.4 Da Criação do Website

Para que todos os *Dashboards* da aplicação *IoT* pudessem ser acessados dentro de uma mesma interface, optou-se pelo desenvolvimento de um *Website* afim de centralizar a navegação entre os *Endpoints* de cada Centro de Trabalho.

Assim, foi desenvolvido um *Endpoint* (mesmo procedimento descrito anteriormente) que apontasse para uma página de *Log-In*, onde os usuários deverão se autenticar para acessar o ambiente e, após isso, serem direcionados à pagina *Home*, que também foi criada como um *Endpoint*.

A diferença com relação ao processo realizado anteriormente é que, ao invés de relacionar o *Endpoint* à um modelo de *Dashboard* do *WEGnology*, foi desenvolvido uma página *WEB* através das linguagens de programação *HTML*, *CSS* e *JavaScript*, para que o usuário realizasse o *Log-In* na interface (Figura116).

Figura 116 – Criação da Interface de *Log-In* da Página *Web*.



Fonte – Elaborado pelo autor (2021).

Na Figura 117 é exibido a página de *Log-In* desenvolvida para a aplicação.

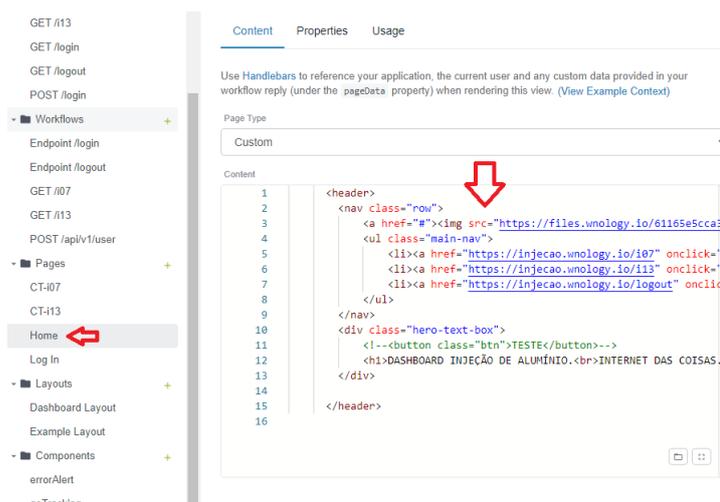
Figura 117 – Página de Log-In.



Fonte – Elaborado pelo autor (2021).

O mesmo processo foi realizado para o desenvolvimento de uma página “Home”, que é exibida após o Log-In. O arquivo CSS utilizado por toda a interface Web em questão foi criado e alocado na aba de “Files” indicada na figura acima (Figura 118).

Figura 118 – Criação da Interface da Página Home da Página Web.



Fonte – Elaborado pelo autor (2021).

Na Figura 119 é exibido a página Home desenvolvida para a aplicação.

Figura 119 – Página *Home*.

Fonte – Elaborado pelo autor (2021).

Os usuários podem acessar a interface *WEB* da aplicação *IoT* através da *URL* padrão (Slug) e serão direcionados à página de *Log-In* da interface caso ainda não estejam autenticados, ou diretamente para a página *Home* caso já estejam autenticados. Após o *Log-In*, a página “seta” um “*cookie*” relacionado à autenticação do navegador utilizado.

5.13 NOVAS FUNCIONALIDADES E INDICADORES DO SISTEMA *IOT*

Nesta seção são descritas algumas das funcionalidades que foram incorporadas à aplicação *IoT* ao longo do desenvolvimento do trabalho, que possibilitaram melhorias na execução da lógica do sistema e na criação de novos indicadores para o processo de Injeção de Alumínio de Rotores.

Cabe aqui ressaltar que a *WEG* possui uma política forte com relação ao desenvolvimento de melhorias (Kaizens) de forma geral, sendo que os trabalhos abordados nos tópicos apresentados a seguir foram desenvolvidos neste contexto.

5.13.1 Implementação de Sistema de Alertas para os Fornos Dosadores das Injetoras de Alumínio

A solidificação do alumínio nos fornos dos centros de trabalho podem ocorrer em decorrência da perda de temperatura do metal. Caso a perda de temperatura ocorra em dias em que não há turno de trabalho, não existe um sistema que dispare avisos alertando do ocorrido, fazendo com que o problema só seja identificado no

início do próximo turno, e que pode levar tempo suficiente para o material se solidificar, impossibilitando o início imediato do trabalho do próximo turno no CT do forno afetado, podendo paralisar a produção por mais de 1 dia.

O controle da temperatura ocorre a partir da existência de um termopar que capta a temperatura do alumínio fundido no interior do forno. Esse valor é enviado para um processador que faz o controle de processo do forno. Neste processador são informados os valores de processo, como os parâmetros de temperatura mínima e máxima. Caso a temperatura do forno atinja o valor mínimo, o processador liga as resistências do forno para elevar e manter a temperatura do banho. No entanto, se essas resistências não ligarem ou a temperatura não voltar a subir, é enviado um alarme para o processador e acionado uma sirena no painel indicando falha de temperatura, ficando visível e audível apenas nas proximidades do painel do forno (Figura 120).

Figura 120 – Forno Dosador da Injetora de Alumínio.



Fonte – Elaborado pelo autor (2021).

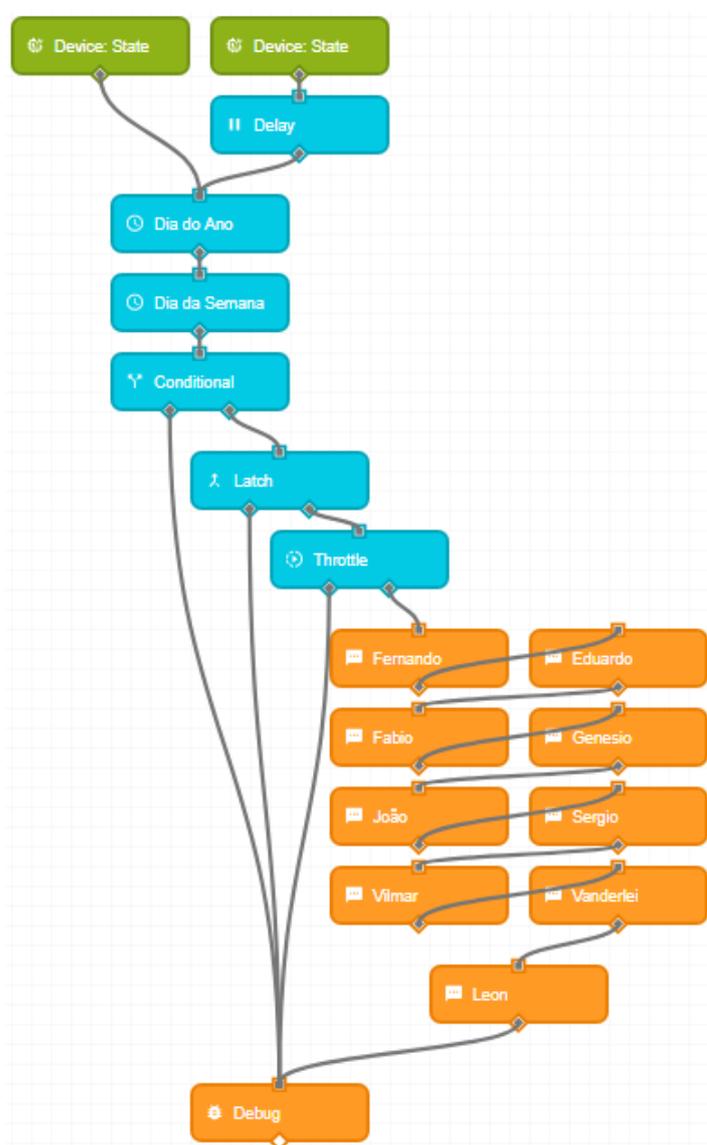
Foi desenvolvido um sistema de monitoramento de temperatura do forno para atuar nos finais de semana e feriados, responsável por emitir alertas via *SMS* caso o valor de temperatura medido for inferior a um valor pré-determinado de referência. A

justificativa da solução é garantir agilidade na notificação de um eventual problema de temperatura no forno, fazendo com que a correção seja iniciada mais rapidamente e concluída antes do metal iniciar o processo de solidificação.

Foi necessário realizar a substituição do termopar utilizado no forno, que possuía apenas 1 saída de sinal e que era ligada ao processador do forno, para um termopar com 2 saídas, para que um sinal pudesse ser conectado ao *CLP* da Injetora e convertido para valores de temperatura. O sistema de monitoramento foi desenvolvido e implementado paralelamente à lógica existente no dispositivo *Edge ED300*.

A Figura 121 demonstra a lógica do sistema de alerta desenvolvida no *WEGno-logy*.

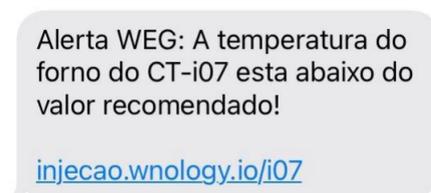
Figura 121 – *Workflow* do Sistema de Alerta.



Fonte – Elaborado pelo autor (2021).

A Figura 122 ilustra a mensagem SMS característica do alerta.

Figura 122 – Modelo de SMS do Alerta.



Fonte – Elaborado pelo autor (2021).

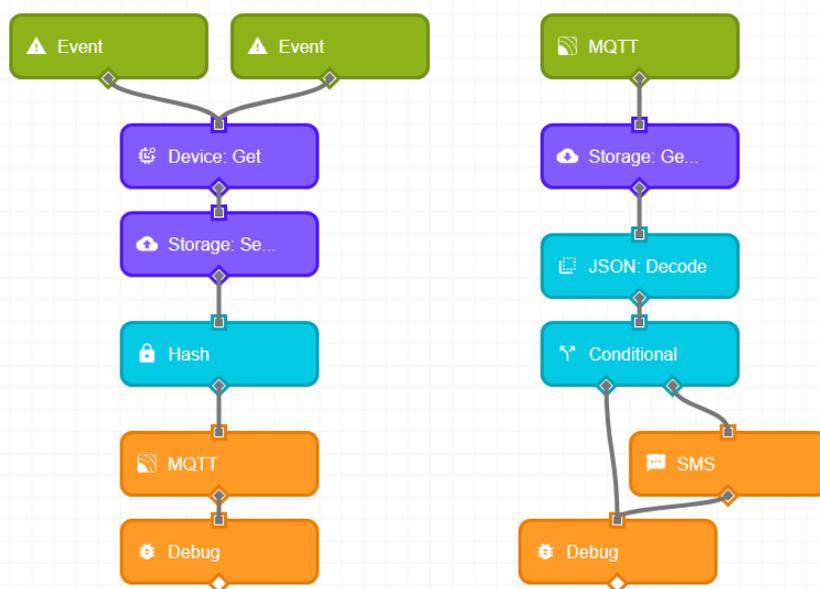
5.13.2 Sistema de *Reboot* Automático para os dispositivos *ED300*

Esta solução teve como finalidade implementar um sistema de *Reboot* automático para os dispositivos *ED300* em caso de perda de conexão com a *internet/WEGnology*. Ao restabelecer a conexão, o *Dashboard* criando para os CT apresentava alguns problemas com relação à atualização dos valores das variáveis exibidas, fazendo-se necessário desligar e religar o dispositivo *ED300* com uma certa frequência.

Foi criada uma lógica que executa na nuvem e identifica eventuais desconexões dos dispositivos e, no momento da reconexão com a plataforma, um comando de *reboot* é enviado automaticamente ao *ED300* em questão para que seja reiniciado e prevenir eventuais problemas no carregamento dos *Dashboards*.

Esta solução se mostrou 100% eficiente desde o momento de sua implementação, fazendo com que não houvesse mais a necessidade de deslocamento até a fábrica de Injeção de Alumínio para reiniciar o dispositivo manualmente, garantindo maior robustez ao sistema *IoT* e aos *Dashboards*.

Na Figura 123 é demonstrado o *workflow* responsável pela identificação das desconexões e pelo *reboot* dos dispositivos.

Figura 123 – Sistema de *Reboot* Automático do *ED300*.

Fonte – Elaborado pelo autor (2021).

5.13.3 API para Identificação de Feriados

A frequência de execução dos *workflows* nos dispositivos *Edge* nos finais de semana e feriados não precisa ser igual à frequência de execução durante os períodos de trabalho, uma vez que não ocorrem variações relevantes dos valores das variáveis de processo quando a máquina não está em funcionamento.

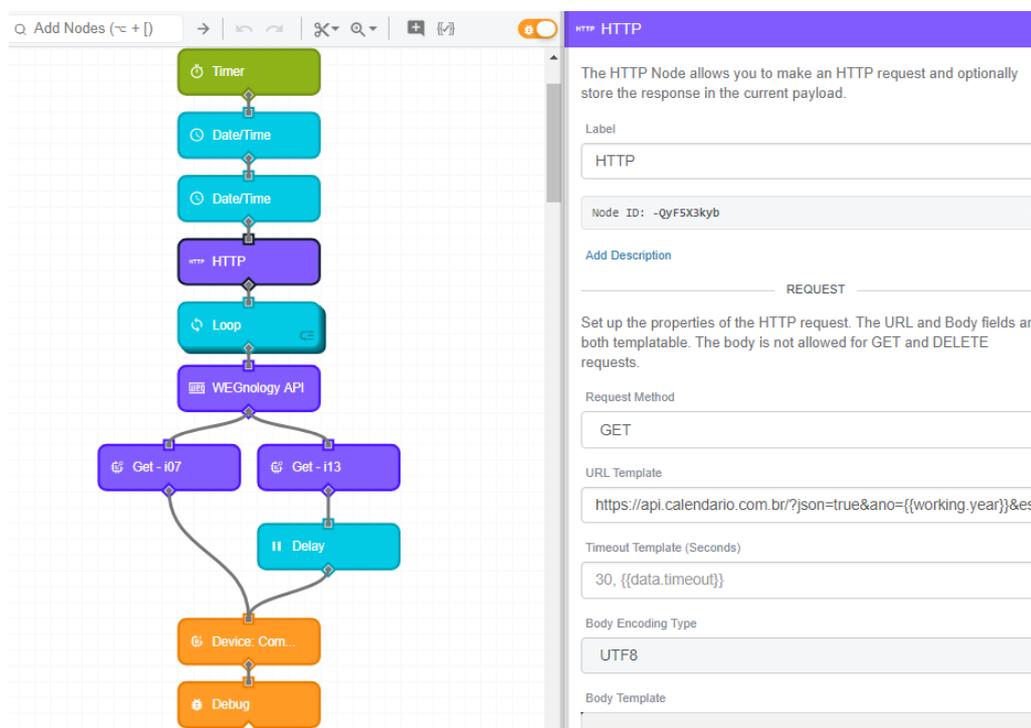
Assim, foi identificada a necessidade de verificar se o dia corrente se trata de um dia útil ou de um dia não útil ou feriado. Foi então desenvolvido uma lógica que executa no *WEGnology* e que estabelece uma *Application Program Interface (API)* com um sistema provedor de informações (*“calendario.com.br”* relacionadas ao calendário de um determinado local em diferentes níveis, ou seja, federal, estadual e municipal.

Assim, a lógica executa diariamente a requisição da informação e atribui valor *“true”* para a variável global *“is.Holiday”* dentro do *WEGnology* caso seja final de semana ou feriado, ou valor *“false”* caso seja um dia útil.

Essa variável é consumida pelo *workflow* que executa localmente do dispositivo *ED300* e irá determinar a frequência de atualização das variáveis coletadas no estado do dispositivo (*Device: State*).

A Figura 124 ilustra o *workflow* que requisita as informações via API e parte das configurações do bloco *HTTP*.

Figura 124 – API para Verificação de Feriados.



Fonte – Elaborado pelo autor (2021).

5.13.4 Criação de Indicador de Excedente de Alumínio no Massalote

Os massalotes resultantes da injeção de alumínio de rotores possuem um valor de altura mínima e máxima estabelecidas de acordo com o molde utilizado no processo, que garantem uma alimentação ideal de alumínio para o rotor.

Eventualmente, devido a problemas inerentes dos fornos, o massalote pode sair maior que o valor de altura máximo determinado, o que significa em custos adicionais para refundir aquele material.

Foi desenvolvida uma lógica que verifica a altura de cada massalote no momento da injeção do rotor e, caso o valor ultrapasse o valor máximo estabelecido para aquele molde, a diferença é armazenada em uma variável e será convertida em termos de massa (quilograma) e financeiros (R\$). A visualização destas informações também foi incorporada aos *Dashboards* das máquinas com o Sistema *IoT*.

A Figura 125 ilustra o código desenvolvido e executado dentro do bloco “*Function*” adicionado ao *workflow* dos dispositivos.

Figura 125 – Função de Excedente de Alumínio no Massalote.

Function (JavaScript)

```
1 // Principal
2
3 let measures = payload.results.modbusData;
4 let massaloteMax = measures.massaloteMax;
5 let massaloteMin = measures.massaloteMin;
6 let massaloteReal = measures.massaloteReal;
7 let massaloteTotal = measures.massaloteTotal;
8 let massaloteMedio = ((massaloteMax+massaloteMin)/2);
9 let volumeTotal = 0;
10 let kilogramaTotal = 0;
11 let custoTotal = 0;
12
13 if (massaloteReal > massaloteMedio) {
14     massaloteTotal = (massaloteTotal + (massaloteReal - massaloteMedio));
15     volumeTotal = Math.PI*(139.7 ** 2)*Number(massaloteTotal);
16     kilogramaTotal = Number((volumeTotal*0.0027)/1000);
17     custoTotal = Number(kilogramaTotal*256.92/1000);
18 } else if (massaloteTotal == null) {
19     massaloteTotal = 0;
20 } else {
21     volumeTotal = Math.PI*(139.7 ** 2)*Number(massaloteTotal);
22     kilogramaTotal = Number((volumeTotal*0.0027)/1000);
23     custoTotal = Number(kilogramaTotal*256.92/1000);
24 }
25
26 payload.results.modbusData.massaloteTotal = Number(massaloteTotal).toFixed(1);
27 payload.results.modbusData.massaloteMedio = Number(massaloteMedio).toFixed(1);
28 payload.results.modbusData.volumeTotal = Number(volumeTotal).toFixed(1);
29 payload.results.modbusData.kilogramaTotal = Number(kilogramaTotal).toFixed(1);
30 payload.results.modbusData.custoTotal = Number(custoTotal).toFixed(2);
31
32
33 payload.results.modbusData = measures;
```

Fonte – Elaborado pelo autor (2021).

A Figura 126 representa a visualização do indicador no *Dashboard* do CT-i07.

Figura 126 – Função de Excedente de Alumínio no Massalote.



Fonte – Elaborado pelo autor (2021).

Até o momento da elaboração deste documento, após todas as implementações de lógicas adicionais à lógica básica de aquisição de dados, bem como de diversas melhorias da própria lógica de aquisição de dados, o *workflow* do Sistema *IoT* do CT-i07 é apresentado na Figura 127.

6 ANÁLISE DE RESULTADOS

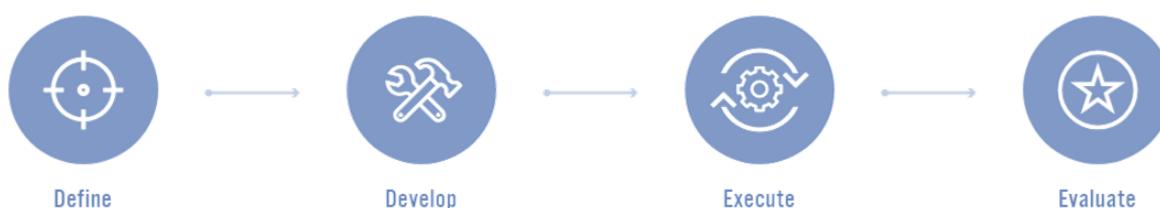
Neste capítulo, é realizada a análise dos resultados obtidos com relação aos objetivos previstos, bem como a explanação de elementos que indicam que o projeto desenvolvido e implementado na Injeção de Alumínio tem gerado resultados satisfatórios em termos de uma proposta de sistema *IoT*.

6.1 *PROOF OF CONCEPT (POC)*

A finalidade de uma Prova de Conceito (*Proof of Concept (PoC)*) é experimentar uma solução em seu ambiente, através da coleta de informações e avaliação de performance dentro de um determinado prazo e orçamento. De forma geral, serve para estudar a viabilidade do projeto, auxiliando na tomada de decisão sobre sua validade.

A Figura 128 ilustra as etapas de uma Prova de Conceito.

Figura 128 – Etapas *Proof of Concept*.



Fonte – Elaborado pelo autor (2021).

A primeira etapa, de definição do escopo do trabalho, metas e objetivos foi realizada e descrita no decorrer dos capítulos 1 e 2 deste documento.

A segunda e terceira etapa, de desenvolvimento e execução, respectivamente, foram detalhadas no Capítulo 5 deste documento, onde foram descritas as arquiteturas dos sistemas, os dispositivos e equipamentos envolvidos, bem como suas configurações, além do desenvolvimento dos *softwares* responsáveis por executar as lógicas do sistema. Por fim, a visualização dos dados do processo por meio de *Dashboards*.

A última etapa, de avaliação, se encaixa neste capítulo atual, onde serão demonstrados os resultados ante os objetivos estabelecidos, bem como alguns *cases* que comprovam que a solução *IoT* desenvolvida já demonstrou sua utilidade.

6.2 OBJETIVOS PREVISTOS E RESULTADOS OBTIDOS

No decorrer do Capítulo 1 foram expostos os objetivos principais e gerais definidos para este trabalho, sendo eles:

- Desenvolvimento e implementação da solução *IoT* nas máquinas injetoras de alumínio;
- Criação de um *Website* para acesso às informações de processo;

Assim, considerando o primeiro objetivo listado, ou seja, o desenvolvimento e implementação de uma solução *IoT* em máquinas injetoras de alumínio, podemos considerar o atingimento da meta prevista, uma vez que a solução foi desenvolvida e implementada em dois centros de trabalho (CT-i07 e CT-i13), além de possuir, no momento da elaboração deste documento, um orçamento aprovado para a expansão para outros 5 centros de trabalho na fábrica de Injeção de Alumínio.

Considerando o segundo item da lista de objetivos, temos a criação de um *Website* para que seja possível o acesso simplificado aos *Dashboards* referentes aos processos de injeção dos diversos centros de trabalho onde a solução *IoT* estivesse implementada. Sendo assim, conforme detalhado no decorrer da seção 5.12 do capítulo 5, o *Website* para tal finalidade foi desenvolvido e está operando no domínio “<https://injecao.wnology.io/login>” como previsto.

6.3 CASES DE APLICAÇÃO DO SISTEMA *IOT*

Após a implementação do Sistema *IoT* nas máquinas injetoras de alumínio, houve ocasiões onde a aplicação se mostrou útil, comprovando o fator de agregação de valor ao processo, que serão descritas a seguir.

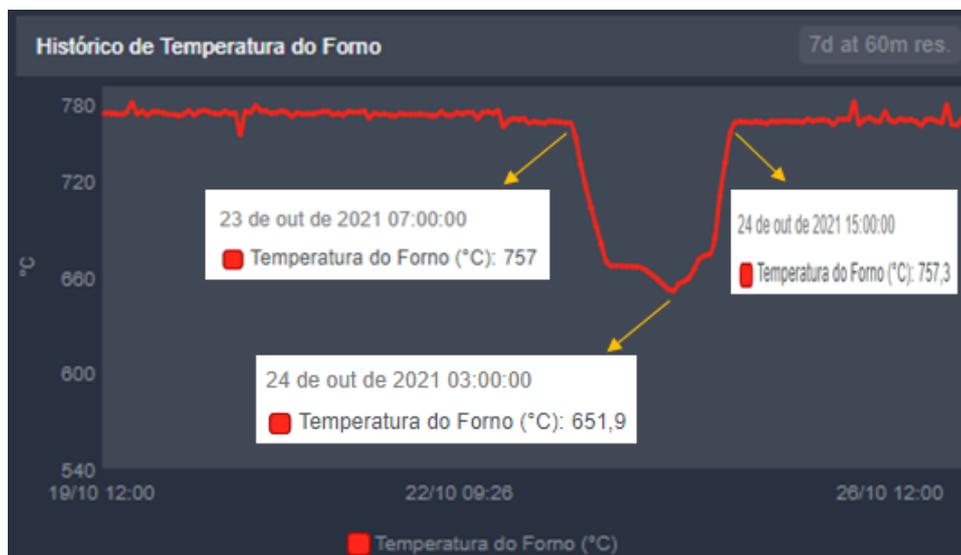
6.3.1 Forno Dosador - Falha na Manutenção da Temperatura do Alumínio

Conforme descrito na Subseção 5.13.1 do capítulo de desenvolvimento, quanto o sistema de monitoramento de temperatura dos fornos dosadores das máquinas injetoras de alumínio, a solidificação do alumínio pode ocorrer em decorrência da perda de temperatura do metal, ocasionada, por sua vez, pela incapacidade do forno em manter o alumínio no estado líquido. Esta incapacidade do forno em garantir a temperatura ideal no forno tem relação a problemas nas resistências dos mesmos e, caso não haja a identificação do fato a tempo, é possível que o alumínio venha a se solidificar no interior do forno.

Após a implementação do Sistema *IoT*, houve a múltipla ocorrência deste problema em diferentes momentos (finais de semana), todas elas sendo devidamente identificadas pela aplicação, que por sua vez sinalizou o fato aos responsáveis pela manutenção dos fornos e também aos chefes do departamento de injeção, permitindo então que uma resposta rápida foi dada para solucionar o problema, evitando que o alumínio no interior do forno viesse a se solidificar, o que resultaria no impedimento da operação do centro de trabalho no início do turno.

A Figura 129 ilustra um destes eventos onde houve a queda da temperatura de um dos fornos dosadores durante um final de semana, fazendo com que os alertas que foram disparados aos responsáveis (Figura 130).

Figura 129 – Queda de Temperatura do Forno Dosador.



Fonte – Elaborado pelo autor (2021).

Figura 130 – Alertas de Temperatura do Forno Dosador.

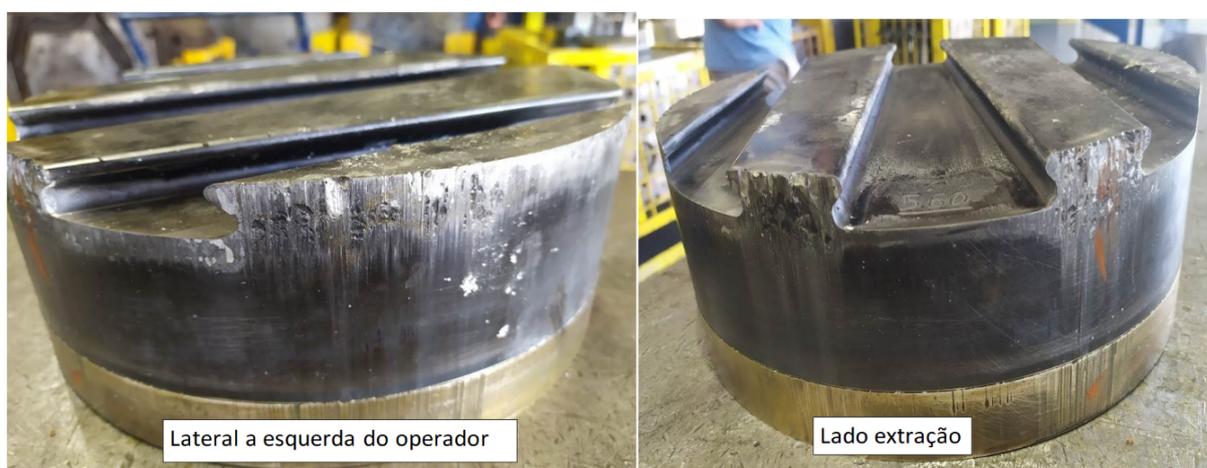


Fonte – Elaborado pelo autor (2021).

6.3.2 Engripamento de Pistão da Injetora de Alumínio

Outro problema que pode ser identificado com o auxílio do Sistema *IoT* foi o caso de engripamento do pistão da máquina injetora de alumínio. Na ocasião, o engripamento se deu pela dilatação da superfície do pistão da máquina, que por sua vez se deu pelo seu aquecimento, ou pela falta de refrigeração. Este dilatamento do pistão ocasionou no atrito entre o componente e a câmara do pistão, onde fica alojado. As figuras 131, 132, 133 e 134 a seguir ilustram o fato ocorrido.

Figura 131 – Pistão.



Fonte – Elaborado pelo autor (2021).

Figura 132 – Pistão (vista 2).



Fonte – Elaborado pelo autor (2021).

Figura 133 – Câmara do Pistão.



Fonte – Elaborado pelo autor (2021).

Figura 134 – Câmara do Pistão (vista 2).



Fonte – Elaborado pelo autor (2021).

O problema ocorreu, a operação da máquina teve que ser interrompida e foram acionadas as equipes de manutenção do departamento para identificação e solução do problema. De forma rápida e prática, com uma análise do histórico das variáveis de vazão e temperatura do pistão através do *Dashboard* daquele Centro de Trabalho, foi possível constatar o seguinte comportamento, indicado na Figura 135.

Figura 135 – Comportamento das Variáveis de Vazão e Temperatura.



Fonte – Elaborado pelo autor (2021).

Com a análise do comportamento das variáveis de vazão e temperatura no tempo, como mostrado na Figura 135, foi possível perceber que a vazão do fluido refrigerante do pistão começou a cair de forma anormal com o passar do tempo e, conseqüentemente, a temperatura começou a aumentar, ou seja, o pistão não estava sendo refrigerado adequadamente durante a operação da máquina. Com este padrão de comportamento, rapidamente foi levantada a hipótese de haver um entupimento no canal de refrigeração do pistão daquela máquina, que poderia estar causando o problema.

Integrantes da equipe de manutenção fizeram a averiguação no local e constataram que, de fato, se tratava de um entupimento naquela mangueira de refrigeração, realizando a correção logo em seguida. Após a correção, a operação naquele centro de trabalho pode ser retomada¹.

Com o entendimento da natureza do problema em questão, foi possível desenvolver e adicionar uma lógica no Sistema *IoT* que identifica, simultaneamente, a queda dos valores de vazão e o aumento dos valores de temperatura dos componentes da

¹ As fotos do pistão e câmara desgastados retratam um evento que ocorreu em outra fábrica da WEG, onde não havia o Sistema *IoT* implementado. A foto do *Dashboard* com o comportamento das variáveis de vazão e temperatura são de uma das máquinas onde a solução foi implementada, em Jaraguá do Sul, que também apresentou o problema do entupimento do canal de refrigeração do pistão. Neste segundo caso, o problema foi constatado e corrigido antes de vir a prejudicar os componentes da máquina.

injetora de alumínio, e sinaliza, por meio do disparo de *e-mails* e mensagens *SMS*, o fato para todos os responsáveis, agilizando na intervenção e correção do problema ainda em estado precoce.

No capítulo a seguir será apresentada a conclusão deste trabalho, onde é colocado um resumo do que foi feito, os principais resultados obtidos, bem como algumas sugestões de trabalhos futuros.

7 CONCLUSÃO

O objetivo deste trabalho foi o desenvolvimento e implementação de uma solução *IoT* em um equipamento legado, neste caso, uma máquina injetora de alumínio de rotores, a fim de coletar e disponibilizar informações relativas às variáveis e indicadores do processo de injeção.

Alinhado com os conceitos da Indústria 4.0, o interesse na aplicação de novas tecnologias no ambiente de chão de fábrica visa, principalmente, no aumento de competitividade no mercado. A *WEG* possui um forte viés de inovação tecnológica, e a possibilidade da realização deste trabalho ocorreu por conta disso.

Os resultados descritos no Capítulo 6 atenderam de forma bastante satisfatória os objetivos previstos para o projeto, sendo que para o ano de 2022 está prevista a expansão do Sistema *IoT* para vários outros centros de trabalho dentro do departamento de Injeção de Alumínio.

Há ainda o interesse de realizar a expansão do projeto para outros departamentos (incluindo em outras unidades da *WEG*), máquinas e processos, que deverão ser devidamente estudadas quanto à sua viabilidade no próximo ano.

Como trabalhos futuros, existe a intenção de envolver outros membros da equipe de Indústria 4.0 da qual faço parte, que desenvolvem trabalhos utilizando Inteligência Artificial e *Machine Learning*. É possível, por exemplo, realizar através da incorporação destas tecnologias, a identificação dos padrões de comportamento normais das variáveis de processo e, caso alguma variável apresente algum comportamento fora destes padrões, alertas seriam enviados às pessoas responsáveis.

De forma geral, este trabalho contribuiu, de forma bastante significativa, com a aproximação e estudo de temas relacionados à Indústria 4.0 e *IoT*, além de diversos assuntos pertinentes ao curso de Engenharia e Controle e Automação, como o de redes de comunicação industrial, programação em diversas linguagens, entre outros.

REFERÊNCIAS

BAHRIN, Mohd Aiman Kamarul; OTHMAN, Mohd Fauzi; AZLI, Nor Hayati Nor; TALIB, Muhamad Farihin. **Industry 4.0: A review on industrial automation and robotic**. v. 78. [S.l.]: Penerbit UTM Press, 2016. P. 137–143. DOI: 10.11113/jt.v78.9285.

BUGHIN, Jacques *et al.* **ARTIFICIAL INTELLIGENCE THE NEXT DIGITAL FRONTIER?** [S.l.], 2017. Disponível em: www.mckinsey.com/mgi..

CAMACHO, Daniel Delgado; CLAYTON, Patricia; O'BRIEN, William J.; SEEPERSAD, Carolyn; JUENGER, Maria; FERRON, Raissa; SALAMONE, Salvatore. Applications of additive manufacturing in the construction industry – A forward-looking review. **Automation in Construction**, Elsevier B.V., v. 89, p. 110–119, mai. 2018. ISSN 09265805. DOI: 10.1016/j.autcon.2017.12.031.

COOPER, Joshua; JAMES, Anne. Challenges for database management in the internet of things. **IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)**, v. 26, p. 320–329, 5 set. 2009. ISSN 02564602. DOI: 10.4103/0256-4602.55275.

HALEEM, Abid; JAVAID, Mohd. Additive Manufacturing Applications in Industry 4.0: A Review. **Journal of Industrial Integration and Management**, World Scientific Pub Co Pte Lt, v. 04, p. 1930001, 04 dez. 2019. ISSN 2424-8622. DOI: 10.1142/s2424862219300011.

KUMAR, Kaushik; ZINDANI, Divya; DAVIM, J Paulo. **SPRINGER BRIEFS IN APPLIED SCIENCES AND TECHNOLOGY MANUFACTURING AND SURFACE ENGINEERING Industry 4.0 Developments towards the Fourth Industrial Revolution**. [S.l.], 2019. Disponível em: <http://www.springer.com/series/10623>.

LEE, Jay; BAGHERI, Behrad; KAO, Hung An. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. **Manufacturing Letters**, Elsevier Ltd, v. 3, p. 18–23, jan. 2015. ISSN 22138463. DOI: 10.1016/j.mfglet.2014.12.001.

MARTINELLI, Arianna; MINA, Andrea; MOGGI, Massimo. **LEM LEM The Enabling Technologies of Industry 4.0: Examining the Seeds of the Fourth Industrial Revolution**, 2019. ISSN 2284-0400.

MCKINSEY GLOBAL INSTITUTE. Industry 4.0 at McKinsey 's model factories: Get ready for the disruptive wave. **McKinsey & Company**, p. 1–12, 2015.

ROBLEK, Vasja; MEŠKO, Maja; KRAPEŽ, Alojz. A Complex View of Industry 4.0. **SAGE Open**, SAGE Publications Inc., v. 6, 2 jun. 2016. ISSN 21582440. DOI: 10.1177/2158244016653987.

SCHUH, Günther; ANDERL, Reiner; DUMITRESCU, Roman; KRÜGER, Antonio. **Industrie 4.0 Maturity Index: Managing the Digital Transformation of Companies**. [S.l.], 2020.

SUNDBLAD, Willem. The Benefits of Connecting Data from Legacy Equipment. **Forbes**, out. 2018. Disponível em: <https://www.forbes.com/sites/willemsundbladeurope/2019/10/01/the-benefits-of-connecting-data-from-legacy-equipment/?sh=79df7e09246b>. Acesso em: 5 out. 2021.

WEF. **Industrial Internet of Things: Unleashing the Potential of Connected Products and Services**. [S.l.], 2015.

WEF. **Insight Report: Readiness for the Future of Production Report 2018**. [S.l.: s.n.], 2018. P. 1–266. ISBN 978-1-944835-16-3. Disponível em: http://www3.weforum.org/docs/FOP_Readiness_Report_2018.pdf.

WEF. Shaping the Future of Production: Four Contrasting Perspectives in 2030. **World Economic Forum**, March, 2017. Disponível em: http://www3.weforum.org/docs/WEF_White_Paper_Shaping_Future_Production_.pdf.

ZHANG, Lin *et al.* Cloud manufacturing: a new manufacturing paradigm. **Enterprise Information Systems**, v. 8, p. 167–187, 2 mar. 2014. ISSN 17517575. DOI: 10.1080/17517575.2012.683812.