

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

EMMANUEL GARBUIO ROMANOVSKI

**IOT: UMA APLICAÇÃO NA INDÚSTRIA DE CORREIAS PARA
TRANSPORTE**

PONTA GROSSA

2021

EMMANUEL GARBUIO ROMANOVSKI

IOT: UMA APLICAÇÃO NA INDÚSTRIA DE CORREIAS PARA TRANSPORTE

IoT: An application in Conveyor Belt industry

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientadora: Profa. Dra. Mauren Louise Sguario Coelho de Andrade.

PONTA GROSSA

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

EMMANUEL GARBUIO ROMANOVSKI

**IOT: UMA APLICAÇÃO NA INDÚSTRIA DE CORREIAS PARA
TRANSPORTE**

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Bacharel em Ciência
da Computação, do Departamento
Acadêmico de Informática, da
Universidade Tecnológica Federal do
Paraná (UTFPR).

Data de aprovação: 23 de novembro de 2021

Prof. Dra. Mauren Louise Sguario Coelho de Andrade
Doutorado
Universidade Tecnológica Federal do Paraná

Prof. Dr. Augusto Foronda
Doutorado
Universidade Tecnológica Federal do Paraná

Prof. Dr. Rui Tadashi Yoshino
Doutorado
Universidade Tecnológica Federal do Paraná

Ponta Grossa

2021

Dedico este trabalho a minha família, pelos
momentos de ausência.

AGRADECIMENTOS

Não há como citar todas as pessoas que me apoiaram nesta fase de grande importância em minha vida. De antemão, peço desculpas as pessoas que não puderam estar presentes nestas palavras, mas elas podem estar certas de que fazem parte do meu pensamento e de minha gratidão.

Agradeço a minha orientadora Profa. Dra. Mauren Louise Sguario Coelho de Andrade, por todo o suporte neste trabalho e fora dele.

Aos meus amigos.

Aos meus colegas da empresa CT Conveying Solutions Brasil.

A Secretaria do Curso, pela cooperação.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

Eu denomino meu campo de Gestão do Conhecimento, mas você não pode gerenciar conhecimento. Ninguém pode. O que você pode fazer, o que a empresa pode fazer é gerenciar o ambiente que otimize o conhecimento.
(Laurence Prusak, 1997).

RESUMO

Os termos e os conceitos de *Internet of Things (IoT)*, Manufatura Inteligente e Indústria 4.0 estão cada vez mais presentes no contexto fabril e, junto com esses conceitos, a demanda de dados aumenta a cada nova tecnologia implementada. Para um melhor controle e rastreabilidade da produção da empresa CT Conveying Solutions Brasil localizada na cidade de Ponta Grossa, é necessário realizar um estudo de caso a fim de conectar as máquinas, junto com o sistema de execução de manufatura, em um ambiente capaz de receber, armazenar, interligar e mostrar os dados de uma forma acessível e fácil. Neste contexto, serão coletados dados de temperatura, velocidade e posição de duas máquinas da produção e que serão enviadas para três ferramentas, o Node-Red que conectará e distribuirá estes dados, o InfluxDB que armazenará e o Grafana como ferramenta para a análise.

Palavras-chave: Internet das coisas. Indústria 4.0. Manufatura Inteligente. Aquisição de Dados.

ABSTRACT

The terms and concepts of Internet of Things (IoT), Manufacturing Intelligence and Industry 4.0 are more and more present into the factory context and, together with these concepts, the data demand increase with each new technology implemented. To a better production control and traceability of CT Conveying Solutions Brazil, is necessary to do a case study to connect the machines, together with the manufacturing execution system, into an environment able to receive, store, interconnect and show the data easily and accessible. In this context, will be collected data as temperature, velocity, and position data from two shopfloor machines, and that will be send to three tools, the Node-red as the distributor and connector of the data, the InfluxDB as the database and the Grafana as the analysis tool.

Keywords: Internet of things. *Industry 4.0. Manufacturing Intelligence. Data Acquisition*

LISTA DE ILUSTRAÇÕES

Figura 1 - Sequência completa do processo	15
Figura 2 - Planta de Conveying Solutions e seus processos.	18
Figura 4 - Dataflow dos dados dentro da infraestrutura da CT	20
Figura 3 - Processo de um controlador	22
Figura 5 - Arquivos do InfluxDB.....	27
Figura 6 - Exemplo de algumas requisições	28
Figura 7 - Comando para a criação de um banco de dados	28
Figura 8 - Comando para a criação de um usuário administrador.....	29
Figura 9 - Nodes do S7 Communication	30
Figura 10 – Atividades configuradas	30
Figura 11 - Configuração do PLC da Calandra	31
Figura 12 - Configuração do PLC da Confecção.....	31
Figura 13 - Status das atividades.....	32
Figura 14 - Configuração do JSON para a Calandra e a Confecção.....	34
Figura 15 - Prompt com informações da Calandra.....	35
Figura 16 - Prompt com informações da Confecção	35
Figura 17 - Fluxo atual no Node-RED	36
Figura 18 – Opções de configurações do banco de dados no Node-RED	37
Figura 19 - <i>Flow</i> dos dados após a integração do banco de dados	37
Figura 20 - Prompt do InfluxDB após a conexão.....	38
Figura 21 - Função "get_mes" para enviarmos a query ao banco.....	38
Figura 22 - Node-RED após implementação.....	39
Figura 23 - Configuração de Data Source no Grafana	40
Figura 24 - <i>Dashboard</i> da Calandra	41
Figura 25 - <i>Dashboard</i> da Confecção	41
Figura 26 - Visualização no padrão Gauge	42
Figura 27 - Gráficos no padrão Stat	43
Figura 28 – Gráfico no padrão Discrete	44
Figura 29 - Visualização alterada para apenas a correia selecionada	44
Figura 30 - Gráfico da velocidade na Calandra	45
Figura 31 - Gráfico de velocidade da linha na Confecção.....	45
Figura 32 - Gráfico da posição dos desbobinadores na confecção.....	45
Quadro 1 - Seleção de máquinas e dados	26
Quadro 2 - Sensores definidos.....	32

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

B.U	Business Units
CAGR	Compound Annual Growth Rate
IoT	Internet of Things
IIoT	Industrial Internet of Things
MI	Manufatura Inteligente
MES	Manufacturing Execution System
PLC	Programmable Logical Controller
T.I	Tecnologia da Informação
API	Application Programming Interface
DNS	Domain Name System
IP	Internet Protocol

SUMÁRIO

1	INTRODUÇÃO	13
1.1	DEFINIÇÃO DO PROBLEMA E MOTIVAÇÃO	15
1.2	OBJETIVOS	15
1.2.1	Objetivo geral	16
1.2.2	Objetivos específicos	16
1.3	ESTRUTURA DO TRABALHO	16
2	REFERENCIAL TEÓRICO	17
2.1	CT CONVEYING SOLUTIONS BRASIL	17
2.2	QUARTA REVOLUÇÃO INDUSTRIAL	18
2.2.1	Industria 4.0 no Brasil	19
2.2.2	Manufatura inteligente	19
2.3	INTERNET DAS COISAS	20
2.3.1	Sensores na manufatura	21
2.3.2	Controlador lógico programável	22
2.3.3	Sistema de execução de manufatura	23
2.3.4	Sistema de gerenciamento de dados	23
2.3.5	Banco de dados	24
2.3.6	Sistema de análise	25
3	DESENVOLVIMENTO	25
3.1	MÁQUINAS	26
3.2	SERVIDOR	26
3.3	INSTALAÇÃO DO INFLUXDB	27
3.3.1	Configuração do banco de dados	28
3.4	COLETANDO DADOS PELO NODE-RED	29
3.4.1	Coleta de dados com o S7 Communication	29
3.4.2	Configuração das máquinas	31
3.4.3	Variáveis	32
3.4.4	Script de endereçamento	34
3.4.5	Conexão com o banco de dados	36
3.5	CONEXÃO COM O MES	38
3.6	CONFIGURAÇÃO DO GRAFANA	40
3.7	VISUALIZAÇÃO DOS DADOS	40
3.7.1	Padrão <i>Gauge</i>	42

3.7.2	Padrão <i>Stat</i>	43
3.7.3	Padrão <i>Discrete</i>	44
3.7.4	Padrão <i>Graph</i>	44
4	CONCLUSÃO	46
5	TRABALHOS FUTUROS	46
	REFERÊNCIAS	47

1 INTRODUÇÃO

Não é de hoje que o ser humano busca o desenvolvimento de novas ferramentas para o benefício próprio. No decorrer da história, a criação de novas tecnologias e de instrumentos de produção, com o objetivo de facilitar desde os processos mais simples até os processos mais complexos se tornou algo comum, decorrente do grande crescimento populacional, e é neste contexto que surgiu a Primeira Revolução Industrial.

A Primeira Revolução Industrial é caracterizada pela mudança de nossa dependência dos animais, esforço humano e fontes primárias de energia para o uso de combustíveis fósseis e a potência mecânica habilitada. A Segunda Revolução Industrial ocorreu entre o século 19 e as duas primeiras décadas do século 20, e trouxe muitos avanços na distribuição de energia, comunicação sem fio/com fio e novas formas de geração de energia. A Terceira Revolução Industrial começou na década de 1950 com o desenvolvimento de sistemas digitais e avanços na capacidade de computação, que permitiram novas formas de geração, processamento e compartilhamento de dados (DAVIS, 2016, p. 2).

Segundo Sarfraz (2021): “A atual situação do COVID-19 acelerou o desenvolvimento da Quinta Revolução Industrial, caracterizado pela customização de serviços em massa e a computação cognitiva”. Isto não significa que a humanidade já dominou todos os conceitos da Quarta Revolução Industrial, que se caracteriza pela utilização de conceitos de Inteligência Artificial, *Big Data*, *Internet of Things* (IoT) dentro da produção.

Nessa revolução, as tecnologias emergentes e as inovações generalizadas são difundidas muito mais rápida e amplamente do que nas anteriores, visto que, por exemplo, a segunda revolução industrial ainda precisa ser vivida por 17% da população mundial, pois quase 1.3 bilhões de pessoas ainda não têm acesso à eletricidade. Isto também é válido para a Terceira Revolução Industrial, já que mais da metade da população não tem acesso à internet (SCHWAB, 2017).

O jornal Facts & Factors (2020) menciona que “o mercado de Indústria 4.0 está estimado em \$ 83 bilhões em 2020 e estima-se que chegará à marca de \$ 267 bilhões até 2026. Espera-se que a Indústria 4.0 cresça no mercado global a uma taxa composta anual de 17% de 2019 a 2026.”

Neste contexto, a Manufatura Inteligente (MI) se encaixa na otimização da linha de produção, ou seja, ela está diretamente ligada com a transformação da produção após a aplicação de novas tecnologias, sendo capaz de combinar os três

pilares de inteligência da manufatura: a inteligência das pessoas, das máquinas e dos processos. Sendo assim, a MI consegue melhorar os resultados do chão de fábrica por meio da otimização da utilização dos recursos e dos custos de fabricação ou até mesmo na agilidade do processo.

Pederneiras (2019) relata que “o alto volume de dados que a produção e as máquinas geram por minuto, [...] torna o *Big Data* como um dos principais pilares da Quarta Revolução Industrial”.

Grandes corporações armazenam múltiplos *petabytes* e mesmo pequenas e médias empresas trabalham com dezenas de *terabytes* de dados. Este volume de dados tende a crescer geometricamente e, em um mundo cada vez mais competitivo e rápido, as empresas precisam tomar decisões baseadas não apenas em palpites, mas em dados concretos (TAURION, 2013).

Em conjunto com o *Big Data*, temos os sensores como uma das maiores fontes de dados dentro do ambiente fabril, enviando informações importantes, como por exemplo temperatura, pressão e tensão, para uma matriz.

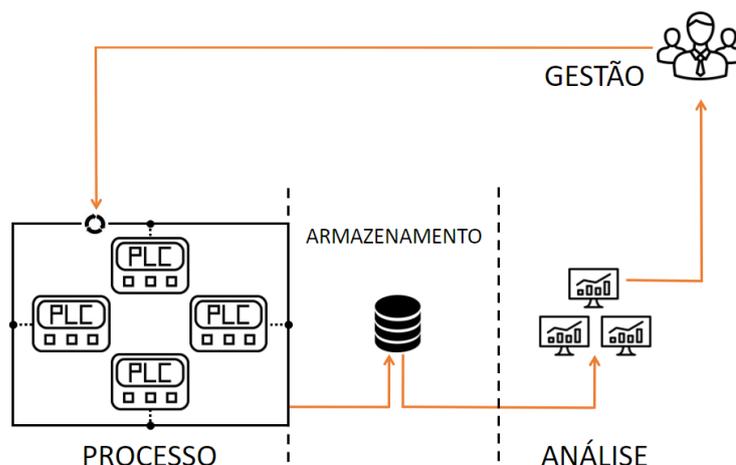
Nos dias atuais, não é possível pensar em um ambiente de Indústria 4.0 sem a utilização destes sensores monitorando de ponta a ponta o processo, e juntamente com os conceitos de *Industrial Internet of Things*, ou IloT, temos o começo da implementação da Manufatura Inteligente.

O IloT pode ser uma conexão entre o PLC e uma plataforma de gerenciamento de dados por meio da utilização da Internet, como ilustrado na Sequência completa do processo (Figura 1). Existem vários sistemas que permitem fazer esta comunicação, os mais utilizados são:

- Node-RED: Originalmente criada pela IBM, é uma ferramenta baseada em fluxos de ações para conexões entre API, hardwares, dispositivos e serviços online.
- Kepware: Plataforma licenciada criada pela *PTC Technology*, é uma ferramenta capaz de conectar todos os dados industriais e distribuí-los conforme o necessário.

Ao estabelecer esta conexão, a manufatura já está disposta a armazenar os dados, podendo ser tanto os dados das máquinas quanto os dados do sistema MES, utilizando bancos de dados como o PostgreSQL, SQL Server ou até mesmo o InfluxDB.

Figura 1 - Sequência completa do processo



Fonte: Autoria própria (2021)

Por fim, é necessário entregar estes dados de uma forma fácil de se compreender. No mercado temos solução como o Microsoft Power BI, Prometheus e o Grafana que mostram as informações em formato de gráficos, desta forma, a gestão consegue identificar os erros no processo e torná-lo mais rápido e eficiente.

1.1 Definição do problema e motivação

Dentro da fábrica da CT Conveying Solutions Brasil situada na cidade de Ponta Grossa, o acesso direto ao PLC da máquina é a única forma de se receber e visualizar os dados, sendo assim, não existe um histórico e uma visualização fácil das informações. Em relação aos dados do processo, o sistema MES da fábrica atende as necessidades, porém quando você correlaciona os dados do MES com os dados das máquinas, o processo se torna mais rastreável no quesito erros.

Com a implementação de um sistema de gerenciamento de dados e um sistema para visualização, será possível introduzir os conceitos de Internet das Coisas, Manufatura Inteligente e Indústria 4.0 dentro da fábrica.

1.2 Objetivos

De acordo com a definição do problema, segue o objetivo geral e os objetivos específicos do projeto:

1.2.1 Objetivo geral

Este trabalho tem como objetivo geral implementar a coleta dos dados de temperatura, velocidade e posição, armazená-los para a análise de dados e criar *dashboards* interativos para a manufatura da fábrica da CT Conveying Solutions Brasil em Ponta Grossa.

1.2.2 Objetivos específicos

Do objetivo geral, criam-se as seguintes metas a serem atingidas:

- Analisar e escolher soluções para a comunicação entre o sistema e o PLC das máquinas, para o armazenamento dos dados e para a análise das informações;
- Implementar as soluções escolhidas no servidor local da CT Conveying Solutions Brasil;
- Estabelecer a comunicação com os PLCs;
- Implementar o gerenciamento dos dados para o armazenamento;
- Conectar o armazenamento no sistema de análise de dados;
- Conectar o sistema de gerenciamento da produção no sistema de análise de dados;
- Criar os dashboards de acordo com a demanda.

1.3 Estrutura do trabalho

No capítulo 1, foi feita a introdução ao tema, comentando um pouco sobre o início da Revolução Industrial até os dias de hoje, sobre alguns pilares da Indústria 4.0 e sobre a Manufatura Inteligente.

No capítulo 2, foi feito um avanço nos principais tópicos do trabalho apresentando a CT, entendendo um pouco sobre o MES que a fábrica de Conveying Solutions trabalha e entrando em tópicos mais específicos sobre os sensores, PLC e softwares utilizados no projeto.

No capítulo 3, será apresentado o desenvolvimento do projeto, explorando um pouco de cada sistema configurado e desenvolvido.

No capítulo 4 será estendido as fronteiras do projeto, comentando um pouco sobre o status atual do mesmo.

Por fim, no capítulo 5 segue uma proposta para os próximos trabalhos.

2 REFERENCIAL TEÓRICO

Este capítulo traz um pouco sobre a CT Conveying Solutions Brasil, sobre a Indústria 4.0 no geral e sobre algumas ferramentas que foram usadas para a conclusão do projeto.

2.1 CT Conveying Solutions Brasil

Criada em 1999 em Ponta Grossa, a CT hoje produz soluções de borracha para o transporte de minérios.

Especificamente na fábrica onde este projeto foi implementado, a produção se divide em 4 processos, como ilustrado na Figura 2:

- **Confecção:** Processo que confecciona os tecidos que podem ou não estar dentro de cada correia.
- **Calandra:** Processo de fabricação do formato da borracha.
- **Vulcanização:** Processo responsável por colocar duas mantas de borracha, uma sobre a outra, podendo ou não ter tecido ou cabos de aço dentro, para ser vulcanizado e se transformar em uma correia.
- **Inspeção:** Processo em que os operadores verificam se há algum defeito na correia, caso haja, é feito a manutenção.

Figura 2 - Planta de Conveying Solutions e seus processos.



Fonte: Autoria própria (2021)

Neste projeto será utilizado apenas os dados da Confeção e da Calandra.

2.2 Quarta revolução industrial

“Ao permitir ‘fábricas inteligentes’, a Quarta Revolução Industrial cria um mundo onde os sistemas físicos e virtuais de fabricação cooperam de forma global e flexível. Isso permite a total personalização de produtos [...]”, diz Schweb (2017).

Neste sentido, muitas empresas começaram a investir em soluções para as demandas antigas da manufatura. Como resultado, diversas tecnologias acabaram se tornando destaque no mercado, tais como:

- **Big Data Analytics:** Possibilita a verificação de uma vasta gama de dados a fim de descobrir padrões, erros, tendências e qualquer outro tipo de informação necessária.
- **Computação em Nuvem:** Possibilita alocar sistemas, arquivos e dados em um servidor não local com uma segurança adequada e em alguns casos mais barato do que um servidor e/ou computador.
- **Inteligência Artificial:** Sistema desenvolvido para possibilitar a tomada de decisões, prever dados precisos ou detectar erros sem a intervenção humana é muito alta, porém essa tecnologia consegue agilizar processos e substituir trabalhos que normalmente se tornavam cansativos para os humanos.
- **Manufatura Aditiva:** Por meio da utilização de impressoras 3D, uma manufatura que possui muitas peças pequenas na sua estrutura consegue imprimir estas peças com um custo menor.

- Internet das Coisas: Esse conceito permite que máquinas e/ou equipamentos transfiram dados entre si utilizando a conexão a Internet.

2.2.1 Indústria 4.0 no Brasil

Segundo a FIRJAN (2016) “grande parte da indústria brasileira está em movimento entre a segunda e a terceira revolução industrial, sendo que aproximadamente apenas 48% das indústrias brasileiras estão com o conceito “4.0”. Neste caso, existem indústrias que ainda estão nos conceitos de linhas de montagem e que não possuem nenhuma automação em sua manufatura.

No Brasil, 43% das empresas consultadas pela CNI (Confederação Nacional da Indústria) não identificam quais tecnologias têm potencial para alavancar a competitividade do setor industrial. Nas pequenas empresas, esse percentual sobe para 57%. Entre as grandes, a fatia recua para 32% (Pequenas empresas e Grandes negócios, 2016).

A implantação da Indústria 4.0 no cenário brasileiro traz desafios como:

- 1) A construção de políticas estratégicas, incentivos governamentais;
- 2) A reunião de empresários e gestores com postura proativa;
- 3) O desenvolvimento tecnológico e formação de profissionais, próximos à indústria.

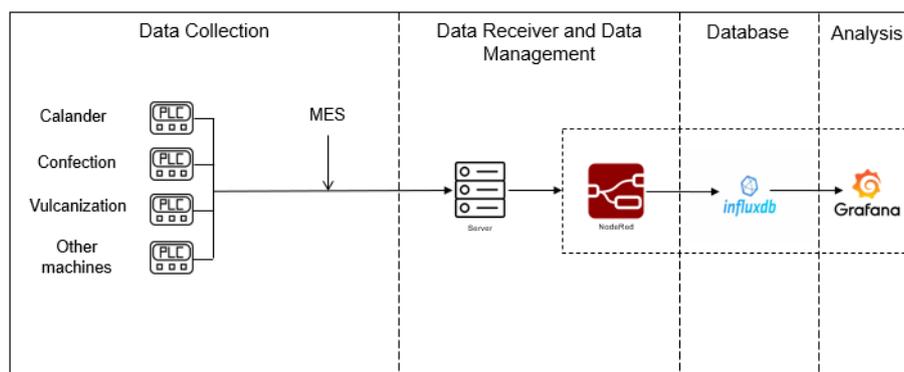
Em 2016 o Governo Federal por meio do Ministério da Indústria, Comércio Exterior e Serviços, e da Associação Brasileira de Desenvolvimento Industrial, lançou a Agenda Brasileira para a Indústria 4.0, sendo um conjunto de iniciativas que visam promover o desenvolvimento da Indústria 4.0.

2.2.2 Manufatura inteligente

Um grande avanço para o começo de qualquer implementação relacionada a manufatura inteligente é já possuir algum método de rastreabilidade da produção, seja ela utilizando um sistema de execução de manufatura ou até mesmo uma planilha no Excel. Deste modo, quando implementado a conexão com o PLC, é possível interligar os dados dos produtos com os dados da máquina.

Com base nessa descrição, a implementação desse projeto dentro da manufatura da CT pode ser observada na Figura 4.

Figura 3 - Dataflow dos dados dentro da infraestrutura da CT



Fonte: Autoria própria (2021)

2.3 Internet das coisas

O mundo inteiro hoje é conectado por sensores e equipamentos de informação, podendo a comunicação ser através de cabeamento (*wired*) ou sem cabeamento (*wireless*). Todas essas interligações de informações resultaram na chamada Internet das Coisas (*Internet of Things*, também conhecido por IoT), “[...] um termo comumente utilizado para tecnologias e sistemas associados a emergente onda de ‘coisas’ baseadas no ambiente físico conectadas a *Internet*.” (HOLLER, 2014, p.14, tradução nossa).

Quando colocado o contexto de *IoT* dentro de uma fábrica, a conexão das máquinas com os sensores que o controlam dentro do ambiente da rede industrial acabou criando uma sub categoria, chamada de Internet das Coisas Industrial (*Industrial Internet of Things*, também conhecido por *IIoT*).

“[...] Isso será benéfico para um processo de manufatura eficiente dentro do ambiente industrial, também ajudando no planejamento de manutenção preventiva dos maquinários. Outro benefício é a identificação de erros na produção o mais rápido possível [...]. Indústria 4.0 e *IIoT* são usados para compartilhamento de complexas tarefas, tomada de decisão baseada nos dados coletados e acesso remoto ao maquinário. Conectividade massiva das ‘coisas’ e a possibilidade de compartilhamento/coleta de dados promove a segurança para os conceitos de *IIoT* e Indústria 4.0.” (BUTUN, 2020, p. 3, tradução nossa)

Para o funcionamento do *IIoT*, são normalmente requeridos três fatores indispensáveis. São eles:

- Rede: Diz respeito a forma em que os dados serão conectados. Neste caso, os dados serão conectados via cabeamento na rede industrial da fábrica.
- Equipamentos: São os equipamentos que estarão conectados entre a rede e os sistemas, como por exemplo os controladores lógicos programáveis que controlam as máquinas da produção e os sensores que mandam os sinais para estes controladores.
- Sistemas: A quantidade de sistemas necessários é variável, podendo ter apenas um sistema capaz de receber os dados, armazenar e mostrar os dados ou também três sistemas independentes.

2.3.1 Sensores na manufatura

A fim de gerir a qualidade da produção e gerenciar a manutenção dos equipamentos, os sensores são utilizados para o monitoramento das atividades das máquinas, sendo capazes de transformar valores físicos, por exemplo temperatura e pressão, em sinais elétricos. Dessa forma, por meio de um controlador, é possível receber essas informações digitalmente.

Atualmente existem mais de 6 tipos de sensores industriais, sendo cada tipo responsável por captar um estado físico e transformá-los em dados. Segue uma lista de alguns deles:

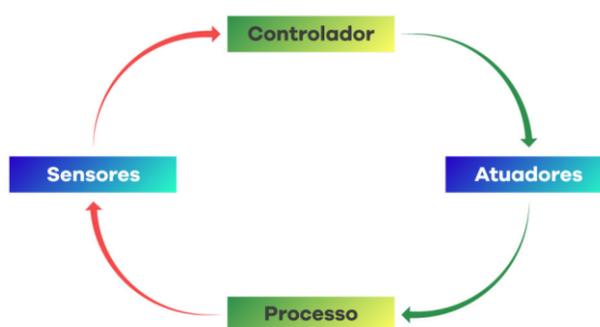
- Sensores Acústicos: recebem informações como a onda (amplitude, fase, polarização), velocidade da onda e espectro.
- Sensores Elétricos: recebem informações como carregamento, corrente e condutividade.
- Sensores Magnéticos: recebem informações como o campo magnético, fluxo magnético e a permeabilidade.
- Sensores Óticos: recebem informações como o índice de refração, absorção, refletividade e a onda (amplitude, fase, polarização).
- Sensores Térmicos: recebem informações como temperatura, fluxo, calor condutividade térmica.
- Sensores Mecânicos: recebem informações como aceleração, força, densidade e posição.

Todas as informações podem ser recebidas por meio de um conjunto de sensores ou até mesmo por apenas um sensor capaz de receber vários estados físicos.

2.3.2 Controlador lógico programável

O Controlador Lógico Programável (PLC – *Programmable Logic Controller*) é um computador capaz de controlar as máquinas e diferentes aplicações dentro da manufatura. Como ilustrado na Figura 3, este computador funciona recebendo informações de uma rede de sensores conectados a ele, processando todos estes dados por meio de uma lógica pré-definida e retornando um estado a um atuador da máquina que também está conectada a ele .

Figura 4 - Processo de um controlador



Fonte: Macrotec (2021)

Para ser possível essa comunicação, o controlador necessita ter módulos de entrada/saída (I/O). Estes módulos permitem a comunicação do cérebro do controlador (CPU) com as informações obtidas pelos sensores para posteriormente transformá-las em sinais de comando para a máquina.

Dependendo do modelo do PLC, pode-se possuir diferentes tipos de entradas e saídas, por um exemplo as entradas digital e analógica (*analog*). Cada um destes tipos define qual sensor este PLC está habilitado a conectar.

A CT possui vários modelos de PLC, dentre eles o SIEMENS SIMATIC S7-1500, o qual possui 8 entradas e 6 saídas do tipo digital e apenas 2 entradas analógicas.

Os PLCs utilizam o protocolo IPv4 para a comunicação na rede. Este protocolo usa um endereçamento 32-bit que equivale a 4 bilhões de endereços

disponíveis para o uso. Ele é composto por 4 grupos de dois dígitos cada um, podendo variar de 0 a 255. Como um exemplo, temos o endereço do Google 216.58.202.132.

2.3.3 Sistema de execução de manufatura

Um sistema de execução de manufatura (MES, *Manufacturing Execution System*) é um sistema que gerencia toda a produção da fábrica. Utilizando-o, é possível ter uma rastreabilidade de cada processo e produto, permitindo que o gestor avalie o planejamento e acompanhe seu desenvolvimento. É interessante destacar que algumas grandes empresas, mesmo nos Estados Unidos e na Europa, ainda utilizam o Microsoft Excel para gerar relatórios da produção, o que não é uma boa prática, já que provavelmente estes dados são inseridos manualmente.

Uma rastreabilidade adequada na manufatura é importante para que caso ocorra qualquer erro dentro do chão de fábrica relacionado ao processo, o responsável consiga corrigir essas falhas antes que prazo ou a qualidade do produto sejam afetadas.

Para um MES gerir a manufatura com qualidade, é necessário ela seguir duas premissas:

- 1) Robustez: permitir que o sistema funcione 24 horas por 7 dias (quando necessário), gerando sempre informações em tempo real;
- 2) Flexibilidade: na chegada de novas demandas, o sistema deve conseguir ser ampliado e/ou se adaptar à nova demanda, sem perder a robustez;

Seguindo essas regras, o sistema será contínuo e que retornará um grande volume de dados, gerando um histórico completo de produção.

A CT atualmente utiliza o sistema da Finer, uma empresa local de Ponta Grossa que criou um MES customizável para a B.U.

2.3.4 Sistema de gerenciamento de dados

Existem métodos manuais para se conseguir retirar os dados dos sensores de dentro dos controladores, por exemplo utilizando o próprio software do PLC. A desvantagem desse método é que não é nada prático, visto que toda vez que surgem demandas é necessário acessar o software do controlador para visualizar os dados.

Neste contexto, ter um sistema que seja capaz de se conectar à máquina e que automaticamente retire os dados para distribuí-los conforme o necessário vem contribuir para o aumento da rastreabilidade.

A fim de sanar as necessidades da empresa, existem duas soluções possíveis: seguir com a utilização do Kepware ou Samsara, ambos softwares robustos de gerenciamento de dados com licenciamento pago ou envolver a utilização de um software gratuito, neste caso o Node-RED. As três soluções suprem as necessidades, porém por questão de demanda financeira neste trabalho, será utilizado o software gratuito Node-RED.

Originalmente desenvolvido pela *IBM Emerging Technology* para criar conexões entre dispositivos de IoT (*Internet of Things*), o Node-RED é uma ferramenta visual de código aberto que atualmente se estende na conexão de APIs, hardwares, o já mencionado IoT etc. A plataforma funciona em qualquer sistema operacional, o que é uma vantagem quando alocada em um servidor que possui muitas restrições.

Em relação a utilização da ferramenta, o Node-RED funciona como um fluxo de aplicações, sendo possível arrastar atividades para este fluxo. Por padrão o software vem com 40 atividades, porém é possível baixar novas aplicações de acordo com a demanda.

2.3.5 Banco de dados

O conceito de banco de dados foi criado a fim de suprir a necessidade de armazenar e agrupar dados de forma segura e consistente. Dessa forma, é capaz de resgatar essas informações com o nível de confiabilidade maior.

Hoje existem vários bancos de dados disponíveis para a utilização, muitos deles gratuitos e outros com um licenciamento. Para a aplicação desse projeto, foram selecionadas três plataformas para a análise: o PostgreSQL, o InfluxDB e o Microsoft SQL Server 2013.

Após uma análise detalhada junto com a equipe de Tecnologia da Informação local da empresa, foi escolhido seguir com a utilização do InfluxDB, já que a comunicação entre ele, o Node-RED e a plataforma de análise de dados escolhida são de fácil implementação e possui uma licença gratuita.

2.3.6 Sistema de análise

Como último recurso necessário para a implementação da Manufatura Inteligente, o sistema de análise tem como finalidade receber os dados resgatados e armazenados no banco de dados, de modo que eles sejam visualizados de forma organizada e fácil.

Para isso foram identificados dois sistemas capazes de resolver o problema de análise, o Grafana que possui licenciamento gratuito para as demandas da empresa, o Prometheus que também é gratuito e o Microsoft Power BI que necessita de licença para atualizar automaticamente os dados. Dentro da CT, já se utiliza o Grafana no sistema MES, por este motivo foi optado por seguir com a utilização dele neste projeto, assim mantendo um padrão de sistema.

Este recurso utiliza sistema de query para resgatar os dados, isto é, uma requisição que é enviada para o banco de dados a fim de receber uma tabela de dados com as informações requeridas. Normalmente se utiliza uma sintaxe como por exemplo “SELECT * FROM exemplo WHERE ‘id_exemplo’ = 0”, que traduzindo significa que o usuário está pedindo ao banco de dados que ele retorne as informações da tabela exemplo com todas os campos em que o id_exemplo seja 0.

Essas queries são feitas em um gráfico que se localiza dentro de um dashboard. Um *dashboard* é uma página que contém vários gráficos juntos a fim de conectá-los para uma visualização mais fácil.

Com todos os conceitos já apresentados, o desenvolvimento do projeto pode ser iniciado.

3 DESENVOLVIMENTO

Este capítulo traz um pouco sobre o desenvolvimento do projeto dentro do ambiente corporativo, passando pelas configurações dos sistemas Node-RED, InfluxDB e Grafana, pela configuração dos fluxos de informações dentro do Node-RED e terminando na criação dos gráficos no Grafana.

3.1 Máquinas

Para iniciar o projeto, foi necessário conversar com as equipes de manutenção e engenharia para definir quais máquinas começaríamos e quais informações seriam interessantes resgatar.

Foi acordado começarmos com apenas 2 máquinas e 16 dados, conforme o Quadro 1.

Quadro 1 - Seleção de máquinas e dados

Máquinas	Dados
1. Calandra	Temperatura da borracha na saída da Calandra
	Temperatura da borracha na saída do Resfriador
	Temperatura dos cilindros (1-4)
	Velocidade dos cilindros (1-4)
2. Confeção	Velocidade da linha
	Posição de alinhamento do desbobinador (0-4)

Fonte: Autoria própria (2021)

A partir disso, foi possível iniciar a rastreabilidade da produção nos processos de Calandragem e confecção.

3.2 Servidor

Todos os sistemas utilizados no projeto foram instalados dentro de um servidor local, visto que é uma exigência da equipe de cyber segurança da empresa. A equipe de T.I local seguiu um padrão interno para nomear o servidor, este nome funciona como um DNS (*Domain Name System*), um sistema que mapeia o endereço do servidor na rede, dessa forma acessaríamos o servidor pelo nome e não pelo IP.

Por questões de segurança, o servidor passa a ser chamado de 'Server017'. Abaixo estão algumas configurações do servidor:

- Sistema Operacional: Windows Server 2016 x64.
- Processador: Intel® Xeon® Gold 6152.
- Memória RAM: 16 GB

Em um primeiro momento, foram disponibilizados apenas 8 GB de RAM, porém como a demanda de implementações, de quantidade de dados recebidos e a

de quantidade de dados enviados foi consideravelmente aumentada, foi necessário adicionar 8 GB.

As versões iniciais dos sistemas que foram instalados dentro deste servidor são:

- Node-RED v1.0.6.
- InfluxDB v1.8.0.
- Grafana v7.0.4.

Neste aspecto, todas as implementações descritas neste documento foram baseadas nessas versões dos softwares.

Por fim, foram necessárias as liberações das portas 3000 e 1880 para o Grafana e o Node-RED, respectivamente. Todos os softwares utilizados seguiram com a instalação e configuração padrão.

3.3 Instalação do InfluxDB

Na instalação do InfluxDB todas as configurações utilizadas seguiram o padrão de especificações do sistema. A Figura 5 ilustra os arquivos criados ao final da instalação.

Figura 5 - Arquivos do InfluxDB

4/12/2020 5:...	influx.exe	Application	51,814 KB
4/12/2020 5:...	influx_inspect.exe	Application	18,823 KB
4/12/2020 5:...	influx_stress.exe	Application	11,340 KB
4/12/2020 5:...	influx_tsm.exe	Application	20,236 KB
4/12/2020 5:...	influxd.exe	Application	63,773 KB
7/17/2020 3:...	influxdb.conf	CONF File	21 KB

Fonte: Autoria própria (2021)

Os arquivos Influxd.exe e o Influx.exe, respectivamente, são responsáveis por inicializar o servidor e realizar a criação do banco de dados e das tabelas necessárias para armazenar as informações.

Iniciando o arquivo Influxd.exe, são mostrados no prompt de comando todas as requisições de leitura e escrita que estão chegando ao servidor do banco de dados. A Figura 6 nos mostra alguns exemplos de requisições que o servidor recebe constantemente.

Figura 6 - Exemplo de algumas requisições

```

C:\Users\uic84782\Desktop\InfluxDB\InfluxDB\influx.exe
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffa941a-f5e7-11eb-9d89-005056a63f9f 4998
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffabb2a-f5e7-11eb-9d8b-005056a63f9f 7004
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffabb2a-f5e7-11eb-9d8a-005056a63f9f 7004
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffae249-f5e7-11eb-9d8c-005056a63f9f 6003
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffae249-f5e7-11eb-9d8d-005056a63f9f 6003
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffae249-f5e7-11eb-9d90-005056a63f9f 8006
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffae249-f5e7-11eb-9d8f-005056a63f9f 8006
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dffae249-f5e7-11eb-9d8e-005056a63f9f 8006
2021-08-05T12:23:00.909367Z info Executing query {"log_id": "0VRfxt9G000", "service": "query", "query": "SELECT 1
ast(Process) FROM CS_Data.autogen.CONFECCAO_FINER WHERE (Process = '-5.069') AND time >= now() - 6h GROUP BY time(15s)"}
[httpd] 10.204.147.68 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data HTTP/1.1" 204 0 "-" "python-req
uests/2.24.0" dfff761d-f5e7-11eb-9d91-005056a63f9f 2006
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" dfff761d-f5e7-11eb-9d92-005056a63f9f 3006
2021-08-05T12:23:00.939369Z info Executing query {"log_id": "0VRfxt9G000", "service": "query", "query": "SELECT 1
ast(Process) FROM CS_Data.autogen.CONFECCAO_FINER WHERE (Process = '-5.07') AND time >= now() - 6h GROUP BY time(15s)"}
2021-08-05T12:23:00.964376Z info Executing query {"log_id": "0VRfxt9G000", "service": "query", "query": "SELECT 1
ast(Process) FROM CS_Data.autogen.CONFECCAO_FINER WHERE (Process = '-5.071') AND time >= now() - 6h GROUP BY time(15s)"}
[httpd] 127.0.0.1 - NodeRedUser [05/Aug/2021:09:23:00 -0300] "POST /write?db=CS_Data&p=%5BREDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" e008ec3e-f5e7-11eb-9d93-005056a63f9f 3001

```

Fonte: Autoria própria (2021)

3.3.1 Configuração do banco de dados

A criação do banco de dados é feita por meio do arquivo Influx.exe. Toda a configuração do usuário, banco de dado e tabela é realizada no prompt de comando do próprio sistema. O comando padrão para a criação do banco de dados é “CREATE DATABASE ‘<database_name>’”, demonstrado na Figura 7.

A fim de deixar um padrão de nome para todos os bancos, caso seja necessário a criação de um banco a mais no futuro, o nome deverá começar com “CS_” (sigla de *Conveying Solution*, nome da B.U em que o projeto está sendo implementado), seguido do nome de identificação que foi definido como “Data”.

Figura 7 - Comando para a criação de um banco de dados

```

C:\Users\uic84782\Desktop\InfluxDB\InfluxDB\influx.exe
Connected to http://localhost:8086 version 1.8.0
InfluxDB shell version: 1.8.0
> CREATE DATABASE "CS_Data"

```

Fonte: Autoria própria (2021)

Após a criação, é necessário usar o comando “USE CS_Data” para entrar no ambiente de programação das tabelas.

Como o banco de dados será acessado pelo Node-RED, é necessário a criação de um usuário administrador exclusivo para a escrita no banco. O comando

padrão do InfluxDB para a criação de um usuário administrador é “*CREATE USER <user> WITH PASSWORD '<password>' WITH ALL PREVILEGES*”, demonstrado na Figura 8. Neste caso este usuário terá acesso a todas as funcionalidades e tabelas.

Figura 8 - Comando para a criação de um usuário administrador

```
Connected to http://localhost:8086 version 1.8.0
InfluxDB shell version: 1.8.0
> use CS_Data
Using database CS_Data
> CREATE USER NodeRedUser WITH PASSWORD "MinhaSenha" WITH ALL PREVILEGES_
```

Fonte: Autoria própria (2021)

Apenas com estas configurações já é possível começar uma implementação dentro do Node-RED.

3.4 Coletando dados pelo Node-Red

Dentro da produção existem vários modelos de PLC conectados nas máquinas, entretanto todos eles são *S7* da SIEMENS. Desta forma, foi necessário achar alguma biblioteca capaz de se comunicar com o PLC dentro do Node-RED. A biblioteca *S7 Communication* permite a retirada destes dados do controlador, passando apenas o IP do PLC. É possível baixá-lo utilizando o comando “`npm install atividade-red-contrib-s7`” no prompt de comando.

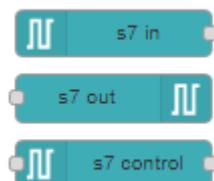
3.4.1 Coleta de dados com o S7 Communication

Para acessar o Node-RED fora do servidor, é possível utilizar o link “`Server017:1880`” no navegador.

Na figura 9 é possível visualizar as atividades que aparecerão quando a biblioteca for instalada. Abaixo está uma breve descrição de cada atividade:

- *s7 IN*: Atividade responsável pela leitura dos dados do PLC.
- *s7 OUT*: Atividade responsável pela escrita de dados no PLC.
- *s7 Control*: Esta atividade habilita controles avançados da conexão e do PLC.

Figura 9 - Nodes do S7 Communication



Fonte: Autoria própria (2021)

Ao arrastar a atividade “s7 IN” para o ambiente de trabalho, é necessário alterar algumas configurações:

- PLC: Configurações de comunicação com o PLC e das variáveis que serão resgatadas.
- *Mode*: Esta função permite a escolha de como os dados serão resgatados.

Ele possui três opções:

- *Single variable*: É possível escolher apenas uma variável já configurada para receber os dados na variável msg.payload.
 - *All variables, one per message*: Recebe os dados de todas as variáveis configuradas, porém um por vez na variável msg.payload.
 - *All variables*: Carrega todas as informações de uma vez só no msg.payload.
- *Name*: Função opcional para renomear a atividade.

Por serem muitas variáveis que o sistema irá resgatar, foi optado por utilizar a opção “*All variables*”.

Figura 10 – Atividades configuradas



Fonte: Autoria própria (2021)

Na Figura 10, como definido anteriormente, temos um fluxo para cada máquina, neste caso um total de 2 atividades configuradas.

3.4.2 Configuração das máquinas

Dentro da opção PLC, é preciso realizar algumas configurações para a comunicação. Essas definições envolvem o IP do controlador, o tempo de requisição e o tipo de transporte dos dados.

Nas Figuras 11 e 12, é possível verificar que o protocolo para acessar o controlador é o *Ethernet (ISO-on-TCP)*, definido pelos endereços de IP e porta 10.204.008.50:102 e 10.204.008.10:102 para a Calandra e a Confecção, respectivamente. Por questões de segurança, os verdadeiros endereços de IP foram alterados.

Figura 11 - Configuração do PLC da Calandra

The screenshot shows the 'Properties' window for a PLC connection. The 'Connection' tab is active. The 'Transport' is set to 'Ethernet (ISO-on-TCP)'. The 'Address' is '10.204.008.50' and the 'Port' is '102'. The 'Mode' is 'Rack/Slot', with 'Rack' set to '0' and 'Slot' set to '2'. The 'Cycle time' is '500 ms', the 'Timeout' is '1000 ms', and the 'Debug' level is 'Default (command line)'. The 'Name' is 'CALANDRA'.

Fonte: Autoria própria (2021)

Figura 12 - Configuração do PLC da Confecção

The screenshot shows the 'Properties' window for a PLC connection. The 'Connection' tab is active. The 'Transport' is set to 'Ethernet (ISO-on-TCP)'. The 'Address' is '10.204.008.10' and the 'Port' is '102'. The 'Mode' is 'Rack/Slot', with 'Rack' set to '0' and 'Slot' set to '1'. The 'Cycle time' is '500 ms', the 'Timeout' is '1000 ms', and the 'Debug' level is 'Default (command line)'. The 'Name' is 'CONFECCAO'.

Fonte: Autoria própria (2021)

Em “*Cycle Time*”, foi definido meio segundo para a requisição de dados ao PLC, isso é, o sistema irá receber uma nova informação a cada 0,5 segundos, um

ótimo tempo visto que um dos objetivos é ter um histórico de informações, dessa forma teremos uma rastreabilidade consistente e, quando eventualmente algum erro ocorrer, será possível ver com mais detalhes as últimas informações obtidas.

Para verificar se a conexão entre o Node-RED e o controlador funcionou, basta apertar o botão de “*Deploy*” do sistema, se posteriormente o status da atividade for “online”, como demonstrado na Figura 13, houve a conexão e a atividade está apta para iniciar a implementação das variáveis.

Figura 13 - Status das atividades



Fonte: Autoria própria (2021)

3.4.3 Variáveis

Com a conexão estabelecida, é preciso especificar os endereços das variáveis dos sensores que estão configuradas dentro do controlador. Esta tarefa de configuração dos endereços atualmente é feita pela equipe de manutenção da fábrica dentro do software responsável pela programação dos controladores.

Os endereços seguem uma lógica de acordo com o tipo de variável. Por um exemplo, quando a variável é do tipo flutuante, ele recebe o endereço “DBx,REALy”, sendo x e y números inteiros.

A equipe responsável forneceu os seguintes endereços especificados no Quadro 2.

Quadro 2 - Sensores definidos

Máquina	Endereço	Sensor	Nome definido
1. Calandra	DB210, REAL326	Velocidade do cilindro 1	VELOCIDADE_CILINDRO_1
	DB210, REAL338	Velocidade do cilindro 2	VELOCIDADE_CILINDRO_2
	DB210, REAL350	Velocidade do cilindro 3	VELOCIDADE_CILINDRO_3

	DB210, REAL362	Velocidade do cilindro 4	VELOCIDADE_CILINDRO_4
	DB210, REAL10	Temperatura do cilindro 1	TEMPERATURA_CILINDRO_1
	DB210, REAL22	Temperatura do cilindro 2	TEMPERATURA_CILINDRO_2
	DB210, REAL34	Temperatura do cilindro 3	TEMPERATURA_CILINDRO_3
	DB210, REAL46	Temperatura do cilindro 4	TEMPERATURA_CILINDRO_4
	DB296, REAL20	Temperatura da borracha na saída da Calandra	TEMPERATURA_BORRACHA_SAIDA_CALANDRA
	DB297, REAL20	Temperatura da borracha na saída do resfriador	TEMPERATURA_BORRACHA_SAIDA_RESFIADOR
2. Confeção	DB58, REAL750	Velocidade da linha	VELOCIDADE_LINHA
	DB58, REAL624	Posição de alinhamento do desbobinador 0	POSICAO_ALINHAMENTO_DESB_0
	DB58, REAL16	Posição de alinhamento do desbobinador 1	POSICAO_ALINHAMENTO_DESB_1
	DB58, REAL90	Posição de alinhamento do desbobinador 2	POSICAO_ALINHAMENTO_DESB_2
	DB58, REAL168	Posição de alinhamento do desbobinador 3	POSICAO_ALINHAMENTO_DESB_3
	DB58, REAL250	Posição de alinhamento do desbobinador 4	POSICAO_ALINHAMENTO_DESB_4

Fonte: Autoria própria (2021)

Os endereços serão registrados dentro da janela “Variables” da atividade das máquinas.

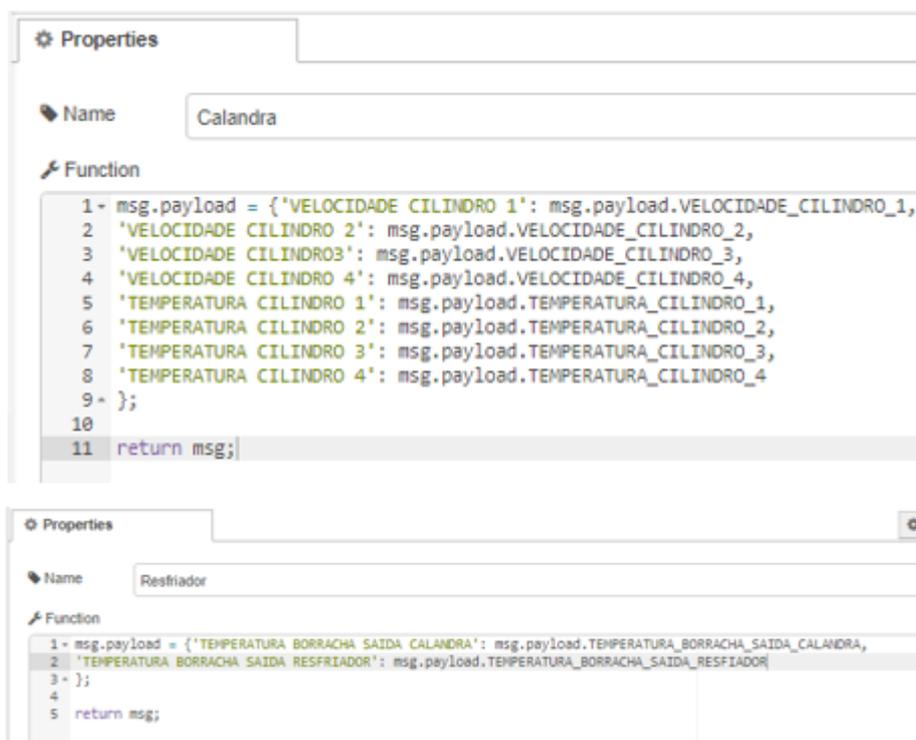
3.4.4 Script de endereçamento

O endereçamento dos dados para a tabela é feito por um script utilizando a atividade “*FUNCTION*” que já vem por padrão no Node-RED. Como definido, o modo de coleta dos dados é “All variables”, neste caso os dados são todos carregados na variável local `msg.payload`.

Para conseguir resgatar o valor de uma variável, é preciso utilizar a variável `msg.payload` e adicionar “.NOME_DA_VARIÁVEL” no final. O padrão de envio das informações do sistema é através de JSON, ao qual deve ser configurado dentro da variável `msg.payload`, como por exemplo: “{“Nome da Variável no Banco de Dados 1”: `msg.payload.NODE_DA_VARIÁVEL1`, “Nome Variável no Banco de Dados 2”: `msg.payload.NODE_DA_VARIÁVEL2`}”.

Finalizando o carregamento da variável com as informações, é necessário retorná-la usando o comando “*return msg*”. Com essas definições e seguindo a conexão “Calandra”, as tabelas “CALANDRA” e “RESFRIADOR” receberam, respectivamente, oito e duas variáveis, como observado na Figura 14.

Figura 14 - Configuração do JSON para a Calandra e a Confeccao



Fonte: Autoria própria (2021)

Para testar se as informações estão chegando, foi adicionado ao final do fluxo a atividade “*DEBUG*”, que recebe a variável `msg.payload` e a imprime na tela, como

demonstrado na Figura 15. Os valores de temperatura são medidos em graus celsius e os de velocidade em m/min.

Por fim, seguindo a conexão “Confeccao”, todas as tabelas receberam apenas uma variável. Ao final do fluxo, também foi adicionado a atividade “DEBUG”, ilustrado na Figura 16. Os valores de posição são medidos em graus e os de velocidade em m/min.

Figura 15 - Prompt com informações da Calandra

```
12 Aug 10:31:29 - [info] [debug:1baf59d5.bcb876]
{
  'TEMPERATURA BORRACHA SAIDA CALANDRA': 55.70022964477539,
  'TEMPERATURA BORRACHA SAIDA RESFRIADOR': 26.04166603088379
}
12 Aug 10:31:29 - [info] [debug:d01395b1.e9ff08]
{
  'VELOCIDADE CILINDRO 1': 22.455135345458984,
  'VELOCIDADE CILINDRO 2': 29.996337890625,
  'VELOCIDADE CILINDRO3': 29.86383628845215,
  'VELOCIDADE CILINDRO 4': 22.510072708129883,
  'TEMPERATURA CILINDRO 1': 65.04000091552734,
  'TEMPERATURA CILINDRO 2': 70.01000213623047,
  'TEMPERATURA CILINDRO 3': 69.73999786376953,
  'TEMPERATURA CILINDRO 4': 64.91999816894531
}
```

Fonte: Autoria própria (2021)

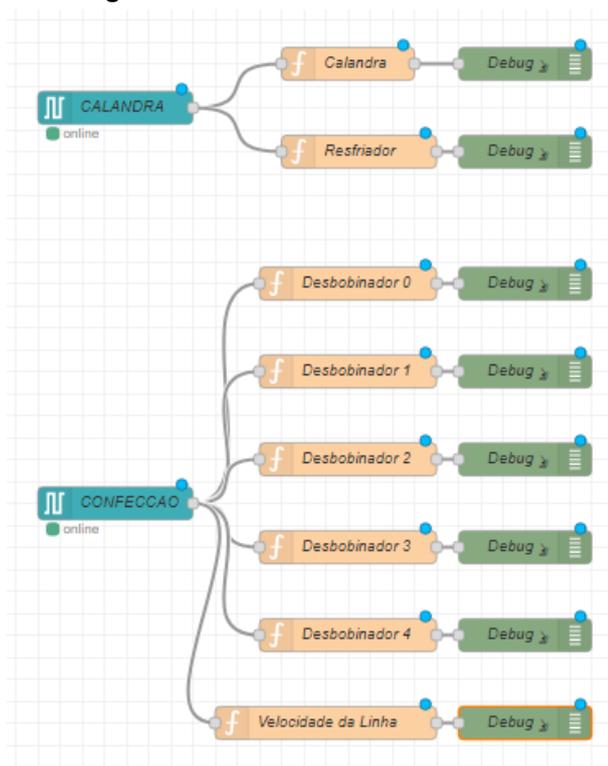
Figura 16 - Prompt com informações da Confecção

```
{ 'POSICAO ALINHAMENTO DESB 2': -53.400001525878906 }
12 Aug 12:51:37 - [info] [debug:db7e28c6.201488]
{ 'POSICAO ALINHAMENTO DESB 4': 0 }
12 Aug 12:51:37 - [info] [debug:4077d7a8.7bf6d8]
{ 'POSICAO ALINHAMENTO DESB 3': 0 }
12 Aug 12:51:37 - [info] [debug:ec3537d5.666288]
{ 'VELOCIDADE LINHA': 0 }
12 Aug 12:51:37 - [info] [debug:d01395b1.e9ff08]
{ 'POSICAO ALINHAMENTO DESB 1': -9.100000381469727 }
12 Aug 12:51:37 - [info] [debug:1baf59d5.bcb876]
{ 'POSICAO ALINHAMENTO DESB 0': 24.200000762939453 }
```

Fonte: Autoria própria (2021)

A Figura 17 mostra a organização atual do Node-RED.

Figura 17 - Fluxo atual no Node-RED



Fonte: Autoria própria (2021)

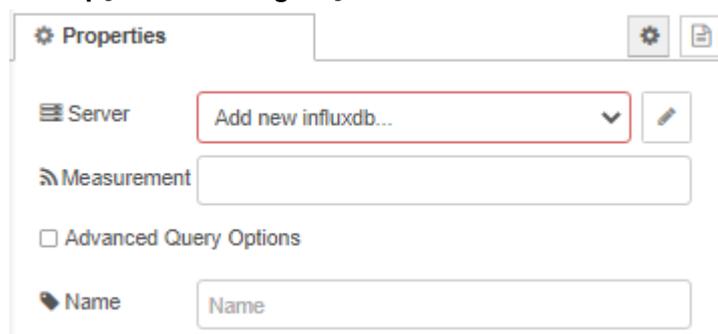
3.4.5 Conexão com o banco de dados

Com os dados já sendo endereçados pelas funções, a próxima etapa é conectar o banco de dados para começar a recebê-los. Essa conexão se dá através da biblioteca do InfluxDB no Node-RED.

Para instalar essa biblioteca, basta digitar comando “*npm install atividade-red-contrib-influxdb*” no prompt de comando. A atividade utilizada para enviar esses dados é o “*influxdb out*”, que recebe a variável msg e o escreve dentro de uma tabela do banco de dados.

Ao arrastar essa atividade para o fluxo, é preciso ajustar duas configurações. Conforme a Figura 18, a primeira opção refere-se a conexão com o banco de dados criado no InfluxDB e a segunda configuração chamada “*Measurement*” é o nome da tabela dentro do banco.

Figura 18 – Opções de configurações do banco de dados no Node-RED



Fonte: Autoria própria (2021)

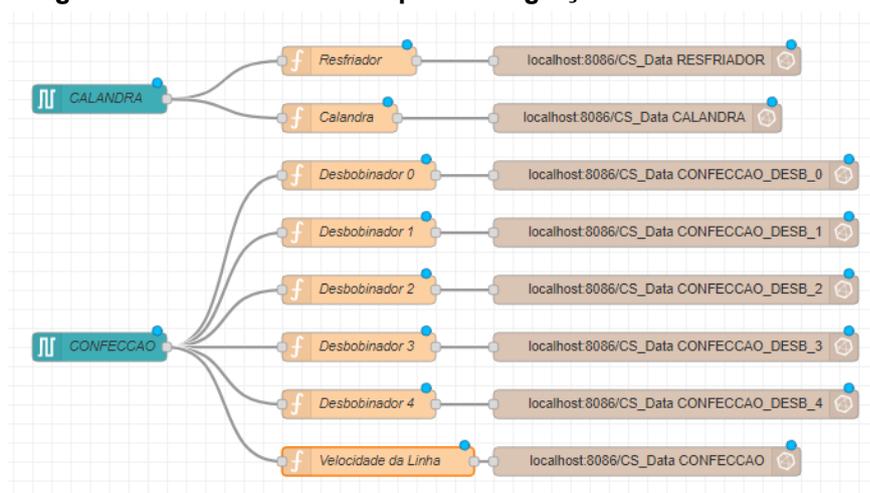
Junto com a equipe da engenharia, foi acordado em criar as tabelas “CALANDRA” e “RESFRIADOR” para a conexão “Calandra” e as tabelas “CONFECÇÃO_DESB_0”, “CONFECÇÃO_DESB_1”, “CONFECÇÃO_DESB_2”, “CONFECÇÃO_DESB_3”, “CONFECÇÃO_DESB_4” e “CONFECÇÃO” para a conexão “Confecção”, a fim de melhor organizar e seguir a estrutura da fábrica dentro do Node-RED.

Neste caso, as tabelas da “Confecção” terão apenas uma variável. Essa estrutura foi pensada para futuramente implementarmos mais variáveis, visto que cada desbobinador possui mais de um sensor.

Na configuração de conexão com o banco de dados foi preciso colocar o nome do banco de dados e do usuário que foi criado para a leitura do banco de dados.

A Figura 19 mostra a organização atualizada do projeto.

Figura 19 - Flow dos dados após a integração do banco de dados



Fonte: Autoria própria (2021)

Para verificar se os dados estavam sendo inseridos no banco de dados, basta olhar o prompt de comando do InfluxDB, como exemplificado na Figura 20.

Figura 20 - Prompt do InfluxDB após a conexão

```
[httpd] 127.0.0.1 - NodeRedUser [13/Aug/2021:09:08:08 -0300] "POST /write?db=CS_Data&p=%5BRDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" 1f6221b3-fc2f-11eb-8d03-005056a63f9f 11205
[httpd] 127.0.0.1 - NodeRedUser [13/Aug/2021:09:08:08 -0300] "POST /write?db=CS_Data&p=%5BRDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" 1f6248b4-fc2f-11eb-8d08-005056a63f9f 10207
[httpd] 127.0.0.1 - NodeRedUser [13/Aug/2021:09:08:08 -0300] "POST /write?db=CS_Data&p=%5BRDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" 1f6248b4-fc2f-11eb-8d0c-005056a63f9f 10207
[httpd] 127.0.0.1 - NodeRedUser [13/Aug/2021:09:08:08 -0300] "POST /write?db=CS_Data&p=%5BRDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" 1f6248b4-fc2f-11eb-8d0b-005056a63f9f 10207
[httpd] 127.0.0.1 - NodeRedUser [13/Aug/2021:09:08:08 -0300] "POST /write?db=CS_Data&p=%5BRDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" 1f6248b4-fc2f-11eb-8d09-005056a63f9f 10207
[httpd] 127.0.0.1 - NodeRedUser [13/Aug/2021:09:08:08 -0300] "POST /write?db=CS_Data&p=%5BRDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" 1f6221b3-fc2f-11eb-8d05-005056a63f9f 11205
[httpd] 127.0.0.1 - NodeRedUser [13/Aug/2021:09:08:08 -0300] "POST /write?db=CS_Data&p=%5BRDACTED%5D&precision=n&rp=&u=
NodeRedUser HTTP/1.1" 204 0 "-" "-" 1f6221b3-fc2f-11eb-8d07-005056a63f9f 17212
```

Fonte: Autoria própria (2021)

3.5 Conexão com o MES

O sistema MES da fábrica possui um banco de dados próprio e restrito. Deste modo só foi recebido o acesso de leitura ao servidor.

Para conectar os dados do sistema com os dados das máquinas, foi desenvolvido um script em Python, resgatando as principais informações do MES. Foram utilizadas as bibliotecas de conexão com os bancos de dados PostgreSQL (MES) e InfluxDB.

Na função principal, o script irá acessar o banco de dados do MES. Como os processos dentro da produção normalmente demoram para trocar, não é necessário um tempo muito pequeno para a atualização, neste caso estimamos que 5 segundos era o tempo ideal para resgatar as informações.

Dentro do MES, existem alguns processos que estão agendados, neste caso precisamos limitar o tempo de busca no banco de dados. Desta forma criamos duas variáveis para guardar a data de dois dias atrás e de um dia no futuro da data atual, respectivamente.

Na função “get_MES”, visto na Figura 21, é passado por parâmetro o id da máquina da Calandra e da Confecção (1 e 8, respectivamente) para enviar uma query ao servidor.

Figura 21 - Função "get_mes" para enviarmos a query ao banco

```
def get_mes(connect, machines, tomorrow, yesterday):
    cur = connect.cursor()
    cur.execute("SELECT * FROM rel_maquina_turno WHERE (data_inicio BETWEEN date %s AND date %s) and (maquina_id = %s) and (final = %s) ORDER BY inicio DESC LIMIT 1;",
                (yesterday, tomorrow, machines, ''))
    row = cur.fetchone()
    try:
        belt_code = row[12]
        name = row[10]
        if row[6] == '12' and row[6] == '6':
            sendinflux(belt_code, "CALANDRA_FINER", name)
        else:
            sendinflux("-", "CALANDRA_FINER", name)
    except:
        print("Problema")
        return
```

Fonte: Autoria própria (2021)

Nesta requisição foi solicitado todas as informações da tabela “rel_maquina_turno” que: possuam o id da máquina, o tempo final seja vazio pois significa que o processo está em andamento e a data atual esteja entre as duas datas definidas anteriormente. Como só pode ter um processo em andamento, o sistema irá nos retornar apenas um vetor de informações.

Os dados que necessários estão nas posições 10 e 12 do vetor, que respectivamente correspondem ao código do produto e o processo que está ocorrendo. A posição 6 refere-se ao status da produção, que se estiver produzindo (código 12) ou em *set-up* (código 6), significa que o sistema possui o código do produto, caso contrário, não possui esta informação.

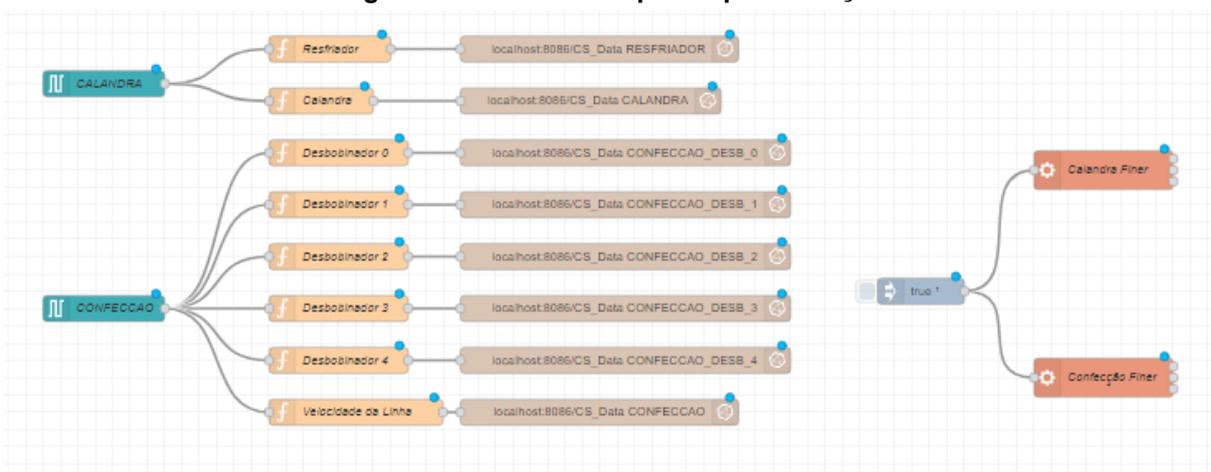
Após receber as informações, é passado para o conector do InfluxDB enviar as informações para a tabela do banco de dados.

Foram produzidos dois scripts iguais para os dois processos, mudando apenas o nome da tabela, colocando como padrão o nome da máquina e concatenando com o nome o nome do sistema MES, neste caso é o Finer.

Deste modo, precisou de um ativador no Node-RED que manda um bit quando o sistema é iniciado, que foi conectado com as duas atividades que gerenciam o prompt de comando, fazendo com que os scripts sempre estejam rodando enquanto o sistema estiver *online*.

A versão final do fluxo do projeto no Node-RED está ilustrada na Figura 22.

Figura 22 - Node-RED após implementação

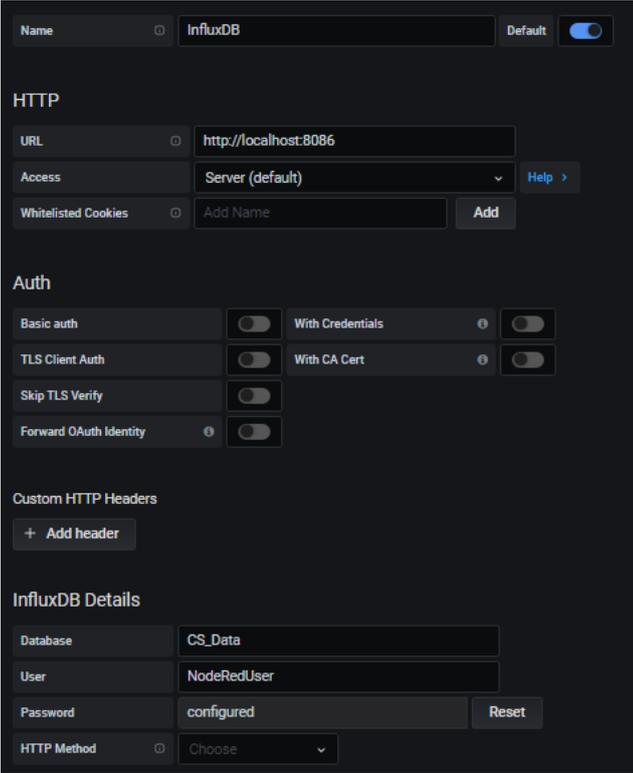


Fonte: Autoria própria (2021)

3.6 Configuração do Grafana

Com o link “Server017:3000” é possível acessar o Grafana fora do servidor. Nele é necessário fazer as configurações de “Data Source”, onde é preciso configurar a base de dados do InfluxDB para coletar as informações. Basta apenas colocar o usuário criado para o acesso, o banco que iremos acessar, o IP do servidor e a porta, demonstrado na Figura 23.

Figura 23 - Configuração de Data Source no Grafana



The image shows the configuration interface for an InfluxDB data source in Grafana. The interface is dark-themed and includes the following sections:

- Name:** InfluxDB (Default is selected, with a toggle switch).
- HTTP:**
 - URL:** http://localhost:8086
 - Access:** Server (default) (with a Help link)
 - Whitelisted Cookies:** Add Name (with an Add button)
- Auth:**
 - Basic auth:** (toggle off) With Credentials (toggle off)
 - TLS Client Auth:** (toggle off) With CA Cert (toggle off)
 - Skip TLS Verify:** (toggle off)
 - Forward OAuth Identity:** (toggle off)
- Custom HTTP Headers:** + Add header (button)
- InfluxDB Details:**
 - Database:** CS_Data
 - User:** NodeRedUser
 - Password:** configured (with a Reset button)
 - HTTP Method:** Choose (dropdown menu)

Fonte: Autoria própria (2021)

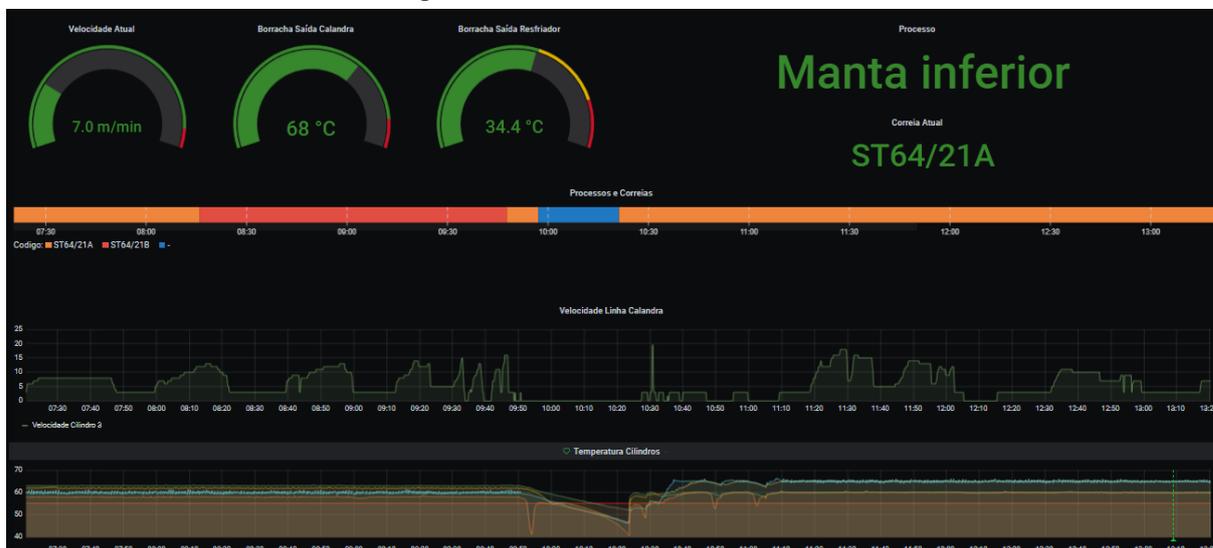
3.7 Visualização dos dados

Após todos os dados estarem conectados com o Grafana, o objetivo agora é criar um dashboard com um fácil entendimento das informações. Para isso, foram criados dois dashboards, um para cada máquina já especificada.

Inicialmente, definimos que o intervalo de tempo que coletaremos os dados nos dashboards seria das últimas 6 horas com atualizações a cada 5 segundos, deste modo o *dashboard* teria uma boa taxa de dados para a análise.

Na Figura 24 é mostrado o primeiro *dashboard*. Ele possui as informações de produção e de máquina da Calandra. Nela foram inseridas informações em histórico de velocidade, temperatura e correias.

Figura 24 - *Dashboard* da Calandra



Fonte: Autoria própria (2021)

Para o *dashboard* da confecção, foram criados gráficos mostrando um histórico da velocidade da linha e da posição dos desbobinadores. Também foram adicionadas informações em tempo real e em histórico das correias e processo, como visualizado na Figura 25.

Figura 25 - *Dashboard* da Confecção



Fonte: Autoria própria (2021)

Para cada gráfico é necessário configurar uma *query* para a tabela no banco de dados e escolher um padrão de visualização, isto é, o design que o gráfico irá possuir.

A *query* dentro do Grafana é bem interativa, basta apenas escolher a tabela e o campo em que a informação que você procura está. Quando configurado o acesso ao banco de dados (*Data Source*), todas as tabelas e campos aparecem automaticamente para você escolher.

No geral, foram utilizados quatro padrões de gráficos para a construção dos *dashboards*. Foram eles os padrões: *Gauge*, *Stat*, *Discrete* e *Graph*.

3.7.1 Padrão *Gauge*

O padrão *Gauge* foi utilizado apenas no *dashboard* da Calandra e representa um valor variável e sua medida. Nele é possível configurar cores para demonstrar se o valor atual passou ou não de um limite especificado.

Na Figura 26 estão ilustrados todos os gráficos que utilizaram este padrão. Eles foram configurados para receber o último valor disponível no banco de dados, isto é, o valor mais atualizado da máquina.

Figura 26 - Visualização no padrão *Gauge*



Fonte: Autoria própria (2021)

Primeiramente foi configurado o gráfico da velocidade atual da produção, que se refere a velocidade da bobina número três da Calandra e é medida em metros por minuto. Com essa informação é possível visualizar se a máquina está em movimento ou não.

Na *query* foi utilizada a tabela CALANDRA e o campo VELOCIDADE CILINDRO3. Nas configurações foi programado para ficar em verde os valores de 0 a 27 e em vermelho qualquer valor acima.

As temperaturas da borracha na saída da Calandra e no resfriador são essenciais para saber se a produção está nos parâmetros corretos de acordo com o material da correia.

Ambas utilizaram a tabela RESFRIADOR, mudando apenas os campos para TEMPERATURA BORRACHA SAIDA CALANDRA e TEMPERATURA BORRACHA SAIDA RESFRIADOR.

Referente as configurações nestes gráficos, para a temperatura na Calandra foi utilizado a cor verde para valores entre 0 e 79 e vermelho para os valores a partir de 80. Para a temperatura no resfriador, verde para valores entre 0 e 34, laranja para valores entre 35 e 49 e vermelho para valores a partir de 80.

3.7.2 Padrão *Stat*

Este padrão foi utilizado nos dois *dashboards* e ele trabalha com informações em formato texto, podendo ou não ser em tempo real.

Na Figura 27 está o gráfico de correia atual, onde é mostrado a ordem de produto da correia em produção. Essa ordem de produto varia de acordo com o tipo de borracha e da ordem de compra. Também é possível visualizar o gráfico de processo que é importante para saber em qual fase da produção a correia está, desta forma caso aconteça algum erro, a rastreabilidade se torna mais concreta.

Figura 27 - Gráficos no padrão *Stat*



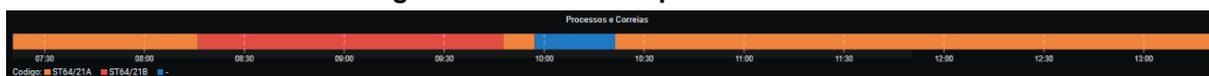
Fonte: Autoria própria (2021)

Ambos *dashboards* usam os campos Processo e CodigoCorreia em suas *queries*, a única mudança é a tabela que cada uma acessa, sendo a tabela CALANDRA_FINER acessada pelo *dashboard* da Calandra e CONFECCAO_FINER acessada pelo *dashboard* da confecção.

3.7.3 Padrão *Discrete*

O *Discrete* é um padrão que coloca todos os valores disponíveis de um determinado *range* de tempo em apenas uma linha. Foi usado este padrão para ter um histórico de correias, como ilustrado na Figura 28. Ele é o único padrão que é necessário instalar dentro do Grafana.

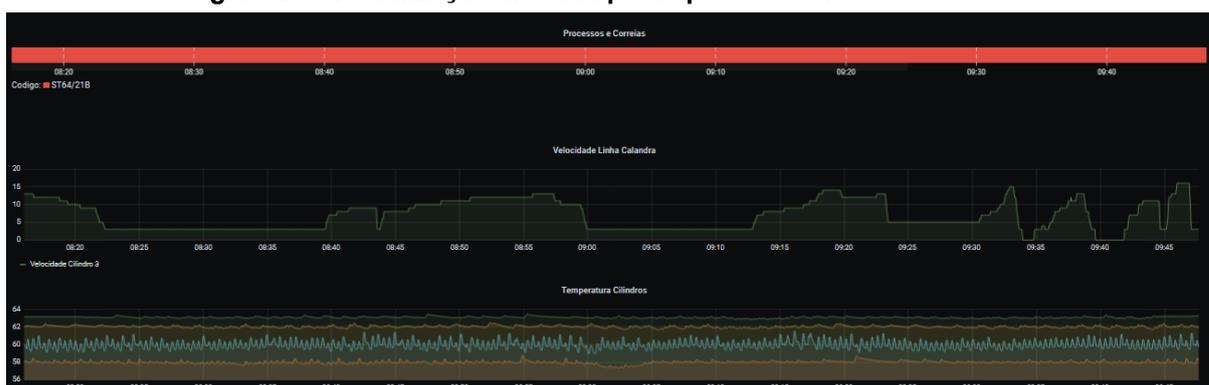
Figura 28 – Gráfico no padrão *Discrete*



Fonte: Autoria própria (2021)

Usando esse padrão é possível filtrar todas as tabelas com uma correia apenas clicando nela, o que torna o *dashboard* mais dinâmico para a pesquisa de incidentes, demonstrado na Figura 29.

Figura 29 - Visualização alterada para apenas a correia selecionada



Fonte: Autoria própria (2021)

Na *query*, usamos o campo “CodigoCorreia” para os dois *dashboards* mudando apenas a tabela para CALANDRA_FINER e CONFECCAO_FINER, respectivamente.

3.7.4 Padrão *Graph*

Por último, o gráfico padrão utilizado para a visualização dos dados de no máximo 6 horas passadas.

Primeiramente foi desenvolvido o gráfico de velocidade da Calandra que armazena um histórico de velocidades, podendo ser útil na rastreabilidade de algum provável erro e até mesmo na prevenção de problemas no processo e na máquina, como visto na Figura 30.

Figura 30 - Gráfico da velocidade na Calandra



Fonte: Autoria própria (2021)

Neste gráfico foi usada a tabela CALANDRA e o campo VELOCIDADE CILINDRO3 na *query*.

Foi produzido também outro gráfico para o *dashboard* da Calandra que nos mostra a temperatura dos cilindros, muito utilizado para verificar se a temperatura está no padrão especificado para a borracha. Para ele, utilizamos a tabela CALANDRA e os campos TEMPERATURA CILINDRO 1, TEMPERATURA CILINDRO 2, TEMPERATURA CILINDRO 3 e TEMPERATURA CILINDRO 4 na *query*.

No *dashboard* da confecção, primeiramente foi desenvolvido o gráfico da velocidade da linha, muito parecido com o gráfico da Calandra. Nela foi utilizado a tabela CONFECÇAO e o campo VELOCIDADE LINHA na *query*, ilustrado na Figura 31.

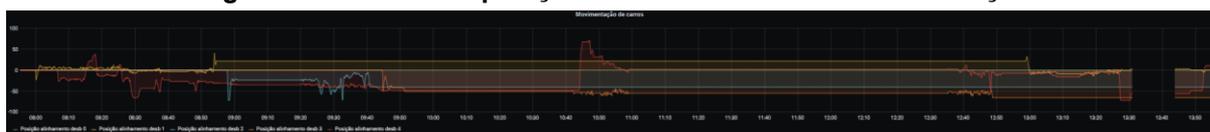
Figura 31 - Gráfico de velocidade da linha na Confecção



Fonte: Autoria própria (2021)

Na Figura 32 é possível observar o gráfico da posição dos carros que nos mostra a posição horizontal, isto é, esquerda ou direita de cada desbobinador dentro da confecção.

Figura 32 - Gráfico da posição dos desbobinadores na confecção



Fonte: Autoria própria (2021)

A *query* deste último gráfico foi definida com a tabela CONFECÇAO_DESB_X e o campo POSICAO ALINHAMENTO DESB Y, sendo X e Y uma sequência de números de 1 a 4.

4 CONCLUSÃO

Após o fechamento do projeto, os dados de temperatura, velocidade e posição foram armazenadas e já estão disponíveis para a visualização. Atualmente o sistema está em ambiente produtivo da fábrica e já possui 1 ano de utilização. Neste meio tempo vários erros na manufatura e nas máquinas já foram observados antes mesmo de ocorrer um problema maior na estrutura de ambos.

Conclui-se que os dados das máquinas podem trazer benefícios para o planejamento e melhoria da manufatura assim como na prevenção de futuros erros no maquinário. Além disso, a rastreabilidade de cada item produzido é ampliada, podendo assim gerar relatórios mensais mais específicos em relação a qualidade do produto.

É importante notar também que houve uma economia de orçamento por parte da fábrica, visto que antes da proposta do trabalho foi cotada a compra de um software para realizar as mesmas atividades. Desta forma, os únicos gastos neste projeto foram os de mão de obra das implementações nos PLCs por parte da equipe de manutenção, na manutenção dos servidores que alocamos os sistemas e no desenvolvimento e configuração das aplicações.

Outro detalhe importante é que após a implementação, o projeto se expandiu para todas as máquinas da manufatura, inclusive na nova expansão da fábrica que entrou em vigor no ano de 2021.

Em relação às dificuldades para a implementação do trabalho, pode-se citar a conexão entre o Node-RED e o PLC da máquina. Esta tarefa em específico foi feito em conjunto com o time de manutenção e acabou tomando mais tempo do que foi planejado.

Atualmente, cerca de 5 equipes utilizam o sistema para a verificação, sendo eles a equipe de engenharia de processos, manutenção, qualidade, produção e a equipe de planejamento. Neste contexto, a fábrica de Conveying Solutions de Ponta Grossa é a única B.U da CT a ter um sistema de monitoramento com integração aos dados do MES como o produzido neste projeto.

5 TRABALHOS FUTUROS

Visto a grande demanda de Indústria 4.0 no Brasil para os próximos anos, a CT sugeriu uma extensão do projeto.

Exemplificando esta extensão, imagine que foi definida a uma máquina usar uma temperatura de 180° C em um determinado material de borracha. Usando os dados coletados, é possível criar uma inteligência capaz de minimizar os recursos.

Voltando ao exemplo, após a inteligência posteriormente criada analisar as informações obtidas, ela descobre que a máquina consegue rodar a 140° C, trazendo os mesmos benefícios que a temperatura anterior. Neste caso a inteligência mandaria um comando avisando a máquina para abaixar a temperatura, assim não sobrecarregando-a. Geralmente este gerenciamento de recursos do maquinário é um grande problema dentro da fábrica.

Por fim, sugere-se encontrar um método de comandar as máquinas utilizando os dados que recebemos neste projeto e uma inteligência agindo por trás. Em resumo, utilizar big data e inteligência artificial para que a manufatura comece a “pensar” sozinha e desviar empecilhos sem interferência humana.

REFERÊNCIAS

BUTUN, I. **Industrial IoT: Challenges, Design Principles, Applications, and Security**. 1. ed. Cham: Springer, 2020.

Citisystems. Sensor: Você sabe o que é Quais os tipos? Jul. 2016. Disponível em: <https://www.citisystems.com.br/sensor-voce-sabe-que-quais-tipos/>. Acesso em: 20 de ago. 2021

DAVIS, N. **What is the fourth industrial revolution?**. World Economic Forum, 2016. Disponível em: <https://alejandroarbelaez.com/wp-content/uploads/2020/10/What-is-the-fourth-industrial-revolution-WEF.pdf>. Acesso em: 13 de mai. 2021.

FIRJAN (Sistema Firjan). **Indústria 4.0: Panorama da Inovação**. 2016.

FNF (Facts & Factors). Industry 4.0 Market By Technology [Artificial Intelligence, 5G, Internet of Things (IoT), Blockchain, Augmented Reality & Virtual Reality, and 3D Printing], By End-User (Energy & Power, Aerospace, Logistics, Automotive, and Others), And By Region: Global Industry Outlook, Market Size, Business Intelligence,

Consumer Preferences, Statistical Surveys, Comprehensive Analysis, Historical Developments, Current Trends, and Forecasts, 2020–2026. Mar. 2020. Disponível em: <https://www.fnfresearch.com/global-industry-40-market-by-technology-artificial-intelligence-963>. Acesso em: 11 de mai. 2021.

HOLLER, J. *et al.* **Internet of Things: Technologies and Applications for a New Age Intelligence**. 1. ed. Cambridge: Academic Press, 2014.

KELVIN, E. Programmable Logic Controllers. Missouri: IEEE Potentials, jan. 1996. E-book. Disponível em: https://scholarsmine.mst.edu/cgi/viewcontent.cgi?article=1684&context=ele_comeng_facwork. Acesso em: 20 de ago. 2021

Macrotec. O que é CLP e qual sua importância na automação industrial. Jan. 2020. Disponível em: <https://macrotec.ind.br/o-que-e-clp-e-qual-sua-importancia-na-automacao-industrial/>. Acesso em: 13 de mai. 2021

Painel Logístico. Por uma indústria mais moderna. Mar. 2019. Disponível em: <https://www.painellogistico.com.br/%EF%BB%BFpor-uma-industria-mais-moderna/>. Acesso em: 13 mai. 2021.

PEDERNEIRAS, G. Big data na Indústria 4.0: como transformar dados em lucro. 2019. Disponível em: <https://www.industria40.ind.br/artigo/17534-big-data-na-industria-40-como-transformar-dados-em-lucro>. Acesso em: 10 de nov. 2021.

Pepperl-Fuchs. Contagem de garrafas em máquinas de enchimento de bebidas com os sensores ultrassônicos. Disponível em: <https://www.pepperl-fuchs.com/brazil/pt/24176.htm>. Acesso em: 25 de ago. 2021.

Pequenas empresas e grandes negócios. Só 48% das indústrias brasileiras são '4.0', diz CNI. Mai. 2016. Disponível em: <https://revistapegn.globo.com/Empreendedorismo/noticia/2016/05/so-48-das-industrias-brasileiras-sao-40-diz-cni.html>. Acesso em: 13 de mai. 2021.

PEREIRA, A.; SIMONETTO, E. O. Indústria 4.0: Conceitos e perspectivas para o Brasil. **Revista da Universidade Vale do Rio Verde**, Três Corações (MG), v. 16, n. 1, p. 1-9, jul. 2018.

SARFRAZ, Z. *et al.* **Is COVID-19 pushing us to the Fifth Industrial Revolution (Society 5.0)?**. Pakistan journal of medical sciences, v. 37, n. 2, p. 591-594, mar. 2021.

SCHWAB, K. **The fourth industrial revolution**. Nova Iorque: Crown Business, 2017.

TAURION, C. **Big Data**. 1. ed. Rio de Janeiro: Brasport, 2013.