

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Engenharia Elétrica
Graduação em Engenharia de Controle e Automação

MARCELA COURY PINTO

**SISTEMA DE MANUTENÇÃO PREDITIVA DE FALHAS EM VÁLVULAS EM UM
PROCESSO INDUSTRIAL UTILIZANDO INTELIGÊNCIA ARTIFICIAL**

UBERLÂNDIA
2021

MARCELA COURY PINTO

**SISTEMA DE MANUTENÇÃO PREDITIVA DE FALHAS EM VÁLVULAS EM UM
PROCESSO INDUSTRIAL UTILIZANDO INTELIGÊNCIA ARTIFICIAL**

Trabalho de Conclusão de Curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Universidade Federal de Uberlândia – UFU
Faculdade de Engenharia Elétrica

Orientador: Prof. Dr. Renato Ferreira Fernandes Júnior

UBERLÂNDIA

2021

Coury Pinto, Marcela

Sistema de manutenção preditiva de falhas em válvulas em um processo industrial utilizando inteligência artificial/ **Marcela Coury Pinto**. – **UBERLÂNDIA, 2021**- 77 p, 30 cm.

Orientador: Prof. Dr. Renato Ferreira Fernandes Júnior

Trabalho de Conclusão de Curso – Universidade Federal de Uberlândia – UFU Faculdade de Engenharia Elétrica. **2021**.

Inclui bibliografia.

1. Manutenção Preditiva 2. Inteligência Artificial 3. Aprendizado de Máquina 4. Inteligência Artificial 4. Processo Industrial I. Orientador Prof. Dr. Renato Ferreira Fernandes Júnior. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Elétrica. IV. Engenharia de Controle e Automação.

MARCELA COURY PINTO

**SISTEMA DE MANUTENÇÃO PREDITIVA DE FALHAS EM VÁLVULAS EM UM
PROCESSO INDUSTRIAL UTILIZANDO INTELIGÊNCIA ARTIFICIAL**

Trabalho de Conclusão de Curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Uberlândia (MG), 16 de junho de 2021.

Banca de Avaliação:

Prof. Dr. Renato Ferreira Fernandes Jr
Orientador

Prof. Dra. Gabriela Vieira Lima
Membro

Prof. Dr. Renato Santos Carrijo
Membro

AGRADECIMENTOS

Agradeço primeiramente à Deus que está sempre me abençoando e encaminhando as coisas que são melhores para o meu crescimento e minha vida.

Agradeço aos meus pais Carlos e Eliane e à minha irmã, que estão sempre me apoiando em todas as decisões e que mesmo de longe, acompanham cada passo e se alegram com cada conquista. Agradeço também ao Luiz Felipe pelo carinho e companhia constante.

Agradeço à minha avó Maria Curi (in memoriam), que sempre me incentivou nos estudos, na pesquisa e principalmente nas redações, um exemplo de mulher guerreira e batalhadora.

Agradeço ao meu professor e orientador Renato Ferreira Fernandes Jr que me ajudou na faculdade e também forneceu ajuda para que esse trabalho fosse concluído com o êxito esperado. E estendo meus agradecimentos também ao professor Aniel Silva de Moraes e a Gabriela Vieira Lima que me deram a oportunidade de me desenvolver na área da pesquisa por meio da iniciação científica.

Agradeço, por fim, aos meus amigos que estiveram presentes na minha formação e tornaram os dias mais felizes.

"Somente quando temos coragem suficiente para explorar a escuridão, descobrimos o poder do infinito de nossa própria luz" Brené Brown

RESUMO

Este trabalho apresenta o desenvolvimento de uma plataforma com inteligência artificial para a predição de falhas em válvulas em um processo industrial, com o objetivo de evitar paradas de linha por conta de quebras não previstas, melhorar a rotina do operador e diminuir custos com manutenção. O sistema foi desenvolvido utilizando conceitos de manutenção, automação, inteligência de negócios e aprendizado de máquina para o desenvolvimento de um sistema de predição.

O projeto foi desenvolvido utilizando ferramentas como o Node-Red, para a coleta de dados, o InfluxDB para o armazenamento dos dados e o *framework* Flask para a criação da página *web*, responsável pela visualização dos dashboards. Para a predição das falhas foram utilizados modelos de aprendizado de máquina como a regressão linear e o K-Means. Os resultados obtidos com a criação da ferramenta e também a predição são apresentados.

Palavras chave: Manutenção preditiva, Business Intelligence, Aprendizado de Máquina, Inteligência Artificial, Processo Industrial.

ABSTRACT

This paper presents the development of an artificial intelligence platform for the valve failure prediction in an industrial process, with the objective of avoiding stop the production line due to unforeseen breaks, improve the operator's routine and reduce maintenance costs. The system was developed using maintenance, automation, business intelligence and machine learning concepts to develop a prediction system.

The project was developed using tools such as Node-Red, for data collection, InfluxDB for data storage and the framework Flask for the creation of a web page, responsible for dashboard visualization. For the failure prediction, machine learning models such as linear regression and K-Means were used. The results obtained with the creation of the tool and also the prediction are presented.

Keywords: *Predictive maintenance, Business Intelligence, Machine Learning, Artificial Intelligence, Industrial Process.*

LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

API – Interface de Programação de Aplicativos

BI – Inteligência de Negócios (*Business Intelligence*)

CLP – Controlador lógico programável

CPU – Unidade Central de Processamento

CSS – *Cascading Style Sheet*

DB – *Data Base*

FC – *Function*

HTML – *HiperText Markup Language*

IA – Inteligência Artificial

IoT – Internet das Coisas (*Internet of Things*)

KNN – *K-Nearest Neighbors* (K vizinhos mais próximos)

MAE – Mean Absolute Error (Erro absoluto médio)

ML – Aprendizado de Máquina (*Machine Learning*)

MSE – Mean Squared Error (Erro quadrático médio)

NA – Normalmente Aberto

NF – Normalmente Fechado

SQL – *Structured Query Language*

SVM – *Support Vector Machine* (Máquina de vetores de suporte)

LISTA DE FIGURAS

Figura 1 - Pirâmide de Automação.....	8
Figura 2 - Curvas características inerentes de válvulas de controle.	10
Figura 3 - Válvula tipo borboleta.....	11
Figura 4 - CLP.....	12
Figura 5 - Estrutura básica de um CLP.	13
Figura 6 - Exemplo de linguagem Ladder.....	14
Figura 7 - Contador incremental.....	15
Figura 8 - Temporizadores.	15
Figura 9 - Power BI dashboard.....	18
Figura 10 - Dashboard em Flask.	19
Figura 11 - Aprendizado supervisionado.	21
Figura 12 - Clusterização com aprendizado não supervisionado.....	22
Figura 13 - Aprendizado por reforço.....	23
Figura 14 - Aprendizado baseado em modelo.....	25
Figura 15 - Aprendizado baseado em instância.	25
Figura 16 - Modelo de regressão e classificação.	26
Figura 17 - Formação do algoritmo K-Means.....	28
Figura 18 - Arquitetura do projeto.....	31
Figura 19 - Simatic Step 7.....	33
Figura 20 - FC do projeto.	34
Figura 21 - DB do projeto.	34
Figura 22 - Exemplo da interface do Node-Red.	35
Figura 23 - Docker com Node-Red.....	36
Figura 24 - Node-Red do projeto.....	37
Figura 25 - Bloco CLP.	37
Figura 26 - Função Node-Red.....	38
Figura 27 - Configurações do InfluxDB no Node-Red.	39
Figura 28 - Docker com o Node-Red e InfluxDB.	40

Figura 29 - Banco de dados InfluxDB.....	41
Figura 30 - Conexão InfluxDB.	41
Figura 31 - Wide x Long.	42
Figura 32 - Dashboard aba corretiva.....	43
Figura 33 - Dashboard: aba manutenção.....	44
Figura 34 - Importação dos dados para regressão linear.....	46
Figura 35 - Aplicação da regressão linear.....	47
Figura 36 - Bibliotecas K-Means.	48
Figura 37 - Dados K-Means.	48
Figura 38 - Dataframe para a predição.....	49
Figura 39 - Válvulas do mês de abril agrupadas.	49
Figura 40 - Dataframe mensal.....	50
Figura 41 - Dados padronizados.	51
Figura 42 - Aplicação do modelo K-Means.....	51
Figura 43 - Válvulas clusterizadas.....	52
Figura 44 - Válvulas clusterizadas sem tempo de abertura.....	53
Figura 45 - Aba Corretiva.	55
Figura 46 - Predição com Regressão Linear.....	56
Figura 47 - Histograma Distribuidor de Frequências do Erro.	57
Figura 48 - Cluster 1.....	58
Figura 49 - Boxplot com tempo de abertura.	59
Figura 50 - Cluster 2.....	60

LISTA DE TABELAS

Tabela 1 - Resultados da regressão linear.....	56
--	----

SUMÁRIO

1. INTRODUÇÃO	3
1.1 Objetivo	4
1.2 Justificativa	4
2. REFERENCIAL TEÓRICO	6
2.1 Manutenção Industrial	6
2.1.1 Manutenção Corretiva	6
2.1.2 Manutenção Preventiva	6
2.1.3 Manutenção Preditiva	7
2.2 Automação em Sistemas Industriais	7
2.2.1 Dispositivos de campo	9
2.2.1.1 Válvulas de Controle	9
2.2.2 Controladores	12
2.2.3 Sistemas de Supervisão	15
2.3 Business Intelligence (BI)	16
2.3.1 Conceito	16
2.3.2 Aplicação do Business Intelligence	17
2.3.3 Ferramentas de BI	17
2.4 Machine Learning (ML)	19
2.4.1 Conceito	20
2.4.2 Tipos de Machine Learning	21
2.4.2.1 Aprendizado Supervisionado e Não Supervisionado	21
2.4.2.2 Aprendizado Online ou por Lote	24
2.4.2.3 Aprendizado Baseado em Instância ou Modelo	24
2.4.3 Algoritmos de Machine Learning	26
2.4.3.1 Regressão Linear	26
2.4.3.2 K-Means	27
2.4.4 Desafios para Aplicações de Machine Learning	28
2.4.5 Validação e Testes	30
3. DESENVOLVIMENTO	31
3.1 Coleta de Dados	32
3.1.1 Lógica do Processo	32

3.1.2	Node-Red.....	35
3.2	Banco de Dados – InfluxDB.....	39
3.3	Dashboard.....	42
3.4	Predição de Falhas.....	44
3.4.1	Regressão Linear.....	45
3.4.2	Clusterização: K-Means.....	47
4.	RESULTADOS.....	54
4.1	Problemas enfrentados.....	54
4.2	Aba de Análise Corretiva.....	55
4.3	Resultados dos Modelos de Predição.....	56
4.3.1	Regressão Linear.....	56
4.3.2	K-Means.....	58
5.	CONCLUSÃO.....	61
6.	REFERÊNCIAS BIBLIOGRÁFICAS.....	62

1. INTRODUÇÃO

Após a Segunda Guerra Mundial houve um desenvolvimento intensificado da industrialização que esteve diretamente ligado a urbanização e ao aumento da demanda por produtos e serviços. A partir desse cenário, as indústrias ficaram voltadas cada vez mais para a busca da competitividade no mercado, com foco em segurança do trabalho, qualidade do produto ou serviço entregue, alta produtividade e geração de lucros. Para que esse objetivo fosse atingido as empresas propuseram metas e indicadores relacionados a zero defeitos, alta produtividade, disponibilidade e eficiência das máquinas, satisfação dos trabalhadores e clientes. Esses indicadores estão ligados diretamente à manutenção e bom estado dos equipamentos, pois quanto melhor seu estado, maior produtividade.

O setor de manutenção que tinha uma imagem de gerador de custos para a empresa passou a ter papel importante na competitividade. A competitividade exige mudanças rápidas, adaptação às exigências do mercado, redução de custos, alto desempenho do processo produtivo (CAMPOS, 2013).

No cenário atual, algumas horas de linha de produção parada por conta de algum defeito em maquinários, pode gerar um prejuízo de milhões de reais em empresas de grande porte, afetando a produtividade e conseqüentemente o lucro da empresa. Um exemplo é a fábrica da BAT Brasil em Uberlândia (MG), na qual segundo dados retirados do site Souza Cruz, tem uma capacidade de produção de 200 milhões de cigarros por dia. Uma parada de 1h de produção pode gerar um prejuízo de 4 milhões de reais em média.

Neste contexto, as indústrias buscam práticas eficientes na manutenção, com o objetivo de melhorar a disponibilidade da produção. Uma técnica proposta na literatura e que está sendo, aos poucos, implementada na indústria é utilizar inteligência artificial para estimar as performances dos equipamentos, prever a necessidade de manutenção, possíveis modos de falhas e aspectos semelhantes com antecedência (BUKHSH et al, 2019).

Tendo por base o objetivo da manutenção, que é evitar que ocorra falhas não previstas, e das oportunidades que uma manutenção bem feita e planejada pode oferecer a um sistema produtivo industrial, e também de técnicas de inteligência artificial, esse

presente trabalho apresentará um sistema de manutenção preditiva de falhas em válvulas.

1.1 Objetivo

Objetivo do projeto é desenvolver uma plataforma com inteligência artificial para a manutenção preditiva de válvulas em um processo industrial, prevendo falhas em válvulas, evitando paradas de linha, reduzindo custos com manutenção e melhorando a rotina do operador.

Os objetivos específicos do trabalho são resumidamente descritos a seguir:

- Criar uma lógica de programação para a coleta de dados;
- Realizar a coleta de dados e armazenamento em um banco de dados;
- Desenvolver um algoritmo de predição de falhas utilizando ferramentas de aprendizado de máquina, em linguagem python;
- Desenvolver uma plataforma de visualização de dados para a manutenção preditiva, utilizando o Flask.

1.2 Justificativa

O presente projeto tem por motivação a oportunidade de diminuir os gastos com manutenção a partir de uma previsão mais assertiva de quando haverá falha na válvula, para que a manutenção seja feita antes e evite paradas de linha em um processo industrial.

No ambiente industrial, nota-se que dentre as modalidades de manutenção industrial, a manutenção preditiva em válvulas tem ganhado espaço significativo no ambiente industrial. Contudo, ainda é recente as publicações e pesquisas sobre o assunto.

Tendo em vista a importância de trazer para o meio acadêmico discussões e problemas que o meio industrial tem enfrentado, esse trabalho tem por finalidade

apresentar um sistema para a modalidade de manutenção preditiva, além de ajudar na disseminação de pesquisas na área de predição.

2. REFERENCIAL TEÓRICO

O objetivo desse capítulo é apresentar os principais fundamentos para a compreensão desse trabalho. Serão abordados conceitos de Manutenção industrial, Automação de Sistemas Industriais, *Bussiness Intelligence* e *Machine Learning*.

2.1 Manutenção Industrial

Manutenção é um conjunto de cuidados e procedimentos técnicos necessários para o bom funcionamento e também reparo de máquinas, equipamentos, peças e ferramentas (ALMEIDA, 2014). A palavra manutenção vem do latim *manus tenere*, que significa “manter o que se tem”.

A manutenção ganhou espaço e importância no setor industrial, evoluindo cada vez mais as técnicas e procedimentos usados nas máquinas e equipamentos, para garantir alta produtividade, disponibilidade e eficiência. A seguir são abordados os principais tipos de manutenção: manutenção corretiva, preventiva e preditiva.

2.1.1 Manutenção Corretiva

De acordo com a Associação Brasileira de Normas Técnicas - ABNT (1994, p. 7), manutenção corretiva é “manutenção efetuada após a ocorrência de uma pane destinada a recolocar um item em condições de executar uma função requerida”.

Esse tipo de manutenção acontece quando ocorre uma falha em um equipamento que prejudique o seu funcionamento. Existem dois tipos de manutenção corretiva: a não programada/emergencial e a programada. A manutenção não programada acontece quando a falha prejudica um processo produtivo e então tem que intervir e fazer a manutenção do equipamento de imediato. Na manutenção programada, quando ocorre uma falha, o equipamento é mantido em funcionamento, com limitações, até que uma parada operacional programada seja realizada.

2.1.2 Manutenção Preventiva

De acordo com a ABNT (1994, p. 7), manutenção preventiva é “manutenção realizada em intervalos pré-determinados, ou de acordo com critérios prescritos,

destinada a reduzir a probabilidade de falha ou a degradação do funcionamento de um item”.

Geralmente, essas manutenções são feitas periodicamente, de acordo com orientações do fabricante ou de acordo com histórico de manutenções dos equipamentos.

2.1.3 Manutenção Preditiva

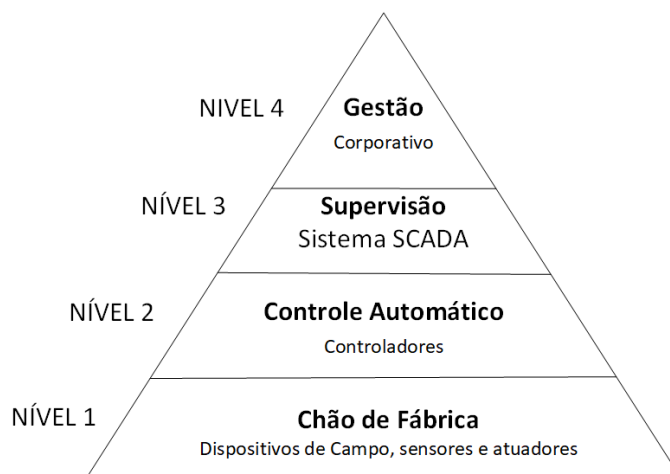
De acordo com a ABNT (1994, p. 7), manutenção preditiva é “manutenção que permite garantir uma qualidade de serviço desejada, com base na aplicação sistemática de técnicas de análise, utilizando-se de meios de supervisão centralizados ou de amostragem, para reduzir ao mínimo a manutenção preventiva e diminuir a manutenção corretiva”.

Esse tipo de manutenção tem o objetivo de prever a situação dos equipamentos a partir de análises de dados do desempenho. Essas análises permitem que seja feita uma manutenção antes que ocorra uma falha e prejudique o processo, diminuindo perdas de produção e gastos com manutenções corretivas e preventivas.

2.2 Automação em Sistemas Industriais

A automação industrial geralmente pode ser classificada de acordo com seus níveis de automação, vindo desde o nível mais baixo, chamado de nível de chão de fábrica ou de campo, até o nível corporativo como mostrado na Fig. 1 (KUMAR, 2014).

Figura 1 - Pirâmide de Automação.



Fonte adaptada: (KUMAR, 2014)

De acordo com a Fig. 1, no nível mais baixo estão localizados os sensores e atuadores responsáveis pelo andamento do processo produtivo. O nível de controle corresponde aos equipamentos que controlam toda a planta sendo principalmente formado pelo *Programmable Logic Controller* (PLC ou CLP). O próximo nível seria formado pelos sistemas de supervisão e controle, é onde o sistema é controlado pelos engenheiros e operadores da planta. O topo da pirâmide é chamado de nível corporativo ou de gestão da planta. Este nível é responsável por processar o escalonamento da produção da planta e por permitir operações de monitoramento estatístico e gestão estratégica do processo de produção, cuidando da qualidade e da gestão do processo. Geralmente, seu funcionamento ocorre por meio de softwares gerenciais (KUMAR, 2014).

Os diferentes níveis têm de lidar com diferentes requisitos peculiares ao nível específico em causa. Por exemplo, o nível da empresa deve lidar com um grande volume de dados que não são nem tempo crítico nem com uso constante. Já nos níveis mais baixos como de sensores e controladores é necessário um determinismo da rede e constância dos processamentos e controles.

No contexto da Manutenção de sistemas, a grande maioria dos eventos de manutenção ocorrem principalmente no nível do chão de fábrica, onde existem mais

quebra de equipamentos ou de falhas em operações devido a própria característica da planta. Nas próximas seções será detalhado cada um destes níveis de automação.

2.2.1 Dispositivos de campo

O nível do chão de fábrica é formado basicamente pelos equipamentos que fazem a interface com o processo industrial. Eles são equipamentos de entrada e saída digitais ou analógicos como sensores, atuadores que fazem todas as medições das diferentes variáveis do sistema e também fazem a atuação no controle dos processos. Exemplos de sensores são transmissores de temperatura, pressão, chaves de níveis, entre outros. Exemplos de atuadores são válvulas, motores, lâmpadas, entre outros.

Como este trabalho tem o intuito de estudar especificamente as válvulas de controle, será concentrado os estudos somente neste tipo de equipamento de campo.

2.2.1.1 Válvulas de Controle

De acordo com Almeida (2018), as válvulas de controle são os segundos elementos mais numerosos dentro da indústria químicas e petroquímicas, perdendo somente para as conexões de tubulação. As válvulas são os elementos mais comuns usados como atuadores sendo usadas em diversas aplicações como: serviço de liga-desliga, controle proporcional, controle e alívio de pressão, controle de vazão direcional, serviço de amostragem, limitação de vazão, selagem de saídas de vasos.

Uma válvula é basicamente um orifício de área variável, através do qual se escoar o fluido, e cuja seção é feita para variar de acordo com a vazão pretendida. Esta obstrução pode ser parcial ou total (CAMPOS & TEIXEIRA, 2008).

Em relação a operação as válvulas podem ser manuais ou automáticas. As válvulas manuais são operadas localmente através de volante, alavanca ou sistema de engrenagens. Já as válvulas automáticas podem operar remotamente ou dispensando a ação externa. Neste caso as válvulas são interligadas aos níveis superiores com os controladores e respondem ao sinal de comando do controlador. Os mecanismos de acionamento das válvulas automáticas podem ser pneumáticos, hidráulicos ou elétricos (CAMPOS & TEIXEIRA, 2008).

Existem diversos tipos de válvulas onde elas podem ser classificadas de acordo com a categoria em válvulas de bloqueio, de controle e de segurança (ALMEIDA, 2018).

As válvulas de bloqueio têm o objetivo de interromper o fluxo e geralmente são válvulas ON/OFF. Exemplo deste tipo de válvulas, são válvulas de gaveta e solenoides.

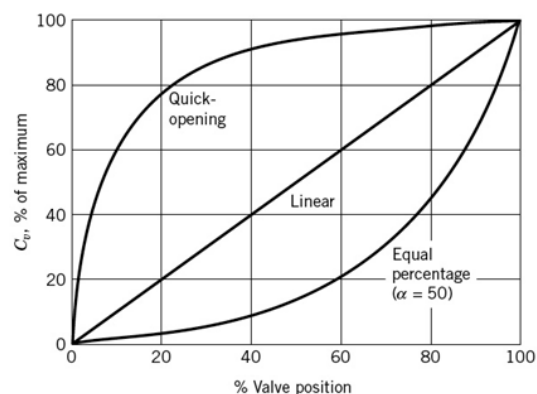
As válvulas de controle são destinadas especificamente para controlar o fluxo, podendo trabalhar em qualquer posição de fechamento. Exemplos deste tipo de válvulas são válvulas globo, de diafragma, borboletas e agulha.

As válvulas de segurança são as que controlam a pressão antes e depois da válvula. Exemplo deste tipo de válvulas são válvulas de alívio, de contrapressão, redutoras de pressão, de quebra-vácuo.

As válvulas de controle consistem do corpo da válvula, das partes internas como o obturador e a sede, um atuador que fornece a força motora para operar a válvula e outros acessórios que incluem posicionadores, transdutores, chaves de posição, reguladores, entre outros.

De acordo com as características de construção, as válvulas de controle possuem características inerentes e instaladas. As características inerentes se referem a característica projetada da válvula e quando instalada sofre algumas alterações quanto a abertura e fechamento da válvula. Desta forma, é obtido a curva característica da válvula, ou seja, qual o comportamento que a válvula vai ter durante a abertura quando se tem uma vazão constante. A Fig. 2 mostra as curvas características inerentes típicas fornecidas pelos fabricantes (SMITH & CORRIPIO, 2008).

Figura 2 - Curvas características inerentes de válvulas de controle.

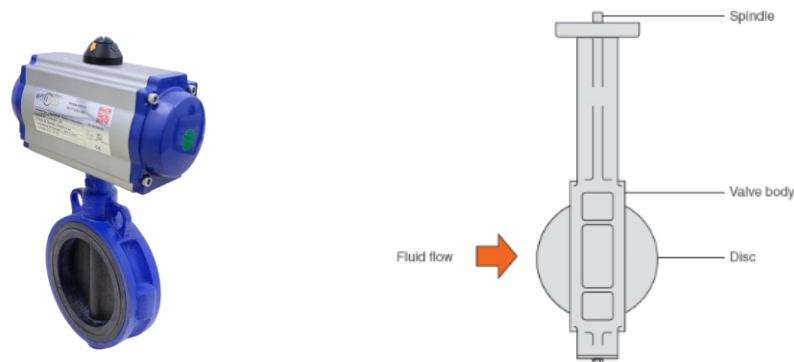


Fonte: SMITH & CORRIPIO (2008).

De acordo com a Fig. 2, as características inerentes das válvulas de controle são tipicamente de abertura rápida, linear e de igual percentagem.

Na válvula de abertura rápida, existe uma grande variação da vazão para uma pequena abertura no início da faixa. Este tipo de válvula possibilita a passagem de quase a totalidade da vazão nominal com apenas uma abertura de 25% do curso total. Não é adequada para controle contínuo, sendo aplicada geralmente em controle do tipo *on-off* (ALMEIDA, 2018). Um exemplo deste tipo de característica seria a válvula borboleta mostrada na Fig. 3. Neste tipo de válvula, para obter a posição aberta ou fechada não é necessário um giro completo, mas apenas um movimento de ângulo 90°.

Figura 3 - Válvula tipo borboleta.



Fonte: Almeida (2018).

De acordo com a Fig. 3, a válvula borboleta é formada por uma haste (*Valve body*) que se liga a sede circular da válvula e a sede se liga a um disco (*Disc*), que fecha o orifício de vazão (*Fluid Flow*). O pino superior (*Spindle*) se liga a um posicionador que faz o controle do sistema.

Na válvula linear, a vazão é diretamente proporcional a sua abertura. Possui ganho constante em todas as vazões, independente do ponto de operação. Utilizada normalmente para controle de nível, controle de pressão de fluidos compressíveis e em sistemas onde a perda de carga na válvula seja elevada (ALMEIDA, 2018).

Na válvula igual percentagem, iguais variações na abertura da válvula correspondem a iguais porcentagens na variação de vazão. Possui uma variação de vazão muito pequena no início de sua abertura e muito alta na próxima da abertura total. Esta válvula perde sua característica inerente quando instalada no processo,

aproximando-se geralmente da linear. Utilizada normalmente no controle de pressão de líquidos, em processos rápidos e quando não se conhece muito bem a dinâmica do processo (ALMEIDA, 2018).

De acordo com Campo & Teixeira (2008), grande parte dos problemas nas instalações industriais tem relação com as válvulas de controle. Os principais fatores que levam a estes problemas são: projeto ou montagem incorreta da válvula, operação indevida devido a desgastes ou incrustação do processo, pressão da linha, problemas no atuador, entre outros. Desta forma, é importante sempre fazer as manutenções preditiva e preventiva para evitar que estes problemas ocorram no sistema.

2.2.2 Controladores

De acordo com a Associação Brasileira de Normas Técnicas (ABNT), o CLP é um equipamento eletrônico digital, com *hardware* e *software* compatível com as aplicações industriais. De acordo com Associação Nacional de Fabricantes de Equipamentos Elétricos dos Estados Unidos da América (National Electrical Manufacturers Association – NEMA), o CLP é um aparelho eletrônico digital, que utiliza uma memória programável para armazenar internamente instruções e para implementar funções específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, controlando, por meio de módulos de entradas e saídas, vários tipos de máquinas ou processos (ZANCAN, 2011). Na Fig. 4 tem um exemplo de um CLP da marca Siemens.

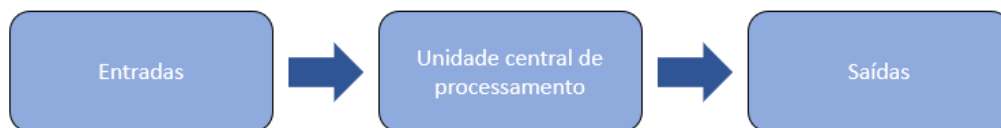
Figura 4 - CLP.



A princípio o CLP surgiu para substituir o relé lógico e as vantagens de um CLP em relação ao relé são maior confiabilidade e flexibilidade, pois o programa elimina uma

boa parte da fiação que seria necessária para o controle de um processo utilizando relés e também é mais fácil criar e modificar um programa. O CLP tem menor custo, consumo de energia e tempo de reposta. Possui maior facilidade de verificação de defeitos, por meio de diagnóstico e também monitoramento de variáveis em tempo real. Tem capacidade de comunicar com outros dispositivos, é fácil fazer a programação e também pode ser utilizável para outros projetos (FRANCHI, 2021). A Fig. 5 mostra o funcionamento do CLP.

Figura 5 - Estrutura básica de um CLP.



Nas entradas são conectados dispositivos elétricos que enviam sinais elétricos para a Unidade Central de Processamento (CPU). Essa por sua vez, analisa as informações das entradas, a lógica programada pelo usuário e ativa ou desliga as saídas do CLP. E as saídas são conectadas a elementos que interagem com o processo, com o objetivo de controlá-lo (ZANCAN, 2011).

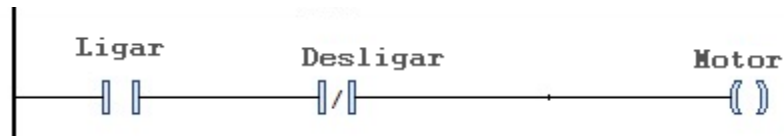
A programação dos CLPs é baseada em cinco linguagens: ST (*Structured Text*), IL (*Instruction List*), Ladder, FBD (*Function Block Diagram*) e SFC (*Sequential Flow Chart*).

As linguagens ST e IL são textuais, porque contém informações em forma de texto. Já as outras são gráficas, pois as informações são passadas em forma de símbolos. Por meio dessas linguagens de programação é possível criar, apagar e alterar as lógicas que são utilizadas para controlar um processo.

Para esse projeto será utilizada a linguagem de programação Ladder, porque é a mais usada nos CLPs, é parecida com a lógica dos comandos elétricos e também é a

linguagem que é utilizada no processo industrial abordado nesse trabalho. A Fig. 6 tem um exemplo de um comando simples.

Figura 6 - Exemplo de linguagem Ladder.



Supondo que nesse exemplo acima, o “Desligar” é um botão em uma máquina de sorvete que tem a intenção de parar a máquina e “Ligar” é outro botão da mesma máquina. Por essa lógica acima, se o botão de ligar estiver apertado e o de desligar não estiver apertado, o motor da máquina vai começar a funcionar e vai conseguir fazer o sorvete. Em uma explicação mais técnica, existem três tipos de elementos na linguagem Ladder: as entradas ou contatos, as saídas ou bobinas e os blocos de funções. A seguir são apresentados alguns exemplos.

Entradas:

- -| |- são contatos NA ou normalmente abertos. Isso significa que o contato fica aberto quando não há energia passando pelo circuito e se fecha quando recebe energia.
- -|/|- são contatos NF ou normalmente fechados. Sua lógica de funcionamento é oposta à NA.
- -(P)- Transição positiva. Se o símbolo mudar de desligado para ligado esse contato transfere um pulso de energia para a linha.

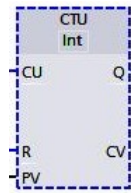
Saídas:

- -()- são representação de bobinas ou saídas. Se todos os contatos da linha estiverem energizados a bobina é ligada.

Blocos de funções:

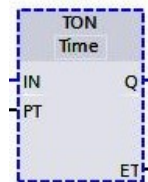
- Contadores: são usados para contagem crescente (incremental), decrescente (decremental) ou ambas. Exemplo:

Figura 7 - Contador incremental.



- Temporizadores: são utilizados para a medir a duração de um evento. Existem temporizadores com *delay* na subida, que significa que ele vai acionar a saída somente após a contagem do tempo. E também com *delay* na descida, que espera a entrada mudar de verdadeira para falso para começar a contar o tempo. Exemplo:

Figura 8 - Temporizadores.



2.2.3 Sistemas de Supervisão

No nível superior da pirâmide estão os sistemas de supervisão, de controle e manutenção da planta pelo operador e pela engenharia (KUMAR, 2014).

Os sistemas supervisórios ou SCADA se comunicam com os níveis de controle e o nível de chão de fábrica e são responsáveis pela interface Homem-Máquina (IHM), onde são monitoradas as variáveis do processo e onde são informado as condições de operação por meio de telas e gráficos do andamento do processo.

No contexto da manutenção do sistema, existem os Asset Management ou gerenciamento de ativos da planta. Estes são usados para gerenciar a planta com a finalidade de se evitar as paradas não programadas do processo. Estes tipos de sistemas são capazes de monitorar as condições dos equipamentos, e possibilitam traçar planos de manutenção.

Os conceitos atuais de indústria 4.0, se baseia na ideia de uma indústria inteligente, com estrutura modular que se integram de forma a prover comunicação rápida

e eficiente da informação visando a melhoria continua dos processos. Desta forma, as fabricas inteligentes não necessitam necessariamente de pessoas para a tomada de decisão e sim de sistemas inteligentes para auxiliar e melhorar cada vez mais o produto em suas diferentes fases (MMTEC, 2021).

2.3 Business Intelligence (BI)

2.3.1 Conceito

Business Intelligence ou Inteligência de negócios pode ser descrito como um processo de negócio orientado por meio da tecnologia para analisar dados e extrair informações relevantes que ajudam a empresa a tomar decisões de negócios.

Segundo Turban (2019), os principais objetivos do BI são permitir acesso interativo aos dados, proporcionar manipulações e tornar a análise do negócio de maneira adequada. Portanto, de maneira geral, o objetivo do BI é fornecer informações de forma confiável tornando o processo de decisão e direcionamento do negócio mais seguro e claro.

Um sistema de Business Intelligence possui as seguintes características, de acordo com Serra (2002):

- extrair e interagir os dados de múltiplas fontes;
- fazer o uso da experiência, analisar dados contextualizados;
- trabalhar com hipóteses;
- procurar relações de causa e efeito;
- transformar os registros obtidos em informações úteis para o conhecimento empresarial.

Atualmente as empresas possuem milhares de dados, porém não conseguem extrair muitas informações úteis desses dados para tomar decisões assertivas e se tornar mais competitiva. Analisando as definições acima, pode-se perceber que o BI surgiu para suprir essa demanda de preparação dos dados para que informações relevantes possam ser extraídas.

2.3.2 Aplicação do Business Intelligence

A aplicação de Inteligência de Negócios possibilita que a organização possa fazer uma análise mais inteligente e mais rápida dos seus dados, percebendo pontos de melhorias do seu processo por meio da análise de indicadores, oportunidade de negócios, identificação de custos sobressalentes ou má utilização dos recursos.

Os principais benefícios que podem se obter com a implementação do BI são a economia de tempo, auxílio na tomada de decisões, melhorias nas estratégias e indicadores, auxílio no planejamento de gestão e redução de custos.

Porém não basta pensar que com quaisquer dados, sem tratamento ou manipulação, será possível obter relatórios completos, precisos e confiáveis, são necessárias algumas etapas que serão descritas a seguir:

Aquisição de dados: É o processo de medição, coleta e armazenamento de dados para que seja possível fazer uma análise. Nessa etapa é necessário que tenha uma fonte de dados que seja confiável e que consiga extrair uma quantidade grande de dados para se fazer análises. Nesse projeto, por exemplo, será abordado a aquisição de dados diretamente do equipamento.

Preparação dos dados: Essa é uma atividade que é responsável por transformar dados brutos e desorganizados em dados limpos, consistentes e de qualidade. Nessa etapa, é feita a limpeza, transformação e organização dos dados coletados. Ela é umas das etapas mais importantes e onde as equipes de dados passam mais tempo trabalhando, cerca de 80% do tempo (IBM, [s.d]).

Modelagem dos dados: Essa etapa é composta pelo agrupamento de dados e exclusão daqueles que não são relevantes para sua aplicação. Também é composta pela construção de métricas que serão utilizadas para fornecer análises importantes para o negócio.

Seguindo essas etapas, os dados vão estar bem preparados, confiáveis e de qualidade para que se possa analisar, de forma mais completa, em uma ferramenta de BI.

2.3.3 Ferramentas de BI

O Business Intelligence engloba uma grande variedade de ferramentas e aplicativos que tornam possível a coleta, organização e análise dos dados. A ferramenta

mais utilizada é o Power BI, desenvolvida pela Microsoft. Nele é possível criar relatórios que podem ser disponíveis online e que permitem fazer a conexão com diversas fontes de dados, como o Excel, SQL Server, arquivos csv, etc. (MORAIS, 2020).

Figura 9 - Power BI dashboard.



Fonte: <https://docs.microsoft.com/pt-br/power-bi/create-reports/service-dashboards>.

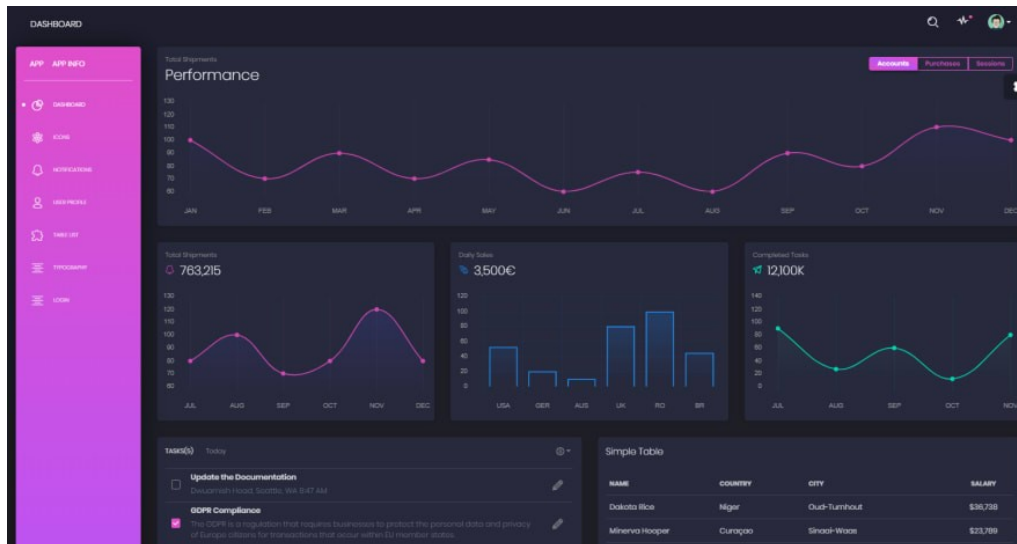
A Fig. 9 mostra um exemplo do Power BI que é uma ferramenta fácil de manipular e criar visualizações. Por ser uma ferramenta desenvolvida pela Microsoft, ela fornece ícones e fórmulas muito semelhantes às outras ferramentas como Word, Power Point e Excel. Porém, não tem um código fonte acessível, o que torna uma ferramenta mais difícil de dar manutenções e de fazer testes.

Essa ferramenta tem três tipos de versões: a versão Power BI Desktop, que pode ser instalada em um computador, é possível fazer conexões com diversos tipos de fontes de dados e é uma ferramenta com alta capacidade analítica. Já a versão Power BI Service (Pro) é paga, mas suas visualizações podem ser compartilhadas com uma equipe ou com a própria organização. E por último, tem o Power BI Mobile, onde é possível acessar os relatórios de forma remota, pelo celular.

Outra ferramenta que vem se destacando nas companhias é o Flask, um *micro-framework* para desenvolvimento *web*, escrito em linguagem python. Ele é utilizado para

criar painéis (dashboards) e aplicativos interativos. Também é possível fazer conexões com várias fontes de dados e análises, porém seu grande diferencial é que o Flask é construído por meio de programação e é código aberto. A Fig. 10 apresenta um exemplo de um dashboard criado em Flask.

Figura 10 - Dashboard em Flask.



O Flask é uma ferramenta totalmente programável em linguagem python, porém também é possível inserir outros tipos de linguagens e estilizações como JavaScript, HTML e CSS para a criação de uma interface com o usuário. Essa ferramenta também fornece vários tipos de gráficos, filtros e manipulações com dados de uma forma que a performance dos painéis não seja afetada com uma grande quantidade de dados. Além de permitir que sejam criados dashboards mais criativos, autênticos e que sejam disponíveis em rede.

Devido a melhor performance com grande quantidade de dados e a maior flexibilidade para fazer manipulações e previsões de dados, o Flask foi a plataforma escolhida para o desenvolvimento desse trabalho.

2.4 Machine Learning (ML)

2.4.1 Conceito

Machine Learning, ou em português, Aprendizado de máquina é uma subárea de estudos dentro de Inteligência Artificial (IA). IA é uma tecnologia que tem a capacidade de criar sistemas que podem simular a inteligência humana, a parte de ML é a que se dedica a encontrar padrões, aprender com os dados e prever resultados futuros.

De acordo com Arthur Samuel, *Machine Learning* é o campo de estudo que dá aos computadores a capacidade de aprender sem serem programados explicitamente (GÉRON, 2017). Por exemplo, o filtro de *spam* que tem nos e-mails. Por meio de um programa de ML é possível fornecer alguns exemplos de e-mails que estariam classificados com *spam*, assim o programa consegue aprender com esses exemplos para classificar novos e-mails que tenham características semelhantes.

Devido à crescente quantidade de dados e também à necessidade de se fazer análises que auxiliam na tomada de decisão, técnicas de aprendizado de máquina vem se tornado cada vez mais populares. A partir de modelos criados utilizando tal técnica é possível obter análises de uma enorme quantidade de dados com maior complexidade e também fazer previsões de resultados de forma mais rápida e precisa. Sendo assim, algumas vantagens do aprendizado de máquina, de acordo com Morais (2020), são:

- capacidade de aprendizado e melhoria com seus próprios erros;
- velocidade na análise de dados;
- melhoria na gestão de dados;
- apresenta soluções para problemas reais;
- permite automatização de processos;
- redução de custos, uma vez que fornece informações que podem apontar desperdícios.

Pode-se perceber a aplicação do aprendizado de máquina em diversas atividades do dia-a-dia, como em resultados de pesquisas na *web*, filmes recomendados em plataformas de *streaming*, previsão de falhas em equipamentos, filtragem de *spam* em e-mail, detecção de invasão ou fraudes, entre muitas outras atividades.

2.4.2 Tipos de Machine Learning

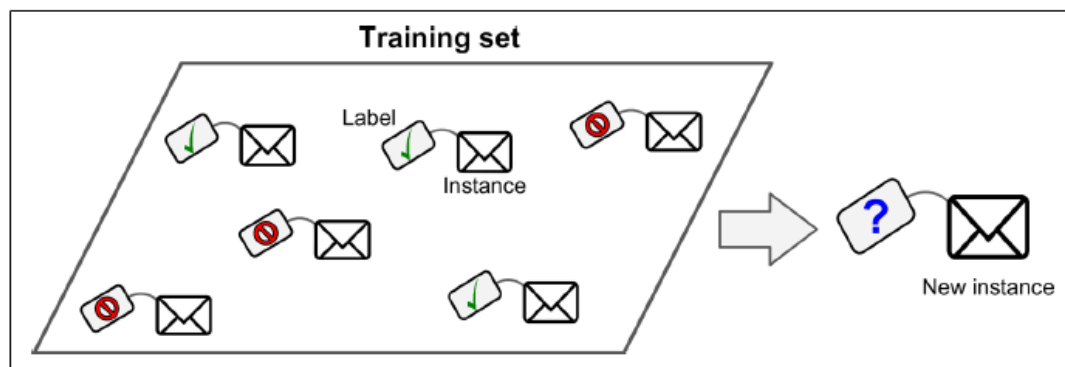
Há vários tipos de algoritmos de ML e eles são classificados em três categorias, segundo Géron (2017):

- Se eles são treinados ou não com uma supervisão humana. Sendo assim podem ser classificados em **aprendizado supervisionado, não supervisionado e por reforço**.
- se eles podem aprender de forma incremental online ou não. Sendo **aprendizado online ou por lote**.
- Se eles podem trabalhar comparando dados ou detectando padrões de treinamento. Sendo **aprendizado baseado em instância ou em modelo**.

2.4.2.1 Aprendizado Supervisionado e Não Supervisionado

Os sistemas supervisionados são os mais comuns. Eles precisam de dados de treino que incluem alguns resultados (*targets*) para que o sistema seja treinado e possa prever os novos dados. A Fig. 11 apresenta um exemplo de aprendizado supervisionado disponível em Géron (2017).

Figura 11 - Aprendizado supervisionado.



Fonte: GÉRON,2017.

Um exemplo são as classificações de e-mails, onde eles podem ser separados em *spam* ou não a partir de determinadas características que contém no corpo do e-mail. Outro exemplo desse tipo de aprendizado é o que está presente em algoritmos de predição do valor de um carro. Os dados de treinamento contêm algumas características

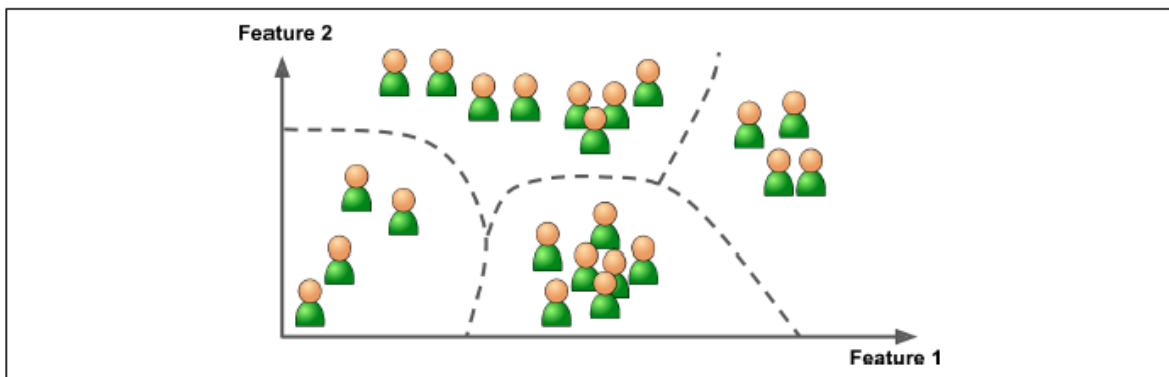
como ano de fabricação, marca, quilometragem, etc, além de receber os preços (rótulos) desses dados. A partir dessas informações, o algoritmo consegue prever os preços de carros futuros.

Alguns dos algoritmos de ML mais importantes que utilizam aprendizado supervisionado são:

- Regressão Linear;
- Regressão Logística;
- SVM (*Support Vector Machine* ou Máquina de Vetores de Suporte);
- KNN (*K-Nearest Neighbors* ou K vizinhos mais próximos);
- Árvores de Decisões;
- Floresta Randômica;
- Redes Neurais.

Já os sistemas não supervisionados são treinados somente com os atributos/ características, sem os resultados e para isso ele encontra padrões e características semelhantes. O aprendizado não supervisionado é um tipo de aprendizado auto-organizado que nos ajuda a descobrir padrões em nossos dados relacionados à vários recursos (GÉRON, 2017). Um exemplo são os algoritmos usados para a recomendação de filmes nas plataformas de *streaming*, em que a partir de escolhas do usuário, o sistema subdivide em grupos com escolhas semelhantes e indica tipos de filmes parecidos. Na Fig.12 é possível identificar os grupos.

Figura 12 - Clusterização com aprendizado não supervisionado.



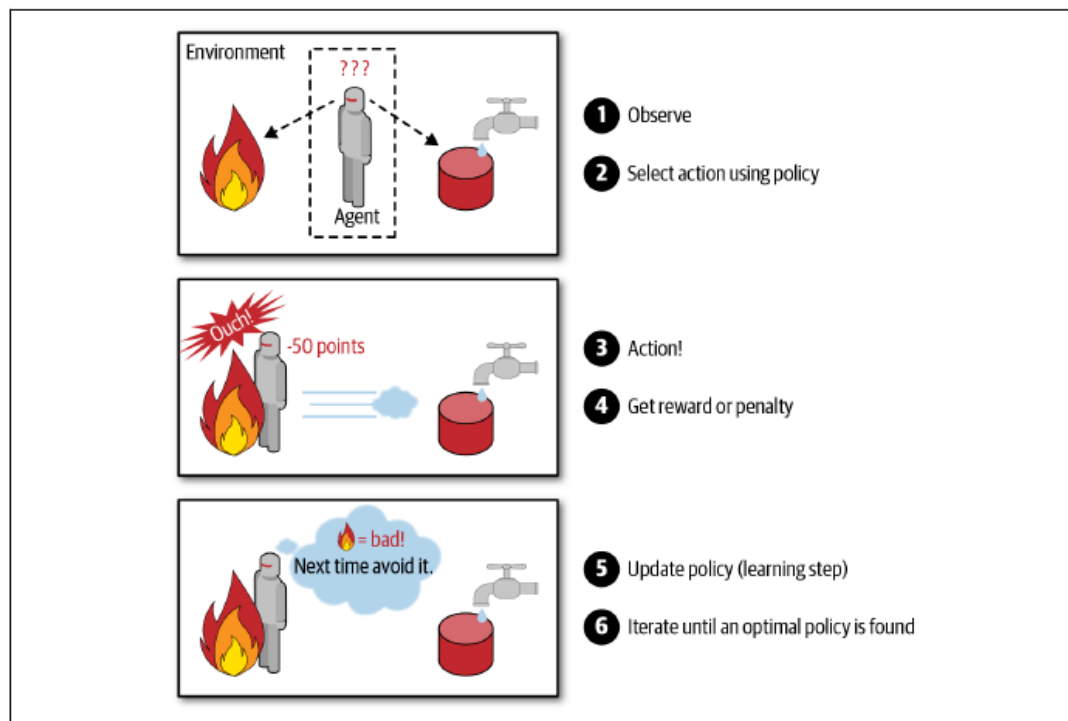
Fonte: GÉRON, 2019.

A clusterização é o processo de dividir o espaço de dados ou pontos de dados em vários grupos, de modo que os pontos de dados nos mesmos grupos sejam mais semelhantes a outros pontos de dados no mesmo grupo e diferentes dos pontos de dados em outros grupos (ILEOH, 2018). Os principais algoritmos de aprendizado não supervisionado são:

- K-Means (K-Médias);
- Análises de Clusters Hierárquicos.

Já os sistemas de aprendizado por reforço são baseados em punição e recompensa, é realizada uma interação entre o sistema de aprendizagem e o seu ambiente, dentro do qual o sistema procura alcançar um objetivo específico apesar da presença de incertezas (GÉRON, 2019). A Fig. 13 apresenta o esquema de penalizações.

Figura 13 - Aprendizado por reforço.



Fonte: GÉRON,2019.

2.4.2.2 Aprendizado Online ou por Lote

Outro tipo de classificação de sistemas de Aprendizado de Máquina é quanto ao seu aprendizado por lotes ou *online*. O aprendizado por lotes utiliza todos os dados de treino, mas de uma forma não incremental. Como deve ser treinado com uma grande quantidade de dados, esse tipo de sistema precisa de um poder computacional maior e, geralmente, é feito *offline*.

Para um sistema que apresenta muitas alterações ou que recebem novos dados constantemente, utiliza-se o aprendizado online. Nesse tipo de aprendizado, o algoritmo é treinado incrementalmente e sempre recebe novos dados ou grupos de dados. Além disso, é ideal para grandes volumes de informações, pois o algoritmo carrega parte dos dados, executa o algoritmo em um grupo de treinamento e repete o processo até tudo tenha passado por esse processo.

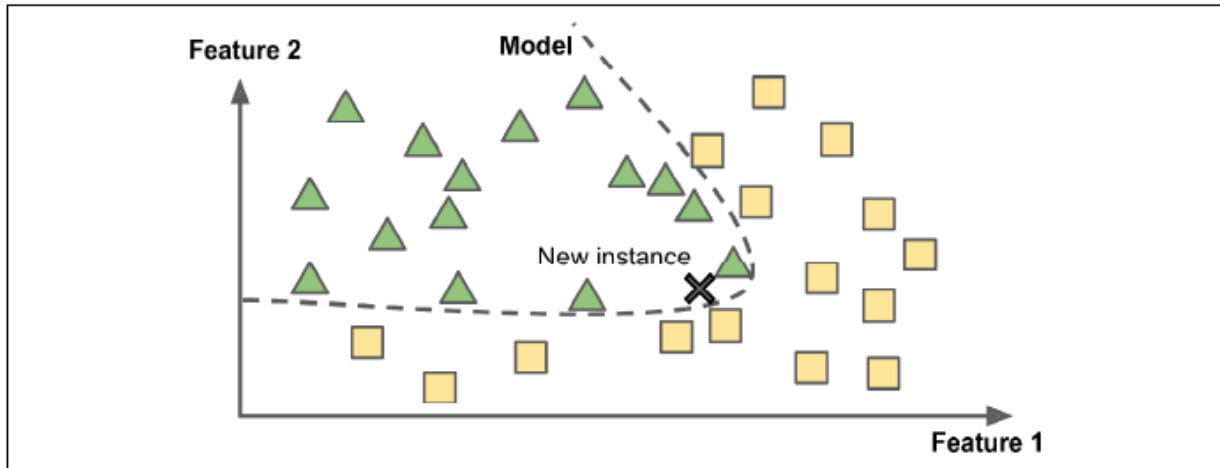
Um parâmetro importante nesse tipo de aprendizado é a taxa de aprendizado, o quanto eles conseguem se adaptar às mudanças de dados. Se a taxa de aprendizado for muito grande, o sistema consegue se adaptar rapidamente, porém ele tende a esquecer os dados antigos facilmente. Já se tiver uma taxa menor, o sistema de aprendizado aprende de forma mais lenta, porém é mais sensível aos dados que não são representativos. Um grande desafio no aprendizado online é que, se incluirmos dados ruins no sistema, seu desempenho diminuirá gradualmente (GÉRON, 2017).

2.4.2.3 Aprendizado Baseado em Instância ou Modelo

Uma outra categoria dos aprendizados de Aprendizado de Máquina é quanto a sua generalização ou capacidade de reconhecer padrões.

Os aprendizados baseados em modelos utilizam um modelo para fazer as previsões. Ou seja, o seu sistema pode ser representado por uma equação matemática que irá prever novos dados a partir desse equacionamento. A Fig. 14 apresenta dois tipos de dados que são divididos de acordo com o modelo utilizado.

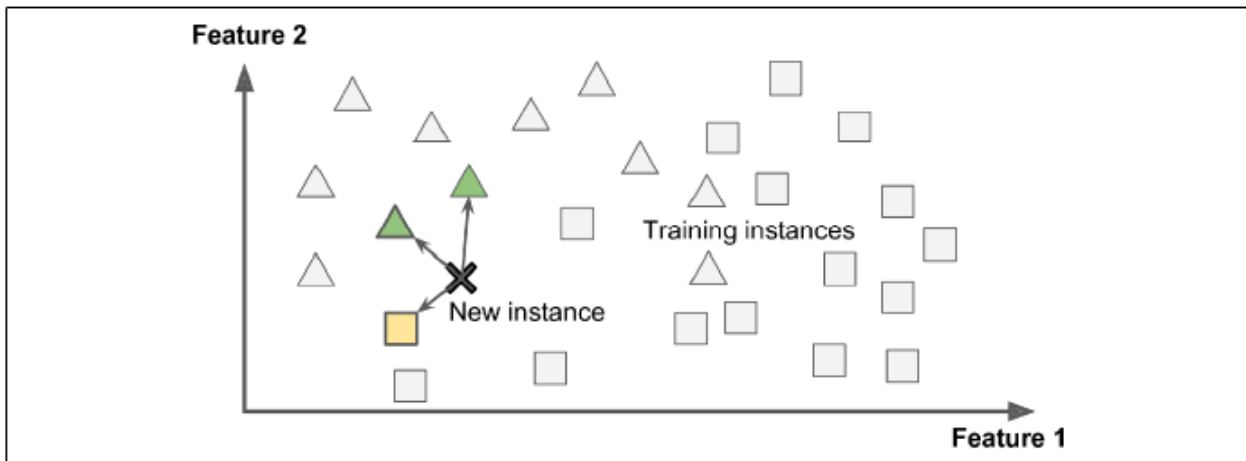
Figura 14 - Aprendizado baseado em modelo.



Fonte: GÉRON, 2019.

Já o aprendizado baseado em instâncias o sistema aprende por meio da memorização e, em seguida, generaliza para novos casos utilizando uma medida de similaridade. Por exemplo, os e-mails identificados como *spam* podem ser classificados por meio de uma similaridade entre os conhecidos. Uma medida seria o número de palavras que um e-mail conhecido como *spam* tem em comum com um e-mail desconhecido, se tiverem muitas palavras parecidas, esse seria classificado com um *spam*. Na Fig. 15 há um exemplo em que os dados novos (*new instance*) são classificados baseados em similaridades.

Figura 15 - Aprendizado baseado em instância.



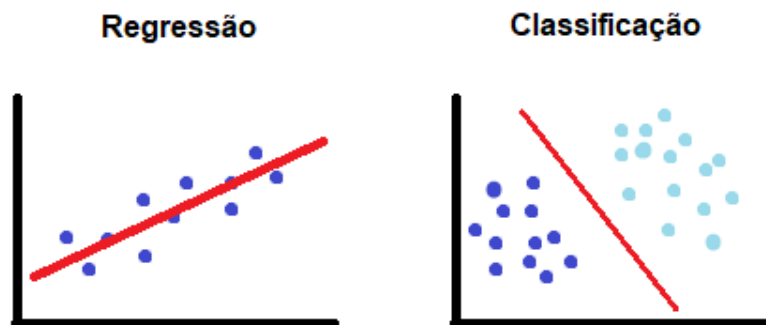
Fonte: GÉRON, 2019.

2.4.3 Algoritmos de Machine Learning

Existem muitos algoritmos de ML diferentes e que conseguem analisar grande quantidade de dados de uma maneira muito mais rápida e precisa que um ser humano. Os algoritmos de predição de aprendizado de máquina fazem uma análise preditiva dos dados, essa tem como objetivo agrupar dados, fornecer *insights* e indicar padrões e tendências, ajudando cada vez mais as pessoas a tomarem decisões.

Na análise preditiva existem dois tipos de modelos de aprendizado de máquina quando se trata do tipo de rótulo dos dados (supervisionados): os modelos de regressão e de classificação. Os modelos de regressão preveem resultados com base em rótulos numéricos, já os de classificação dividem seus dados em conjuntos finitos (azul claro e azul escuro), como pode ser observado na Fig.16.

Figura 16 - Modelo de regressão e classificação.



A seguir serão detalhados dois algoritmos de ML que serão utilizados nesse projeto. O primeiro é o modelo de regressão linear e o outro modelo é de clusterização (não supervisionado).

2.4.3.1 Regressão Linear

A regressão linear é representada por uma equação matemática que descreve a relação entre duas ou mais variáveis. A regressão linear simples se dá quando tem duas variáveis. Quando tem mais variáveis é chamada de regressão linear múltipla. Esse trabalho utilizará a regressão linear simples conforme na equação 1.

$$y = \theta_1 x + \theta_0 + \varepsilon \quad (1)$$

Sendo:

y é a variável dependente;

x a variável independente;

θ_1 é o coeficiente de inclinação da reta que fornece a variação de y a partir de uma variação correspondente de x ;

θ_0 é a intersecção da reta com o eixo y ;

ε representa o erro da relação entre as duas variáveis.

Para treinar o modelo de regressão linear é necessário achar os valores das variáveis de θ que minimizam o erro. Para isso serão feitas algumas manipulações na equação. Primeiramente será feito o agrupamento dos θ , como na equação 2.

$$y = \theta x + \varepsilon \quad (2)$$

Isolando o erro e fazendo a transposta de ambos os lados se obtém as equações 3 e 4.

$$\varepsilon = y - \theta x \quad (3)$$

$$\varepsilon^T = (y - \theta x)^T \quad (4)$$

Para saber o valor que minimiza o θ , deve ser calculada a derivada do erro quadrático e igualada a zero, calculando o ponto mínimo e chegando na equação 5.

$$\theta = (x^T \cdot x)^{-1} \cdot x^T \cdot y \quad (5)$$

A partir dessa equação é possível calcular os valores de θ que minimiza o erro e que melhor se encaixa a equação da regressão linear.

2.4.3.2 K-Means

K-Means (K-Médias) é um algoritmo de aprendizado de máquina não supervisionado que é utilizado para fornecer uma clusterização dos dados. Esse

algoritmo é capaz de agrupar os dados em “k” grupos por meio da identificação de padrões e similaridade entre eles.

A forma mais comum e mais utilizada entre os cientistas de dados para a identificação das similaridades é por meio da distância entre os valores de um atributo e os demais. A seguir serão detalhados o passo a passo desse algoritmo:

Na primeira etapa acontece a geração aleatória dos “k” clusters/grupos. Nessa divisão cada grupo é formado a partir de um centroide, também chamado de centro geométrico, que é o ponto central do cluster.

Na segunda etapa é feita o cálculo da distância euclidiana entre todos os pontos de dados e os centroides. Os pontos que estiverem mais próximos de cada centroide são formados os grupos.

Posteriormente, é recalculado os centroides novamente para reajustar os clusters. Nessa etapa é calculada a média dos valores dos pontos até o centroide e assim o novo centroide será no valor médio (SANTANA,2017). Como mostrado na Fig. 17 onde tem 2 clusters, azul e a laranja.

Figura 17 - Formação do algoritmo K-Means.



Fonte: SANTANA, 2017.

As etapas anteriores são repetidas várias vezes até que nenhum dos dados do cluster se alterem ou atinja um número máximo de interações.

2.4.4 Desafios para Aplicações de Machine Learning

A aplicação de algoritmos de Aprendizado de Máquina requer muitos cuidados durante a preparação dos seus dados para que não ocorra erros e nem apareça

informações que não condizem com a realidade do processo. A seguir estão listados os principais problemas que podem afetar os resultados do aprendizado de máquina, segundo Géron (2017):

- Quantidade insuficiente de dados para treinar o algoritmo: É necessária uma grande quantidade de dados para treinar o algoritmo de uma forma que ele consiga funcionar adequadamente, reconhecer os padrões e prever um resultado coerente;
- Baixa qualidade dos dados: Se os dados não forem de qualidade e tiverem erros, outliers e ruídos, será muito mais difícil para o algoritmo prever os resultados e ter uma boa performance. A etapa de limpeza dos dados é mais importante para quem trabalha com análise de dados.
- Dados não representativos: A intenção é obter um modelo que seja geral e representativo caso sejam acrescentados mais dados, portanto, os dados de treinamento precisam contemplar boa parte do problema. Caso contrário, será treinado um modelo que dificilmente fará previsões precisas.
- Características irrelevantes: Dados irrelevantes podem prejudicar o desempenho do algoritmo. Portanto é importante selecionar os atributos que serão úteis e combinar características que podem produzir uma outra mais relevante.
- *Overfitting* dos dados de treino: quando o modelo é muito complexo em relação à quantidade e ao ruído dos dados de treinamento. Isso pode gerar uma generalização que não condiz com o sistema. Para resolver esse problema pode-se simplificar o modelo, coletar mais dados de treinamento ou reduzir o ruído.
- *Underfitting* dos dados de treino: É o oposto do item anterior, onde o modelo é simples e não representa o sistema. Para resolver esse problema pode-se selecionar um modelo mais complexo, fazer novas combinações de características ou reduzir as restrições do modelo como os hiperparâmetros.

Assim que treinar o modelo com os dados de treinamento é necessário avaliá-lo para saber se esse modelo é geral para todos os dados ou se precisa treiná-lo novamente com uma otimização.

2.4.5 Validação e Testes

Durante o processo de criação de um modelo de aprendizado de máquina é preciso medir a qualidade dele de acordo com o objetivo da tarefa. Existem funções matemáticas que ajudam a avaliar a capacidade de erro e acerto dos modelos.

Para que se tenha um modelo generalizado e que represente o sistema é necessário fazer testes em novos casos. Para que o teste não seja feito apenas quando chega ao usuário final, a melhor opção é dividir os dados em dois conjuntos: dados de treinamento e dados de teste.

O conjunto de dados de treinamento é utilizado para treinar o modelo de aprendizado de máquina. Já o conjunto de dados de teste é utilizado para testar o modelo treinado com os dados de treino e assim conseguir fazer uma avaliação se o modelo foi generalizado ou não.

Segundo Filho ([s.d]), uma referência mundial em ML, podemos descrever algumas métricas que podem ser usadas na validação de um modelo. Para modelos de classificação temos as seguintes métricas mais usadas:

- **Acurácia:** é proximidade entre o valor obtido experimentalmente e o valor verdadeiro. É basicamente o número de acertos (positivos) dividido pelo número total de exemplos.
- **Precisão:** É o número de exemplos classificados como pertencentes a uma classe, que realmente são daquela classe (positivos verdadeiros), dividido pela soma entre este número, e o número de exemplos classificados nesta classe, mas que pertencem a outras (falsos positivos).

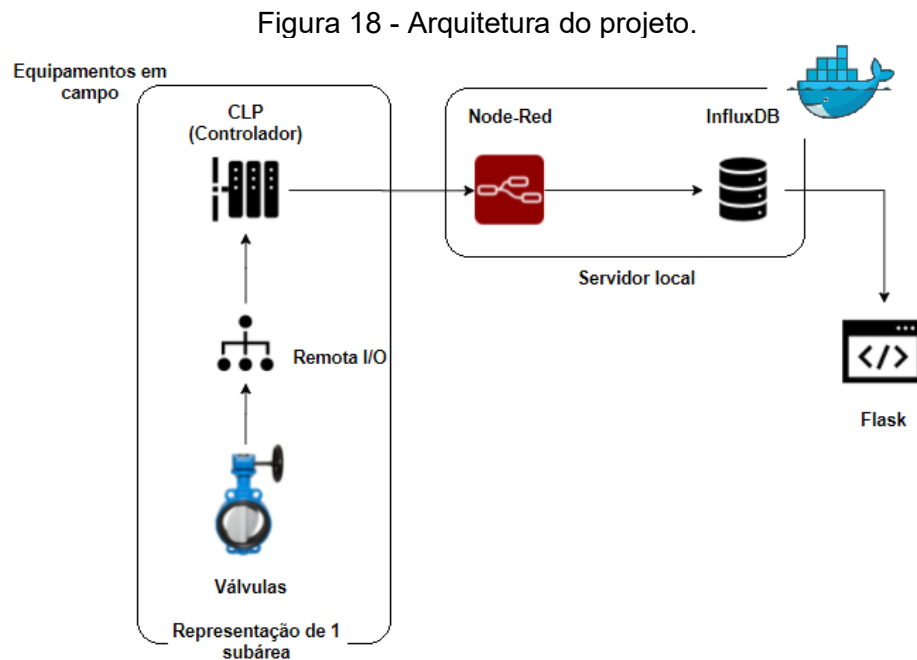
E as métricas mais comuns para modelos de regressão são:

- **Erro absoluto médio (MAE):** É o valor absoluto entre a previsão do modelo e o valor real.
- **Erro quadrático médio (MSE):** É a média dos erros do modelo ao quadrado. Nesse caso, as diferenças menores tem menos importância que as maiores.

3. DESENVOLVIMENTO

Esse trabalho propõe desenvolver uma ferramenta capaz de fornecer uma previsão de falhas em válvulas a partir da análise de dados dos equipamentos, gerando uma visualização das válvulas críticas para que se faça uma manutenção.

Para o desenvolvimento do projeto serão necessárias algumas etapas, como a criação de uma lógica de programação para a coleta de variáveis das válvulas, coleta dos dados e armazenamento em um banco de dados, tratamento e análise das variáveis coletadas, predição do comportamento do equipamento e desenvolvimento de uma plataforma *web* para a visualização e análise. A Fig.18 a seguir ilustra a arquitetura do projeto e as ferramentas que serão utilizadas.



De acordo com a Fig. 18, as válvulas estão conectadas em uma remota que fica no campo e é ligada a um CLP localizado em uma sala climatizada. Assim, os dados são coletados por um CLP e armazenados em variáveis internas (DB), durante o dia todo. Em seguida, o Node-Red irá coletar esses dados das variáveis do CLP uma vez ao dia, todo os dias às 18h. Foi escolhido esse horário pois é horário padrão para zerar as variáveis dos CLPs na empresa. Após isso, os dados serão armazenados em um banco de dados

InfluxDB, localizado dentro de um Docker em um servidor local. Posteriormente, os dados serão disponibilizados para visualização.

A visualização da Fig. 18 representa 1 subárea, porém todas as subáreas seguem o mesmo fluxo de dados. As válvulas, os CLPs e as remotas estão localizadas em campo. Já o Node-Red e o InfluxDB funcionam dentro de Docker em um servidor local na empresa. E o Flask é possível acessar pelo navegador por um endereço na rede industrial.

O desenvolvimento do projeto e sua implementação duraram sete meses ao todo, porém durante a construção do dashboard houveram testes com o usuário para fazer melhorias e adaptar o projeto da melhor forma. Todas as etapas estão detalhadas a seguir.

3.1 Coleta de Dados

A primeira etapa para o projeto é a coleta de dados das válvulas do processo. Essa etapa é importante para obter os dados que serão utilizados para a predição de falhas. Os dados são coletados das válvulas por meio de uma lógica no CLP e enviadas para o Node-Red como será explicado.

Para esse projeto foram analisadas 3000 válvulas de controle tipo borboleta com atuador pneumático de diferentes dimensões dentro do processo industrial.

3.1.1 Lógica do Processo

O CLP é responsável por executar a lógica de controle automático dos equipamentos. O CLP utilizado foi do fabricante Siemens, modelos S7-300. Para fazer a interação e a lógica de programação foi utilizado um programa chamado SIMATIC Step 7, do mesmo fabricante. A Fig. 19 mostra a página inicial que contém os blocos do CLP selecionado.

Figura 19 - Simatic Step 7.

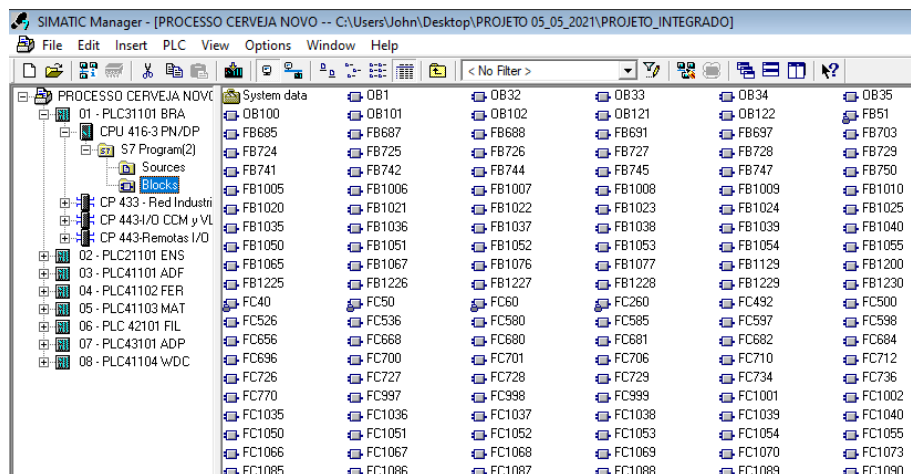


Figura 20 - FC do projeto.

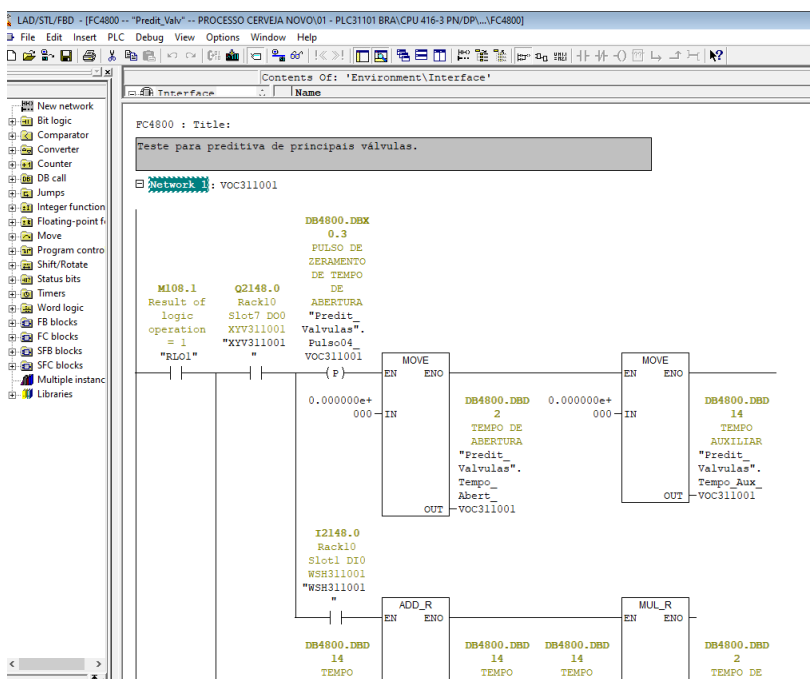


Figura 21 - DB do projeto.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Pulso01_VOC311001	BOOL	FALSE	PULSO DE TEMPO DE ABERTURA
+0.1	Pulso02_VOC311001	BOOL	FALSE	PULSO DE CONTAGEM DE ACIONAMENTOS
+0.2	Pulso03_VOC311001	BOOL	FALSE	PULSO DE CONTAGEM DE FALHAS
+0.3	Pulso04_VOC311001	BOOL	FALSE	PULSO DE ZERAMENTO DE TEMPO DE ABERTURA
+0.4	Pulso05_VOC311001	BOOL	FALSE	PULSO DE ZERAMENTO DE FALHA E ACIONAMENTO
+0.5	Pulso06_VOC311001	BOOL	FALSE	PULSO DE COMANDO MANUAL
+0.6	Pulso07_VOC311001	BOOL	FALSE	reserva
+0.7	Pulso08_VOC311001	BOOL	FALSE	reserva
+2.0	Tempo_Abert_VOC311001	REAL	0.000000e+000	TEMPO DE ABERTURA
+6.0	Cont_Acion_VOC311001	REAL	0.000000e+000	CONTAGEM DE ACIONAMENTO
+10.0	Cont_Falha_VOC311001	REAL	0.000000e+000	CONTAGEM DE FALHAS
+14.0	Tempo_Aux_VOC311001	REAL	0.000000e+000	TEMPO AUXILIAR
+18.0	Pulso01_VOC311002	BOOL	FALSE	
+18.1	Pulso02_VOC311002	BOOL	FALSE	
+18.2	Pulso03_VOC311002	BOOL	FALSE	
+18.3	Pulso04_VOC311002	BOOL	FALSE	
+18.4	Pulso05_VOC311002	BOOL	FALSE	
+18.5	Pulso06_VOC311002	BOOL	FALSE	
+18.6	Pulso07_VOC311002	BOOL	FALSE	
+18.7	Pulso08_VOC311002	BOOL	FALSE	
+20.0	Tempo_Abert_VOC311002	REAL	0.000000e+000	
+24.0	Cont_Acion_VOC311002	REAL	0.000000e+000	
+28.0	Cont_Falha_VOC311002	REAL	0.000000e+000	
+32.0	Tempo_Aux_VOC311002	REAL	0.000000e+000	
+36.0	Pulso01_VOC311003	BOOL	FALSE	
+36.1	Pulso02_VOC311003	BOOL	FALSE	

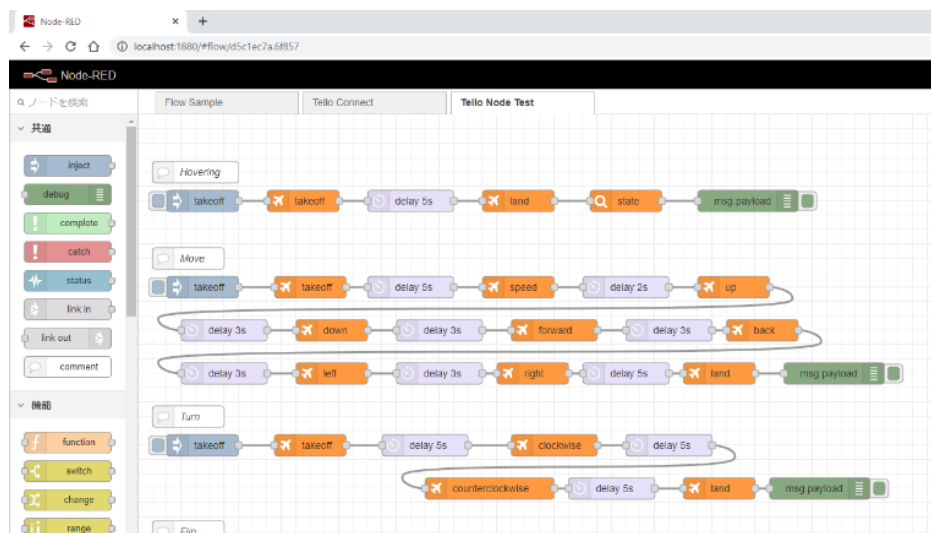
Ao todo foram cadastradas 3000 válvulas de um processo de produção industrial, sendo divididas em 8 subáreas. Cada subárea tem de 1 a 2 CLPs e algumas remotas (média de 10 por área) e foram feitos uma DB e FCs (algumas áreas precisaram de mais

de uma FC por conta da quantidade de válvulas cadastradas), porém com a mesma lógica mostrada na Fig. 20.

3.1.2 Node-Red

De acordo com o próprio site, o Node-Red é uma ferramenta de programação para conectar dispositivos de *hardware*, APIs (Interface de Programação de Aplicativos, em português) e serviços online. Ele fornece uma interface fácil e intuitiva para criar protótipos, fazer conexões com dispositivos IoT (*Internet of Things* ou Internet das coisas), sensores, CLPs, etc. A Fig. 22 mostra um exemplo da interface do Node-Red.

Figura 22 - Exemplo da interface do Node-Red.



Fonte: <https://flows.nodered.org/node/node-red-contrib-tello>.

O Node-Red foi utilizado para fornecer a ponte entre a coleta de dados do CLP e o armazenamento deles em um banco de dados local. Abaixo estão listadas algumas vantagens:

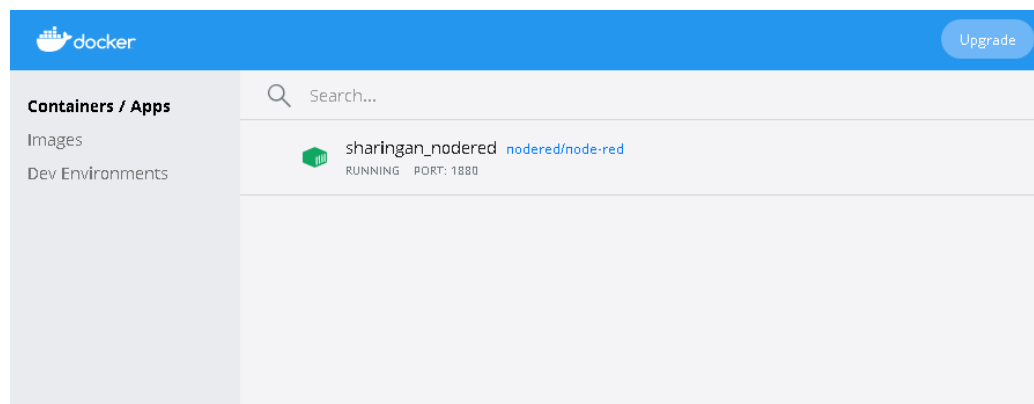
- Pode ser executado em *hardware* de baixo custo;
- Pode ser executado na nuvem;
- Facilidade em criar as conexões (baseado em arrastar e mover blocos);
- Interface intuitiva e amigável;
- Não tem a necessidade de programação, porém aceita diferentes tipos de linguagens de programação como JavaScript (padrão), Python e Swift;
- Pode ser executado em um Docker;

- Qualquer pessoa pode criar seus nós, fazer as conexões e adaptar os fluxos às necessidades.

Para a aplicação do projeto, o Node-Red foi criado dentro de um container em uma plataforma chamada Docker e dentro de um servidor localizado localmente na empresa. O Docker é uma plataforma para a criação e execução de containers e o container é uma forma de empacotar a aplicação e as bibliotecas de forma padronizada e simplificada. Essa escolha foi feita pela razão de que em um Docker pode rodar várias aplicações isoladas e também porque é mais fácil e simples executar em outro computador sem fazer alterações, permitindo a escalabilidade do projeto para outras indústrias.

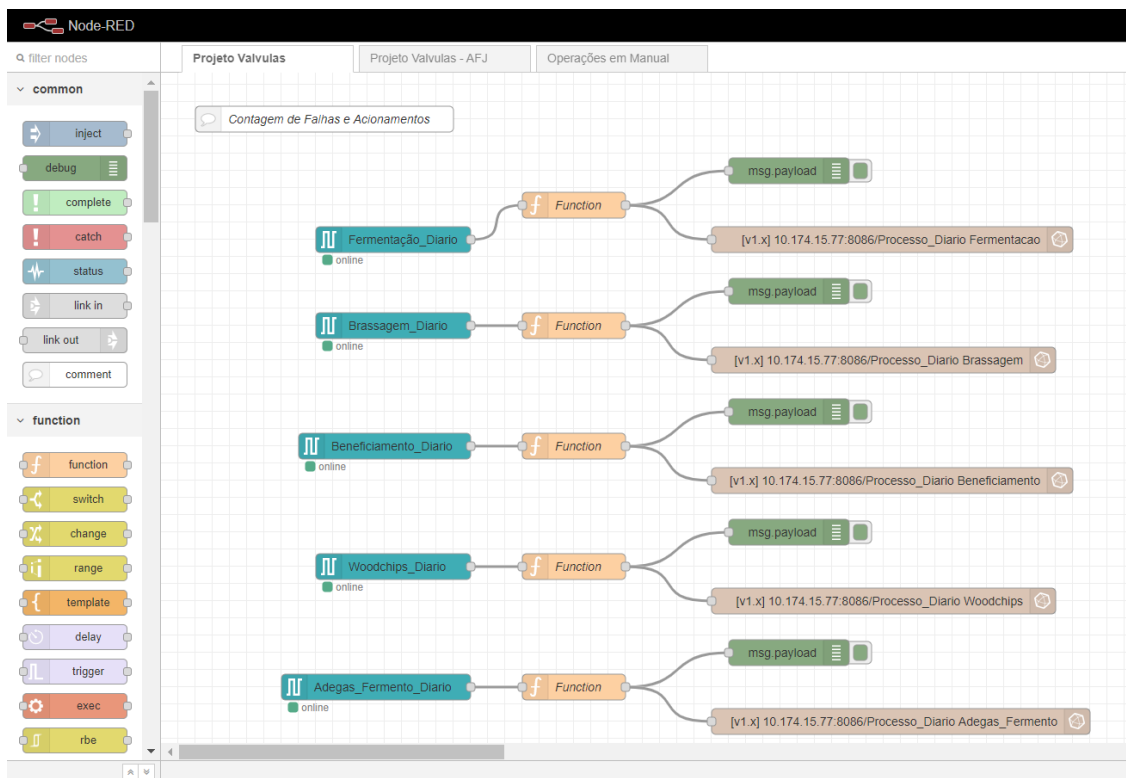
Primeiramente, foram instalados o Docker e a imagem do Node-Red, como na Fig. 23.

Figura 23 - Docker com Node-Red.



Com o container do Node-Red feito, foi possível acessar a página na web onde tem a interface para a programação. A Fig. 24 contém um exemplo da página com as configurações utilizadas no projeto.

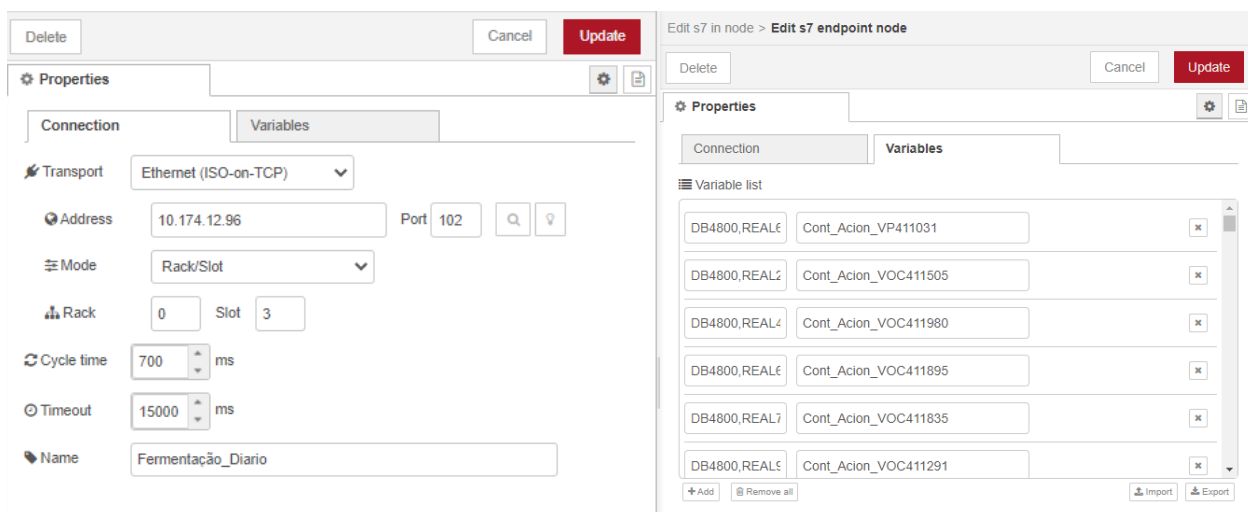
Figura 24 - Node-Red do projeto.



Foram criados vários blocos de conexão com o CLP (azul), funções em JavaScript (amarelo) para que os dados fossem coletados às 18h e também blocos de conexão com o banco de dados InfluxDB (marrom). São coletados três variáveis de cada válvula todos os dias, contagem de acionamentos, contagem de falhas e tempo de abertura das válvulas. Os blocos foram divididos por subáreas, porém com as configurações semelhantes.

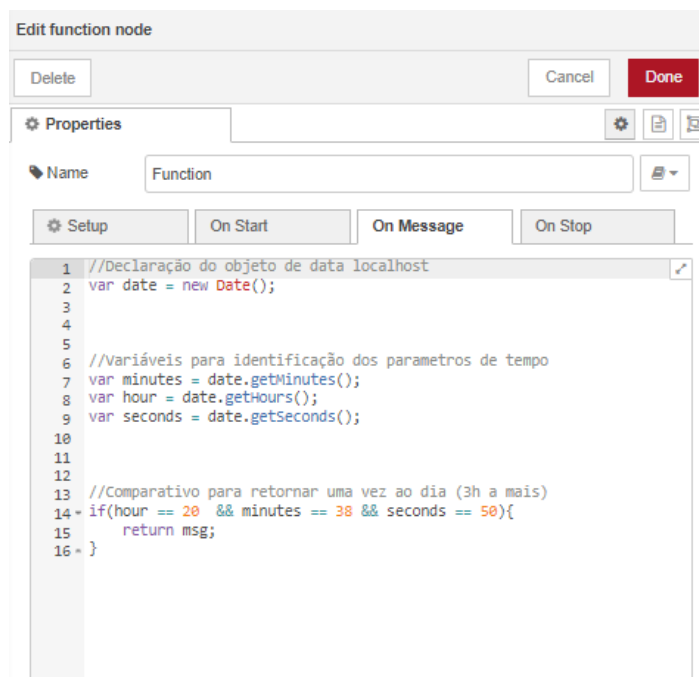
Para o Node-Red fazer a coleta de dados dos CLPs é necessário configurar o driver Profinet (ISO) com o respectivo IP do equipamento, e também as variáveis internas do CLP conforme o exemplo mostrado na Fig. 25.

Figura 25 - Bloco CLP.



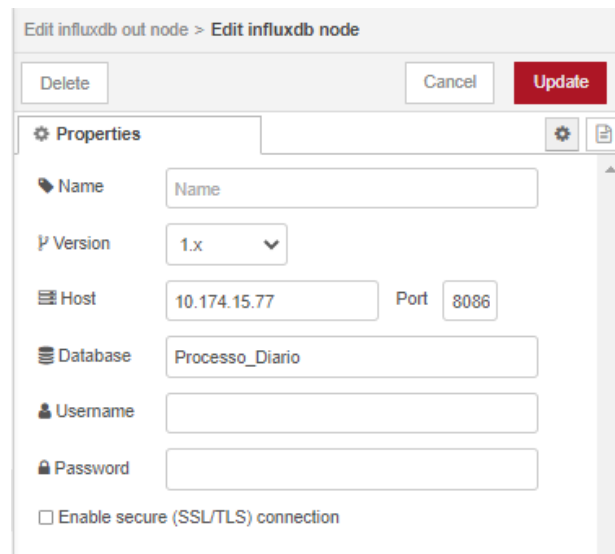
Na função de coletar os dados todos os dias as 18h foi feito o código da Fig. 26, onde o horário do Node-Red é 3h a mais do horário de Brasília.

Figura 26 - Função Node-Red.



E para que os dados coletados pelo Node-Red fossem armazenados, foi criado um bloco para a conexão com o banco de dados, utilizando um bloco do InfluxDB e as configurações mostradas na Fig. 27.

Figura 27 - Configurações do InfluxDB no Node-Red.



The image shows a configuration window titled "Edit influxdb out node > Edit influxdb node". At the top, there are three buttons: "Delete", "Cancel", and "Update". Below this is a "Properties" section with a gear icon and a document icon. The properties are as follows:

- Name:** A text input field containing "Name".
- Version:** A dropdown menu set to "1.x".
- Host:** A text input field containing "10.174.15.77".
- Port:** A text input field containing "8086".
- Database:** A text input field containing "Processo_Diario".
- Username:** An empty text input field.
- Password:** An empty text input field.
- Enable secure (SSL/TLS) connection:** An unchecked checkbox.

Para fazer essa conexão foi necessário criar e configurar o banco de dados, como será detalhado no próximo tópico.

3.2 Banco de Dados – InfluxDB

InfluxDB é um banco de dados de código aberto do tipo NoSQL (*Not Only SQL*), ou seja, banco de dados não relacional. Ele permite que sejam realizadas consultas a partir de uma sintaxe muito parecida com a utilizada em bancos de dados SQL (*Structured Query Language*), ou seja, banco de dados relacional baseado em tabelas. A partir de alguns comandos é possível criar, excluir, adicionar e modificar dados do banco. A seguir estão listadas algumas características desse banco de dados.

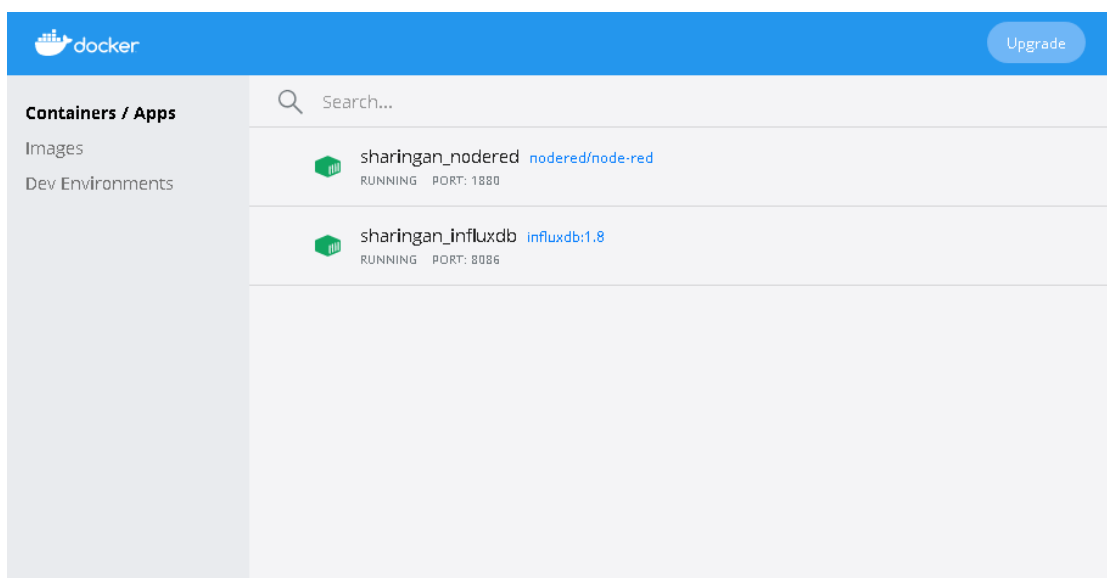
- Alta escalabilidade horizontal;
- Monitoramento em tempo real;
- Armazenamento de grande quantidade de dados;
- Pode ser executado em Docker;
- Tolerante ao particionamento de informações;
- Flexibilidade de armazenar qualquer tipo de informação;

- Agilidade para consultar os dados.

A principal característica que levou à escolha desse banco de dados foi a capacidade que ele tem de armazenar grande quantidade de dados, já que serão armazenados dados de três mil válvulas diariamente.

O banco de dados InfluxDB também foi criado em um container no Docker juntamente com o Node-Red. Após a criação do container foi feita a conexão do banco de dados conforme a Fig. 28.

Figura 28 - Docker com o Node-Red e InfluxDB.



E em seguida foi criado o banco de dados chamado Válvulas e também as tabelas das diferentes áreas do processo. A Fig. 29 mostra como foi feita a estruturação do banco de dados, onde os dados foram divididos em várias tabelas, uma para cada subárea do processo.

Figura 29 - Banco de dados InfluxDB.

The screenshot shows the InfluxDB Studio interface. The query editor contains the SQL query: `SELECT * FROM "Adegas_Fermento"`. The results pane displays a table with 15 columns and 15 rows of data. The first column is labeled '#', the second is 'time', and the remaining columns are labeled 'Cont_Acion_VOC401002' through 'Cont_Acion_VOC401015'. The data shows values for each column over a period from 26/03/2021 to 05/04/2021.

#	time	Cont_Acion_VOC401002	Cont_Acion_VOC401003	Cont_Acion_VOC401004	Cont_Acion_VOC401005	Cont_Acion_VOC401007	Cont_Acion_VOC401008	Cont_Acion_VOC401009	Cont_Acion_VOC401...	Co ^
1	26/03/2021 20:38:50	0	0	0	0	0	0	0	10	0
2	27/03/2021 20:38:50	0	0	2	14	0	1	0	8	0
3	27/03/2021 20:38:50	0	0	2	14	0	1	0	8	0
4	28/03/2021 20:38:50	1	0	0	6	0	2	1	20	0
5	29/03/2021 20:38:50	0	0	2	16	0	1	0	0	0
6	29/03/2021 20:38:50	0	0	2	16	0	1	0	0	0
7	30/03/2021 20:38:50	0	5	1	14	0	0	1	20	0
8	31/03/2021 20:38:50	0	0	0	0	0	2	0	0	1
9	01/04/2021 20:38:50	1	0	0	15	0	0	0	0	0
10	02/04/2021 20:38:50	0	0	0	0	0	0	2	0	0
11	03/04/2021 20:38:50	0	0	0	0	1	0	0	24	0
12	03/04/2021 20:38:50	0	0	0	0	1	0	0	24	0
13	04/04/2021 20:38:50	0	0	0	0	0	0	0	0	0
14	04/04/2021 20:38:50	0	0	0	0	0	0	0	0	0
15	05/04/2021 20:38:50	0	0	2	8	0	0	2	19	0

Para ter essa visualização do banco de dados InfluxDB, foi utilizada um programa chamado InfluxDB Studio, desenvolvido pelo CymaticLabs e disponível no github. Foi necessário apenas criar uma conexão com o endereço e a porta do servidor que estava com o banco de dados, como na Fig. 30.

Figura 30 - Conexão InfluxDB.

The screenshot shows the 'Connection Settings' dialog box in InfluxDB Studio. The fields are filled with the following information:

- Name: Valvulas
- Address: 10.174.15.77 : 8086
- Database: (empty)
- User Name: (empty)
- Password: (empty)
- Security: Use SSL

Buttons at the bottom include Test, Ping, Save, and Cancel.

Uma vez realizada a conexão do Node-Red com o banco de dados InfluxDB os dados coletados no Node-Red serão armazenados automaticamente no banco de dados.

3.3 Dashboard

O dashboard é uma parte importante onde é possível fazer análises mais visuais do projeto além de fornecer uma interface mais amigável para quem irá utilizar. Para esse projeto, optou-se por fazer um dashboard mais intuitivo para que a operação do processo industrial tivesse mais facilidade em utilizá-lo. Além disso, optou-se por colocar um apelido ao dashboard para que instigasse curiosidade das pessoas para conhecer sobre o projeto, o apelido foi dado de Sharingan. Esse nome surgiu por causa do anime Naruto em que um personagem que possuía essa habilidade conseguia prever os movimentos do oponente, comparando com a previsão das falhas no projeto em questão.

O Flask foi utilizado para criar uma visualização e análise dos dados das válvulas do processo, possibilitando a previsão das falhas.

Devido o Flask ser uma plataforma totalmente programável foram feitas várias tratativas nos dados durante a criação da plataforma. As etapas para a criação da plataforma foram as seguintes:

- Conexão do Flask com o banco de dados;
- Transformar a tabela de formato *wide* para *long* (Fig. 31). O formato *wide* é aquele em que as variáveis estão separadas umas das outras. Por exemplo, cada válvula está em uma coluna separada. E o formato *long* é aquele em que as variáveis estão empilhadas, ou seja, tem uma coluna onde estão todas as válvulas. O formato *long* é melhor para fazer manipulações com os dados;

Figura 31 - Wide x Long.

Wide			Long		
Data	Voc1	Voc2	Data	Válvulas	Acionamentos
01/04	5	10	01/04	Voc1	5
02/04	2	8	01/04	Voc2	10
			02/04	Voc1	2
			02/04	Voc2	8

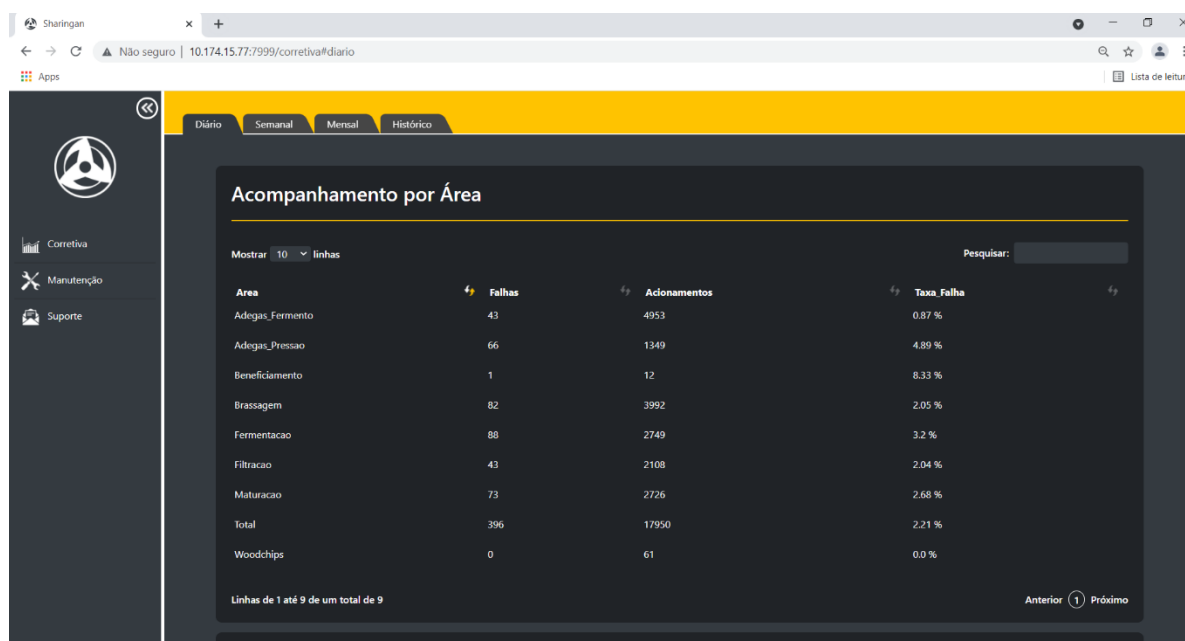
- Separar os dados das válvulas entre número de acionamentos e número de falhas;
- Calcular a porcentagem de taxa de falhas das válvulas, fazendo a divisão entre número de falhas e número de acionamento. Assim é possível identificar as

válvulas que estão falhando mais e precisam de maior atenção por parte do operador para evitar perdas;

- Desenvolver arquivos HTML, JavaScript e CSS para a criação de uma página web que contemplem as visualizações.
- Separar arquivos com funções, manipulações de tabelas e gráficos para que haja uma organização dos arquivos;
- Desenvolver um arquivo chamado `__main__.py` onde será chamado todas as funções e arquivos que foram criadas.

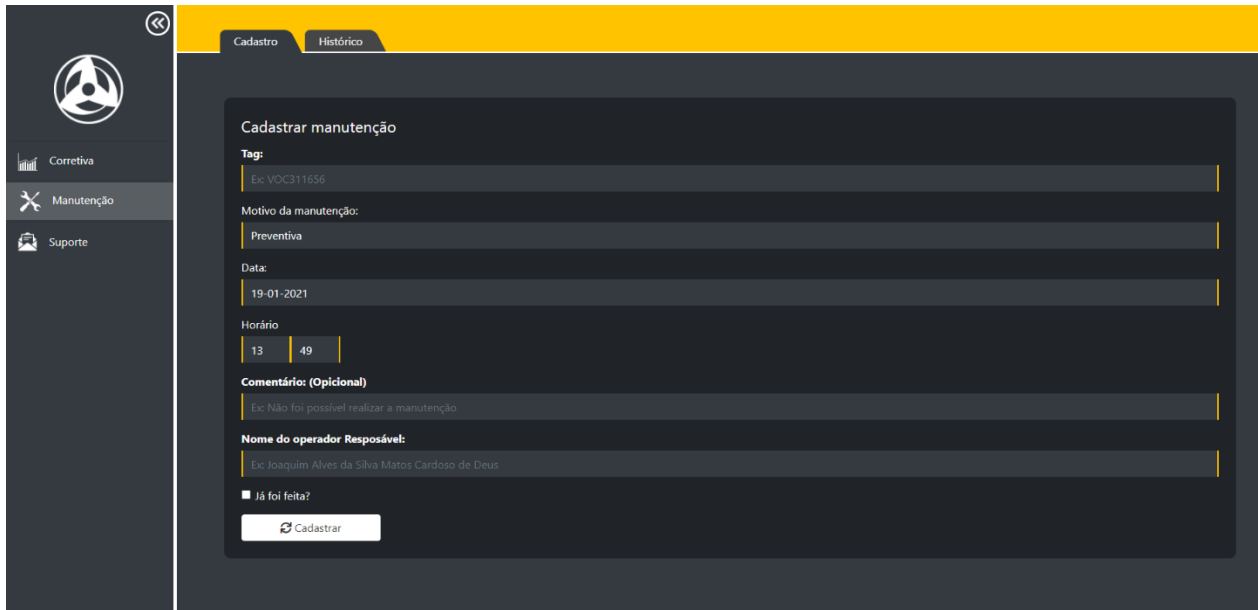
A Fig. 32 mostra a página web que foi criado após todos os procedimentos listados acima.

Figura 32 - Dashboard aba corretiva.



Esse dashboard foi criado primeiramente com a visualização da taxa de falhas das válvulas para se tornar possível uma análise prévia das falhas das válvulas e das válvulas mais críticas (com mais falhas). Foi chamado de aba corretiva porque tem uma amostragem das falhas das válvulas. Também foi acrescentado uma aba em que é possível fazer o cadastro da manutenção das válvulas, conforme a Fig. 33.

Figura 33 - Dashboard: aba manutenção.



The image shows a web dashboard with a dark theme. On the left is a sidebar with a logo and three menu items: 'Corretiva', 'Manutenção', and 'Suporte'. The main area has a yellow header with 'Cadastro' and 'Histórico' tabs. Below the tabs is a form titled 'Cadastrar manutenção'. The form fields are: 'Tag' (with example 'Ex: VOC311656'), 'Motivo da manutenção:' (with example 'Preventiva'), 'Data:' (with example '19-01-2021'), 'Horário' (with dropdowns for '13' and '49'), 'Comentário: (Opcional)' (with example 'Ex: Não foi possível realizar a manutenção'), and 'Nome do operador Responsável:' (with example 'Ex: Joaquim Alves da Silva Matos Cardoso de Deus'). There is a checkbox 'Já foi feita?' and a 'Cadastrar' button at the bottom.

O Flask tem uma característica importante de que pode ser hospedado em um servidor e com seu endereço, a página *web* pode ser acessada por vários computadores ao mesmo tempo que estejam conectados na mesma rede. Sendo assim, esse dashboard foi colocado em um servidor local e qualquer pessoa que esteja na mesma rede pode acessá-lo pelo navegador.

Em seguida foi iniciado um estudo sobre os modelos de predição, assim como os testes para prever as falhas em válvulas, como será abordado no tópico seguinte.

3.4 Predição de Falhas

Depois das etapas de coleta, armazenamento e criação de uma visualização para os dados, iniciou-se a etapa de estudos e desenvolvimento de um modelo de predição de falhas em válvulas que refletisse em tomada de decisão para a empresa.

Para isso foram utilizados modelos preditivos de aprendizado de máquina e as variáveis coletadas do CLP e obtidas com a preparação dos dados, como tempo de abertura, número de acionamentos, número de falhas e taxa de falha das válvulas.

Para o desenvolvimento do algoritmo de predição também foi utilizada a linguagem Python. Utilizou-se esse tipo de linguagem para o desenvolvimento do algoritmo pois já era a linguagem utilizada no projeto, é a linguagem mais utilizada na área de Ciência de

Dados, possui uma comunidade grande e ativa e é uma linguagem que a autora tinha mais facilidade.

Para o desenvolvimento do algoritmo, exploração dos dados, testes e análises foi utilizado a plataforma de código aberto chamada Jupyter Notebook.

Inicialmente optou-se por desenvolver um algoritmo de regressão linear, porém após os resultados do modelo, foi necessário desenvolver outro método de aprendizado de máquina que se encaixasse melhor no sistema. O modelo de classificação não foi utilizado nesse projeto, o motivo principal é em relação as poucas conexões que o CLP tinha disponível para a realização do trabalho, dificultando a inserção de novas variáveis como tipo de líquido e temperatura que passa pela válvula. Sendo assim foi escolhido o modelo não supervisionado K-Means, pois ele não necessita de dados com resultados para treinar o modelo e nem a inserção de novas variáveis.

3.4.1 Regressão Linear

Para a construção do algoritmo de predição foi utilizado as bibliotecas Pandas, Numpy e Sklearn e a biblioteca Matplotlib para a visualização dos dados. A Fig. 34 mostra o código, em python, desenvolvido para a predição.

Figura 34 - Importação dos dados para regressão linear.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split
import math
from sklearn.metrics import r2_score

#Importando dados do arquivo csv da VOC311403
data = pd.read_csv("Dados_Acionamento_2.csv")

data.head()
```

	Tag	Tempo	Dataa	Dia
0	VOC311403	100	13/11/2020	318
1	VOC311403	600	13/11/2020	318
2	VOC311403	300	13/11/2020	318
3	VOC311403	700	13/11/2020	318
4	VOC311403	200	13/11/2020	318

De acordo com a Fig. 34, a primeira etapa para a predição foi a importação das bibliotecas mencionadas anteriormente e em seguida a importação dos dados das *tags* das válvulas com o tempo de abertura das válvulas que estavam armazenados no banco de dados.

Para os testes de regressão linear, foi escolhida uma válvula que atua constantemente para fornecer uma boa quantidade de dados. Foi coletado o tempo de abertura dessa válvula no período de novembro de 2020 até abril de 2021, gerando 1592 dados.

Após algumas análises dos dados, correlação de variáveis, correção de valores nulos e ausentes, foi acrescentado na tabela uma variável que incrementasse uma unidade a cada acionamento da válvula para criar um gráfico que mostrasse o tempo de abertura por cada acionamento da válvula. No início da coleta de dados ocorreram alguns problemas com dados nulos e incorretos e, portanto, foram utilizados 505 dados para a construção do modelo de regressão linear.

Para a regressão, os dados foram divididos entre variáveis dependentes e independentes e posteriormente, aplicados na equação 5 descrita no capítulo 2. A Fig.

35 mostra o código em python, utilizando a equação normal da regressão linear, e também os resultados da predição.

Figura 35 - Aplicação da regressão linear.

```

- Utilizando equação normal

In [12]: # theta = inv(xT.x).xT.y
x_2 = val1["Acionamento"]
y_2 = val1["Tempo"]

x_b = np.c_[np.ones((505,1)),x_2]
theta_best = np.linalg.inv(x_b.T.dot(x_b)).dot(x_b.T).dot(y_2)
theta_best

Out[12]: array([140.07340486,  0.26717579])

In [13]: y_predict = x_2.apply(lambda x : theta_best[0] + theta_best[1]*x)
y_predict

Out[13]: 1087    430.493484
1088    430.760660
1089    431.027836
1090    431.295011
1091    431.562187
...
1587    564.081377
1588    564.348553
1589    564.615729
1590    564.882904
1591    565.150080
Name: Acionamento, Length: 505, dtype: float64

```

Onde:

val1: tabela de dados já com o pré-processamento de dados;

x_2 e y_2: variáveis independente e dependente;

theta_best: θ que minimiza o erro quadrático;

y_predict: valores da predição.

Após a aplicação da regressão linear, foi feita a predição dos valores aplicando o θ encontrado na equação da regressão linear. Assim, foi possível fazer a validação da regressão linear utilizando o erro absoluto médio e erro quadrático médio. Os resultados obtidos serão discutidos no próximo capítulo.

3.4.2 Clusterização: K-Means

Para a construção do algoritmo de clusterização K-Means foram utilizadas as mesmas bibliotecas principais como numpy, pandas, matplotlib e sklearn. Tanto as bibliotecas quanto as variáveis que foram utilizadas no código podem ser observadas na Fig. 36.

Figura 36 - Bibliotecas K-Means.

```
In [39]: # Import das bibliotecas

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime

In [40]: # df: dataframe de acionamentos
# df_abert: dataframe tempo de abertura da valvula
# df_abert_new: tempo de abertura a partir de 22 de marco (equivalente quando comecou a coletar acionamentos)
# dataframe_abert: df_abert_new agrupado por Tag e Data
# dataframe: dataframe final
```

Em seguida foram importados os *dataframes* com o tempo de abertura da válvula e também com o número de acionamentos e falhas, como na Fig. 37. Para esse algoritmo foram utilizados dados de 34 válvulas do processo industrial, as válvulas consideradas mais críticas para o sistema.

Figura 37 - Dados K-Means.

```
df = pd.read_csv("Acionamentos.csv")
df.head()
```

Out[5]:

Unnamed: 0		DateTime	Tags	Acionamentos	Falhas
0	1	2021-05-10 20:38:50	VOC221001	1	0
1	2	2021-05-09 20:38:50	VOC221001	3	0
2	4	2021-05-08 20:38:50	VOC221001	3	0
3	5	2021-05-07 20:38:50	VOC221001	3	0
4	6	2021-05-06 20:38:50	VOC221001	7	0

```
In [6]: # Import dataset open valve time

df_abert = pd.read_excel("Tempo_Abertura.xlsx")
df_abert.head()
```

Out[6]:

	Data	Tag	Tempo
0	2020-11-12 14:38:50	VOC311053	300.0
1	2020-11-12 14:40:32	VOC311053	300.0
2	2020-11-12 14:58:10	VOC311053	300.0
3	1899-12-30 00:00:00	?	0.0
4	2020-11-12 15:00:51	VOC311251	1000.0

Para que os dados sejam mais confiáveis e a predição possa ter uma boa exatidão, foi necessário fazer algumas tratativas de dados. Como por exemplo: retirar células com ponto de interrogação devido a um erro de coleta, dados nulos, arrumar a data para que fosse possível uma manipulação adequada dos dados, além de agrupar as duas tabelas que tínhamos. A Fig. 38 mostra o resultado do *dataframe* com dados de todas as válvulas,

a data de coleta do dado, *tags* das válvulas, tempo médio de acionamento (ms), tempo mínimo de acionamento (ms) daquele dia, tempo máximo (ms), número de acionamentos e número de falhas.

Figura 38 - Dataframe para a predição.

```
In [24]: # Merge dataframes
dataframe = dataframe_abert.merge(df_new, left_on=['Tag', 'Data'], right_on=['Tag', 'Data'], how='left' )
dataframe
```

```
Out[24]:
```

	Data	Tag	tempo_mean	tempo_min	tempo_max	Acionamentos	Falhas
0	2021-03-22	VOC311251	0.000000	0.0	0.0	13.0	0.0
1	2021-03-22	VOC311252	233.333333	100.0	400.0	10.0	0.0
2	2021-03-22	VOC311301	50.000000	0.0	100.0	7.0	0.0
3	2021-03-22	VOC311303	50.000000	0.0	100.0	5.0	0.0
4	2021-03-22	VOC311481	250.000000	200.0	300.0	10.0	0.0
...
599	2021-04-30	VOC311635	200.000000	100.0	300.0	14.0	0.0
600	2021-04-30	VOC311704	250.000000	200.0	300.0	9.0	0.0
601	2021-04-30	VOC311705	300.000000	200.0	400.0	9.0	0.0
602	2021-04-30	VOC311708	150.000000	100.0	200.0	14.0	1.0
603	2021-04-30	VOC311732	57.142857	0.0	100.0	20.0	1.0

604 rows x 7 columns

Posteriormente, foi filtrado as válvulas apenas do mesmo mês para que fosse feita uma predição mensal dos valores. Isso foi necessário para que houvesse apenas *tags* únicas para cada válvula, a média do mês. A Fig. 39 contém uma amostra dos dados das válvulas durante o mês de abril.

Figura 39 - Válvulas do mês de abril agrupadas.

```
df_abril_new = df_abril.groupby(['Tag']).agg({'tempo_mean': 'mean', 'tempo_min': 'min', 'tempo_max': 'max', 'Acionamentos': 'sum',
df_abril_new = df_abril_new.reset_index()
print(df_abril_new)
```

	Tag	tempo_mean	tempo_min	tempo_max	Acionamentos	Falhas
0	VOC311053	0.000000	0.0	0.0	20.0	0.0
1	VOC311201	64.285714	0.0	100.0	78.0	0.0
2	VOC311203	83.333333	0.0	100.0	56.0	0.0
3	VOC311251	59.122807	0.0	200.0	207.0	0.0
4	VOC311252	271.608310	0.0	5800.0	175.0	1.0
5	VOC311301	53.571429	0.0	100.0	85.0	0.0
6	VOC311303	62.564103	0.0	100.0	96.0	1.0
7	VOC311304	100.000000	0.0	600.0	29.0	0.0
8	VOC311401	25.000000	0.0	100.0	148.0	0.0
9	VOC311402	0.000000	0.0	0.0	102.0	0.0
10	VOC311403	40.476190	0.0	100.0	184.0	0.0
11	VOC311441	95.555556	0.0	800.0	60.0	0.0
12	VOC311481	230.864198	100.0	300.0	241.0	0.0
13	VOC311482	1254.934762	100.0	6300.0	491.0	14.0
14	VOC311501	1429.166667	0.0	10600.0	189.0	2.0
15	VOC311502	42088.957100	0.0	2620000.0	2541.0	1.0
16	VOC311503	0.000000	0.0	0.0	49.0	0.0
17	VOC311601	0.000000	0.0	0.0	13.0	0.0

Os valores de tempo de abertura mínimos e máximos foram criados apenas para a análise de valores como outliers e não são necessários para a análise de predição, portanto, foram tirados do *dataframe*, como pode ser observado na Fig. 40.

Figura 40 - Dataframe mensal.

```
df_abril_new.drop(['tempo_min', 'tempo_max'],axis = 1, inplace = True)
df_abril_new
```

	Tag	tempo_mean	Acionamentos	Falhas
0	VOC311053	0.000000	20.0	0.0
1	VOC311201	64.285714	78.0	0.0
2	VOC311203	83.333333	56.0	0.0
3	VOC311251	59.122807	207.0	0.0
4	VOC311252	271.608310	175.0	1.0
5	VOC311301	53.571429	85.0	0.0
6	VOC311303	62.564103	96.0	1.0

Após toda a preparação dos dados, foi iniciada a parte de clusterização das válvulas de acordo com o modelo de K-Means. O modelo fará a clusterização dos dados das válvulas com o objetivo de separá-las em 3 grupos: (1) grupo de válvulas críticas, onde é necessário fazer a manutenção para evitar paradas de linha; (2) válvulas que precisam de atenção, porém não estão críticas; e (3) válvulas que estão boas.

Esse modelo se baseia em similaridades das informações que são fornecidas, como no caso: tempo médio de abertura, número de acionamentos e número de falhas

das válvulas. Dessa forma, para que o modelo tenha uma performance adequada, é necessário fazer a padronização dos dados, como mostrado na Fig. 41.

Figura 41 - Dados padronizados.

```
scale = StandardScaler()
x_scale = scale.fit_transform(x)
x_scale
array([[ -0.25373467, -0.50803986, -0.33674346],
       [ -0.25156576, -0.37058539, -0.33674346],
       [ -0.25092312, -0.42272329, -0.33674346],
       [ -0.25173995, -0.06486768, -0.33674346],
       [ -0.24457098, -0.14070463, -0.24745542],
       [ -0.25192724, -0.35399605, -0.33674346],
       [ -0.25162384, -0.3279271 , -0.24745542],
       [ -0.25036081, -0.48671072, -0.33674346],
       [ -0.25289121, -0.20469206, -0.33674346],
       [ -0.25373467, -0.31370767, -0.33674346],
       [ -0.25236906, -0.11937549, -0.33674346],
       [ -0.25051076, -0.41324367, -0.33674346],
       [ -0.24504563,  0.01570908, -0.33674346]
```

Em seguida, a clusterização dos dados aplicando o modelo K-Means, os resultados dos centroides (*clusters_centers_*) e também dos grupos (*labels_*) das válvulas, como na Fig. 42. Foram utilizados 3 clusters e a inicialização dos pontos dos centroides foi randômica.

Figura 42 - Aplicação do modelo K-Means.

```
# Modelo KMeans
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters= 3, init = 'random')
kmeans.fit(x_scale)

KMeans(init='random', n_clusters=3)

kmeans.cluster_centers_
array([[ -0.20766939, -0.19039618, -0.23881464],
       [ -0.17373237,  0.31668698,  3.94908239],
       [ 3.39260787,  2.6344538 , -0.24745542]])

kmeans.labels_
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1])
```

E a seguir (Fig. 43) pode-se analisar o código para a construção do gráfico de clusterização das válvulas.

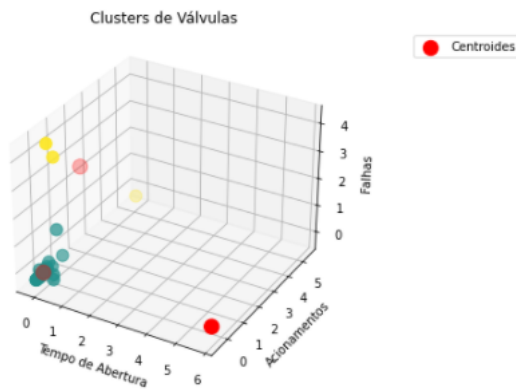
Figura 43 - Válvulas clusterizadas.

```

### 3D
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(x_scale[:, 0], x_scale[:,1],x_scale[:,2], c = kmeans.labels_, s = 100)
ax.scatter(kmeans.cluster_centers[:, 0], kmeans.cluster_centers[:, 1],kmeans.cluster_centers[:, 2], color = 'red', s = 150, la

plt.title('Clusters de Válvulas')
ax.set_xlabel('Tempo de Abertura')
ax.set_ylabel('Acionamentos')
ax.set_zlabel('Falhas')
plt.legend(bbox_to_anchor=(1.2, 1), loc='upper left')
plt.show()

```

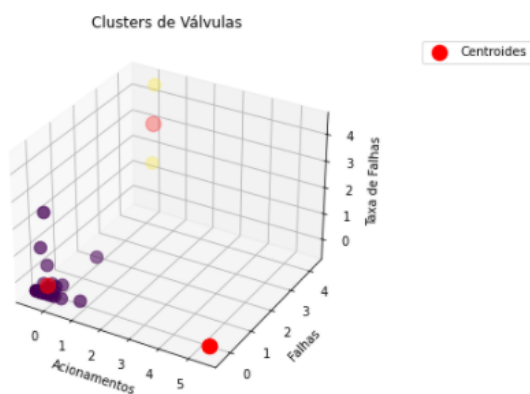


Para uma melhor análise e comparação dos dados de válvulas clusterizadas, decidi fazer o teste do mesmo modelo utilizando apenas número de acionamentos, número de falhas e taxa de falhas das válvulas. O código para a construção do gráfico para mostrar a clusterização pode ser analisado na Fig. 44.

Figura 44 - Válvulas clusterizadas sem tempo de abertura.

```
### 3D
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(x_4_scale[:, 0], x_4_scale[:, 1], x_4_scale[:, 2], c = kmeans_4.labels_, s = 100)
ax.scatter(kmeans_4.cluster_centers[:, 0], kmeans_4.cluster_centers[:, 1], kmeans_4.cluster_centers[:, 2], color = 'red', s = 100)

plt.title('Clusters de Válvulas')
ax.set_xlabel('Acionamentos')
ax.set_ylabel('Falhas')
ax.set_zlabel('Taxa de Falhas')
plt.legend(bbox_to_anchor=(1.2, 1), loc='upper left')
plt.show()
```



Os resultados das análises e discussões vão ser detalhadas no próximo capítulo.

4. RESULTADOS

O sistema foi desenvolvido em conjunto com uma equipe de desenvolvedores e técnicos de automação. A validação de cada etapa do processo foi acompanhada pelos supervisores, gerente da área e também pelos operadores que estavam em contato direto com a ferramenta.

Durante a construção do programa foram entregues dashboards para que os operadores pudessem ir utilizando, gerando críticas e sugerindo melhorias. Como um dos objetivos era melhorar a rotina do operador, a opinião deles foi extremamente importante em cada etapa. O primeiro dashboard foi entregue no mês de janeiro, com 100 válvulas e foi passando por adaptações até ter a sua validação por completo no mês de maio, com 3000 válvulas. Algumas dessas adaptações foram a inserção de abas com análise diária, semanal e mensal da quantidade de acionamentos e falhas e também um histórico das falhas totais de todos os meses, porque eram as informações que eles precisavam.

4.1 Problemas enfrentados

Durante a realização do projeto foram encontrados alguns problemas com a coleta de dados e com o armazenamento no banco de dados, mas que foram solucionados no decorrer do projeto.

Um problema que foi encontrado e que gerou um aprendizado importante foi com relação a utilização do Node-Red e InfluxDB. A empresa já utilizava essas plataformas em uma máquina virtual dentro de um servidor, porém muitas pessoas tinham acesso ao mesmo tempo e também muitos projetos foram feitos em conjunto. Como não havia um controle sobre o acesso, as vezes um usuário fazia umas alterações que geravam erros, e, se não fossem retirados, a máquina virtual que hospedava os programas desligava. Conseqüentemente, não eram feitas as coletas dos dados, que gerava inconfiabilidade no sistema.

Para solucionar esse problema com a coleta de dados foi criado um Node-Red e um InfluxDB em um container no Docker dentro de um servidor local, onde poucas pessoas tinham acesso. Assim, os erros de conexão pararam de acontecer e os dados começaram a ser coletados diariamente, conforme o planejado.

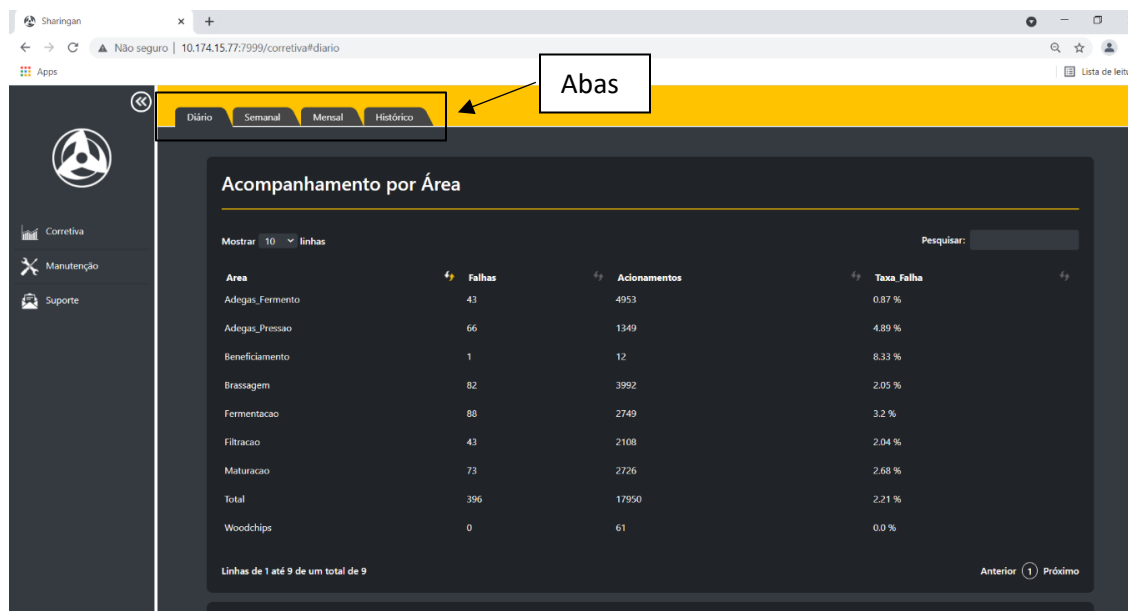
4.2 Aba de Análise Corretiva

Uma das etapas da criação dos dashboards foi a aba chamada corretiva onde é possível verificar os acionamentos, falhas e a taxa de falha das válvulas do processo industrial. A ideia surgiu devido a necessidade dos operadores de acompanharem a quantidade de falhas que estavam ocorrendo nas válvulas para que fosse feito um acompanhamento das áreas mais críticas.

Se as válvulas estão falhando em uma quantidade muito elevada em comparação com o número de vezes em que ela abre, é sinal de que elas precisam ser monitoradas mais de perto e precisam de uma manutenção. Portanto, essa análise foi essencial no início do projeto.

Para essa etapa do projeto não foi utilizado nenhum método de aprendizado de máquina e também nenhuma análise crítica e estatística dos dados, apenas foi feita uma aba informativa dos dados de acionamentos, falhas e taxa de falhas das válvulas. Na Fig. 45 é possível verificar as áreas que estão ocorrendo maiores taxas de falhas e qual a quantidade de acionamentos e de falhas das mesmas. Também é possível ver que os dados estão divididos abas com dados diários, semanais e mensais, além de um histórico com informações de todos os meses.

Figura 45 - Aba Corretiva.



The screenshot shows a web application interface for 'Corretiva'. At the top, there is a navigation bar with tabs for 'Diário', 'Semanal', 'Mensal', and 'Histórico'. A yellow box labeled 'Abas' points to these tabs. Below the navigation bar is a table titled 'Acompanhamento por Área'. The table has columns for 'Área', 'Falhas', 'Acionamentos', and 'Taxa Falha'. The data is as follows:

Área	Falhas	Acionamentos	Taxa Falha
Adegas_Fermento	43	4953	0.87 %
Adegas_Pressao	66	1349	4.89 %
Beneficiamento	1	12	8.33 %
Brassagem	82	3992	2.05 %
Fermentacao	88	2749	3.2 %
Filtracao	43	2108	2.04 %
Maturacao	73	2726	2.68 %
Total	396	17950	2.21 %
Woodchips	0	61	0.0 %

At the bottom of the table, it says 'Linhas de 1 até 9 de um total de 9' and 'Anterior 1 Próximo'.

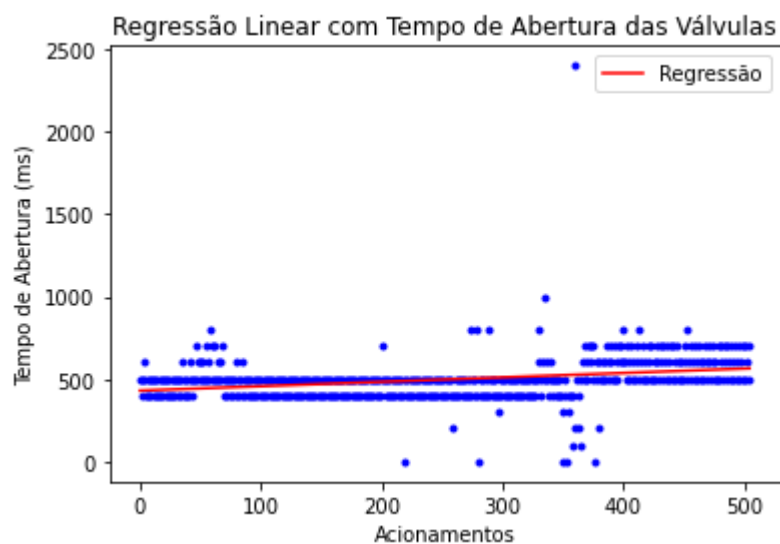
4.3 Resultados dos Modelos de Predição

Conforme discutido anteriormente foram utilizados dois tipos de modelos de aprendizado de máquina, regressão linear e K-Means.

4.3.1 Regressão Linear

No método de regressão linear foi feito a predição das falhas das válvulas a partir do tempo de abertura delas. A Fig. 46 mostra um exemplo dos dados de tempo de acionamento de uma válvula e também a reta da regressão linear construída pelo modelo.

Figura 46 - Predição com Regressão Linear.



Na Fig. 46 é possível perceber que os dados de tempo de abertura da válvula não são lineares e apresentam muitos outliers. A partir do cálculo dos erros, foram obtidos os dados da Tab. 1.

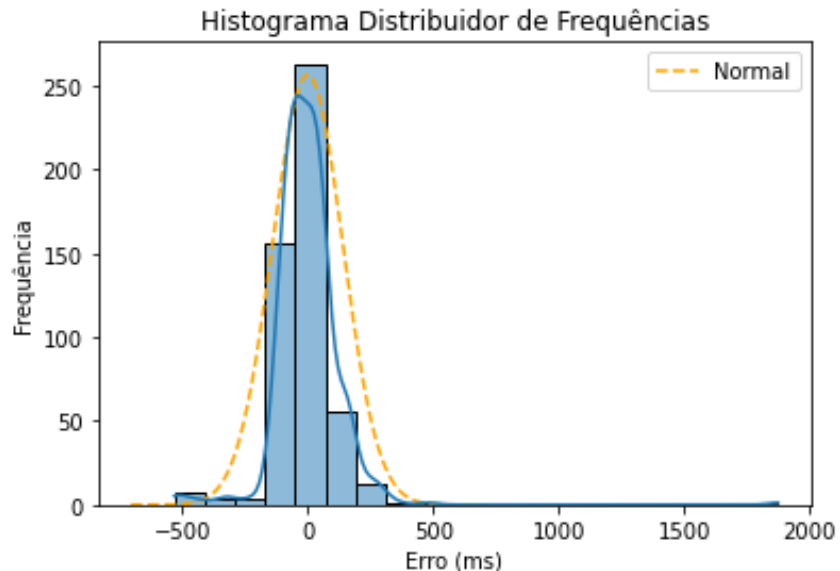
Tabela 1 - Resultados da regressão linear.

Parâmetros	Valores
MAE	83,03 ms
MSE	19448,52 ms
RMSE	139,46 ms
R²	0,072

O R-quadrado é uma medida estatística que indica o quão próximos os dados estão da linha de regressão ajustada pelo modelo de predição. Ele também é conhecido como o coeficiente de determinação e também foi utilizado para a validação, ou não, da regressão linear. Ele varia entre 0 e 1, quanto mais explicativo for o modelo linear, mais próximo de 1 será o R^2 .

O R^2 de 0,072 quer dizer que o modelo linear explica apenas 7,2% da variância das variáveis, o que indica que o modelo não explica a variabilidade dos dados. Para a análise do erro foi feito um histograma distribuidor de frequências do erro (Fig. 47).

Figura 47 - Histograma Distribuidor de Frequências do Erro.



No gráfico da Fig. 47 os erros são representados pelas barras azuis, a curva de distribuição dos dados está em azul e a curva de distribuição normal está representada em laranja. Podendo assim, comparar e afirmar que a curva de distribuição do erro é equivalente a uma curva de distribuição normal. Apesar dessa afirmação, erros de 500 milissegundos são praticamente o tempo de abertura da válvula, considerando que os tempos de abertura são em torno de 400 e 500 milissegundos (Fig. 46), o que é considerável inviável para o sistema.

Com essas considerações, o modelo de predição utilizando a regressão linear não foi validado e assim descartado como modelo de predição de falhas em válvulas.

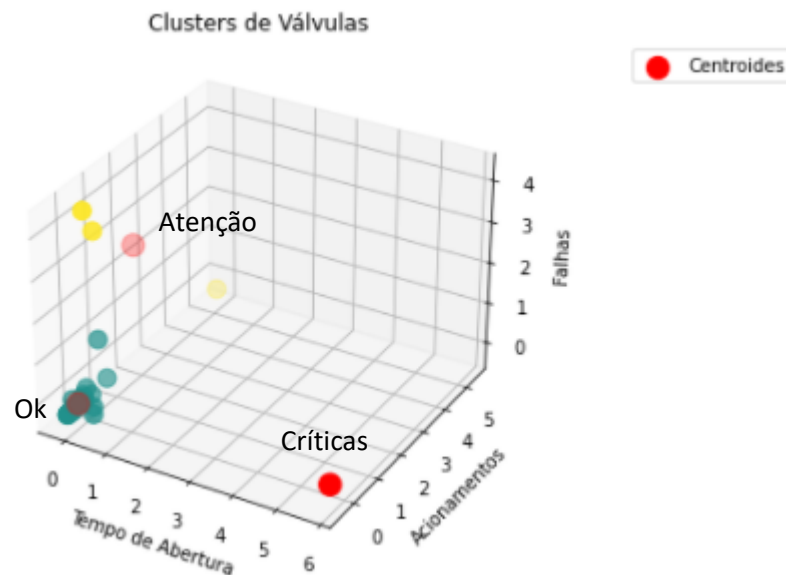
Devido a não validação da regressão linear como método de predição de falhas em válvulas, foi necessário desenvolver outro algoritmo de aprendizado de máquina.

4.3.2 K-Means

O algoritmo de aprendizado de máquina K-Means é capaz de agrupar clusters/grupos de válvulas considerando suas similaridades, como foi explicado no capítulo 2. Sendo assim, utilizou atributos como número de acionamentos, número de falhas e tempo de abertura das válvulas para clusterizá-las de acordo com a sua criticidade. Como foi utilizado as próprias falhas para fazer essa clusterização, o modelo tem a finalidade de prever se as válvulas irão falhar mais ou chegar a quebrar.

Em um primeiro momento foi feita uma clusterização das válvulas utilizando tempo de abertura, número de acionamentos e número de falhas. Na Fig. 48 é possível observar a divisão dos grupos de válvulas.

Figura 48 - Cluster 1.



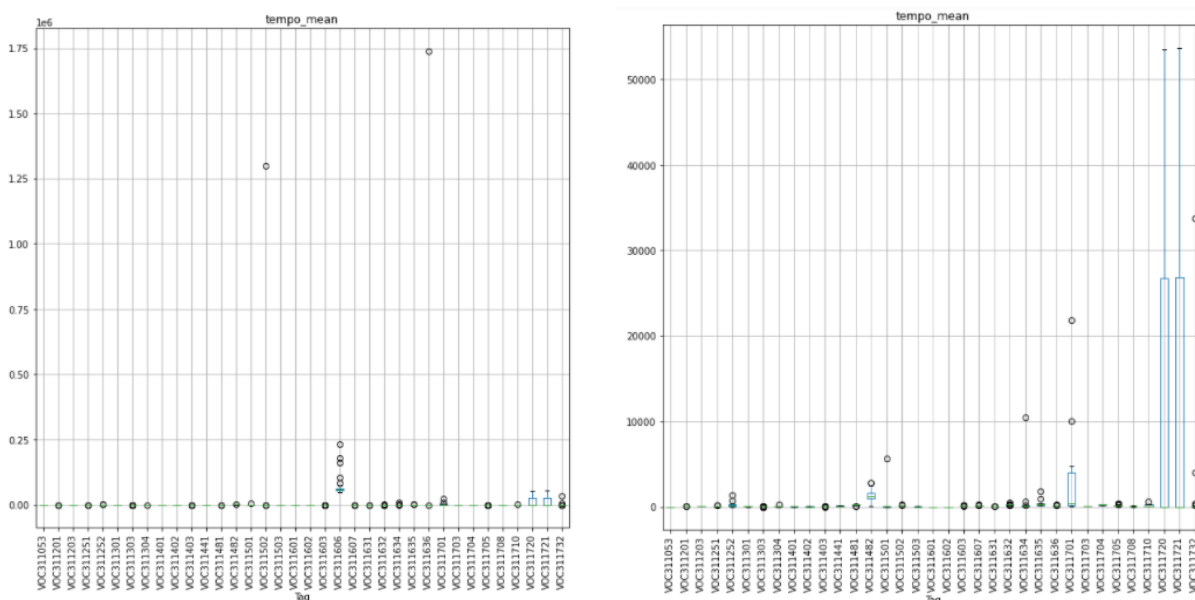
As válvulas em azul foram consideradas “ok” por terem um baixo número de falhas, de acionamentos e de tempo de abertura. As válvulas em amarelo foram consideradas “atenção” por apresentarem um número de falhas grande, porém com tempo de abertura ainda baixo. E por fim, a válvula em roxo (na imagem não aparece por ser apenas uma)

foi considerada “crítica”, por apresentar um elevado tempo de abertura. Lembrando que os círculos em vermelho são os centroides dos clusters.

A partir de uma análise mais crítica das divisões das válvulas foi possível perceber que os grupos de válvulas “ok” e “atenção” estão relativamente próximos, comparados com o outro grupo. Essa observação pode dar margem para divisões errôneas quanto a clusterização dos grupos.

Considerando apenas o tempo de abertura das válvulas, foi construído um gráfico boxplot (Fig. 49) antes de fazer qualquer tratamento nos dados e depois da retirada de alguns outliers.

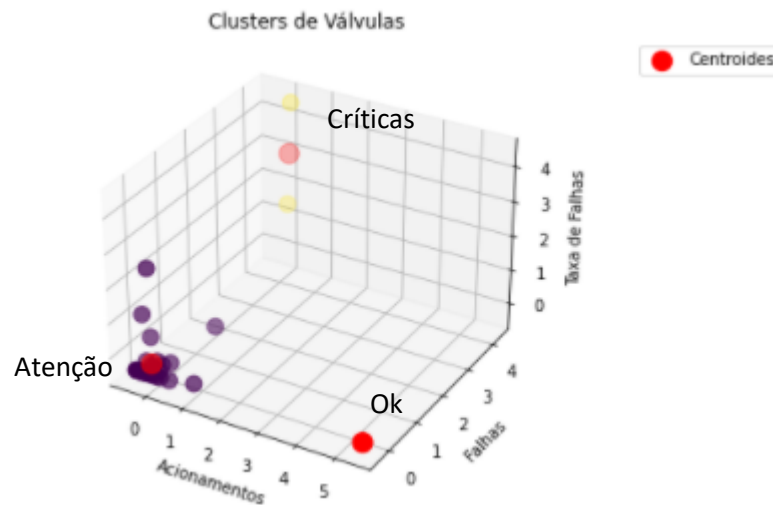
Figura 49 - Boxplot com tempo de abertura.



Mesmo com a retirada de uns outliers muito fora do contexto, os tempos de abertura das válvulas não tinham uma similaridade e eram muito diferentes, o que poderiam influenciar na clusterização das válvulas. Por esse motivo foram feitos mais alguns testes com diferentes combinações de variáveis para o modelo K-Means.

Assim, outro modelo obtido foi utilizando o número de acionamentos, número de falhas e taxa de falhas das válvulas. O resultado pode ser observado na Fig. 50.

Figura 50 - Cluster 2.



Na Fig. 50 a classificação já foi um pouco diferente. As válvulas em roxo tiveram poucos acionamentos e poucas falhas, mas a taxa de falha quando tem poucos acionamentos é maior se comparada com aquela que tem muitos acionamentos, portanto, foram consideradas válvulas de “atenção”. As válvulas em amarelo falham muito e são consideradas “críticas”. E a válvula em azul (na mesma posição do centroide) foi considerada “ok” por ter uma taxa de falha menor e muitos acionamentos.

Retirando o tempo de abertura do *dataframe*, o modelo se performou melhor em relação ao que continha esse tempo e conseguiu clusterizar melhor as válvulas. Esse tipo de modelo de aprendizado de máquina não supervisionado não tem uma métrica para medição de erros ou acurácia do sistema, a validação ou não do modelo é feita a partir da clusterização visual dos grupos. Dessa forma, o modelo de clusterização K-Means utilizando taxa de falha foi o mais adequado para a clusterização das válvulas pela sua criticidade e previsão de quebra delas.

5. CONCLUSÃO

Esse trabalho propôs a criação de um sistema utilizando inteligência artificial para a manutenção preditiva de válvulas em um processo industrial, para prever falhas, evitar paradas de linha, reduzir custos de manutenção e melhorar a rotina do operador. Essa plataforma foi implantada na empresa e com a utilização desse sistema, os operadores sentiram uma melhora na rotina, pois já conseguiam dar um foco maior nas válvulas que eram consideradas mais críticas e que já mostravam na tela.

Os objetivos específicos do trabalho foram cumpridos. A lógica de programação foi criada no CLP, a coleta de dados foi feita pela Node-Red e armazenadas no InfluxDB. Foi desenvolvido uma plataforma de visualização e análise dos dados. Além de um algoritmo de predição para as válvulas, utilizando aprendizado de máquina.

A análise dos custos de manutenção precisa de mais tempo para conseguir ter resultados mais sólidos, porque as manutenções nas válvulas são feitas, em média, com 3 anos de funcionamento. Como o projeto foi colocado em prática há sete meses, ainda não foi possível obter resultados de redução de custos com manutenção.

Com relação a predição das falhas em válvulas não foi possível chegar a um modelo que conseguisse prever quando a válvula iria falhar, devido às limitações de conexões e memória do CLP para adquirir novas variáveis e também devido a não validação do modelo de regressão linear. Porém foi possível chegar a um modelo de predição de quebra e piora das válvulas também utilizando inteligência artificial, mais precisamente, aprendizado de máquina.

Ainda seguirá em desenvolvimento um outro modelo utilizando inteligência artificial que aprimorará o sistema e que consiga prever as falhas em válvulas antes que elas aconteçam, para que o reparo seja feito de forma preditiva e evite quaisquer problemas em relação a parada do processo industrial em questão.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, C. F. M. **Válvulas de Controle** – notas de aula, Universidade de São Paulo, 2018. Disponível em: < <https://edisciplinas.usp.br/mod/resource/>. Acesso em 07 de junho de 2021.

ALMEIDA, P. S. **Manutenção mecânica industrial: conceitos básicos e tecnologia aplicada**. São Paulo: Editora Érica, 2014.

AMARAL, A. N. **Critérios para Análise de viabilidade para manutenção preditiva em válvulas de controle em indústrias de processo**. 2013. 78p. Monografia (Especialização em Automação Industrial) - Programa de Pós-Graduação do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR-5462: **Confiabilidade e Manutenibilidade**. Rio de Janeiro. 1994.

BARROS, B. A. **A Importância da Manutenção Industrial Como Ferramenta Estratégica de Competitividade**. Faculdade Redentor, Miracema – Rio de Janeiro. Disponível em:< http://redentor.inf.br/files/brenoalvimbarros-artigo_16092016111003.pdf >. Acesso em 17 de fevereiro de 2021.

BUKSH, Z. A. et al, **Predictive maintenance using tree-based classification techniques: A case of railway switches**, *Transportation Research Part C: Emerging Technologies*, Volume 101, 2019.

BUKSH, Z. A.; SAEED, A.; STIPANOVICA, I. **A machine learning approach for maintenance prediction of railway assets**. Proceedings of 7th Transport Research Arena TRA 2018, April 16-19, 2018, Vienna, Austria

CAMPOS, F. H. L. F. de; FILHO, O. R. **Aplicação de Manutenção Preditiva em um sistema de descarregamento de caminhões e vagões tanque**. 2013. 93f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial),

Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

CAMPOS, M.C.M.M; TEIXEIRA, H. C. G. – **Controles Típicos de equipamentos e processos industriais**, Editora Edgard Blucher, 2008.

CINAR, Z. M. et al. **Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0**. 2020.

CYMATICLABS. **InfluxDB Studio**, c2021. Disponível em: <<https://github.com/CymaticLabs/InfluxDBStudio>. Acesso em 02 de junho de 2021>.

FILHO, M. **As Métricas Mais Populares para Avaliar Modelos de Machine Learning** [s.d]. Disponível em: < <https://www.mariofilho.com/as-metricas-mais-populares-para-avaliar-modelos-de-machine-learning/> >. Acesso em 21 de abril de 2021.

FLASK. **Flask web development, one drop at a time**, c2010. Disponível em: <https://flask.palletsprojects.com/en/2.0.x/>. Acesso em: 06 de março de 2021.

FRANCHI, C.; CAMARGO, V. **Controladores Lógicos Programáveis: sistemas discretos**. 2021. Editora Érica - São Paulo.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow**. 2017. Published by O'Reilly Media, Inc.,1005 Gravenstein Highway North, Sebastopol, CA 95472.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. 2019. Published by O'Reilly Media, Inc.,1005 Gravenstein Highway North, Sebastopol, CA 95472.

IBM. **Preparação de dados** [s.d]. Disponível em: < <https://www.ibm.com/br-pt/analytics/data-preparation#:~:text=A%20prepara%C3%A7%C3%A3o%20de%20dados%20%C3%A9,o organiza%C3%A7%C3%A3o%20e%20coleta%20de%20dados> >. Acesso em 07 de abril de 2021.

ILEOH. Introdução A Clusterização E Os Diferentes Métodos. **Portal Data Science**. 2018. Disponível em: <<https://portaldatascience.com/introducao-a-clusterizacao-e-os-diferentes-metodos/>>. Acesso em: 20 de maio de 2021.

JAKRA A. Husain. " **Modelling of Industrial Machine Structure for Predictive Maintenance Using IoT Sensor Data and Machine Learning: A Review.**" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 14.2 (2019): 16-21.

KALATHAS, I.; PAPOUTSIDAKIS, M. **Predictive Maintenance Using Machine Learning and Data Mining: A Pioneer Method Implemented to Greek Railways.** Designs 2021

KUMAR, S. S. – **Fieldbus and Networking in Process Automation**, CRC Press, 2014.

MMTEC, **Manutenção Preditiva e Industria 4.0**, Artigo técnico, MMTec Inspeções Industriais, Disponível em: <<http://www.mmtec.com.br/manutencao-preditiva-e-industria-4-0/>>, Acesso em: 07 junho de 2021.

MORAIS, Bruna Corrêa. **Sistema integrado de análise e previsão de indicadores de desempenho de um processo industrial.** 2019. 90 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) - Universidade Federal de Uberlândia, Uberlândia, 2020.

NAKAJIMA, S. **Introdução ao TPM - Total Productive Maintenance.** São Paulo: IMC Internacional Sistemas Educativos Ltda., 1989. PEREIRA, Décio Martins. **Manutenção Industrial.** São Paulo: Fundação de Apoio à Escola Técnica Centro de Ensino Técnico e Profissionalizante Quintino – E.T.E.R. – Mecânica, 2004.

PRUDENTE, F. - **Automação industrial: PLC, teoria e aplicações: curso básico.** Editora LTC, 2007.

SANTANA, F. **Entenda o Algoritmo K-means e Saiba como Aplicar essa Técnica. 2017. Minerando Dados.** Disponível em: <<https://minerandodados.com.br/entenda-o-algoritmo-k-means/>>. Acesso em 26 de abril de 2021.

SMITH, C. A; CORRIPIO, A. B. – **Princípios e Práticas de Controle Automático de Processos**, Editora LTC 2008.

TURBAN, E. et al. **Business intelligence: um enfoque gerencial para a inteligência do negócio**. Porto Alegre: Bookman, 2009.

ZANCAN, M. D. **Controladores programáveis** / Marcos Daniel Zancan. – 3. ed. – Santa Maria: Universidade Federal de Santa Maria: Colégio Técnico Industrial de Santa Maria, 2011