



**UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”**

Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

VITOR INFORZATO GUERMANDI

**INTELIGÊNCIA DE NEGÓCIO APLICADO À ANÁLISE DE PROCESSOS
DA INDÚSTRIA 4.0 ATRAVÉS DO SOFTWARE POWER BI**

Sorocaba

2022

VITOR INFORZATO GUERMANDI

**INTELIGÊNCIA DE NEGÓCIO APLICADO À ANÁLISE DE PROCESSOS DA
INDÚSTRIA 4.0 ATRAVÉS DO SOFTWARE POWER BI**

Trabalho de Graduação apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador: Me. Ricardo Pasquati
Pontarolli
Coorientador: Prof. Dr. Eduardo
Paciência Godoy

Sorocaba

2022

G932i Guermandi, Vitor Inforzato
Inteligência de negócio aplicado à análise de processos da indústria 4.0 através do software Power BI / Vitor Inforzato Guermandi. -- Sorocaba, 2022
78 p.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Ricardo Pasquati Pontarolli
Coorientador: Eduardo Paciência Godoy

1. Inteligência de negócio. 2. Power BI. 3. Framework. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

INTELIGÊNCIA DE NEGÓCIO APLICADO À ANÁLISE DE PROCESSOS DA
INDÚSTRIA 4.0 ATRAVÉS DO SOFTWARE POWER BI

VITOR INFORZATO GUERMANDI

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO
COMO PARTE DO REQUISITO PARA OBTENÇÃO DO GRAU DE
BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Prof. Dr. Maurício Becerra Vargas
Coordenador

BANCA EXAMINADORA:

Prof. Me. Ricardo Pasquati Pontarolli

Prof. Dr. Eduardo Verri Liberado

Prof. Dr. Everson Martins

Julho de 2022

AGRADECIMENTOS

Aos meus familiares por sempre apoiarem e me incentivarem a estudar dando todo o suporte necessário para isso. Especialmente aos meus pais, Argemiro e Deily, que sempre batalharam e não mediram esforços para que esse momento fosse possível. Sem vocês, nada disso teria acontecido. Às minhas duas irmãs, Júlia e Isabela, por sempre me ajudarem e me auxiliarem em todos os momentos de minha vida, desde toda a minha juventude até o término da graduação, sem vocês esse caminho teria sido muito mais difícil.

À Carol, minha namorada, que me acompanha há mais de quatro anos nessa trajetória. Agradeço por toda ajuda, conselho, trocas de experiência e compreensão em todos os momentos vividos.

Ao Prof. Dr. Eduardo Paciência Godoy pela oportunidade de trabalho desde o início de minha iniciação científica, à sua coordenação, dicas, conselhos e ajudas prestadas. Às suas aulas que me despertaram interesse em buscar uma iniciação científica, à sua orientação durante a iniciação científica e coorientação durante o Projeto Final de Curso (PFC). Especialmente à sua paciência e compreensão em todos os momentos, imagino que não tenha sido fácil.

Ao Prof. Me. Ricardo Pasquati Pontarolli por toda ajuda e suporte prestados desde o primeiro contato, se mostrando sempre muito prestativo e com vontade em ajudar sempre que possível. À orientação durante o PFC, à troca de experiência e aos conselhos dados que me fizeram pensar e refletir em muitos momentos deste trabalho. Com seu conhecimento e força de vontade, você ainda vai mais longe.

A todos os professores do curso de Engenharia de Controle e Automação da UNESP/Sorocaba, de substitutos a adjuntos, que tive contato ao longo da minha trajetória acadêmica. Desde aqueles do início do curso, do ciclo básico, no qual sem esse tipo de conhecimento nada seria possível de ser criado e compreendido, até aqueles com conhecimento voltado para as áreas de aplicação, pois sem eles nenhum tipo de conhecimento seria possível de ser praticado. Ambos foram igualmente fundamentais para a minha formação.

A todos os outros funcionários da UNESP: aos servidores técnicos e administrativos da seção de graduação, à assistente social do campus, aos funcionários terceirizados conhecidos como o “pessoal da limpeza”, à portaria e todos os outros. Vocês foram importantes nessa minha trajetória para chegar onde estou hoje.

À UNESP de Sorocaba por fornecer um ensino e educação de excelentes qualidades, desde à infraestrutura das salas de aula, aos laboratórios, biblioteca aos recursos humanos.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo nº 2018/19984-4, pelo apoio à realização dessa pesquisa.

Às minhas segundas famílias que encontrei aqui em Sorocaba: todos os amigos e colegas que morei durante esse período. Tanto ao pessoal da república *Space*, que morei no primeiro ano de minha graduação, quanto ao pessoal da república *Magnatas*, que morei o restante do tempo. Com todo suporte emocional, espírito de grupo, cooperação, ajuda em matérias da graduação, vocês me fizeram crescer e compreender ainda mais o sentido da vida.

GUERMANDI, V.I. **Inteligência de negócio aplicado à análise de processos da indústria 4.0 através do software PowerBI**. 2022. 78 f. Trabalho de Graduação (Bacharel em Engenharia de Controle e Automação) - Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista, Sorocaba, 2022.

RESUMO

O atual contexto global exige buscas constantes por maior eficiência, redução de custos e aumento da produtividade em curto período. Estes fatores propulsionaram o surgimento da Indústria 4.0, que tem como objetivo o aumento global de receitas e a melhoria da qualidade de vida da população através de tecnologias interconectadas. Um dos principais pilares da Indústria 4.0 é a tomada de decisão baseada em dados, no qual é possível analisar processos industriais constantemente a fim de obter indicadores que refletem o desempenho da área de negócio. Uma das principais tecnologias que facilita a modularidade e escalabilidade das soluções industriais é a arquitetura de comunicação orientada a serviços, no qual cada serviço desempenha sua funcionalidade de maneira específica e desacoplada dos demais serviços. Pensando nisso, esse trabalho propõe a utilização da arquitetura orientada a microsserviços (MOA) a fim de levantar indicadores que avaliam os processos industriais. Para isso, utiliza-se uma planta piloto que simula um contexto industrial no qual são coletados os dados relacionados aos parâmetros do controlador e aos valores dos sensores e atuadores. Destaca-se que o *framework* utilizado para executar a aplicação é o *Molecular* e o banco de dados onde as informações são salvas é o *MySQL*. O software LabVIEW é responsável por realizar o controle da malha desejada (pressão de linha, pressão do tanque, nível e vazão). E o *Power BI* é utilizado como ferramenta de inteligência de negócio a fim de mensurar o desempenho dos ciclos executados. Este trabalho avaliou parâmetros estatísticos para analisar as etapas estacionárias e indicadores relacionados à integral do erro a fim de mensurar todas as etapas do processo. Três contextos industriais diferentes foram simulados, sendo um processo sem distúrbios e outros dois com distúrbios em etapas distintas. Por fim, os KPIs analisados foram exibidos em um *dashboard* e publicado no serviço do *Power BI*. O trabalho mostrou-se eficiente na caracterização dos processos com e sem distúrbios através dos KPIs estudados. Tal fato pode ser observado na 2ª aba do *dashboard*, onde o 2º processo, que apresentou 100% para os índices IAE, ISE, ITAE e ITSE teve o melhor resultado, seguido pelo 3º e 1º processo. Quanto menor for a porcentagem desses índices, pior é o desempenho do processo.

Palavras-chave: Indústria 4.0; MOA; inteligência de negócio; KPIs.

GUERMANDI, V.I. **Business intelligence applied to industry 4.0 process analysis using PowerBI software.** 2022. 78 f. Undergraduate final work (Bachelor in Control and Automation Engineering) – Institute of Science and Technology of Sorocaba, São Paulo State University, Sorocaba, 2022.

ABSTRACT

The current global context requires constant search for greater efficiency, cost reduction and increased productivity in a short period. These factors propelled the emergence of Industry 4.0, which aims to increase global revenues and improve the population's quality of life through interconnected technologies. One of the main pillars of Industry 4.0 is data-based decision making, in which it is possible to constantly analyze industrial processes in order to obtain indicators that reflect the performance of the business area. One of the main technologies that facilitates the modularity and scalability of industrial solutions is the service-oriented communication architecture, in which each service performs its functionality in a specific way and is decoupled from other services. With that in mind, this work proposes the use of microservices-oriented architecture (MOA) in order to raise indicators that evaluate industrial processes. For this, a pilot plant is used that simulates an industrial context in which data related to controller parameters and values of sensors and actuators are collected. It is noteworthy that the framework used to run the application is Moleculer and the database where the information is saved is MySQL. LabVIEW software is responsible for performing the desired loop control (line pressure, tank pressure, level and flow). And Power BI is used as a business intelligence tool to measure the performance of the cycles performed. This work evaluated statistical parameters to analyze the stationary steps and indicators related to the error integral in order to measure all the steps of the process. Three different industrial contexts were simulated, one process without disturbances and the other two with disturbances in different stages. Finally, the analyzed KPIs were displayed on a dashboard and published to the Power BI service. The work proved to be efficient in the characterization of processes with and without disturbances through the KPIs studied. This fact can be observed in the 2nd tab of the dashboard, where the 2nd process, which presented 100% for the IAE, ISE, ITAE and ITSE indexes had the best result, followed by the 3rd and 1st process. The lower the percentage of these indexes, the worse the performance of the process.

Keywords: Industry 4.0; MOA; business intelligence; KPIs.

LISTA DE FIGURAS

Figura 1 - Arquitetura orientada a microsserviços.	23
Figura 2 - Arquitetura orientada a microsserviços da planta piloto.	25
Figura 3 - Planta piloto orientada a serviços.	28
Figura 4 - Estrutura das interconexões da planta piloto.	29
Figura 5 - Placa Raspberry Pi 3 Modelo B +.	30
Figura 6 - Valor da pressão de linha desejado ao longo do processo.	34
Figura 7 - Gráfico <i>boxplot</i> .	39
Figura 8 - Cálculo da integral pela aproximação da regra do trapézio.	41
Figura 9 - Fluxograma com os dados do processo.	44
Figura 10 - Definição das bibliotecas e das variáveis globais.	47
Figura 11 - Definição do serviço e da tabela do banco de dados.	48
Figura 12 - Ação do serviço <i>pressures</i> para escrever os valores no banco de dados.	49
Figura 13 - Chamadas executadas em loop a cada 500 ms.	50
Figura 14 - Credenciais para acessar o banco de dados.	52
Figura 15 - Query em SQL realizada no <i>Power BI</i> .	53
Figura 16 - Colunas calculadas no <i>Power Query</i> .	54
Figura 17 - Relacionamento dos dados através do modelo star schema.	56
Figura 18 - Colunas calculadas na linguagem DAX para os KPIs IAE, ISE, ITAE e ITSE.	57
Figura 19 - Medidas calculadas na linguagem DAX para a variabilidade.	58
Figura 20 - Medidas calculadas na linguagem DAX para os índices de capacidade.	59
Figura 21 - Medidas de agregação de colunas pré-programadas e disponíveis no <i>Power BI</i> .	60
Figura 22 - Campos para serem preenchidos do gráfico <i>Box and Whisker</i> by <i>MAQ Software</i>	61
Figura 23 - Medidas calculadas para o 1º processo.	64
Figura 24 - Medidas calculadas para o 2º processo.	65
Figura 25 - Medidas calculadas para o 3º processo.	65
Figura 26 - Análise comparativa entre os três processos.	66
Figura 27 - Análise comparativa das etapas estacionárias de cada processo.	68

LISTA DE TABELAS

Tabela I - Definição de um processo e caracterização das etapas estudadas.	34
Tabela II - KPIs escolhidos para avaliar o desempenho dos processos.	36

LISTA DE ABREVIATURAS E SIGLAS

"	Unidade de comprimento em polegadas
API	<i>Application Programming Interface</i>
bar	Unidade de pressão
BI	<i>Business Intelligence</i>
CP	<i>Process capability index</i>
CPK	<i>Process capability index</i>
DAQ	<i>Data Acquisition</i>
DAX	<i>Data Analysis Expressions</i>
DB	<i>Data Base</i>
ETL	<i>Extract, Transformation and Load</i>
GASI	Grupo de Automação e Sistemas Integráveis
IAE	<i>Integrated Absolute Error</i>
ISE	<i>Integrated Squared Error</i>
ITAE	<i>Integrated Time Absolute Error</i>
ITSE	<i>Integrated Time Squared Error</i>
KPI	<i>Key Performance Indicator</i>
LSE	Limite superior
LSI	Limite inferior
mA	Miliampere
MOA	<i>Microservice Oriented Architecture</i>
NATS	<i>Native Messaging System</i>
mmH ₂ O	Milímetro de coluna d'água
ms	Milissegundos
PID	<i>Proportional, Integral and Derivative</i>

s Segundos

SQL *Standard Query Language*

SUMÁRIO

1. INTRODUÇÃO	15
1.1 OBJETIVO	18
1.2 ESTRUTURA DO TRABALHO	19
2. REVISÃO CONCEITUAL	20
2.1 BUSINESS INTELLIGENCE	20
2.2 POWER BI	21
2.3 FRAMEWORK MOLECULER	22
3. MATERIAIS E MÉTODOS	25
3.1 PROPOSTA DO TRABALHO	25
3.2 MATERIAIS	26
3.2.1 ESTRUTURA FÍSICA	27
3.2.2 HARDWARE	30
3.2.3 SOFTWARE	31
3.3 MÉTODO	33
3.3.1 DELINEAMENTO EXPERIMENTAL	33
3.3.2 INDICADORES	35
4. DESENVOLVIMENTO	43
4.1 FLUXOGRAMA DOS DADOS E VISÃO GERAL DO DESENVOLVIMENTO DA PROPOSTA	43
4.2 CRIAÇÃO DO BANCO DE DADOS	46
4.3 DESENVOLVIMENTO DA QUERY	50
4.4 PROCESSO DE ETL NO POWER QUERY	53
4.5 MODELO DE RELACIONAMENTO DOS DADOS	55
4.6 CÁLCULO DOS KPIs NA LINGUAGEM DAX	57
4.7 ELABORAÇÃO DO DASHBOARD	60
5. RESULTADOS E DISCUSSÃO	63
5.1 RESULTADOS	63
5.1.1 VISÃO GERAL DOS PROCESSOS COM E SEM DISTÚRBIOS ANALISADOS	63

5.1.2 ANÁLISE COMPARATIVA DOS PROCESSOS	66
5.1.3 ANÁLISE COMPARATIVA DAS ETAPAS ESTACIONÁRIAS	67
5.2 DISCUSSÃO	69
6. CONCLUSÃO	71
REFERÊNCIAS	73

1 INTRODUÇÃO

A busca por maior eficiência, redução de custos e aumento da produção em um curto período, atrelado a um contexto global com elevado crescimento demográfico e recursos escassos que exige maior rendimento energético, propiciou o surgimento da Indústria 4.0. Esta tem o propósito de aumentar o nível global de receitas e melhorar a qualidade de vida da população mundial através de uma tecnologia interconectada (SCHWAB, 2016).

A Indústria 4.0, ou Quarta Revolução Industrial, que incorpora tecnologias das revoluções anteriores (mecanização, eletricidade, automação e tecnologia da informação), caracteriza-se por um conjunto de técnicas que permitem a fusão dos mundos físico, digital e biológico (DIAS, 2018). Dessa forma, ela acrescenta a implementação de novas tecnologias voltadas à Internet das Coisas e Serviços no ambiente de manufatura, elevando a autonomia de processos operacionais (com a incorporação da Robótica Autônoma/Cooperativa, Internet das Coisas, Realidade Aumentada/Virtual e Manufatura Aditiva) e promove a interconectividade, desde a cadeia de produção aos níveis de gestão (com a utilização da Computação em Nuvem e *Big Data*).

No modelo da indústria 4.0 destacam-se os seguintes princípios dos sistemas de produção inteligente: i. capacidade de operação em tempo real (com a aquisição e tratamento de dados quase instantâneo, permitindo tomada de decisão em tempo real); ii. Virtualização através de cópia virtual das fábricas inteligentes (permitindo monitoramento remoto de todos os processos); iii. Descentralização nas tomadas de decisão em que o sistema cyber-físico poderá atuar de acordo com as necessidades da produção em tempo real (máquinas não apenas receberão comandos, mas poderão fornecer informações sobre seu ciclo de trabalho); iv. Modularidade que oferece maior flexibilidade nas linhas de produção, atuando de acordo com a demanda (acoplando/desacoplando módulos de produção) e v. orientação a serviços (com a maior utilização de arquiteturas orientadas a serviços) (KAGERMANN; WAHLSTER; HELBIG, 2013; SILVEIRA, 2018).

A análise de dados é um dos principais pilares da revolução industrial 4.0, como é frequentemente evidenciado em setores empresariais. Isto porque tais setores produzem e processam dados em ritmos acelerados, o que produz resultados sem

necessariamente ter o tempo para interpretá-los na mesma velocidade das análises (EIDIZADEH; SALEHZADEH; ESFAHANI, 2017).

Atualmente, as empresas enfrentam cenários cada vez mais competitivos dentro do mundo corporativo. A tomada de decisões assertivas ainda é uma grande dificuldade para o setor empresarial, já que as organizações necessitam de informações claras, seguras e eficientes para elaborar resoluções estratégicas de problemas de negócios (SHARMA; MITHAS; KANKANHALLI, 2014).

Por conta disso, saber tomar decisões de maneira inteligente é uma solução em potencial para a sobrevivência de novos negócios (AHMAD, 2015). Nesse contexto, profissionais *data driven* que saibam utilizar ferramentas de *Business Intelligence* (BI) são cada vez mais necessários dentro do ambiente organizacional (MISAL, 2019; ZHOU et al., 2021). Isto pode ser observado tanto em setores com um elevado número de interferências, como na saúde (BASILE et al., 2022), como para setores industriais, com mecanismos comparativamente mais controlados, como a indústria de processos (TAO et al., 2021).

A indústria de processos transforma matérias primas de baixo valor agregado em produtos como bens de consumo e produtos intermediários (i.e, com maior valor agregado) através de processos físico-químicos (SANTOS et al., 2014). Ela engloba os setores industriais de química e petroquímica, papel e celulose, fármacos e outros tipos de indústria que possuem diversos equipamentos e dispositivos em cada etapa de transformação da matéria prima. Elementos atuadores, sensores, transmissores, controladores entre outros estão envolvidos, caracterizando assim a indústria de processos (KLEMES, 2011).

Como se trata de um fenômeno complexo, qualquer falha, perturbação ou erro ao longo do processo pode ser crítico para a segurança da operação. Isto porque existem inúmeros equipamentos interconectados em diferentes níveis hierárquicos e funções distintas com alto valor de implementação e manutenção (GARCIA, 2021). Além disso, pode gerar grandes prejuízos financeiros por manutenções ou falhas. Dessa forma, torna-se necessário a gestão dos ativos a fim de evitar perdas, garantir a segurança da operação, maximizar a produção e reduzir desperdícios (TAVARES, 2018).

Um exemplo é a indústria de papel e celulose, em que mensurar, monitorar e analisar constantemente a temperatura da água nas caldeiras, os níveis nos reservatórios, a pressão nos reservatórios e pressão de linha torna-se extremamente relevante ao

processo de fabricação. Tanto para garantir a segurança da operação quanto para elevar a produção e reduzir perdas (AHLÉN et al., 2019).

Pensando nisso, inúmeros *softwares* surgiram com o intuito de monitorar e analisar os processos industriais. Num contexto global com crescente demanda de produtos e serviços, aumento na automatização dos processos, aproximação das tecnologias da informação com tecnologias de automação, a avaliação constante de parâmetros que mensuram o desempenho industrial mostra-se cada vez mais necessário e possível de serem realizados (JASPERNEITE; SAUTER; WOLLSCHLAEGE, 2020).

Através de indicadores é possível diagnosticar problemas antes mesmo que eles ocorram, além de identificar pontos de melhoria ao longo do processo (SCHUMACHER; SIHN, 2020). Portanto, é plausível supor que mensurar constantemente parâmetros de avaliação dos processos industriais podem evitar paradas indesejadas nas indústrias. Isto é muito custoso, pois interrompe a produção e gera gastos com manutenção. Sendo assim, pressupõe-se que com avaliações constantes é possível programar as paradas para manutenção, além de realizar a manutenção preditiva a partir de possíveis falhas indicadas nos equipamentos.

Outra mudança do setor industrial com grande impacto na indústria de processos está relacionada com a arquitetura de comunicação entre os diversos dispositivos. Atualmente, os elementos dos mais baixos níveis hierárquicos na clássica arquitetura de comunicação possuem capacidade de se comunicar com níveis hierárquicos superiores (KANNISTO et al, 2022). Nesta pirâmide, uma informação de um sensor ou atuador do chão de fábrica poderia se comunicar diretamente com sistemas de informação, por exemplo.

Dessa forma, o aumento e a busca por uma arquitetura menos engessada e mais escalável foi possível graças às soluções comercializadas como serviço, e assim houve um crescimento exponencial daqueles serviços e microserviços que realizam funcionalidades de formas separadas. A MOA surge como uma alternativa ao tradicional padrão arquitetural monolítico, em que a saída de um serviço é usada como uma entrada para outro em uma orquestração de serviços independentes. Como é uma arquitetura agnóstica a dispositivos e plataformas, os microserviços fornecem maior flexibilidade para o desenvolvimento de soluções, além do recurso de possibilitar um banco de dados individual (BIGHETI et al., 2018).

O aumento no número de dados armazenados pelos microsserviços estimulou o interesse pela necessidade em ferramentas de análise. Neste contexto de Indústria 4.0, as tecnologias que possibilitaram a interconexão de dispositivos e a quantidade de dados e de trocas de informações cresceram exponencialmente (ASAITHAMBI; VENKATRAMAN; VENKATRAMAN, 2020)

Inúmeros *softwares* de análises surgiram a fim de aumentar os ganhos e a produtividade em diversos setores industriais, que agora são capazes de transformar dados em informações refletindo o desempenho das organizações (ESPINEL-VELASCO, 2022). Um dos *softwares* mais utilizado nos últimos anos que ganhou grande destaque na área de análise é o *Power BI*, ferramenta de *self-service BI* da *Microsoft*.

De acordo com o quadrante mágico da *Gartner*, que publica constantemente provedores de tecnologia para as mais diversas aplicações, o *Power BI* é a ferramenta de BI da *Microsoft* na posição de líder disparada no mercado. Portanto, também se apresenta como uma ferramenta promissora para análises de processos. O *software* possui grande semelhança aos demais produtos da companhia pela sua interface amigável e pela fácil navegabilidade, justificando a posição de liderança (SMART CONSULTING, 2022).

Através do *Power BI* é possível realizar todo o processo de ETL (*Extract, Transformation and Load* - extração, transformação e carregamento) dos dados, sendo realizado por meio do *Power Query*. Além disso, o *software* da *Microsoft* possibilita criar e gerenciar o relacionamento dos dados (BECKER; GOULD, 2019). Por fim, vale destacar que as medidas e os índices podem ser calculados através da linguagem DAX (*Data Analysis Expressions* - expressões de análise de dados) e o *dashboard* depois de criado pode ser publicado na *web* a fim de ser acessado remotamente.

1.1 OBJETIVO

Dessa forma, neste trabalho objetiva-se levantar KPIs (*Key Performance Indicator* - indicador chave de desempenho) capazes de mensurar o desempenho de processos industriais. Para isso, utiliza-se a MOA, no qual dispositivos do “chão de fábrica” (controlador, sensores e atuadores) transmitem informações até o *software* de análise capaz de identificar falhas, distúrbios e pontos de melhoria ao longo do processo.

1.2 ESTRUTURA DO TRABALHO

O trabalho de conclusão de graduação contém 6 capítulos e o resumo. O Capítulo 1 aborda uma introdução ao tema, apresenta os objetivos definidos e a estrutura de como o trabalho está documentado. No Capítulo 2, foi apresentado o levantamento bibliográfico em que abordou-se conceitos e revisões de literatura sobre o tema, além de justificar a escolha utilizada de algumas tecnologias.

O Capítulo 3 contém a proposta do trabalho, apresentando a solução através de esquemas. Nele, o leitor também encontra os materiais utilizados como o *hardware*, a explicação da estrutura física da planta piloto e os *softwares* para a construção da solução proposta. Por fim, foi apresentado o delineamento experimental e a metodologia, no qual é caracterizado o objeto de estudo e os KPIs propostos.

No Capítulo 4, há a documentação do desenvolvimento da proposta de maneira detalhada. O leitor encontrará a estruturação da solução, os dados selecionados, a criação do banco de dados, a construção da *query* para filtrar os dados e o processo de limpeza, o modelo de relacionamento entre as tabelas e o desenvolvimento do *dashboard* para a análise dos dados.

O capítulo 5 destina-se aos resultados e discussão. Portanto, os resultados do trabalho obtidos através da metodologia explicada são apresentados. Ainda, existem tabelas e gráficos para visualização dos resultados encontrados. Na discussão, o trabalho aborda as limitações e potencialidades encontradas com a pesquisa e aponta caminhos para trabalhos futuros.

O Capítulo 6 é o último deste trabalho. Nele, o leitor encontrará a conclusão da pesquisa desenvolvida a partir dos objetivos estabelecidos. Por fim, estão as referências bibliográficas utilizadas para o embasamento teórico do trabalho.

2 REVISÃO CONCEITUAL

Para início do desenvolvimento do trabalho, a compreensão do tema abordado e conhecimento sobre as principais ferramentas, *softwares* e conceitos utilizados, neste capítulo será explicado em detalhe o termo *business intelligence*, a ferramenta de análise *Power BI* e o *framework Molecular*.

2.1 BUSINESS INTELLIGENCE

Durante a década de 1980, o termo “*Business Intelligence*” surgiu como uma nova área, voltada para as ações *data driven* em análises de negócios (PRIMAK, 2008, p.02). Isto é, a tomada de decisões de negócios é baseada na interpretação do contexto organizacional e na análise dos dados. Em termos práticos, pode-se afirmar que se trata de uma gestão empresarial orientada por dados.

De acordo com Misal (2019), ferramentas de BI ajudam no crescimento dos negócios e são substanciais para novos *insights* em decisões eficientes baseadas em dados (principalmente ao lidar com metadados e um aporte grande de KPIs) dentro do mundo corporativo. As ferramentas estão presentes em todas as etapas de análise: no planejamento da coleta, na sua organização, na busca da melhor análise e durante o compartilhamento e monitoração de dados que ficam contidos em *Data Warehouse/Data Mart* (PRIMAK, 2008, p.02).

Por conta disso, a capacitação de recursos humanos em BI e o desenvolvimento de novas ferramentas analíticas relacionadas aumentaram exponencialmente nas últimas décadas (AHMAD, 2015). As ferramentas de BI permitem mais do que o armazenamento de dados: elas proporcionam a recuperação, modelagem e análise de metadados e informações. O principal objetivo é a criação de sistemas de informação computacionais, disponíveis para gerentes, executivos e outras lideranças importantes do negócio (MATHEUS; PEREIRA, 2004, p.04). A partir da interpretação dos seus resultados, permitem novas percepções de clientes ou avaliações de desempenho dentro de organizações (MIRONYANNIDIS; THEODOULIDIS, 2010).

Assim, além de ferramentas em potencial para a tomada inteligente de decisões nos negócios, é necessário treinamentos para a formação de profissionais *data driven* dentro do contexto organizacional (CHOU, 2020). Por conta deste e dos outros motivos

supracitados, ferramentas interativas, acessíveis e de fácil navegabilidade possuem potencial inovador para o desenvolvimento em BI, como é o caso do *software Power BI*, que ganha cada vez mais adeptos por ser um diferencial (ALON, 2020).

2.2 POWER BI

O *software Power BI* desenvolvido pela *Microsoft* se destacou no mercado como uma das principais soluções para o desenvolvimento de projetos de BI. Este software de análise de dados pode ser utilizado com várias finalidades, inclusive para gerar relatórios, e possui 3 opções de planos: gratuito, *Pro* ou *Premium*.

Durante o momento de tomada de decisões de forma estratégica, ter acesso aos seus dados compilados e realizar análises em um *dashboard* pode ser uma ferramenta útil, por exemplo. Além disso, o *Power BI* tem um ambiente simples, de fácil acessibilidade em sistemas corporativos e acessível aos desenvolvedores, tanto para resoluções de cenários complexos como para a modelagem dos dados. Todas essas características foram fortemente corroboradas com a ampla utilização do software que, em menos de 5 anos, dominou o mercado (SÁ et al, 2020, p.22).

Outra grande vantagem que faz com que o *software* da *Microsoft* ocupe a posição de liderança entre seus concorrentes é a sua qualidade de tratamento e otimização com conjuntos de dados robustos e extensos (LAGO; ALVEZ, 2018). Por conta disso, se tratando de plataformas de análises com BI, a *Microsoft* ganha grande destaque.

O surgimento do *Power BI* foi desenvolvido a partir de suplementos pensados para o *Excel* (LAGO; ALVEZ, 2018). Com a implementação dos suplementos *Power Query Editor*, *Power View* e *Power Pivot*, as análises de dados ganharam outro nível de qualidade. Outra forma de organizá-los é pela classificação em três ambientes: *Power Query Editor*, Modelo e Relatório (SÁ et al, 2020).

Quando cria-se um novo projeto, a fase de padronização é feita no ambiente *Power Query Editor*. Então é nele que a importação e o tratamento dos dados é feita, além de integrar outros relatórios e outros sistemas também. A linguagem de programação deste ambiente é a linguagem M (SÁ et al, 2020).

O próximo ambiente em que a etapa de modelagem acontece é o no ambiente Modelo. Trata-se da criação e gerenciamento do relacionamento entre as diferentes

tabelas existentes. Comparando com o *Excel*, seria a ferramenta “*Power Pivot*” com linguagem DAX (SÁ et al, 2020).

A terceira e última fase diz respeito à visualização dos dados. Ela acontece no ambiente Relatório e integra o *Power View* (SÁ et al, 2020). Aqui, monta-se a aparência visual do *dashboard*. Para isto, o *Power BI* oferece uma gama de opções e de formas de visualizar os dados, como matrizes, mapas, tabelas, gráficos, etc. Nesta etapa também acontece a integração com outros softwares de programação e de análise de dados, como *R Project* e o *Python*, além de outras fontes de dados.

2.3 FRAMEWORK MOLECULER

De acordo com Bigheti, Fernandes e Godoy, pode-se definir *framework* como “uma estrutura base ou uma plataforma de desenvolvimento que contém ferramentas, bibliotecas, sistemas e componentes que agilizam o processo de desenvolvimento de uma solução específica” (BIGHETI; FERNANDES; GODOY, 2019). O *framework Moleculer* é uma estrutura de código aberto (*open source*) utilizada em aplicações orientadas a microsserviços que possui a linguagem *JavaScript* para desenvolvimento (MOLECULER, 2020).

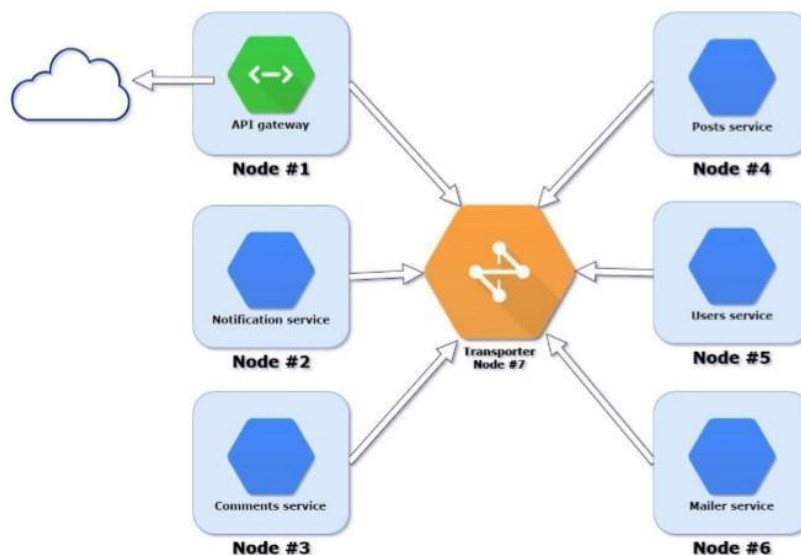
Na MOA do *framework* (Figura 1), a latência da comunicação entre os serviços não é insignificante, mas permite que serviços sejam replicados a fim de proporcionar resiliência e tolerância a falhas. Uma das principais vantagens na arquitetura do *framework Moleculer* é não haver prioridade ou hierarquia entre os microsserviços e existir um recurso de descoberta e registro automático.

Isto cria a possibilidade de outros microsserviços serem informados após a criação de um novo serviço ou a disponibilização de uma nova funcionalidade. Outro recurso importante é o balanceamento automático de carga, com a função de distribuir dinamicamente a carga da comunicação de maneira uniforme entre os microsserviços (BIGHETI; FERNANDES; GODOY, 2019).

Os quadrados de cor azul claro (Figura 1) representam os nós (*Node*) de gerenciamento do *framework Moleculer* para o desenvolvimento, descoberta e configuração dos microsserviços. Cada nó pode conter um ou mais serviços, que são responsáveis pelo gerenciamento e armazenamento de seus dados com seu próprio banco de dados. Os hexágonos azuis representam os próprios microsserviços, que

podem executar diferentes tarefas (ação). No caso, para a aplicação proposta, os hexágonos azuis são os dados coletados dos sensores da bancada de processos.

Figura 1 - Arquitetura orientada a microsserviços.



Fonte: (MOLECULER, 2020)

Já o hexágono verde (Figura 1) representa um (micro) serviço de *Gateway* (API *Gateway*) com a finalidade de interface de conexão dos microsserviços internos com aplicações externas através de chamadas de rede ou da arquitetura REST. “Um *gateway* de API é uma maneira de desacoplar a interface do cliente da sua implementação de *back-end*. Quando um cliente faz uma solicitação, o *gateway* de API a divide em várias solicitações, as direciona para os locais adequados, produz uma resposta e faz o monitoramento” (RED HAT, 2020).

O hexágono laranja (Figura 1) indica um serviço de mensagens (*Transporter*) responsável pela comunicação entre os microsserviços, sendo um dos componentes mais importantes na arquitetura devido a sua capacidade de enfileirar e manter as mensagens até que elas sejam processadas pelos consumidores. Isto permite que uma mensagem seja enviada para o *Transporter* sem a necessidade de um mecanismo de descoberta de serviços.

Uma das principais vantagens de tal serviço no *framework Moleculer* é que, caso um mesmo serviço seja disponibilizado em múltiplas instâncias em diferentes nós (maior disponibilidade), as requisições recebidas pelo *Transporter* serão

automaticamente balanceadas entre os nós que estão *online* naquele momento e que podem executar o serviço requisitado.

Ademais, outro grande benefício dessa abordagem é o desacoplamento entre produtores e consumidores de eventos e dados, uma vez que promove a simplificação do desenvolvimento e aumenta a disponibilidade em comparação ao uso de transações distribuídas. O serviço *Transporter* é uma técnica que busca solucionar a comunicação entre sistemas completamente diferentes de uma maneira confiável. O *Broker* de mensagem torna possível integrar tais sistemas para que eles possam trocar dados de forma desacoplada.

O protocolo de comunicação NATS foi utilizado com o *Transporter*. Este protocolo possui inúmeras aplicações para IIoT, além de fornecer um método leve de transmitir mensagens através do modelo *publish/subscribe*. Nele, o *Broker* funciona como um servidor que recebe as mensagens publicadas pelos clientes e distribui para os assinantes, de acordo com os tópicos de interesse. Tal tipo de comunicação demonstra-se muito eficaz, já que possibilitou o desacoplamento direto entre quem publica a mensagem e quem assina para receber o conteúdo.

Além disso, o NATS permite que os programas compartilhem código comum de manipulação de mensagens, isolem recursos e interdependências e escalem ao lidar facilmente com um aumento no volume de mensagens, sejam elas solicitações de serviço ou dados de fluxo. (NATS, 2022).

Para o presente trabalho utilizou-se o *framework Molecular* com a função de gerenciar e coordenar o envio e recebimento de mensagens dos microsserviços através do *Transporter*, um *broker* mensageiro, conforme representado na arquitetura MOA do *framework* (Figura 1).

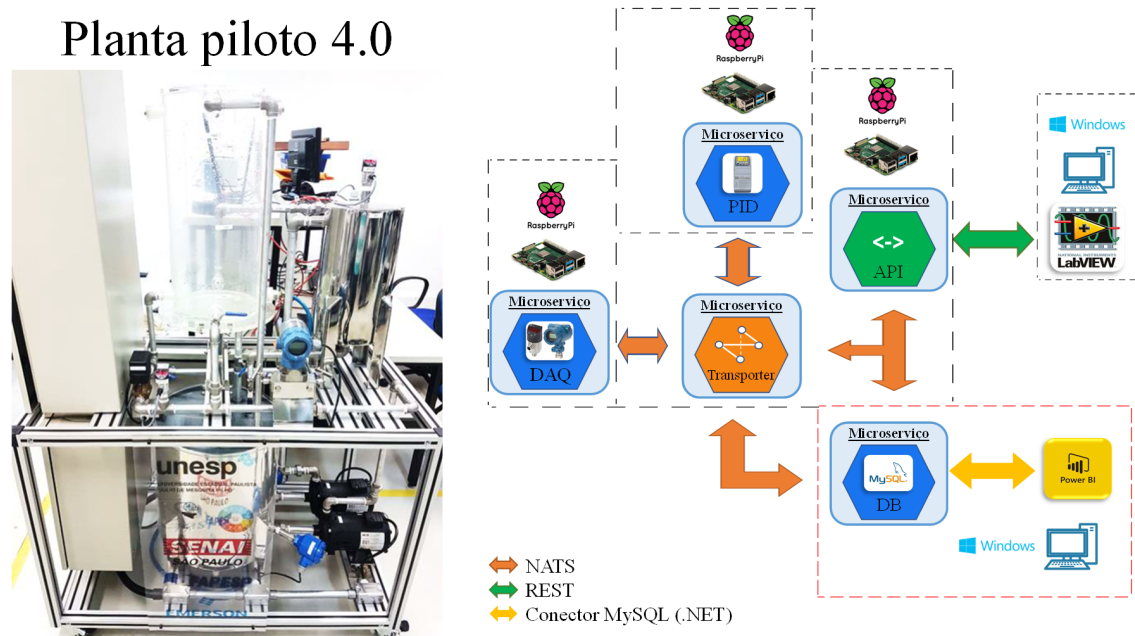
3 MATERIAIS E MÉTODOS

Nessa seção será abordado a proposta do trabalho, a explicação da planta piloto e os materiais que a compõem, os métodos utilizados como a caracterização de um ciclo e o delineamento experimental para realizar os cálculos e encontrar os KPIs.

3.1 PROPOSTA DO TRABALHO

Este trabalho se propõe a analisar processos industriais através de uma arquitetura orientada a microserviços, conforme mostrado na Figura 2. Para isso, utiliza-se o *software Power BI* como ferramenta de inteligência de negócio para obter relatórios a respeito dos ciclos executados, o *software LabVIEW* para executar o controle dos processos industriais e o banco de dados *MySQL* para armazenar os dados coletados. Destaca-se que a linha tracejada em vermelho (microserviço DB e a aplicação com o *Power BI*) foi totalmente desenvolvida neste trabalho.

Figura 2 - Arquitetura orientada a microserviços da planta piloto.



Fonte: Autoria própria.

É possível notar que os serviços são alocados em diferentes *hardwares*: o serviço DAQ está em uma *Raspberry Pi*, o serviço *Transporter* e API estão em outra

Raspberry Pi e uma terceira placa contém o serviço PID; dois computadores com sistema operacional do *Windows* são utilizados: em um está o serviço DB e o *Power BI* e em outro está o *LabVIEW*.

Dois objetivos principais destacam-se na arquitetura deste trabalho: a execução do controle de um processo e o armazenamento de dados através de uma comunicação e orquestração entre os serviços. De acordo com a Figura 2, a comunicação entre os diferentes serviços para realizar o controle ocorre da seguinte forma: 1) a aplicação externa (software *LabVIEW*) requisita o serviço API através de uma chamada REST; 2) este, por sua vez, aciona o *Transporter* que irá requisitar os dados do serviço DAQ; 3) o serviço DAQ comunica-se com os sensores através da conexão com o pino I2C da placa *Raspberry Pi*.

Após a coleta dos dados necessários, o processo contrário ocorre: 1) o serviço DAQ envia uma resposta ao *Transporter*; 2) este transmite a informação ao serviço *API Gateway*; 3) por fim, o software *LabVIEW* recebe uma resposta através da comunicação com a aplicação externa. Para realizar o controle, o software *LabVIEW* realiza processo análogo ao descrito anteriormente, acionando o serviço PID que irá adequar os valores dos atuadores de acordo com o algoritmo de PID implementado. Logo após, o serviço PID aciona o *Transporter*, que transmite as informações ao serviço DAQ. Este último irá enviar o ajuste necessário aos atuadores através da comunicação I2C.

De maneira independente, o serviço DB (Figura 2) armazena os dados do processo a cada 500 ms em um processo análogo ao descrito anteriormente. Ou seja, o serviço DB aciona o *Transporter* que por sua vez requisita os dados do serviço DAQ. Após o retorno dos dados dos dispositivos de interesse, o serviço DAQ retorna uma resposta ao serviço mensageiro que enviará as informações para serem salvas no banco de dados. Posteriormente os dados serão acessados pelo *software Power BI* através do conector do *software MySQL .NET*).

3.2 MATERIAIS

Nesta subseção será explicado a planta piloto e seus materiais utilizados, sua estrutura física, além do *hardware* e do *software*.

3.2.1 PLANTA PILOTO 4.0

A planta piloto 4.0 utilizada é uma bancada de processos e está localizada no Laboratório do Grupo de Automação e Sistemas Integráveis (GASI) do Instituto de Ciência e Tecnologia de Sorocaba (UNESP). Ela foi desenvolvida através de uma parceria entre a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), a UNESP/Sorocaba, o Serviço Nacional de Aprendizagem Industrial (SENAI) de Lençóis Paulista-SP e a empresa *Emerson Automation Solutions* (EUA). A sua operação é baseada totalmente no uso de serviços, o que constitui seu diferencial em relação a outras plantas industriais.

Para melhor compreensão e entendimento da Planta Piloto, decidiu-se explicar separadamente a estrutura física (que será detalhada nesta subseção) e a parte elétrica e os *softwares* que serão explicados mais adiante. Na parte física é possível compreender as interconexões dos tanques e válvulas, localização dos sensores e atuadores. A estrutura elétrica foca no entendimento do microprocessador utilizado e suas principais características. A parte de *softwares* aborda tanto a criação do banco de dados em *JavaScript* como a parte de análise na ferramenta *Power BI*.

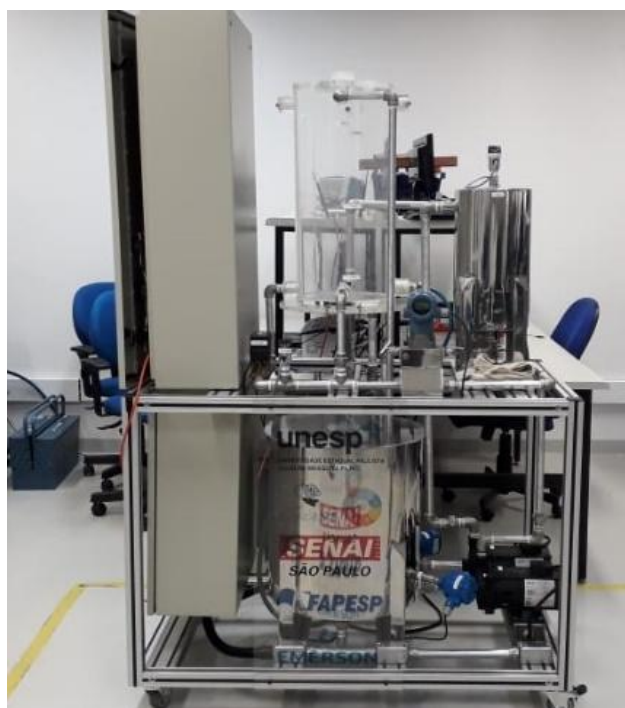
A planta piloto (Figura 3 e Figura 4), é composta em sua estrutura física por um tanque principal (TQ02) em aço inox com capacidade de 83 litros localizado na parte inferior e dois tanques na parte superior, sendo um de acrílico (TQ01) com capacidade de 38 litros e outro de inox para o reservatório de pressão (R01) de 15 litros. O líquido pode ser bombeado do reservatório inferior para qualquer um dos reservatórios superiores através de tubos de inox por duas rotas distintas: uma via para o reservatório de inox e outra via para o reservatório de acrílico, utilizando a bomba P1 e a bomba P2, respectivamente.

Na parte frontal da bancada de processos, dois painéis elétricos são utilizados para a organização das ligações e conexões entre dispositivos. O painel inferior é utilizado para circuitos de comando e potência enquanto o painel superior é utilizado para alocação das interfaces de conexões de entradas e saídas e placas programáveis. Além disso, o painel superior possui um *display LCD touch* de 10", no qual é possível monitorar e supervisionar o processo em tempo real.

A ideia principal da planta piloto é o controle de processos e para isto é disposta de sensores de nível do tanque (LIT125) e vazão (FIT116) com o transmissor de pressão coplanar de 0 a 5000 mmH₂O, dos sensores de pressão de linha (PI118) e

pressão do reservatório (PIT129) com o pressostato eletrônico de 0 a 2,5 bar e do sensor de temperatura (TIT127). Como dispositivos atuadores, a planta possui duas bombas de P1 e P2, controladas por inversores de frequência *WEG*. Outrossim, uma chave de nível (LSL110) é utilizada para evitar que as bombas não funcionem sem água no reservatório e a válvula proporcional motorizada (LV122) é de $\frac{1}{2}$ ", assim como os demais tubos de conexão (PONTAROLLI et al., 2021).

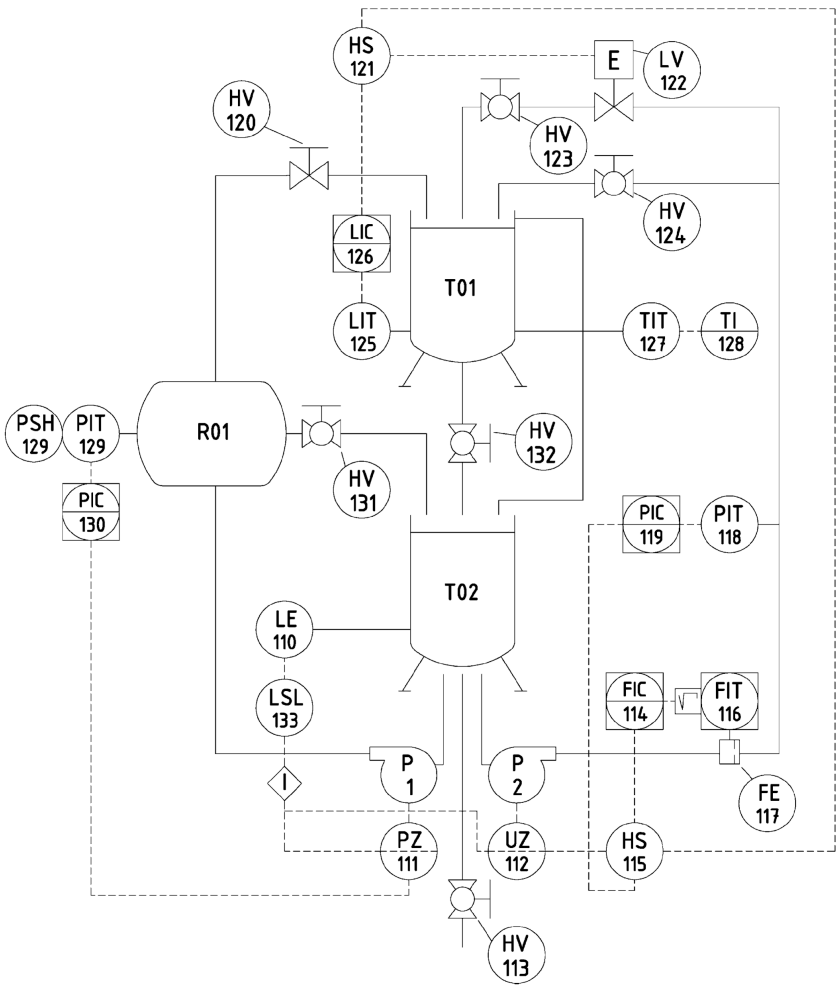
Figura 3 - Planta piloto orientada a serviços.



Fonte: Autoria própria.

É possível realizar diferentes processos de acordo com a necessidade de cada aplicação. Quatro malhas de controle podem ser utilizadas para o controle das diferentes variáveis: 1) nível (LIT125); 2) vazão (FIT116); 3) pressão de linha (PIT118) e 4) pressão do reservatório (PIT129). A fim de realizar a tarefa mencionada, foi utilizado o software *LabVIEW*, o qual possui um microserviço PID focado na execução do controle da variável desejada.

Figura 4 - Estrutura das interconexões da planta piloto



Instrumentos discretos				Geral	
Montado no campo	Montado no campo	Montado no painel de controle principal			
HV 120	Válvula Manual	LIT 125	Transmissor indicador de nível	TI 128	Indicator de temperatura
HV 123	Válvula Manual	PIT 118	Transmissor indicador de pressão	Montado atrás do painel de controle principal	
HV 124	Válvula Manual	PIT 129	Transmissor indicador de pressão	PZ 111	Pressão com controle final elemento não classificado
HS 121	Chave manual	TIT 127	Transmissor indicador de pressão	UZ 112	Multivariável com controle final elemento não classificado
HS 115	Chave manual	PSH 129	Chave de pressão alta	Display compartilhado montado no campo	
HV 113	Válvula manual esférica	LSL 133	Chave de nível baixo	FIT 116	Transmissor indicador de vazão
HV 131	Válvula manual esférica	LE 110	Elemento de nível	Display compartilhado montado no painel de controle principal	
HV 132	Válvula manual esférica	FE 117	Elemento de vazão	FIC 114	Controlador indicador de vazão
LV 122	Válvula de nível Operada eletricamente			LIC 126	Controlador indicador de nível
				PIC 119	Controlador indicador de pressão
				PIC 130	Controlador indicador de pressão
					Sinal Hidráulico
					Sinal elétrico
					Intertravamento
					Tanque
					Reservatório
					Moto bomba

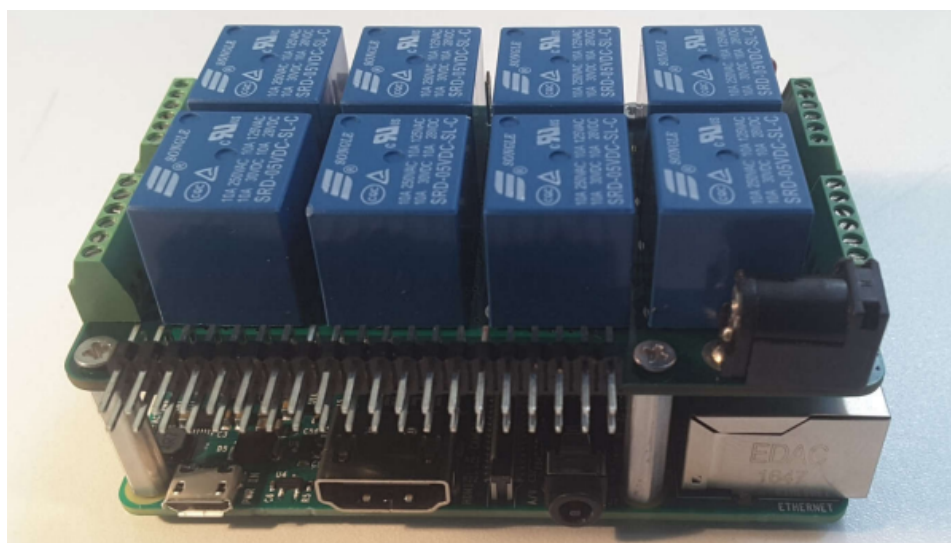
Fonte: (PONTAROLLI et al, 2021)

3.2.2 HARDWARE

O *hardware* utilizado para a alocação e orquestração dos serviços foram três *Raspberry Pi 3B+* (Figura 5). Esta placa possui suporte para redes *wireless* no padrão AC, proporcionando mais velocidade para a conexão e melhorando a performance da placa. Além disso, o embarcado possui um processador *Broadcom BCM2837B0 64bits ARM Cortex-A53 Quad-Core* com as seguintes características (RASPBERYPi, 2020):

- Clock 1.4 GHz;
- Memória RAM: 1GB;
- Adaptador *Wi-Fi* 802.11 b/g/n/AC 2.4GHz e 5GHz integrado;
- *Bluetooth* 4.2 BLE integrado;
- Conector de vídeo HDMI;
- 4 portas USB 2.0;
- Conector Gigabit Ethernet over USB 2.0;
- Interface para câmera;
- Interface para *display*;
- *Slot* para cartão *here*;
- Conector de áudio e vídeo;
- GPIO de 40 pinos;
- Certificado de homologação Anatel: 01598-18-10629;
- Dimensões: 85 mm x 56 mm x 17mm.

Figura 5 - Placa *Raspberry Pi 3* Modelo B+ com o *shield MegaIO industrial*



Fonte: *Sequent Microsystems* (2022).

Um *shield* de expansão (*MegaIO industrial*) foi utilizado com a placa *Raspberry Pi* de maneira a expandir os *slots* de entrada e saída. Isto possibilitou a coleta de dados dos sensores através da medição da corrente com um range de 4 a 20 mA, assim como acionar as bombas (P1 e P2) através de um sinal de 0 a 10V enviado para os inversores. A placa *Raspberry Pi* em conjunto com a placa de expansão *MegaIO industrial* foi onde desenvolveu-se o serviço DAQ, pois através dele é possível fazer a leitura dos sensores (entrada) e atualizar o valor dos atuadores (saídas).

3.2.3 SOFTWARE

Nesta parte será explicado os *softwares* necessários para realização do trabalho. Será abordado o ambiente de desenvolvimento para criação dos serviços, as bibliotecas e linguagem de programação utilizada, o *software* para realizar o controle dos processos, o banco de dados e a ferramenta de análise.

Visual Studio Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela *Microsoft* para *Windows*, *Linux* e *macOS*. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código (CODE, 2022). Inicialmente ele só era possível de ser utilizado no sistema operacional do *Windows*, porém com o passar dos anos estendeu-se aos demais sistemas. Esse editor foi utilizado para o desenvolvimento dos *scripts* dos microsserviços.

Node.js e NPM

O Node.js é um *software* de código aberto, multiplataforma e que permite a execução de *scripts* em *JavaScript* fora de um navegador *web*. Além disso, sua alta capacidade de escalabilidade, arquitetura, flexibilidade e baixo custo o tornam uma boa escolha para a implementação de microsserviços. (LENON, 2018). Outra grande vantagem desse ambiente de desenvolvimento é o gerenciador de pacotes de biblioteca disponível, o NPM (*Node Package Manager*), em que é possível a criação de módulos,

executar instruções ou conjunto de instruções através de um comando criado pelo usuário conforme a necessidade (NPM, 2022)

LabVIEW

O *LabVIEW* é um ambiente de programação gráfica usado para desenvolver sistemas automatizados de pesquisa, validação e teste de produção. Ele destaca-se por ter milhares de funções de análise disponíveis, possuir elementos de exibição interativos e configuráveis, ter *drivers* para automatizar cada instrumento e *hardware* de aquisição de dados e por possuir conectividade com outros idiomas e protocolos padrão. Ele foi utilizado para controlar a pressão de linha do processo (NATIONAL INSTRUMENTS, 2022).

MySQL

MySQL Workbench, sistema gerenciador de banco de dados do *MySQL*, é uma plataforma unificada de desenvolvimento e administração visual que inclui ferramentas para modelagem e design de banco de dados, desenvolvimento e teste de consulta, configuração de servidor e monitoramento, administração de usuários e segurança, automação de *backup* e recuperação, inspeção de dados de auditoria e migrações de banco de dados orientadas por assistente. Além disso, o *MySQL* está disponível para os sistemas operacionais *Windows*, *Linux* e *Mac OS X* (*MySQL*, 2022). Ele foi utilizado para armazenar os dados dos processos.

Power BI

O *Power BI* é uma coleção de serviços de *software*, aplicativos e conectores que trabalham juntos para transformar as fontes de dados em informações coerentes, visualmente envolventes e interativas. Os dados podem estar em uma planilha do *Excel*, em banco de dados relacionais ou não relacionais, em uma coleção de *data warehouses* híbridos locais ou baseados na nuvem. Com o *Power BI*, é possível se conectar facilmente a fontes de dados, visualizar e descobrir conteúdos importantes e compartilhá-los com todas as pessoas desejadas. (HART et al., 2022).

Ele possui uma versão gratuita e duas versões pagas, sendo as versões pagas o *Power BI Pro* e o *Power BI Premium*. O *Power BI Premium* é uma licença baseada em capacidade, enquanto o *Power BI Pro* e o PPU (*Premium* por Usuário) são licenças

baseadas em usuário. O *Power BI Pro* é voltado para usuários que publicam relatórios, compartilham *dashboards*, colaboram com colegas em *workspaces* e participam de outras atividades relacionadas. Uma licença PPU (*Premium* por Usuário) inclui toda a funcionalidade de um usuário Pro, além de capacidades adicionais como inúmeros serviços disponíveis na nuvem. (SHARABI, 2022). Para esse trabalho a versão gratuita foi escolhida pois atende as necessidades desejadas.

3.3 MÉTODO

Nesta subseção será abordado o delineamento experimental, através da caracterização do comportamento matemático de um ciclo e, posteriormente, serão exibidos todos os KPIs escolhidos para avaliar o desempenho dos processos. Também será explicado de forma detalhada como os cálculos foram realizados e o propósito de tais indicadores.

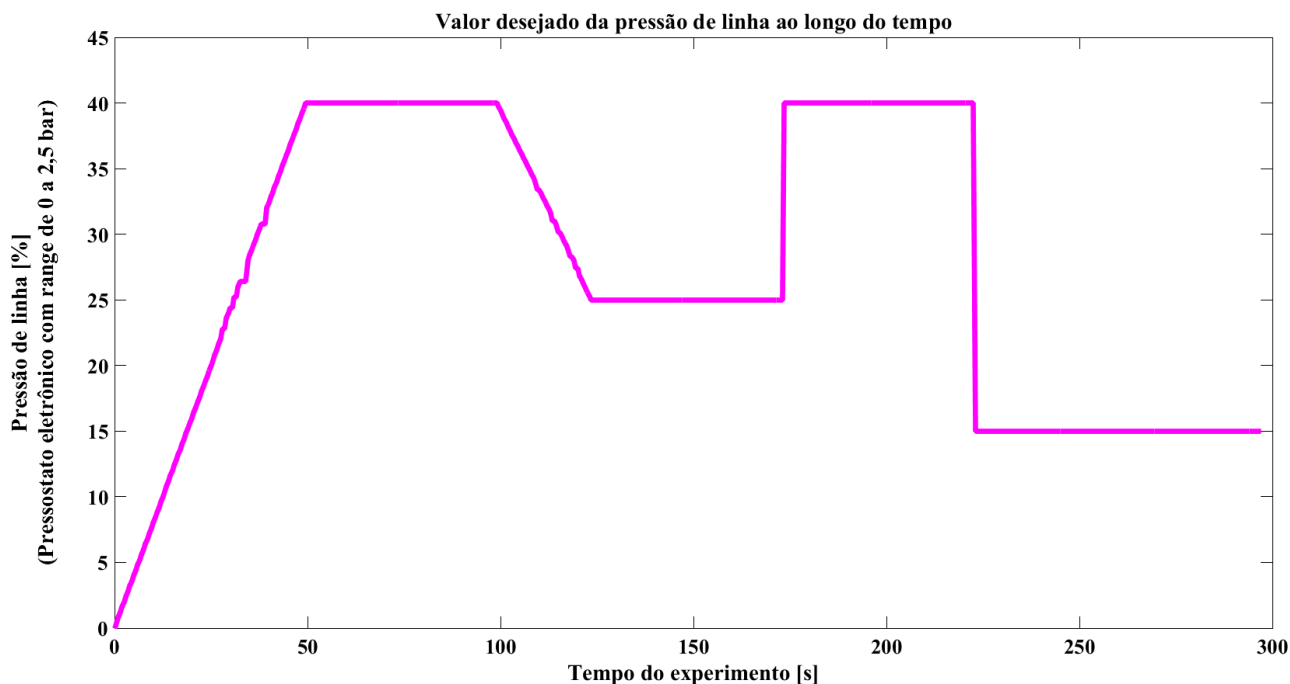
Vale destacar que monitorar e analisar constantemente KPIs na indústria de processos são de fundamental importância na inteligência de negócio, pois são capazes de traduzir os dados coletados do “chão de fábrica” em informações valiosas que refletem os resultados financeiros da organização. Sendo assim, os KPIs estudados são capazes de medir a qualidade de processos industriais através de avaliações quantitativas.

3.3.1 DELINEAMENTO EXPERIMENTAL

Entende-se por processo um ciclo no qual ocorrem transformações ao longo de um período com a finalidade produzir ou controlar algo. De maneira análoga a aplicações industriais, pode-se considerar um processo como um ciclo de fabricação de um produto, matéria prima ou transformações físico-químicas, por exemplo (PETERSON et al, 2008).

Optou-se por analisar processos que controlam a pressão de linha ao longo do tempo, porém destaca-se que qualquer malha de controle poderia ser escolhida. Um processo é representado na Figura 6. É possível observar que para esse trabalho um processo é composto por um ciclo de 300 segundos.

Figura 6 - Valor da pressão de linha desejado ao longo do processo.

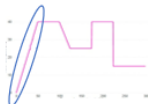
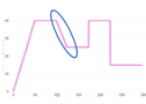
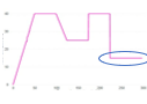


Fonte: Autoria própria.

A malha de controle possui comportamentos matemáticos bem definidos ao longo do processo e pode ser caracterizado em 6 etapas distintas: 1) há quatro trechos com valor desejado constante - trechos estacionários (o pressostato eletrônico possui um range de 0 a 2,5 bar e os valores foram representado em porcentagem: [50 => 100 s]: 40% (2ª etapa), [125 => 175 s]: 25% (4ª etapa), [175 => 225 s]: 40% (5ª etapa), [225 => 300 s]: 15% (6ª etapa); 2) dois trechos com mudança de valor desejado em rampa (função do 1º grau), sendo de 0 a 50 s (1ª etapa) uma função crescente e de 100 a 125 s (3ª etapa) uma função decrescente; e 3) duas mudanças de valor desejado em degrau, com alteração na pressão de linha de 25% para 40% e de 40% para 15%.

A tabela I, indica um resumo da caracterização das etapas do processo de forma bem clara.

Tabela I - Definição de um processo e caracterização das etapas estudadas.

CARACTERIZAÇÃO DO OBJETO DE ESTUDO						
Etapa	1ª	2ª	3ª	4ª	5ª	6ª
Período	0 - 50s	50 - 100s	100 - 125s	125 - 175s	175 - 225s	225 - 300s
Comportamento matemático	Função crescente do primeiro grau	Função constante	Função decrescente do primeiro grau	Função constante	Função constante	Função constante
Representação						

Fonte: Autoria própria.

É possível observar que a 1ª, 2ª, 4ª e 5ª etapa possuem período de 50 segundos, enquanto a 3ª etapa corresponde a 25 segundos e a 6ª etapa possui 75 segundos. Além disso, a tabela I indica o comportamento matemático e a representação gráfica de cada etapa.

Três processos distintos foram utilizados como objeto de estudo a fim de serem analisados e comparados entre si. Para simular 2 tipos de processos com distúrbio, optou-se por realizar perturbações em etapas específicas dos processos para caracterização de dois cenários diferentes de distúrbios. Essas perturbações foram provocadas através da abertura de válvulas manuais para observar o comportamento do controlador perante os distúrbios. Ainda, as métricas também tiveram o propósito de validar os índices de desempenho calculados (métricas de comparação entre 3 cenários: o processo sem distúrbio e outros processos com distúrbios em fases diferentes: 3ª e 5ª etapa no 1º processo e 4ª e 6ª etapa no 3º processo).

3.3.2 INDICADORES

Segundo Garcia, os indicadores que monitoram e avaliam o desempenho das malhas de controle devem ser simples o suficiente para não exigir intervenção humana e não ser invasivo. Isso quer dizer que os indicadores não devem fazer alterações no sistema que está sendo monitorado e que devem usar o mínimo de informação possível, dando preferência apenas às disponíveis nos sistemas de controle digital (GARCIA, p. 250-305, 2017).

Para avaliar o desempenho dos processos analisados neste trabalho, dois critérios foram utilizados: o primeiro está relacionado a parâmetros estatísticos descritivos como a média aritmética, mediana, variância, desvio padrão, variabilidade e índice de capacidade e o segundo critério relacionado à integral do erro ao longo do período. A tabela I apresenta os KPIs utilizados para avaliar o desempenho dos processos.

A estatística descritiva é um ramo de estudo da matemática na qual várias técnicas são aplicadas com o intuito de descrever e sumarizar um conjunto de dados. Normalmente são usadas medidas de tendência central, de variabilidade e de dispersão (FIELD; MILES; FIELD, 2002). Desta forma, vale destacar que nesse trabalho os parâmetros estatísticos foram calculados para a análise da variável de processo e a variável manipulada com o propósito de mensurar o desempenho das etapas estacionárias.

Tabela II - KPIs escolhidos para avaliar o desempenho dos processos.

Média aritmética
Mediana
Variância
Desvio padrão
Variabilidade
Índice de capacidade (CP e CPK)
Integral do erro absoluto (IAE)
Integral do erro quadrático (ISE)
Integral do erro absoluto pelo tempo (ITAE)
Integral do erro quadrático pelo tempo (ITSE)

Fonte: Autoria própria.

A representação matemática das funções utilizadas para calcular a média, o desvio padrão e a variância são apresentadas nas equações (1), (2), (3) respectivamente. A média representa o somatório dos valores da amostra dividido pelo número total de amostras. O desvio padrão e a variância são medidas que indicam como os valores analisados oscilam em torno da média. A seguir, a descrição das equações:

$$\mu = \sum_{i=1}^{i=n} \frac{x_i}{n} \quad (1)$$

$$\sigma = \sqrt{\sum_{i=1}^{i=n} \frac{(\mu - x_i)^2}{n-1}} \quad (2)$$

$$\sigma^2 = \sum_{i=1}^{i=n} \frac{(\mu - x_i)^2}{n-1} \quad (3)$$

A variância é o somatório da diferença entre a média e cada ponto da amostra elevado ao quadrado, dividido pelo número total de amostras menos um. Caso fosse calculada a variância da população, toda a divisão seria apenas pelo número total da população. O desvio padrão é a raiz quadrada da variância. Logo, caso a variância for menor que um o desvio padrão será maior que a variância; caso contrário, a variância será menor.

Além disso, a mediana também foi calculada, sendo que essa variável indica o valor central do conjunto de dados, ou seja, separa a metade maior e a metade menor de uma amostra ou população. Caso a faixa amostral analisada for par, então a mediana é a média dos dois valores centrais. Ou seja, soma os dois valores centrais e divide por 2. Caso o número de elementos for ímpar, então a mediana é o próprio valor central.

Tais parâmetros estatísticos foram analisados nos trechos estacionários, uma vez que o valor desejado foi sempre o mesmo em todo o período, os valores devem oscilar pouco em torno da média. Caso a variância e o desvio padrão apresentem elevados valores, é possível identificar distúrbios, perturbações ou problemas no processo.

O próximo índice avaliado foi a variabilidade do processo, conforme representado na equação (4). Tal parâmetro foi representado por duas vezes o desvio padrão dividido pela média e multiplicado por 100 para obter o valor em porcentagem. Ao comparar este índice com mais de um processo, foi possível obter uma estimativa da oscilação da malha de controle.

$$var = \frac{2 \cdot \sigma}{\mu} \cdot 100 \quad (4)$$

Segundo Garcia, há várias causas que podem gerar a variabilidade: sintonia inadequada do controlador, instrumento atuador mal selecionado, mal dimensionado ou com atrito excessivo, processo extremamente sensível, não linearidades na malha, etc. Distúrbios como mudança nos valores de referência ou alteração na sintonia dos controladores podem dar início à variabilidade no sistema. Logo, é notório que a variabilidade depende das perturbações ou do ruído na malha, da constante de tempo dominante do sistema em malha fechada, do tempo morto da malha e da zona morta do instrumento de controle (GARCIA, p. 250-305, 2017).

Para o cálculo do índice de capacidade é necessário estabelecer os limites inferior e superior conforme a necessidade da aplicação. Os índices CP e CPK, proposto por Kane, permitem determinar se a média de uma amostra ou processo está centrada em relação aos limites superior (LSE) ou inferior (LIE) da especificação, quantificando a capacidade de estabilidade do processo (MOYNE et al., 2020).

Num contexto de fabricação industrial esse índice mensura a qualidade do processo de acordo com os limites estabelecidos. Na indústria de manufatura, por exemplo, ele é utilizado quando um dispositivo é fabricado e seu tamanho medido para saber se atende as especificações desejadas de acordo com a margem de erro para que o produto possa ser vendido. Caso contrário, o produto deve ser descartado.

Optou-se em adotar os limites de maneira que a oscilação deve ser 10% do valor alvo; logo, quando o valor desejado é de 40% o limite inferior deve ser 36% e o limite superior deve corresponder a 44%. A escolha dos limites inferiores e superiores foi escolhida apenas para efeito de cálculos e a fim de gerar uma análise comparativa entre os processos.

Na prática, os limites devem ser escolhidos de acordo com a necessidade da aplicação, levando em conta condições críticas e relevantes tanto para o processo como para a regra de negócio. Por exemplo, em indústrias de processo com trechos críticos que a pressão de linha não deve oscilar muito em relação aos limites estabelecidos senão ocorrerá algum problema de segurança ou risco iminente. Portanto, caso necessário, os limites podem ser alterados de acordo com os valores desejados.

A equação (5) e (6) representam como os índices de capacidade (CP e CPK) foram calculados. O CP compara a tolerância especificada com a variação potencial do

processo, ou seja, descreve a relação entre o espaço disponível para variação de acordo com as especificações e o espaço ocupado pela variação do processo. Para caracterizar a localização do processo utiliza-se o CPK, que avalia a distância da média do processo com a especificação mais próxima dela

$$CP = \frac{LSE - LIE}{6.\sigma} \quad (5)$$

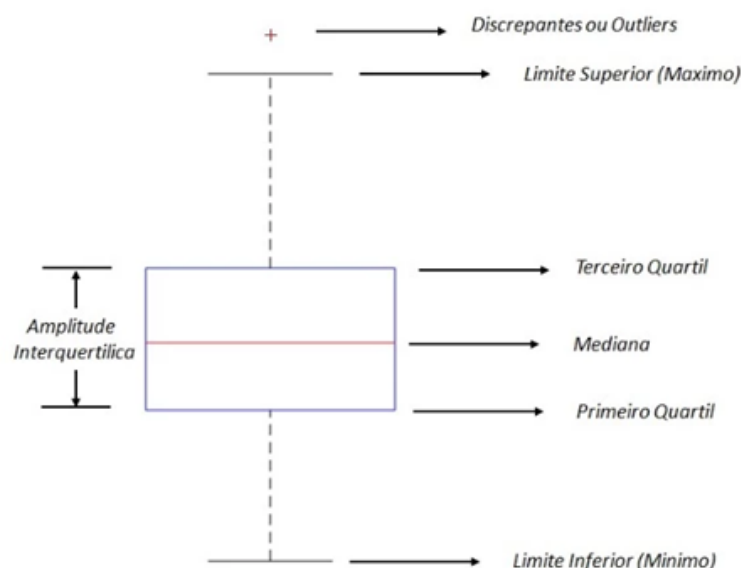
$$CPK = \min\left(\frac{LSE - \mu}{3.\sigma}; \frac{\mu - LIE}{3.\sigma}\right) \quad (6)$$

Quando o processo opera centralizado na nominal (alvo da especificação), os dois índices (CP e CPK) terão o mesmo valor, ou valores muito próximos. À medida que a média do processo se desvia da nominal, o CPK vai ficando menor que o CP (NARCISO-FILHO, 2017).

Um valor elevado para o índice de capacidade indica alta probabilidade do controlador atender às especificações limites do processo. Para trechos com os índices maiores que 1,33, denomina-se estes como altamente capazes, enquanto etapas com o parâmetro entre 1 e 1,33 são capazes e valores abaixo de 1 indicam processos incapazes. Entretanto, vale ressaltar que a análise de apenas um parâmetro, sem levar em consideração os demais, não é conclusivo e pode apresentar falhas. A análise de todos os parâmetros em conjunto permite avaliar o desempenho dos processos analisados.

Vale ressaltar que os índices de capacidade foram calculados apenas para a variável de processo, analisando a relação da pressão de linha nos trechos estacionários. Caso existisse alguma oscilação seria esperado que a variável manipulada fosse sujeita a fortes variações para que a variável de processo fosse mantida próxima ao valor desejado. Enquanto oscilações na pressão de linha são críticas ao processo e podem levar a problemas relacionados à segurança da operação, por exemplo.

Para averiguar a distribuição dos dados em cada trecho analisado, a fim de entender o comportamento matemático de cada período do processo, foi utilizado o gráfico *boxplot*, conforme Figura 7. Assim foi possível observar os limites máximos e mínimos da faixa amostral, os *outliers* (pontos que se encontram fora dos limites estabelecidos), a mediana e os intervalos interquartis.

Figura 7 - Gráfico *boxplot*

Fonte: (site da *Oper*, 2019).

A grande vantagem é que o diagrama identifica onde estão localizados 50% dos valores mais prováveis da amostra, a mediana e os valores extremos, além de dar uma noção de como é a distribuição dos dados. A representação é composta por uma caixa retangular que representa o intervalo interquartilico, em que o lado inferior indica primeiro quartil e o lado superior indica o terceiro quartil, já uma reta dentro do retângulo representa a mediana.

As hastes representam os limites (superior e inferior) e a reta entre as hastes e o retângulo aponta a distribuição dos dados acima do terceiro quartil e abaixo do primeiro quartil. Os *outliers* são os pontos extremos que se encontram fora dos limites estabelecidos na distribuição dos dados. Vale destacar que neste trabalho os *outliers* não foram desconsiderados das amostras analisadas.

Os índices baseados na integral do erro, como IAE, ISE, ITAE e ITSE permitem avaliar o comportamento do erro do processo, sendo calculado através da diferença entre o valor desejado e a variável de processo. As equações (7), (8), (9) e (10) representam como esses parâmetros são obtidos.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (7)$$

$$ISE = \int_0^{\infty} e^2(t) dt \quad (8)$$

$$ITAE = \int_0^{\infty} |e(t)| \cdot t dt \quad (9)$$

$$ITSE = \int_0^{\infty} e^2(t) \cdot t dt \quad (10)$$

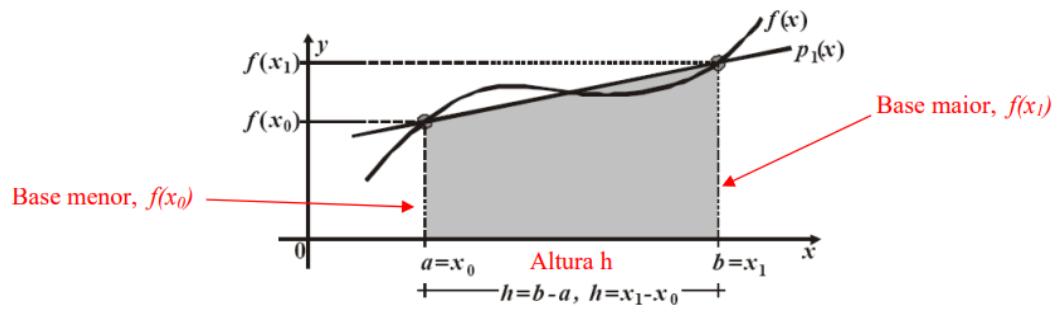
Trata-se de indicadores comparativos com o intuito de avaliar o erro ao longo do processo. Logo, tais critérios só fazem sentido quando mais de um processo é analisado. Quanto menor for o valor destes índices, melhor será a performance do processo.

O IAE representa a integral do erro absoluto e permite mensurar o erro ao longo do processo. Já o ISE é a integral do erro quadrático e tem como propósito penalizar mais gravemente erros maiores do que 1 e minimizar erros menores que 1, já que números menores que 1 elevado ao quadrado apresentam resultados menores. O ITAE é a integral do erro absoluto pelo tempo. Dessa forma, esse critério penaliza gravemente erros apresentados no final das etapas. O ITSE é a integral do erro quadrático pelo tempo, em que penaliza erros maiores do que 1 e ainda pondera mais gravemente erros nos finais das etapas.

Portanto, os parâmetros relacionados à integral do erro permitem diagnosticar erros ao longo do processo com ponderação e peso de acordo com erros maiores/menores que 1, além de levar em consideração o instante de ocorrência dos erros (início/final das etapas).

Vale destacar que para realizar os cálculos das integrais no *Power BI* utilizou-se a aproximação pela regra do trapézio, conforme pode ser explicado pelo gráfico da Figura 8. A ideia da regra do trapézio é aproximar a função $f(x)$ por um polinômio de ordem 1 (reta). Através de alguns cálculos e aproximações, como um polinômio interpolador e a fórmula de *Lagrange*, pode-se encontrar a expressão representada abaixo.

Figura 8 - Cálculo da integral pela aproximação da regra do trapézio.



Fonte: Universidade do Vale do Paraíba (UNIVAP, 2017).

$$\int_a^b f(x)dx \approx \int_{a=x_0}^{b=x_1} p_1(x)dx = \int_{x_0}^{x_1} \left[\frac{(x-x_1)}{-h} f(x_0) + \frac{(x-x_0)}{h} f(x_1) \right] dx = \frac{h}{2} [f(x_0) + f(x_1)]$$

Ressalta-se que o resultado foi obtido no *Power BI* através de cálculos entre dois pontos consecutivos (linha a linha) e o valor do KPI obtido através do somatório desses valores. Outrossim, com o intuito de apresentar uma análise comparativa entre os três processos analisados, os índices IAE, ISE, ITAE e ITSE foram normalizados com base no menor valor, ou seja, utilizando como referência o melhor processo. Sendo assim, o melhor valor é indicado com 100% e quanto menor for a porcentagem pior será o indicador do processo.

4 DESENVOLVIMENTO

Nesta etapa do trabalho será abordado os principais aspectos para o desenvolvimento do tema proposto. Inicialmente será apresentado um fluxograma explicando desde a coleta dos dados na planta piloto até a exibição do *dashboard*, características e justificativas na escolha das ferramentas utilizadas e aspectos positivos e negativos ao longo do desenvolvimento.

Por fim, será abordado a sequência de execução do trabalho de maneira técnica e detalhada a fim de explicar as etapas de desenvolvimento para a criação do banco de dados por meio da linguagem *JavaScript*, a *query* em SQL utilizada para filtrar os dados carregados no *Power BI*, o processo de ETL através do *Power Query*, o modelo de relacionamento dos dados, a criação dos KPIs na linguagem DAX e por fim a montagem do *dashboard*.

Além disso, vale ressaltar que não será explicado o desenvolvimento no *LabVIEW*, já que esta etapa não foi realizada neste trabalho. Sendo necessário apenas a escolha de uma das quatro malhas de controle disponíveis (controle da pressão de linha, pressão do reservatório, do nível ou da vazão).

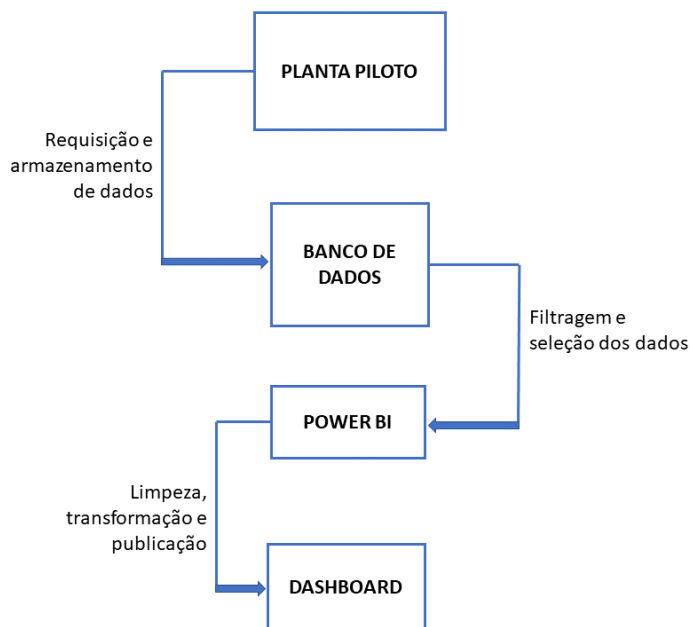
4.1 FLUXOGRAMA DOS DADOS E VISÃO GERAL DO DESENVOLVIMENTO DA PROPOSTA

A Figura 9 apresenta o fluxograma da transformação dos dados para o desenvolvimento deste trabalho. Inicialmente, os dados da planta piloto (controlador, sensores e atuadores) são requisitados e armazenados no banco de dados, posteriormente as informações são selecionadas e filtradas antes de serem carregadas para o *Power BI*. Por fim, o *software* da *Microsoft* executa a limpeza, transformação e publicação dos dados em um *dashboard* que pode ser acessado remotamente.

Vale destacar que a requisição e o armazenamento de dados foram realizados através de um microsserviço executado pelo *framework Moleculer*. O processo de filtragem e seleção foi feito por meio de uma *query* em SQL executada no *Power BI*. A limpeza, transformação e carregamento dos dados foi realizado através do *Power Query*, ferramenta de ETL do *Power BI*. A criação dos KPIs é feita por meio da

linguagem DAX, utilizada para calcular medidas na ferramenta de análise da *Microsoft*. Por fim, a publicação do *dashboard* foi realizada através do serviço do *Power BI*.

Figura 9 - Fluxograma com os dados do processo.



Fonte: Autoria própria.

Um aspecto interessante observado no trabalho foi a facilidade na comunicação dos dados através de diferentes *softwares* e ambientes de desenvolvimento. O *framework Molecular* possui conectores de banco de dados *MySQL* e o *Power BI* também consegue se comunicar com o banco de dados facilmente para consumir os dados da aplicação. Dessa forma, enfatiza-se que a consistência dos dados no processo de transformação e publicação dos *dashboard* não foi um problema no desenvolvimento do trabalho.

Além disso, outro fato observado ao longo do desenvolvimento do trabalho foi em relação a faixa amostral dos dados coletados. É importante salientar que inúmeros testes foram realizados na planta piloto a fim de entender o comportamento e o padrão de repetição amostral por ciclo executado. Ressalta-se que 595 dados por ciclo foi o espectro amostral mais recorrente entre todos os processos executados. Ademais, como nenhum ciclo apresentou 600 amostras, isso indica falha na requisição dos dados entre os serviços da planta piloto.

Dessa forma, uma das limitações encontradas no desenvolvimento do trabalho foi em não apresentar uma solução para o tratamento dos dados que não possuem 595 amostras.

A utilização do *Framework Moleculer* e o ambiente de desenvolvimento *Node.js* facilitaram o processo de criação do serviço do banco de dados, uma vez que possuem inúmeras bibliotecas e conectores disponíveis. Entretanto, vale ressaltar que uma limitação apresentada no pacote de bibliotecas disponíveis no *Moleculer* é o fato da biblioteca do *MySQL* permitir somente a criação de uma única tabela por serviço. Ou seja, caso houvesse a necessidade de criar mais de uma tabela para armazenar as informações dos processos, mais de um serviço precisaria ser criado.

Neste trabalho foi utilizado uma única tabela para armazenar as informações dos processos executados, nela estão contidas as informações relativas aos parâmetros do controlador, dos sensores e dos atuadores. Outras duas tabelas foram criadas no *Power BI* a fim de permitir aplicar filtros nos dados oriundos do *MySQL* e com o propósito de segmentar as informações da melhor forma possível para serem exibidas nos visuais do *dashboard*.

Ao longo do desenvolvimento do trabalho, uma das limitações encontradas no *software* da *Microsoft* é em relação a manipulação dos dados coletados pelo *MySQL*. Trata-se de uma limitação da ferramenta, tanto na versão paga como na gratuita. Apesar do *Power BI* possuir a ferramenta de ETL, o *Power Query*, o processo de separação dos ciclos executados na planta piloto que possuem a mesma faixa amostral se demonstrou muito lento de ser realizado.

Dessa forma, como o *Power BI* apresenta a possibilidade de executar uma consulta em linguagem nativa do banco de dados (SQL), essa solução foi desenvolvida. Ao se conectar com o banco de dados é possível inserir a *query* a fim de filtrar os dados do processo de acordo com a necessidade da aplicação. Além disso, destaca-se que a *query* selecionou apenas as colunas desejadas. Ou seja, somente os dados referente a execução da malha da pressão de linha, logo informações a respeito do sensor de nível, por exemplo, não foram carregadas ao *Power BI*.

Ademais, o *Power Query* demonstrou-se útil na seleção dos últimos 3 processos executados na planta piloto. Além disso, através dele também foi possível criar uma coluna contendo o período de integração de cada etapa e três colunas

calculadas: uma com os valores do erro ao longo do processo, outra com o erro absoluto e a última coluna contendo o erro quadrático.

Para a criação do relacionamento dos dados entre as diferentes tabelas, o *Power BI* apresenta uma interface amigável e fácil de ser utilizada. Basta selecionar a coluna comum entre as tabelas e apontar a direção do filtro entre elas. Algumas vezes a ferramenta é capaz de detectar automaticamente a relação entre os dados. Porém, é necessário conferir o modelo de relacionamento para garantir que a integridade dos dados estejam corretas.

Por fim, a linguagem DAX (disponível no *Power BI*, *Excel* e em outras ferramentas da *Microsoft*) foi utilizada para calcular os KPIs estudados. A principal vantagem na utilização da linguagem DAX é que ela inclui funções, operadores e valores para realizar cálculos avançados e consultas em dados nas tabelas e colunas relacionadas nos modelos de dados tabulares (MICROSOFT, 2022).

No desenvolvimento do trabalho, os índices IAE, ISE, ITAE e ITSE foram obtidos através de colunas calculadas e que posteriormente foram exibidos como a soma total da coluna. Outrossim, os parâmetros estatísticos utilizados para mensurar as etapas estacionárias foram obtidos através de medidas calculadas. Essa ferramenta do *Power BI* mostrou-se capaz de realizar os cálculos conforme a necessidade, sem apresentar maiores problemas.

Por fim, cada etapa do desenvolvimento do trabalho será explicada em detalhes a seguir.

4.2 CRIAÇÃO DO BANCO DE DADOS

Para realizar a coleta de dados foi necessário baixar e instalar o *software MySQL* (utilizou-se a versão 8). Após as configurações necessárias, um banco de dados foi criado localmente (*localhost*) com uma senha adequada e com o usuário padrão “*root*”. Ao abrir o *prompt* de comando do *software MySQL* e digitar a senha de acesso, foi possível criar o banco de dados para armazenar as informações do processo através do comando “CREATE DATABASE daq_db”.

As confecções da tabela e de todas as demais estruturas necessárias para o banco de dados foram criadas através da linguagem de programação *JavaScript* e para

isso, instalou-se o ambiente de desenvolvimento *Visual Studio Code* e a plataforma *Node.js*. Inicialmente, foi necessário instalar o *framework* utilizado e escolheu-se uma pasta como diretório, acessada através de “cd <local da pasta>” no *prompt* de comando da ferramenta *Visual Studio*. A seguir o *framework Molecular* foi instalado na pasta desejada através do comando “npm i molecular-cli-g”; e por fim foi criado o projeto digitando “molecular init heret <nome do projeto>”.

Após essa etapa, iniciou-se a criação do *script* do serviço do banco de dados, optou-se por habilitar a comunicação com outros nós e foi escolhido o *Transporter NATS*. As demais funcionalidades (como o serviço *API Gateway* e o *software Docker*, por exemplo) não foram utilizadas para essa aplicação. A fim de iniciar o desenvolvimento, foi essencial a instalação de algumas bibliotecas utilizadas através do gerenciador de pacotes oficial NPM do *framework Molecular*. As principais dependências instaladas foram para o banco de dados *MySQL* (*molecular-db*, *molecular-db-adapter-sequelize*, *sequelize* e *mysql2*).

O passo inicial para o desenvolvimento do projeto foi carregar as bibliotecas utilizadas e definir as variáveis globais, conforme consta na Figura 10. A biblioteca “*dotenv*” foi utilizada para armazenar informações sobre o *Transporter* e sua rota de comunicação (com um endereço IP e a porta de comunicação) em um arquivo no diretório “.env”. O mesmo mecanismo poderia ser feito para armazenar a senha e o usuário do banco de dados, já que tal estratégia é segura por não deixar visível as informações relevantes na troca de informações.

Figura 10 - Definição das bibliotecas e das variáveis globais.

```
require('dotenv').config()
const { ServiceBroker } = require("molecular")
const DbService          = require("molecular-db")
const SqlAdapter          = require("molecular-db-adapter-sequelize")
const Sequelize           = require("sequelize")

let broker = new ServiceBroker({transporter: process.env.TRANSPORTER})
let requisition1 = {}
let requisition2 = {}
let requisition3 = {}
```

Fonte: Autoria própria.

O *broker* é um dos principais componentes do *framework Moleculer*, sendo utilizado para tratar os serviços, chamar as ações, emitir eventos, comunicar com outros nós, etc. Os objetos “requisition1”, “requisition2” e “requisition3” foram criados para armazenar respectivamente as requisições dos dados do sensor, atuador e controlador.

Na Figura 11 é possível observar o nome do serviço criado (“pressures”). As credenciais do banco de dados como nome, usuário e senha também são informados ao adaptador utilizado. O *localhost* define que o banco de dados é local, o UTF8 é o padrão de caracteres utilizado, *timezone* define o fuso horário em relação ao meridiano de Greenwich e *port* define a porta de comunicação utilizada pelo banco de dados. Em “model” (Figura 8), trata-se da definição da tabela e das colunas que foram criadas.

Figura 11 - Definição do serviço e da tabela do banco de dados.

```
broker.createService({
  name: 'pressures',
  mixins : [DbService],
  adapter: new SqlAdapter('daq_db', 'root', 'l2qazX90plkM', {
    host: 'localhost',
    dialect: 'mysql',
    charset: 'utf8',
    collate: 'utf8_general_ci',
    timezone: '-03:00',
    port: '3307',
  }),
  model: {
    name: "pressure",
    define: {
      pitl129: Sequelize.FLOAT,
      pitl118: Sequelize.FLOAT,
      litl125: Sequelize.FLOAT,
      fitl116: Sequelize.FLOAT,
      ruout1: Sequelize.FLOAT,
      ruout2: Sequelize.FLOAT,
      ruout3: Sequelize.FLOAT,
      ruout4: Sequelize.FLOAT,
      controll1: Sequelize.FLOAT,
      control2: Sequelize.FLOAT,
      control3: Sequelize.FLOAT,
      control4: Sequelize.FLOAT,
      control5: Sequelize.FLOAT,
      control6: Sequelize.FLOAT
    },
    options: {
      logging: false
    }
  },
},
```

Fonte: Autoria própria.

Quando é informado apenas o “name” do “model” e não é especificado o nome da tabela, por padrão, o adaptador *Sequelize* define o plural do “name” do “model” como o nome da tabela (dessa forma, “pressures” é o nome da tabela). Todos os dados que serão armazenados são do tipo *float*, sendo os quatro primeiros os dados dos sensores, os quatro “ruout” os dados dos atuadores e, por último, seis variáveis de controle. Vale ressaltar que a coluna do identificador único (ID) e a coluna com data e hora foram criadas automaticamente pelo adaptador do banco de dados utilizado.

Após a construção da estrutura do banco de dados, foi possível criar uma ação no *framework Molecular* para escrever as informações coletadas no banco de dados. Para isto, a função criada deve ser assíncrona, conforme a Figura 12. Dentro da função, foi criada uma variável local a fim de encapsular e armazenar as informações das três requisições. Através do “await” (função assíncrona) a promessa aguarda até que seja resolvida e assim é possível armazenar as variáveis coletadas nas colunas estabelecidas.

Figura 12 - Ação do serviço *pressures* para escrever os valores no banco de dados.

```
actions: {
  async getEscreve(ctx) {
    let data = {requisition1, requisition2, requisition3}
    await ctx.call("pressures.create", {
      pitl29: data.requisition1.riin[0],
      pitl18: data.requisition1.riin[1],
      litl25: data.requisition1.riin[2],
      fitl16: data.requisition1.riin[3],
      ruout1: data.requisition2.ruout[0],
      ruout2: data.requisition2.ruout[1],
      ruout3: data.requisition2.ruout[2],
      ruout4: data.requisition2.ruout[3],
      control1: data.requisition3.control.inputs.setpointInPercentage,
      control2: data.requisition3.control.inputs.proportionalConstant,
      control3: data.requisition3.control.inputs.resetTimeInMs,
      control4: data.requisition3.control.inputs.derivativeTimeInMs,
      control5: data.requisition3.control.inputs.processVariableInPercentage,
      control6: data.requisition3.control.outputs.manipulatedVariableInPercentage
    }, {time: 100}).then(async(res)=>{
      this.logger('Success! ${res.message}')
      data = Object.assign(data, res)
    })
    .catch(async (err) => {
      await this.logger.error('Error occurred in posts! ${err.message}')
    })
  },
}
```

Fonte: Autoria própria.

O processo de armazenamento de todas as variáveis coletadas no banco de dados deve demorar menos de 100 ms. Logo, os dados foram salvos e o objeto “data” conteve todas as informações do processo, caso contrário uma mensagem de erro poderia aparecer e as informações não seriam salvas. Vale ressaltar que nem todas as

informações coletadas foram utilizadas na análise. Na seção seguinte será explicado de forma mais detalhada quais foram as colunas utilizadas.

As chamadas que foram executadas em *loop* foram representadas na Figura 13. Inicialmente o serviço DAQ é requisitado, seguindo a ordem: 1) chamou-se a ação *riin* e salva seus valores em “*requisition1*”; 2) a ação *ruout* foi executada e seus valores são salvos em “*requisition2*” e 3) salvou-se as variáveis do serviço de controle (*control-mesh-3.data* foi utilizado já que a malha de controle analisada possui o propósito de controlar a pressão de linha, caso outra malha de controle fosse executada seria necessário alterar para *mesh-1*, *mesh-2* ou *mesh-4*) em “*requisition3*”.

Figura 13 - Chamadas executadas em loop a cada 500 ms.

```
broker.start()
.then(() => {
  setInterval(() => {
    broker.call("daq.riin",{time: 100})
    .then(res =>{requisition1 = res})

    .then(() => broker.call("daq.ruout",{time: 100})
    .then(res =>{requisition2 = res})
    .catch(err => console.error('Error occurred! ${err.message}'))

    .then(() => broker.call("control-mesh-3.data",{time: 100})
    .then(res =>{requisition3 = res})
    .catch(err => console.error('Error occurred! ${err.message}'))

    .then(() => broker.call("pressures.getEscreve",{time: 100})
    .then(res =>{data = res})
    .catch(err => console.error('Error occurred! ${err.message}'))

  }, 500)
})
```

Fonte: Autoria própria.

Após requisitar todas as informações desejadas, a ação “*getEscreve*” do serviço *pressures* foi chamada a fim de salvar os valores requisitados no banco de dados, conforme esquema já explicado anteriormente. Cada chamada de serviço possui tempo máximo de 100 ms para ser executado, caso contrário uma mensagem de erro é exibida. De acordo com a função “*setInterval*”, foi definido que os dados seriam coletados a cada 500 ms.

4.3 DESENVOLVIMENTO DA QUERY

Finalizado a construção do banco de dados, a próxima etapa foi desenvolver uma *query* em SQL a fim de filtrar os dados antes de serem analisados no *Power BI*. Para isso deve-se levar em consideração a malha de controle escolhida (pressão de linha) com o intuito de selecionar somente as colunas de interesse.

Dentre as 16 colunas do banco de dados (4 valores referente aos sensores: pit129, pit118, lit125, fit116; 4 relacionadas aos atuadores: ruout1, ruout2, ruout3, ruout4; 6 em relação aos parâmetros do controlador: control1, control2, control3, control4, control5 e control6; além da coluna de ID e createdAt: referente a data e hora), foram escolhidas 7 colunas: ID, pit118, ruout2, control1, control5, control6 e createdAt, com valores relativos ao identificador único, pressão de linha medido pelo sensor, sinal da bomba d'água medido pelo atuador, valor desejado da pressão de linha ao longo do processo, variável de processo (pressão de linha) medido pelo controlador, sinal de controle (bomba d'água) medido pelo controlador e data e hora, respectivamente.

Após a escolha das colunas desejadas foi possível iniciar o desenvolvimento no *Power BI*. Inicialmente, para conectar os dados do *MySQL* na ferramenta de análise, foi necessário realizar o *download* do conector de dados através do site *MySQL*: Download Connector/NET. Ao instalar no computador foi possível conectar as informações do banco de dados com o *Power BI*.

Feito isso, para carregar as informações do banco de dados basta clicar em “Obter dados” no menu superior da barra de ferramentas do *Power BI* e selecionar a opção *MySQL*. Em seguida, foi informado o servidor (*localhost*) e a porta de comunicação (3307), o nome do banco de dados (*daq_db*) e inserido a *query* que será explicada mais adiante, conforme Figura 14.

O propósito da *query* é selecionar os processos que possuem a mesma faixa amostral para serem analisados. Além disso, outra vantagem da *query* é aumentar a performance e diminuir as inúmeras transformações no *Power BI*, já que muitos processos de limpeza na ferramenta de análise podem demorar mais tempo para ser processado.

Para elaboração da *query*, o primeiro passo foi definir a faixa amostral estudada. Através de inúmeros testes realizados na planta piloto observou-se que a faixa amostral mais recorrente é de 595 dados por processo, lembrando que os dados são coletados a cada 500 ms. Além disso, destaca-se que nenhum dos ciclos realizados apresentou 600 amostras, demonstrando falha em alguma das requisições entre os diferentes serviços.

Figura 14 - Credenciais para acessar o banco de dados.

MySQL database

Server
localhost:3307

Database
daq_db

Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

```
use daq_db;

## Prepara os dados com as amostras desejadas
SET @CT = (SELECT control1 FROM pressures LIMIT 1);
SET @INDICE = 0;
SET @RNUMBER = 0;
```

☒ Include relationship columns

☐ Navigate using full hierarchy

OK Cancel

Fonte: Autoria própria.

Para filtrar os dados através da *query*, o primeiro passo foi a criação de uma coluna nomeada de “INDICE” cuja função foi diferenciar os ciclos. Essa coluna incrementa 1 toda vez que a coluna control1 (Valor desejado) apresenta o valor 0, já que cada ciclo inicia-se com o Valor desejado em 0. Dessa forma, essa coluna consegue separar cada processo executado. A *query* é apresentada na Figura 15.

O próximo passo foi a criação de uma outra tabela, chamada de “tabela_temp”, cuja função é filtrar apenas os ciclos com a faixa amostral de interesse. Essa tabela possui apenas duas colunas, a primeira indica o processo e a segunda aponta a faixa amostral dos processos. A variável nomeada de TT foi utilizada para escolher a faixa amostral de interesse. Para o presente trabalho optou-se por 595 amostras, conforme já

mentionado, no entanto caso desejado a quantidade pode ser alterada de acordo com a necessidade da aplicação.

Figura 15 - Query em SQL realizada no *Power BI*.

```
use daq_db;

## Prepara os dados com a faixa amostral desejada
SET @CT = (SELECT control1 FROM pressures LIMIT 1);
SET @INDICE = 0;
SET @RNUMBER = 0;

truncate tabela_temp;

insert into tabela_temp (
select * from (select indice,0 from (
select indice,count(1) as TT
FROM (
SELECT
if(control1 = 0, @INDICE := @INDICE + 1 , @INDICE := @INDICE ) AS indice,
@CT := control1 FROM pressures
) a group by indice
) b where b.TT = 595 ) b group by indice); ## Altera a faixa amostral - TT

## Lista apenas os indices constantes na tabela_temp
SET @CT = (SELECT control1 FROM pressures LIMIT 1);
SET @INDICE = 0;
SET @RNUMBER = 0;

select id, pit118, ruout2, control1, control5, control6, createdAt,indice

FROM (
SELECT
id, pit118, ruout2, control1, control5, control6, createdAt,
if(control1 = 0, @INDICE := @INDICE + 1 , @INDICE := @INDICE ) AS indice,
if(control1 = @CT, @RNUMBER := @RNUMBER + 1 , @RNUMBER ) AS rnumber,
@CT := control1 FROM pressures
) a where a.indice in
(
select indice from tabela_temp
)
```

Fonte: Autoria própria.

Por fim, a tabela “tabela_temp” foi utilizada para filtrar os dados da tabela “pressures” criada em *JavaScript*. Além das colunas que foram escolhidas (conforme já explicado anteriormente), a *query* cria uma nova coluna com o nome de INDICE, no qual indica o número do processo.

4.4 PROCESSO DE ETL NO POWER QUERY

Com as informações desejadas carregadas para o *Power BI*, foi possível realizar a limpeza, transformação e carregamento dos dados. O primeiro passo no *Power Query* foi escolher a quantidade de ciclos a serem analisados dentre todos os processos com 595 amostras.

Optou-se por analisar os 3 últimos ciclos a fim de realizar uma comparação através dos KPIs. Dois ciclos analisados apresentam distúrbios em etapas distintas (um processo possui distúrbio na 3ª e 5ª etapa enquanto outro ciclo apresenta distúrbio no 4º e 6º trecho) provocados através da abertura de uma das válvulas manuais e 1 ciclo sem distúrbios. Dessa forma, a tabela apresenta 1785 linhas. Caso necessário todos os ciclos com 595 amostras poderiam ser analisados, ou quantos ciclos desejados.

A seguir, a tabela foi renomeada para *Análise* e as colunas *pit118*, *ruout2*, *control1*, *control5*, *control6*, *createdAt* e *indice* foram renomeadas para Sensor de pressão, Bomba d'água, Valor desejado, Variável de processo, Variável manipulada, Data e hora e Processo, respectivamente. Por fim, a coluna *Aux* (iniciada em 0 e incrementando 1 a cada linha) foi criada com o intuito de auxiliar nos cálculos futuros (como subtrair linhas subsequentes), a coluna *Tempo* somente com dados referentes ao tempo dos processos foi criada e normalizada com base na coluna *Data e hora*. Além disso, a coluna *IDProcesso* também foi criada iniciando em 1 e incrementando 1 cada linha com a condição de reiniciar a contagem cada vez que o valor desejado for 0.

Por último, três colunas calculadas foram criadas: uma com o valor do erro ao longo do processo (Valor desejado subtraído da Variável de processo, vale destacar que essa coluna foi criada no intuito de observar o comportamento do erro, apesar de não ser utilizada para nenhum cálculo e nem ser exibida no *dashboard*); uma segunda coluna do erro absoluto, ou seja, o módulo do erro, para calcular essa coluna utilizou-se uma função condicional para averiguar o maior valor entre o Valor de desejado e a Variável de processo e realizar a subtração correta; e por último a coluna relacionada ao erro quadrático, ou seja, o Valor desejado subtraído da Variável de processo elevado ao quadrado. As três colunas calculadas são mostradas na Figura 16.

Antes de finalizar todo o processo de ETL no *Power Query*, duas outras tabelas foram criadas com o intuito de filtrar os dados conforme a necessidade e facilitar o trabalho de exibição no *dashboard*. A primeira tabela criada foi chamada de *Processos* e teve como propósito filtrar um dos três processos analisados. A segunda tabela foi nomeada de *Etapas* e através dela foi possível filtrar os 6 trechos dos processos.

Como a função Valor desejado pelo tempo é descontínua, com etapas bem definidas e com comportamento matemático distintos, o objetivo da tabela foi filtrar cada trecho do processo. Dessa forma, existem 6 etapas para cada processo, sendo 4

trechos com valor constante e 2 períodos representados por uma função do primeiro grau (crescente e decrescente).

Figura 16 - Colunas calculadas no Power Query.

The figure displays three sequential screenshots of the 'New column' dialog box in Power Query, each showing a different custom formula:

- Top Screenshot:** The 'New column name' field contains 'Erro absoluto'. The 'Custom column formula' field contains the formula: `= if [Valor desejado] >= [Variável de processo] then [Valor desejado] - [Variável de processo] else [Variável de processo] - [Valor desejado]`.
- Middle Screenshot:** The 'New column name' field contains 'Erro quadrático'. The 'Custom column formula' field contains the formula: `= ([Valor desejado] - [Variável de processo])*([Valor desejado] - [Variável de processo])`.
- Bottom Screenshot:** The 'New column name' field contains 'Erro'. The 'Custom column formula' field contains the formula: `= [Valor desejado] - [Variável de processo]`.

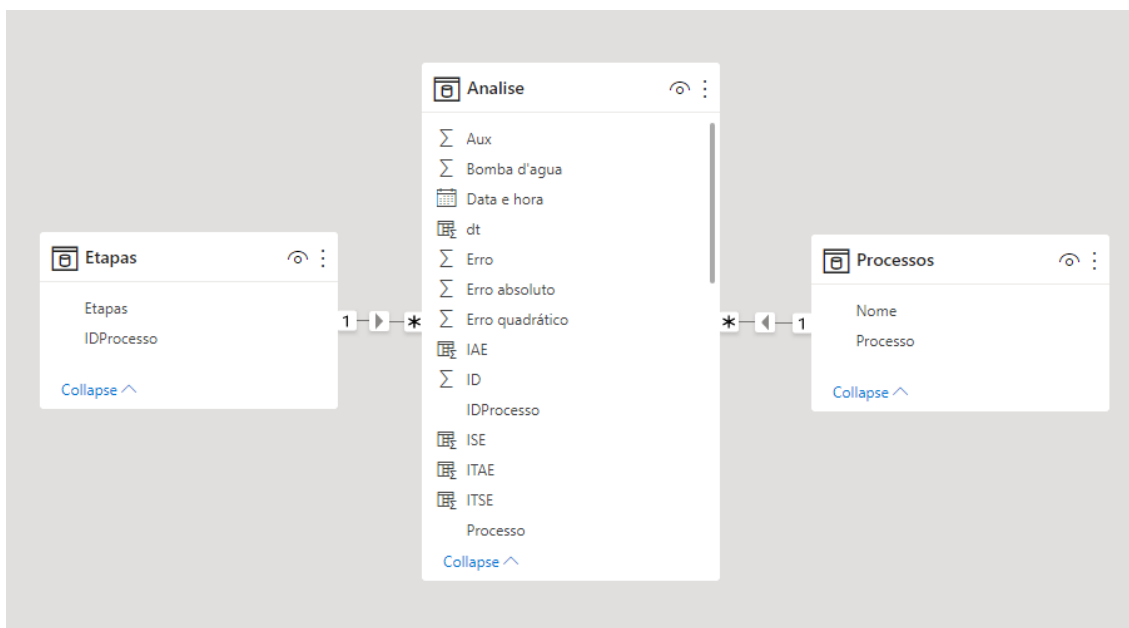
Fonte: Autoria própria.

4.5 MODELO DE RELACIONAMENTO DOS DADOS

Finalizado todas as etapas no *Power Query*, o próximo passo foi a criação do relacionamento entre as tabelas, sendo esse processo realizado na aba “Modelo” do *Power BI*. Adotou-se o modelo *star schema*, conforme Figura 17. Nesse esquema, as tabelas foram classificadas como dimensão ou fato: na tabela fato encontraram-se dados quantitativos relacionados às observações ou eventos que de fato ocorreram, nesse caso todos os dados do processo encontram-se nessa tabela; já na tabela de dimensão encontram-se dados que descreveram as entidades de negócio, sendo as informações relativas aos processos e as etapas do processo.

Em geral, as tabelas de dimensões contêm um número relativamente pequeno de linhas, enquanto as tabelas de fato podem conter um número muito grande de linhas e continuar crescendo ao longo do tempo. Como cada elemento visual de relatório do *Power BI* gera uma consulta que é enviada para o modelo do *software*, essas consultas são utilizadas para filtrar, agrupar e resumir dados do modelo.

Figura 17 - Relacionamento dos dados através do modelo *star schema*.



Fonte: Autoria própria.

Assim, um modelo bem projetado fornece tabelas para filtragem e agrupamento e tabelas para resumo. Neste caso, as tabelas Processo e Etapas foram utilizadas para filtrar a tabela Análise.

Outrossim, uma relação de modelo estabelece um caminho de propagação de filtro entre duas tabelas, e é a propriedade de cardinalidade da relação que determina o tipo de tabela. Uma cardinalidade em uma relação comum é uma relação de um para muitos ou seu inverso, de muitos para um. O lado "um" é sempre uma tabela de tipo de dimensão, enquanto o lado "muitos" é sempre uma tabela de tipo de fato. (MICROSOFT, 2022).

De acordo com a Figura 17, é possível notar que a tabela Processos possui duas colunas, sendo a primeira com os valores 1, 2 ou 3 e a segunda coluna com os 1º, 2º e 3º processo e o 3º Processo.

Já a tabela Etapas possui a coluna IDProcesso. Ela é um identificador único de cada linha desta tabela e a coluna Etapas representando os seis trechos do processo, conforme já explicado anteriormente. Além disso, é possível observar que a tabela Análise possui algumas colunas como dt, IAE, ISE, ITAE e ITSE que foram criadas através da linguagem DAX e que serão explicadas mais adiante.

4.6 CÁLCULO DOS KPIs NA LINGUAGEM DAX

O primeiro passo para realizar os cálculos dos indicadores apresentados foi criar colunas adicionais auxiliares através da linguagem DAX. A primeira coluna criada foi a dt. Esta coluna possui o objetivo de armazenar o período de integração de cada etapa do processo. Como a variável desejada (pressão de linha) é uma função descontínua, cada trecho deve possuir seu período de integração bem definido.

Posteriormente, criou-se as colunas que vão representar os índices IAE, ISE, ITAE, ITSE. Vale ressaltar que para calcular as integrais a aproximação pela regra do trapézio foi utilizada. Dessa forma, o cálculo é feito linha a linha e depois o resultado é obtido através do somatório da coluna. As colunas calculadas são representadas na Figura 18.

Figura 18 - Colunas calculadas na linguagem DAX para os KPIs IAE, ISE, ITAE e ITSE.

```
IAE = VAR IAE = (('Analise'[dt] - CALCULATE(SUM('Analise'[dt]),
    FILTER('Analise', 'Analise'[ID] = EARLIER('Analise'[Aux]))))*
    ('Analise'[Erro Absoluto] + CALCULATE(SUM('Analise'[Erro Absoluto]),
    FILTER('Analise', 'Analise'[ID] = EARLIER('Analise'[Aux])))))/2
    return
    IF('Analise'[dt] = 0, 0, IAE)

ISE = VAR ISE = (('Analise'[dt] - CALCULATE(SUM('Analise'[dt]),
    FILTER('Analise', 'Analise'[ID] = EARLIER('Analise'[Aux]))))*
    ('Analise'[Erro Quadrático] + CALCULATE(SUM('Analise'[Erro Quadrático]),
    FILTER('Analise', 'Analise'[ID] = EARLIER('Analise'[Aux])))))/2
    return
    IF('Analise'[dt] = 0, 0, ISE)

ITAE = VAR ITAE = (('Analise'[dt] - CALCULATE(SUM('Analise'[dt]),
    FILTER('Analise', 'Analise'[ID] = EARLIER('Analise'[Aux]))))*
    (('Analise'[Erro Absoluto]*'Analise'[dt]) + (CALCULATE(SUM('Analise'[dt]),
    FILTER('Analise', 'Analise'[ID] = EARLIER('Analise'[Aux]))))*
    CALCULATE(SUM('Analise'[Erro Absoluto]),
    FILTER('Analise', 'Analise'[ID] = EARLIER('Analise'[Aux])))))/2
    return
    IF('Analise'[dt] = 0, 0, ITAE)
```


quando o valor desejado é de 40% o limite inferior deve ser 36% e o limite superior deve corresponder a 44%. A escolha dos limites inferiores e superiores foi escolhida apenas para efeito de cálculos e a fim de gerar uma análise comparativa entre os processos. Os parâmetros foram obtidos conforme apresentados na Figura 20.

Figura 20 - Medidas calculadas na linguagem DAX para os índices de capacidade.

```
CP = ((1.1*MAX(Analise[Valor desejado]))-(0.9*MAX(Analise[Valor desejado])))/
      (6*STDEV.S(Analise[Variável de processo]))

CPK = MIN(
  ((MAX(Analise[Valor desejado])*1.1)-AVERAGE(Analise[Variável de processo]))/
    (3*STDEV.S(Analise[Variável de processo])),
  (AVERAGE(Analise[Variável de processo])-(MAX(Analise[Valor desejado])*0.9))/
    (3*STDEV.S(Analise[Variável de processo]))
)
```

Fonte: Autoria própria.

Na Figura 22, observou-se que quatro principais funções foram utilizadas: MIN, MAX, AVERAGE e STDEV.S. A função MIN tem como objetivo retornar o menor dos dois valores calculados. A função MAX tem como objetivo retornar apenas um valor da coluna Valor desejado e não possui o propósito de retornar o valor máximo.

Como um filtro foi aplicado nos visuais para cada etapa do processo, os trechos analisados possuem sempre o mesmo valor para a coluna Variável desejada (por exemplo, a segunda etapa do processo possui Valor desejado igual a 40% durante todo o período), logo a função MAX retorna apenas um valor do trecho escolhido. A função AVERAGE e STDEV.S calcula a média e desvio padrão, respectivamente, conforme já explicado anteriormente.

Por fim destaca-se que os parâmetros estatísticos relacionados a média, mediana, desvio padrão e variância podem ser obtidos através de medidas pré programadas no *Power BI*. Para isso basta selecionar a coluna desejada e escolher a medida de interesse.. Dessa forma, esses indicadores foram obtidos nos próprios visuais do *dashboard*.

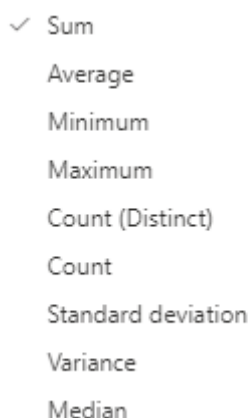
4.7 ELABORAÇÃO DO DASHBOARD

Por fim, a última etapa foi a elaboração do *dashboard*. Optou-se em escolher cartões para exibir as medidas estatísticas, gráficos de barra para apresentar os valores

do IAE, ISE, ITAE e ITSE dos 3 processos (além dos cartões que exibem os valores normalizados em relação ao melhor processo); 2 gráficos de linha, sendo um para representar o valor desejado e a variável de processo ao longo do tempo e outro com a variável manipulada ao longo do tempo; 1 gráfico de área para representar o erro absoluto ao longo do tempo e um gráfico *boxplot*. Além disso, é possível utilizar filtros no *dashboard* para filtrar cada processo e etapa de interesse.

Conforme mencionado anteriormente, valores como a média, mediana, desvio padrão e variância podem ser obtidos na etapa de montagem do dashboard, conforme Figura 21. Basta selecionar a coluna desejada e escolher como a informação será exibida.

Figura 21 - Medidas de agregação de colunas pré-programadas e disponíveis no *Power BI*.



Fonte: Autoria própria.

Para criar o *boxplot* no *Power BI* existem algumas possibilidades, como utilizar um *script* em *python* ou linguagem R ou obtê-lo através da *Microsoft Store*, já que esse gráfico não está inserido como padrão na ferramenta. A limitação em utilizar *scripts* consiste na atualização automática dos visuais do *dashboard*, caso ocorram novos processos e mais informações entrem no banco de dados não haverá consistência nas informações exibidas no serviço do *Power BI*.

Dessa forma, o visual *Box and Whisker by MAQ Software* foi escolhido (através da *Microsoft Store*) para exibição do *boxplot*. Para isso, foi selecionado o ícone “Obter mais visuais”, realizado o login com o e-mail institucional para observar mais visuais disponíveis e selecionado o gráfico desejado.

Ele possui alguns campos para serem preenchidos a fim de que o visual seja exibido corretamente, conforme Figura 22. O primeiro campo (*Axis*) deve ser preenchido com o identificador único de cada amostra, logo a coluna ID foi colocada; existem dois campos de eixo categórico (*Axis category I* e *Axis category II*) sendo um preenchido pela coluna Etapas e outro pela coluna Processo.

Por fim, o campo com os valores desejados, que no caso foi a coluna Variável de Processo para um dos gráficos e Variável manipulada para outro gráfico é colocado no campo Value. Vale destacar que os demais campos como *Dots size*, *Legend* e *Tooltips* não foram preenchidos.

Figura 22 - Campos para serem preenchidos do gráfico *Box and Whisker* by *MAQ Software*

The image shows a vertical configuration panel for a 'Box and Whisker' chart. It contains the following sections and fields:

- Axis:** A dropdown menu with 'ID' selected.
- Axis category I:** A dropdown menu with 'Etapas' selected.
- Axis category II:** A dropdown menu with 'Processo' selected.
- Value:** A dropdown menu with 'Variável manipulada' selected.
- Dots size:** A text input field containing 'Add data fields here'.
- Legend:** A text input field containing 'Add data fields here'.
- Tooltips:** A text input field containing 'Add data fields here'.

Fonte: Autoria própria.

Optou-se por exibir o *dashboard* com 3 abas: a primeira foi respectiva aos índices calculados e um *overview* dos processos, sendo cada processo escolhido mediante um filtro contido no visual, assim como cada etapa pode ser analisada em detalhe por um filtro das etapas.

A segunda aba foi relacionada a análise comparativa entre os processos, sendo que os índices IAE, ISE, ITAE e ITSE foram representados. A terceira aba faz uma comparação entre as etapas estacionárias dos processos, ou seja, indicando o *boxplot*, a média, a variância, o desvio padrão, a variabilidade e os índices de capacidade (CP e CPK).

Finalizada a construção do *dashboard*, o resultado foi publicado no serviço do *software Power BI* e o seu acesso pode ser feito através do link: <https://app.powerbi.com/view?r=eyJrIjoiMWFIMTkyNmItMGNI00Y2E2LWFkNzctZjMzMTE3NWY2ODIyIiwidCI6ImZlODc4N2JjLWw5MTQtNDY2NS04NTQ3LTl2OGUxNWNiMGQ5YSJ9>.

5 RESULTADOS E DISCUSSÃO

Nesta seção serão apresentados inicialmente os resultados obtidos no *Power BI* por meio do *dashboard*. Por fim, será apresentada uma discussão a respeito dos resultados obtidos a fim de entender os aspectos positivos e os pontos de melhoria em cada um dos processos e etapas analisadas.

5.1 RESULTADOS

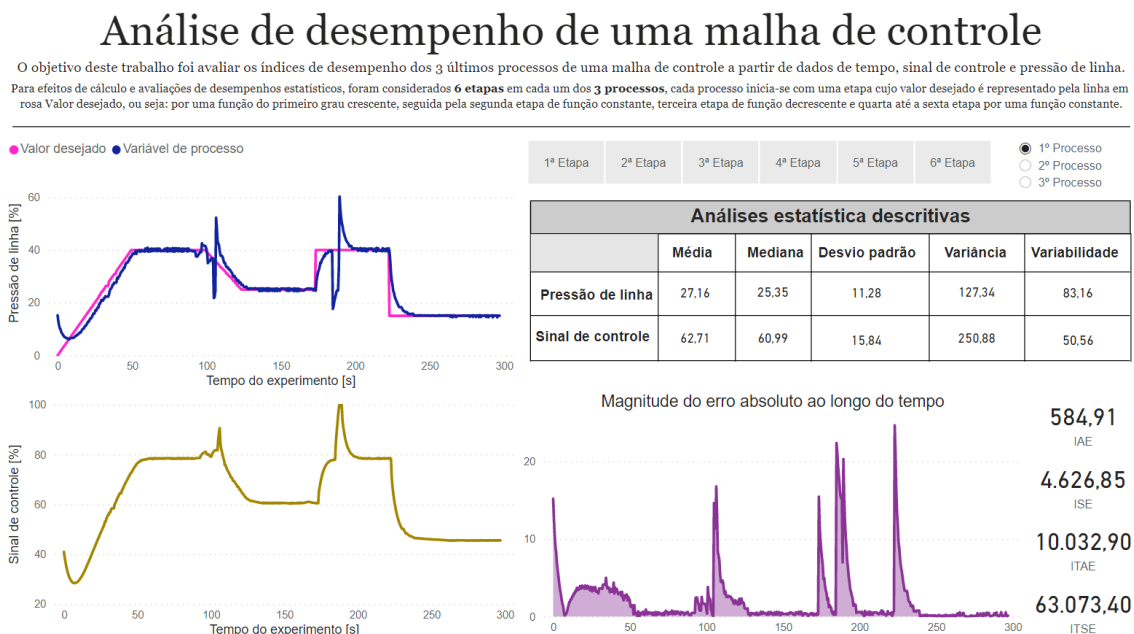
Nessa subseção serão explicadas as três abas do *dashboard* em detalhe, assim como será descrito o conteúdo contido em cada parte do visual. A 1ª aba apresenta um *overview* de cada um dos três processos (2 processos com distúrbios em duas etapas distintas e 1 processo sem distúrbios), além da possibilidade de analisar cada uma das etapas; a 2ª aba dedica-se a análise comparativa entre os processos e a 3ª aba apresenta uma análise comparativa entre as etapas estacionárias dos processos.

5.1.1 VISÃO GERAL DOS PROCESSOS COM E SEM DISTÚRBIOS ANALISADOS

A primeira aba do *dashboard* apresenta todas as medidas calculadas por processo, exceto o índice de capacidade retratado na última aba. Através da primeira aba observa-se que é possível analisar apenas um processo por vez ao selecionar o processo desejado no filtro superior à direita, além de ser possível escolher as etapas de acordo com a necessidade.

A Figura 23 apresenta a primeira aba do *dashboard* para as medidas do 1º processo indicadas por um gráfico de pressão de linha por tempo de experimento para o valor desejado e para a variável de processo, outro gráfico de sinal de controle por tempo de experimento e mais um gráfico demonstrando a magnitude do erro absoluto ao longo do tempo para este processo. A figura ainda indica o filtro relacionado a seleção da etapa (1ª, 2ª, 3ª, 4ª, 5ª ou 6ª), o filtro referente ao processo (1º, 2º ou 3º), a tabela contendo as análises estatísticas descritivas para a pressão de linha e sinal de controle (média, mediana, desvio padrão, variância e variabilidade) e os quatro índices estatísticos (IAE, ISE, ITAE, ITSE).

Figura 23 - Medidas calculadas para o 1º processo (com distúrbios na 3ª e 5ª etapa).



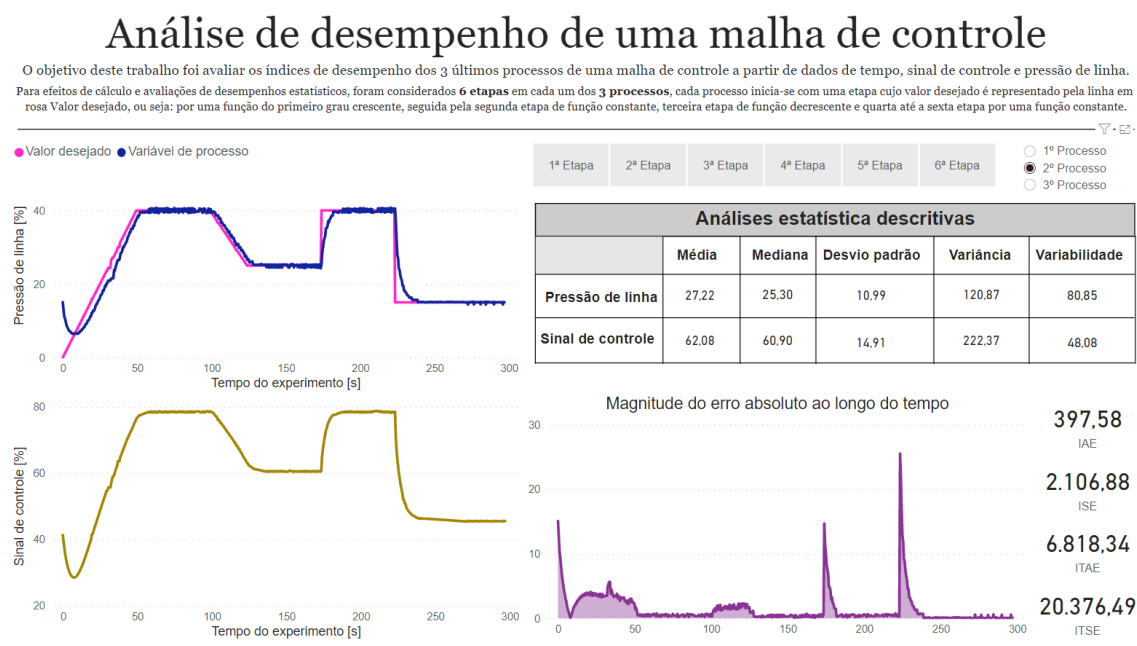
Fonte: Autoria própria.

É possível observar na Figura 23 dois distúrbios causados pela abertura de uma das válvulas manuais: o primeiro ocorre na 3ª etapa enquanto o segundo ocorre na 5ª etapa.

Para a Figura 24 e Figura 25 relativas à primeira aba do *dashboard* para o 2º e 3º processo respectivamente as medidas indicadas são as mesmas do 1º processo conforme a Figura 23, ou seja: gráfico de pressão de linha por tempo, gráfico de sinal de controle por tempo, gráfico da magnitude do erro absoluto pelo tempo, filtro para seleção de etapa, filtro para seleção de processo, análises estatísticas descritivas e índices estatísticos.

Assim a Figura 24 demonstra a primeira aba do *dashboard* para as medidas do 2º processo.

Figura 24 - Medidas calculadas para o 2º processo (sem distúrbios).



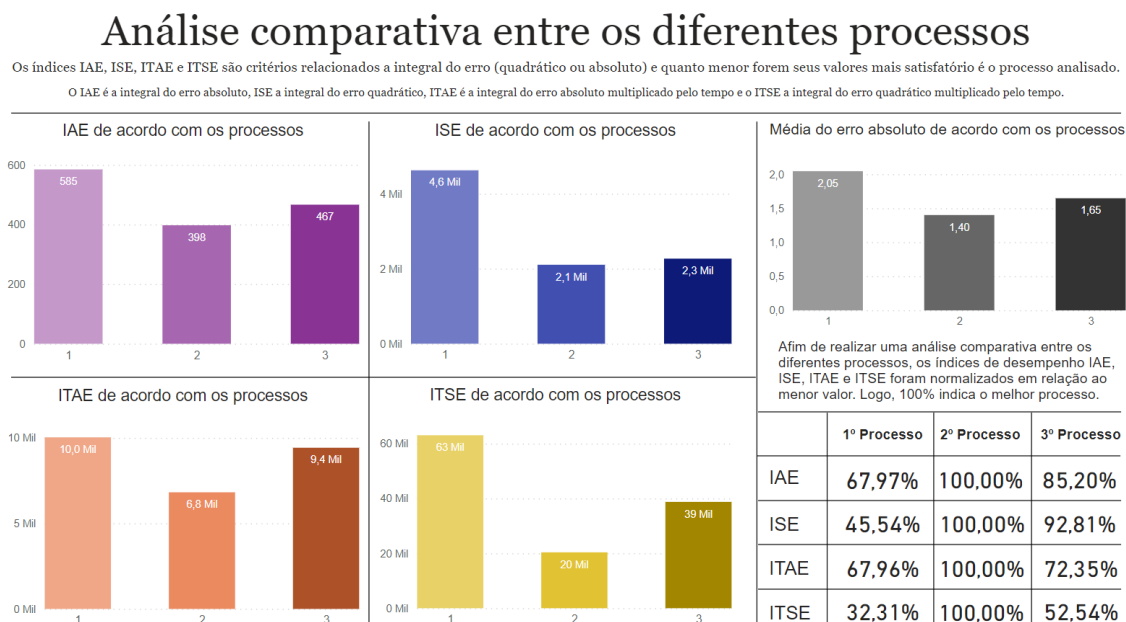
Já através da Figura 25 são constatados dois distúrbios: o primeiro ocorre na 4ª etapa enquanto o segundo distúrbio na 6ª etapa.

Ademais, é possível observar que os maiores erros são respectivos às transições em degrau, no início do processo, devido à influência do processo anterior, e nas transições em rampa. Os quatro cartões à direita do gráfico do erro trazem informações da integral do erro absoluto (IAE), da integral do erro ao quadrado (ISE), da integral do erro absoluto pelo tempo (ITAE) e da integral do erro quadrático pelo tempo (ITSE).

5.1.2 ANÁLISE COMPARATIVA DOS PROCESSOS

A segunda aba do *dashboard* apresenta uma análise comparativa dos índices IAE, ISE, ITAE e ITSE para os três processos, conforme indicado na Figura 26. Nela é apresentado 5 gráficos de barra, sendo os 4 gráficos relacionados a integral do erro na parte esquerda e 1 gráfico da média do erro absoluto na parte superior a direita, além de uma tabela com os valores normalizados referente às medidas da integral do erro, localizado na parte inferior à direita. Vale ressaltar que o índice de 100% representa o melhor desempenho para o parâmetro analisado.

Figura 26 - Análise comparativa entre os três processos.



Fonte: Autoria própria.

Observando os gráficos e a tabela da Figura 26 é possível notar que o melhor desempenho foi respectivo ao 2º processo, seguido pelo 3º processo e, por último, o 1º processo em todos os parâmetros analisados. Para o IAE, o 3º processo, apesar dos distúrbios, apresentou valores satisfatórios quando comparado com o 2º processo, enquanto o 1º processo foi o pior de todos.

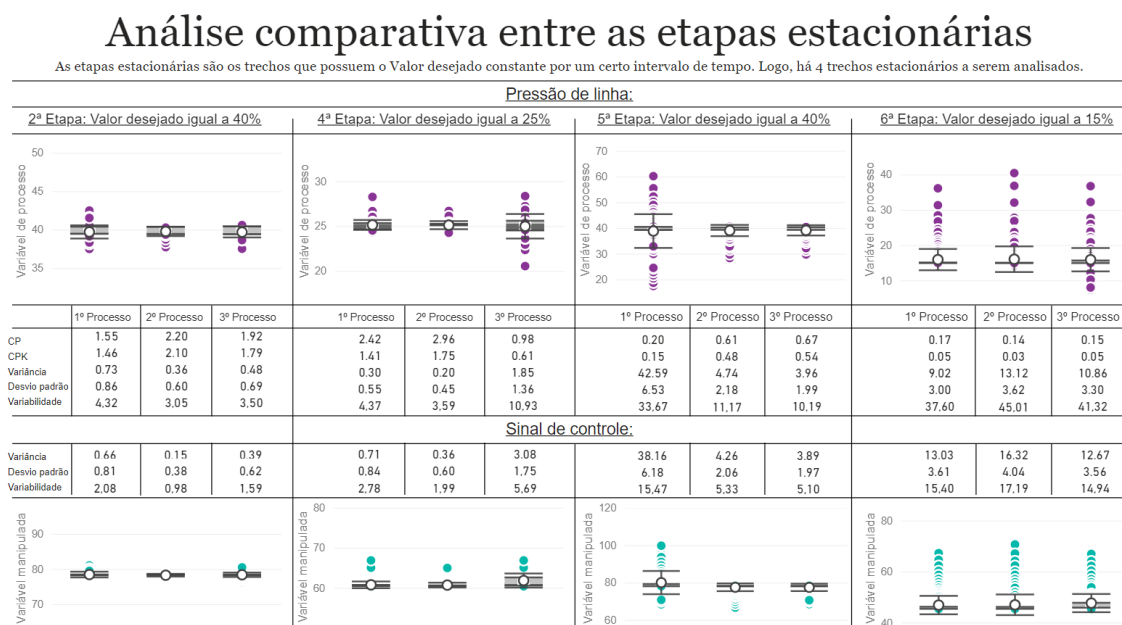
Já no que se refere ao ISE, o 3º processo apresentou valores muito próximos ao 2º processo, com 92.81%, valor mais próximo ao 2º processo entre todos os índices apresentados. Quando comparado o 1º com o 2º processo, percebe-se que o 1º processo não foi capaz de atingir nem a metade do valor do 2º processo, demonstrando um resultado insatisfatório.

Com relação ao ITAE, é possível observar que foi o índice no qual o 1º e 3º processos foram mais próximos, apresentando uma diferença próximo a 5%. Ademais, o 2º processo foi em torno de 1/3 melhor que o 1º processo. Já no último índice relacionado à integral do erro analisado, o ITSE, foi o que apresentou os piores resultados para o 1º e 3º processo, uma vez que penaliza gravemente erros maiores do que um.

O 2º processo foi em torno de duas vezes melhor que o 3º processo e três vezes melhor que o 1º processo. Outrossim, o gráfico da média do erro absoluto corrobora que o 2º processo apresentou o melhor desempenho.

5.1.3 ANÁLISE COMPARATIVA DAS ETAPAS ESTACIONÁRIAS

Por fim, a última aba do *dashboard* apresenta a análise comparativa dos trechos estacionários de cada processo, tanto para a pressão de linha como para o sinal de controle, conforme Figura 27. Logo é possível observar que o visual apresenta 8 gráficos *boxplot* e duas tabelas. Os 4 gráficos superiores e a tabela acima representam os valores para a pressão de linha, enquanto os gráficos e a tabela inferior os resultados para o sinal de controle.

Figura 27 - Análise comparativa das etapas estacionárias de cada processo.

Fonte: Autoria própria.

Analisando os índices CP e CPK é possível observar que os seus valores apresentaram resultados insatisfatórios para as etapas 5 e 6 de todos os processos. Na 2ª etapa, trecho com valor desejado igual a 40%, todos os processos apresentaram valores satisfatórios para os índices CP e CPK. Já na 4ª etapa, o 3º processo apresentou valores insatisfatórios para os índices, devido a uma oscilação nesse trecho. Outro aspecto interessante de ser observado é que o 1º processo, mesmo após sofrer uma grande oscilação na 3ª etapa, apresentou resultados satisfatórios para os parâmetros CP e CPK.

A variância e o desvio padrão, tanto da pressão de linha como do sinal de controle, da 2ª etapa de todos os processos apresentou resultados com valores abaixo de um, indicando poucas oscilações nesse trecho. Os melhores processos da 2ª etapa foram, respectivamente, os processos 2º, 3º e 1º.

Na 4ª etapa, os melhores indicadores foram respectivos ao 2º processo. O 3º processo sofreu uma forte oscilação nesse período e apresentou resultados ruins para a variância, desvio padrão e variabilidade. O 1º processo apresentou bons resultados nesse trecho, com valores muito próximos ao 2º processo.

O 3º processo foi o que apresentou os melhores indicadores na 5ª etapa. Apesar de todos os processos apresentarem mal desempenho para o CP e CPK nesse trecho, os demais índices se mostraram bons para esse processo. O 2º processo apresentou resultados semelhantes ao 3º processo. Já o 1º processo foi o que apresentou os piores valores devido a grande oscilação presente nesse trecho.

5.2 DISCUSSÃO

Ao comparar os três processos, através da Figura 23, 24 e 25, é possível notar que os piores desempenho ocorreram no primeiro e terceiro processo. No 1º processo ocorrem dois grandes distúrbios provocados pela abertura de uma das válvulas manuais, sendo na 3ª e 5ª etapa do processo. No 3º processo ocorrem dois pequenos distúrbios, sendo um na 4ª etapa e outro na 6ª etapa. Já o 2º processo não indicou nenhum distúrbio ao longo do tempo, sendo assim o melhor processo analisado.

Outro fato interessante a ser analisado é em relação aos índices IAE, ISE, ITAE e ITSE, conforme Figura 26. Nota-se que todos os índices apresentaram resultados mais satisfatórios para o 2º, 3º e 1º processo, respectivamente. Porém, não necessariamente isso acontecerá sempre. Pode haver casos em que alguns índices são melhores para um processo e outros índices piores quando comparado aos demais.

Observando a Figura 27, na última etapa analisada os três processos apresentaram péssimos indicadores, e neste trecho os índices dos diferentes processos apresentaram resultados que mais se assemelham. Em alguns parâmetros o 1º processo foi melhor, em outros o 2º processo foi melhor e em outro indicador, como a variabilidade do sinal de controle, o último processo foi o melhor.

Além disso, outro fator que merece destaque é a recuperação do processo (de acordo com o valor desejado) nas etapas subsequentes aos distúrbios provocados. Observando as Figuras 23 e 27, é possível notar que o 1º processo, mesmo após sofrer uma grande oscilação na 3ª etapa, apresentou resultados satisfatórios para os parâmetros CP e CPK na 4ª etapa, indicando boa recuperação do processo.

A escolha dos limites para o cálculo do índice de capacidade também merece ser discutida. Conforme mencionado anteriormente, a escolha do limite de 10% do valor alvo, foi apenas para efeito de cálculos e com o intuito de apresentar uma análise

comparativa. Dessa forma, os limites devem ser muito bem escolhidos tendo em vista as regras de negócio e as normas industriais vigentes.

Uma das limitações presentes neste trabalho é o fato de não haver um mecanismo de sincronismo entre a execução dos ciclos e o armazenamento de dados. Ou seja, para a coleta dos dados é necessário iniciar o serviço do banco de dados antes de executar a malha de controle pelo *LabVIEW*. Assim, os dados iniciais coletados no *LabVIEW* antes da execução dos processos são irrelevantes para a análise.

Outra limitação do trabalho é na limpeza das variáveis de controle caso o botão de emergência seja acionado. Ou seja, ao acionar o botão de emergência e, posteriormente, selecionar uma malha de controle para ser executada, o valor desejado não iniciará com o valor zero, mas sim com valor em relação ao último ciclo de execução antes da parada de emergência.

Além disso, uma das principais limitações no que refere-se à análise dos processos é o fato de ignorar os ciclos que não possuem 595 amostras. Logo, esses processos são ignorados e nenhum tipo de tratamento dos dados foi apresentado. Portanto, para os trabalhos futuros recomenda-se apresentar alguma solução para essas questões que não foram contempladas nesse estudo.

Ademais, nota-se que o trabalho atendeu ao tema proposto sendo possível realizar uma análise de desempenho dos processos executados na planta piloto. Destaca-se que o processo de análise é realizado de forma automática. Ou seja, caso novos ciclos sejam executados o *dashboard* é atualizado com novos valores.

6 CONCLUSÃO

Através dos KPIs utilizados foi possível mensurar distúrbios nos processos e identificar o mais eficiente entre os três últimos ciclos executados na planta piloto. Isso pode ser observado na 2ª aba do *dashboard* (através da comparação dos índices IAE, ISE, ITAE e ITSE), onde o 2º processo apresentou-se como o mais satisfatório, seguido pelo 3º e 1º processo. Ressalta-se que ambos os índices apresentados nesse estudo devem ser levados em consideração ao avaliar os processos, sendo que a análise de apenas um parâmetro pode ser inconclusivo e levar a mal entendimento dos fatos.

Como acontece se apenas o índice de capacidade para a 5ª e 6ª etapa de todos os processos analisados fossem analisados. Ele indicaria essas etapas como incapazes de atender aos limites de especificação. Destaca-se algumas soluções para o índice de capacidade nessas etapas: alterar os limites de tolerância nesse trecho; ajustar o cálculo do parâmetro de maneira a ignorar alguns instantes iniciais das etapas subsequentes a mudança no valor desejado em degrau (5ª e 6ª etapa); ajustar os parâmetros do controlador.

Outro fato relevante observado no trabalho é a recuperação da malha de controle em estabelecer a pressão de linha de acordo com o valor desejado. Sendo observado na 4ª etapa do 3º processo, trecho com valor desejado igual a 25%, onde esse trecho apresenta bons resultados dos KPIs mesmo após a ocorrência de distúrbios na etapa anterior.

Vale destacar que mais índices poderiam ser propostos a fim de complementar os resultados apresentados. Parâmetros preditivos a fim de diagnosticar possíveis desgastes nos elementos de controle, sensores e atuadores trariam uma complementação maior ao estudo, por exemplo. Logo, uma sugestão a trabalhos futuros seria investigar a manutenção preditiva dos elementos da planta piloto.

Destaca-se também a dificuldade em encontrar índices capazes de avaliar todo o processo de forma eficiente e conclusiva. Talvez encontrar mais índices relacionados a problemas na área de negócio e a aplicabilidade seria interessante.

Ademais, a automatização da análise de todos os processos com 595 amostras é outro ponto que deve ser destacado, uma vez que possibilita a execução contínua de

inúmeros processos na malha de controle e somente os últimos três serão avaliados e comparados entre si.

Portanto, conclui-se que o *Power BI* apresenta-se como uma boa ferramenta de visualização de análise de processos industriais. Ressalta-se porém algumas limitações na manipulação dos dados, tratamento e limpeza. Observa-se que uma ferramenta de *pipeline* de dados poderia ter sido utilizada como elemento intermediário entre o *MySQL* e o *Power BI* a fim de tratar os dados com faixas amostrais distintas. Dessa forma, a ferramenta da *Microsoft* seria utilizada apenas como elemento final na visualização dos dados e o processamento das informações realizado em outro *software*.

REFERÊNCIAS

AHLÉN, A., et al. **Toward wireless control in industrial process automation: A case study at a paper mill.** IEEE Control Systems Magazine, v. 39, n. 5, p. 36-57, 2019.

AHMAD, A. **Business intelligence for sustainable competitive advantage.** In: Sustaining competitive advantage via business intelligence, knowledge management, and system dynamics. Emerald Group Publishing Limited, 2015.

ALON. **PRÉ-REQUISITOS DO POWER BI: PRECISO SABER EXCEL?** 2020. Disponível em: <<https://www.hashtagtreinamentos.com/pre-requisitos-do-power-bi>>. Acessado em : 05 de jul. 2022.

ASAITHAMBI, S. P. R.; VENKATRAMAN, R.; VENKATRAMAN, S. **MOBDA: Microservice-oriented big data architecture for smart city transport systems.** Big Data and Cognitive Computing, v. 4, n. 3, p. 17, 2020.

BASILE, L. J. et al. **Business intelligence in the healthcare industry: The utilization of a data-driven approach to support clinical decision making.** Technovation, p. 102482, 2022.

BECKER, L.T.; GOULD, E. M. **Microsoft power BI: extending excel to manipulate, analyze, and visualize diverse data.** Serials Review, v. 45, n. 3, p. 184-188, 2019.

BIGHETI, J. A.; FERNANDES, M. M.; GODOY, E. P. **Control as a Service: A Microservice Approach to Industry 4.0.** 2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019 - Proceedings. Anais...Institute of Electrical and Electronics Engineers Inc., 1 jun. 2019.

BIGHETI, J. A.; et al. **Proposta De Arquitetura Orientada a Microserviços Para Aplicações De Internet Das Coisas Industrial.** Proceedings XXII Congresso Brasileiro de Automática, 2018.

CHOU, L. **Top 10 Key Features of BI Tools in 2020.** 2020. Disponível em:<<https://towardsdatascience.com/top-10-key-features-of-bi-tools-in-2020-fd02c1dd4c71>>. Acessado em: 05 de jul. 2022.

CODE. **Code editing.** Empresa Microsoft. 2022. Disponível em: <<https://code.visualstudio.com/>>. Acessado em: 30 de jun. 2022.

DB-ENGINES. **DB-Engines Ranking of Time Series DBMS**. 2022. Disponível em: <<https://db-engines.com/en/ranking/time+series+dbms>>. Acessado em: 30 de jun. 2022.

DIAS, V. **Automação rompe limites entre digital, físico e biológico**. Jornal da USP. Jornal da USP. São Paulo, SP, BrasilTecnologia, , 2018. Disponível em: <jornal.usp.br/?p=152479>. Acessado em: 17 de jun., 2022.

EIDIZADEH, R.; SALEHZADEH, R.; ESFAHANI, A.C. **Analysing the role of business intelligence, knowledge sharing and organisational innovation on gaining competitive advantage**. Journal of Workplace Learning, v. 29, n. 4, p. 250-267, 2017.

ESPINEL VELASCO, E. F. **Análisis de los datos de perforación del Banco de Información Petrolera del Ecuador usando Microsoft Power Bi Desktop**. 2022. Trabalho de Conclusão de Curso. Quito: UCE.

FERNANDES, M. **Banco de Dados Relacional (SQL) e Não Relacional (NoSQL) – O que são, para que servem e qual a diferença?** 2020. Disponível em: <<https://marquesfernandes.com/tecnologia/banco-de-dados-relacional-sql-e-nao-relacional-nosql-o-que-sao-para-que-servem-e-qual-a-diferenca/>>. Acessado em: 25 de jun. 2022.

FIELD, A.; MILES, J.; FIELD, Z. **Discovering Statistics Using R**. 2012. Great Britain: Sage Publications, Ltd, v. 958.

FIELDING, T. **Architectural Styles and the Design of Network-based Software Architectures**. Irvine, EUA: University of California, 2000.

FONSECA, M. O.; TORRES, B. S.; SEIXAS FILHO, C. **Avaliação de desempenho e auditoria de malhas de controle**. Revista InTech Brasil, ano VI, n. 63, p. 32-37, 2004.

GARCIA, C., **CONTROLE DE PROCESSOS INDUSTRIAIS, Estratégias convencionais**, volume 1, cap. 5, pág. 250-305, 2017.

GARCIA, Claudio. **Controle de processos industriais: estratégias convencionais**. Editora Blucher, 2021.

GETTI. **Entenda os motivos para você investir em um bom banco de dados**. 2018.

GetTI Services. Disponível em:

<<https://getti.net.br/2018/07/19/entenda-os-motivos-para-voce-investir-em-um-bom-banco-de-dados/#:~:text=As%20principais%20vantagens%20do%20banco,permitir%20o%20uso%20em%20camadas>> . Acessado em: 02 de jul. 2022.

HART, M. et al. **O que é o PowerBI?** 2022. Microsoft. Disponível em:

<<https://docs.microsoft.com/pt-br/power-bi/fundamentals/power-bi-overview>>.

Acessado em 24 de jun. de 2022.

JASPERNEITE, J.; SAUTER, T.; WOLLSCHLAEGER, M. **Why we need automation models: handling complexity in industry 4.0 and the internet of things**. IEEE

Industrial Electronics Magazine, v. 14, n. 1, p. 29-40, 2020.

KAGERMANN; WAHLSTER, W.; HELBIG, J. **Recommendations for implementing the strategic initiative INDUSTRIE 4.0**. Final report of the Industrie 4.0 WG, n. April, 2013.

KANE, V. E. “**Process Capability indices**”, Journal of Quality Technology, Vol. 18. Nº. 1, p. 41-52, 1986.

KLEMES, J. **Sustainability in the process industry: integration and optimization**.

McGraw-Hill Education, 2011.

KRONZ, A. et al. **Magic Quadrant for Analytics and Business Intelligence Platforms**. 2022. Gartner. Disponível em:

<<https://www.gartner.com/doc/reprints?id=1-292LEME3&ct=220209&st=s>>. Acessado em: 30 de jun. 2022.

LAGO, K.; ALVES, L. **Dominando o Power BI**. [S.l.]: DATAB, 2018. v. 1

LENON. **Node.js – O que é, como funciona e quais as vantagens**. 2018. Opus

Software. Disponível em: <<https://www.opus-software.com.br/node-js/>>. Acessado em: 02 de jul. de 2022.

LIPTAK, B. G. **Instrument Engineers' Handbook**, Volume 2: Process Control Optimization, Fourth Edition, Taylor & Francis Group, 2006.

LORENZI, L. **Power BI: o que é, por que usar e quais as vantagens**. 2022. Incidium Tech. Disponível em:

<<https://blog.indicium.tech/power-bi-o-que-e-vantagens/#:~:text=O%20Power%20BI%20%C3%A9%20uma,mais%20baixos%20para%20as%20empresas>>. Acessado em: 03 de jul. 2022.

MATHEUS, R. F.; PARREIRAS, F. S. **Inteligência empresarial versus business intelligence: abordagens complementares para o apoio à tomada de decisão no Brasil**. São Paulo: Sociedade Brasileira de Gestão do Conhecimento, 2004.

MICROSOFT. **Entenda o esquema em estrela e a importância para o Power BI**. Disponível em: <<https://docs.microsoft.com/pt-br/power-bi/guidance/star-schema>>. Acessado em: 18 de jun. 2022

MISAL, D. A **Look at the evolution of BI platform over the years**. 2019.. Disponível em: <<https://analyticsindiamag.com/a-look-at-the-evolution-of-bi-platform-over-the-years/>>. Acessado em: 05 de jul. 2022.

MIKROYANNIDIS, A.; THEODOULIDIS, B. **Ontology management and evolution for business intelligence**. [S.l.]: International Journal of Information Management, 2010

MOLECULER. **Fast & powerful microservices framework for Node.js**. Disponível em: <<https://moleculer.services/>>. Acesso em: julho, 2020.

MOYNE, J. et al. **A requirements driven digital twin framework: Specification and opportunities**. IEEE Access, v. 8, p. 107781-107801, 2020.

MySQL. **Why MySQL?** Disponível em: <<https://www.mysql.com/why-mysql/>>. Acessado em: 30 de jun. 2022

NATS. **Welcome to the Official NATS Documentation**. Disponível em: <<https://docs.nats.io/>>. Acesso em: junho, 2022

NARCISO-FILHO, P. **Capacidade e Performance: entenda os índices Cp, Cpk, Pp e Ppk**. 2017. HarboR. Disponível em; <<https://www.harbor.com.br/harbor-blog/2017/07/06/capacidade-performance-significa>>

do/#:~:text=A%20interpreta%C3%A7%C3%A3o%20dos%20valores%20do,continue%20a%20fazer%20no%20futuro>. Acessado em: 27 de jun. 2022.

NATIONAL INSTRUMENTS. **What's new in LabVIEW?** 2022. National Instruments. Disponível em: <<https://www.ni.com/pt-br/shop/labview.html>>. Acessado em: 28 de jun. de 2022.

OLIVEIRA, B. **Boxplot: como interpretar?** OperData, 2019. Disponível em: <<https://operdata.com.br/blog/como-interpretar-um-boxplot/>>. Acesso em: setembro, 2021.

ORACLE. **Oracle Hardware**. 2022. Disponível em: <<https://www.oracle.com/it-infrastructure/>>. Acessado em: 02 de jul. 2022.

PETERSON, A.A.; et al. **Thermochemical biofuel production in hydrothermal media: a review of sub-and supercritical water technologies**. Energy & environmental science, v. 1, n. 1, p. 32-65, 2008.

PONTAROLLI, R.P. et al. **Automação e Controle de Processos usando Microserviços: Uma solução para a aplicação da Indústria 4.0**. Brazilian Journal of Development, v. 7, n. 3, p. 23249-23268, 2021.

PRIMAK, F.V. **Decisões com BI (business intelligence)**. Fabio Vinicius Primak, 2008. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=cGBneHPeLvKC&oi=fnd&pg=PA11&dq=business+intelligence&ots=14cP7Zr_ol&sig=kn72yLTKJliPozdjq9ODcFnW4YE#v=onepage&q=business%20intelligence&f=false>. Acesso em: julho, 2020.

SÁ, M. P. B. d. et al. **Lei de benford e regras de associação no Power BI: ferramentas estatísticas aplicadas à auditoria**. Universidade Federal da Paraíba, 2020.

SALENIUS, M. **Designing A Data Solution For A Game Company – Case InfluxDB and Grafana**. 2017. Disponível em: <<https://www.theseus.fi/handle/10024/137958>>. Acessado em: 01 de jul. 2022

SANTOS, C.F.V., et al. **CONTROLES AUTOMÁTICOS APLICADOS À FABRICAÇÃO DO AÇÚCAR CRISTAL: LEVANTAMENTO DO ESTADO DA ARTE**. Ciência & Tecnologia, v. 6, n. 1, 2014.

SHARMA, R.; MITHAS, S.; KANKANHALLI, A. **Transforming decision-making processes: a research agenda for understanding the impact of business analytics on organisations**. European Journal of Information Systems, v. 23, n. 4, p. 433-441, 2014.

SCHWAB, K. **The Fourth Industrial Revolution: what it means and how to respond**. World Economic Forum, p. 1–7, 2016.

SCHUMACHER, A.; SIHN, W. **Development of a monitoring system for implementation of industrial digitalization and automation using 143 key performance indicators**. Procedia CIRP, v. 93, p. 1310-1315, 2020.

SHARABI, Kesem. **Perguntas frequentes do Power BI Premium**. 2022. Microsoft.

Disponível em:

<<https://docs.microsoft.com/pt-br/power-bi/enterprise/service-premium-faq>>. Acessado em: 01 de jul. de 2022.

SILVEIRA, M. **O que é a Indústria 4.0 e como ela vai impactar o Mundo**. Acesso em 15 de julho de 2020.

SMART CONSULTING. **O que é o Quadrante Mágico de Gartner?** SMART

Consulting. 2022. Disponível em:

<<https://blog.smartconsulting.com.br/quadrante-magico-gartner/>>. Acessado em: 01 de jun. 2021.

TAO, F. et al. **Data-driven smart manufacturing**. Journal of Manufacturing Systems, v. 48, p. 157-169, 2018.

TAVARES, P. **Logística Lean Aplicando: as ferramentas lean na cadeia de suprimentos para gestão e geração de valor**, ebook, 2018. p. 35-70.

ZHOU, C. et al. **A data-driven business intelligence system for large-scale semi-automated logistics facilities**. International Journal of Production Research, v. 59, n. 8, p. 2250-2268, 2021.