



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE DE AUTOMAÇÃO

VINICIUS CAMPELO PIERRI

AUTOMATIZAÇÃO DO MONITORAMENTO DO CONSUMO DE GÁS NATURAL EM UMA
PLANTA INDUSTRIAL USANDO TECNOLOGIAS DE CÓDIGO ABERTO

Recife
2025

VINICIUS CAMPELO PIERRI

AUTOMATIZAÇÃO DO MONITORAMENTO DO CONSUMO DE GÁS NATURAL EM UMA
PLANTA INDUSTRIAL USANDO TECNOLOGIAS DE CÓDIGO ABERTO

Trabalho de Conclusão do Curso de
Engenharia de Controle e Automação da
Universidade Federal de Pernambuco,
como requisito da disciplina de Trabalho de
Conclusão de Curso (EL403)

Orientador(a): Prof. Dr. Douglas Contente Pimentel Barbosa

Recife
2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Pierri, Vinicius Campelo .

Automatização do monitoramento do consumo de gás natural em uma planta industrial usando tecnologias de código aberto / Vinicius Campelo Pierri. - Recife, 2025.

73 p : il., tab.

Orientador(a): Douglas Contente Pimentel Barbosa

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e Automação - Bacharelado, 2025.

Inclui referências, apêndices.

1. Automação industrial. 2. Monitoramento energético. 3. Código aberto. 4. InfluxDB. 5. Grafana. I. Barbosa, Douglas Contente Pimentel . (Orientação). II. Título.

620 CDD (22.ed.)

VINICIUS CAMPELO PIERRI

AUTOMATIZAÇÃO DO MONITORAMENTO DO CONSUMO DE GÁS NATURAL EM UMA
PLANTA INDUSTRIAL USANDO TECNOLOGIAS DE CÓDIGO ABERTO

Trabalho de Conclusão do Curso de
Engenharia de Controle e Automação da
Universidade Federal de Pernambuco,
como requisito da disciplina de Trabalho de
Conclusão de Curso (EL403)

Aprovado em: 11/04/2025.

BANCA EXAMINADORA

Prof. Dr. Douglas Contente Pimentel Barbosa (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Rafael Cavalcanti Neto (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Davidson da Costa Marques (Examinador Interno)
Universidade Federal de Pernambuco

Este trabalho é dedicado a Deus e a minha família.

RESUMO

Este trabalho propõe e implementa uma solução integrada para automatizar o monitoramento do consumo de gás natural em uma planta industrial, utilizando ferramentas de código aberto. A proposta surge da necessidade de substituir um processo manual, suscetível a falhas humanas e pouco eficiente, por um sistema automatizado capaz de coletar, armazenar e visualizar dados em tempo real. A arquitetura desenvolvida integra dispositivos industriais, como CLPs e *dataloggers*, por meio dos protocolos MQTT e HTTP, com os dados sendo armazenados em um banco de dados *InfluxDB* e visualizados por meio da plataforma *Grafana*. O sistema proposto possibilita a centralização das informações, melhora a confiabilidade dos dados e fornece suporte à tomada de decisões baseada em dados. Os resultados obtidos demonstram ganhos significativos em eficiência operacional e acessibilidade da informação. A aplicação mostrou-se escalável e flexível, evidenciando o potencial das tecnologias abertas na modernização de processos produtivos.

Palavras-chave: Automação industrial; Monitoramento energético; Código aberto; *InfluxDB*; *Grafana*.

ABSTRACT

This work proposes and implements an integrated solution to automate the monitoring of natural gas consumption in an industrial plant using open-source technologies. The proposal addresses the need to replace a manual, error-prone, and time-consuming process with an automated system capable of collecting, storing, and visualizing data in real time. The developed architecture integrates industrial devices such as PLCs and dataloggers through the protocols MQTT and HTTP. Data is stored in an InfluxDB time-series database and visualized via the Grafana platform. The proposed system enables centralized data access, improves data reliability and supports data-driven decision-making. The results demonstrate significant improvements in operational efficiency and information accessibility. The solution proved to be scalable and flexible, highlighting the potential of open technologies in modernizing industrial processes.

Keywords: Industrial automation; Energy monitoring; Open source; InfluxDB; Grafana.

LISTA DE ILUSTRAÇÕES

Figura 1 : Fluxo de gás natural.....	15
Figura 2 : Medidor de vazão do tipo vórtex.	20
Figura 3 : Turbina para líquidos, com retificador de fluxo incorporado.....	21
Figura 4: Representação 3-way <i>handshake</i>	23
Figura 5: Formato geral de uma mensagem de requisição <i>HTTP</i>	24
Figura 6 : Exemplo de aplicação com <i>MQTT</i>	25
Figura 7 : Sequência de comunicação do protocolo <i>MQTT</i>	26
Figura 8 : Duas trajetórias para a temperatura de um certo local num dia.	29
Figura 9 : Visão superficial de um SGBD e seus relacionamentos.	31
Figura 10 : Comparação do tempo de latência de consultas entre <i>TimescaleDB</i> e <i>PostgreSQL 14.4</i>	34
Figura 11: Exemplo de dashboard com o <i>Grafana</i>	38
Figura 12 Fluxo do Power BI.....	39
Figura 13: Editor do <i>Node-RED</i> como uma ferramenta de programação baseada em fluxos.....	40
Figura 14: Interface do <i>Node-RED</i>	41
Figura 15 : Diagrama do fluxo de dados.	42
Figura 16: Transmissão de vazão de gás da Copergás.	44
Figura 17: Tubulação de distribuição de vapor.....	44
Figura 18: Equipamentos disponíveis na indústria.	46
Figura 19 : Realização de 3 requisições <i>HTTP</i> utilizando <i>Node-RED</i> e <i>Python</i>	48
Figura 20: Instalação do <i>broker</i> no <i>Node-RED</i>	51
Figura 21 : Bloco de <i>broker MQTT</i> no <i>Node-RED</i>	51
Figura 22: Fluxograma de publicação de dados <i>CLP</i> da linha A.....	52
Figura 23 : Configuração do IP do <i>broker</i> e da identificação do <i>CLP</i>	52
Figura 24 : Nó <i>MQTT</i> inscrito no tópico do totalizador de gás natural do processo da linha 2.....	53
Figura 25 : Configuração do nó de entrada <i>MQTT</i>	53
Figura 26: Configuração da entrada analógica para receber sinal do transmissor de vazão de vapor da linha B.....	55

Figura 27: Configuração da entrada analógica para receber sinal do transmissor de vazão de vapor da linha A.	55
Figura 28: Configuração de comunicação MQTT do <i>Logbox</i>	56
Figura 29: Configuração do nó de recebimento no <i>Node-RED</i>	56
Figura 30: <i>Url</i> para obtenção de token e autenticação na documentação da Copergás.	57
Figura 31: <i>Url</i> para consulta de variáveis mensuradas pela Copergás.	57
Figura 32: Nós para leitura de variáveis da Copergás.	58
Figura 33: Configuração da requisição do tipo <i>GET</i> para obter medida da Copergás.	58
Figura 34: Nós para comunicação com servidor com <i>InfluxDB</i>	59
Figura 35: Integração entre fonte de dados e o banco de dados.	60
Figura 36: Opções para fontes de dados no <i>Grafana</i>	61
Figura 37: Adicionando endereço do banco de dados.	61
Figura 38: Adicionando a base de dados no <i>Grafana</i>	61
Figura 39: Consulta gráfica no <i>Grafana</i>	62
Figura 40 : Diagrama da solução final.	65
Figura 41: Gráficos do consumo de vapor no <i>Grafana</i>	66
Figura 42: Gráfico do consumo de gás no <i>Grafana</i>	66
Figura 43: Exportando dados pelo <i>Grafana</i>	67
Figura 44: Autenticação no <i>Grafana</i>	67
Figura 45 : Código em Ladder desenvolvido para publicação de mensagem MQTT referente ao totalizador de gás natural usado no processo da linha A.	73

LISTA DE TABELAS

Tabela 1 : Comparação entre envio e recebimento de dados sequenciais entre dispositivos e um servidor.	27
Tabela 2: Códigos de função do protocolo <i>Modbus</i>	28
Tabela 3: Interesses comuns no estudo de series temporais.....	30
Tabela 4 - Bancos de dados integrados ao Grafana.	37
Tabela 5: Parâmetros para envio de dados do CLP para o <i>Node-RED</i>	53
Tabela 6: Tipos de gráfico no <i>Grafana</i>	63

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
CLP	Controlador Lógico Programável
CSV	<i>Comma Separated Values</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IOT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
SGBD	Sistema de Gerenciador de Banco de Dados
SQL	<i>Sequencial Query Language</i>
TCP	<i>Transmission Control Protocol</i>
TI	Tecnologia da Informação
URL	<i>Uniform Resource Locator</i>
WSN	<i>Wireless Sensor Network</i>

LISTA DE SÍMBOLOS

α	Ângulo das pás com o eixo do motor
v	Velocidade do fluido
R	Raio das pás

SUMÁRIO

1	INTRODUÇÃO	14
1.1	LIMITAÇÃO DE ESCOPO	15
1.2	OBJETIVOS	16
1.2.1	Geral	16
1.2.2	Específicos	16
1.3	ORGANIZAÇÃO DO TRABALHO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	INSTRUMENTAÇÃO	18
2.1.1	Transmissores de Vazão do Tipo Vórtex	19
2.1.2	Transmissores de Vazão do Tipo Turbina	20
2.2	PROTOCOLOS DE COMUNICAÇÃO	21
2.2.1	HTTP	22
2.2.2	MQTT	25
2.2.3	Modbus TCP	27
2.3	SERIES TEMPORAIS	29
2.4	BANCO DE DADOS	30
2.4.1	InfluxDB	32
2.4.2	TimescaleDB	33
2.5	ANÁLISE DE DADOS INDUSTRIAIS	35
2.6	VISUALIZAÇÃO DE DADOS	36
2.6.1	Grafana	36
2.6.2	Power BI	38
2.7	NODE-RED	39
3	DESENVOLVIMENTO DO TRABALHO	42
3.1	ESPECIFICAÇÕES E REQUISITOS	42
3.2	ANÁLISE DO CONTEXTO	43
3.3	ESCOLHA DOS PROTOCOLOS	46

3.4	ESCOLHA DA FERRAMENTA DE COLETA DE DADOS	47
3.5	ESCOLHA DO BANCO DE DADOS.....	48
3.6	ESCOLHA DE FERRAMENTA DE VISUALIZAÇÃO	49
3.7	COMUNICAÇÃO MQTT COM O <i>NODE-RED</i>	50
3.8	COMUNICAÇÃO COM CLP <i>COMPACTLOGIX</i>	51
3.9	COMUNICAÇÃO MQTT COM <i>LOGBOX WI-FI</i>	54
3.10	COMUNICAÇÃO HTTP COM <i>NODE-RED</i>	57
3.11	INTEGRAÇÃO DO <i>INFLUXDB</i>	59
3.12	VISUALIZAÇÃO DE SÉRIES TEMPORAIS COM O <i>GRAFANA</i>	60
4	RESULTADOS E ANÁLISES	65
5	CONCLUSÕES E PROPOSTAS DE CONTINUIDADE	69
	REFERÊNCIAS.....	71
	APÊNDICE A – CODIGO CLP ROCKWELL.....	73

1 INTRODUÇÃO

Automação é um conceito que vem crescendo de importância junto com a informática. Entende-se por automação qualquer sistema apoiado em computadores que almeje substituir tarefas de trabalho humano e/ou que almeje soluções rápidas e econômicas para as indústrias e os serviços modernos (AGUIRRE, 2007). Com o avanço das tecnologias digitais e a chegada da Indústria 4.0, novos métodos de coleta, armazenamento e visualização de dados vêm sendo incorporados ao ambiente fabril. Os requisitos técnicos para construir e gerenciar plataformas de dados deram origem a novas ferramentas de infraestrutura de código aberto e protocolos projetados para facilitar as crescentes necessidades de dados de sistemas em tempo real em escala (SOTTILARE, GRAESSER, *et al.*, 2017).

Essas ferramentas de código aberto reduzem significativamente as barreiras para a adoção de soluções baseadas em séries temporais, uma vez que eliminam ou minimizam os custos iniciais de implementação. Ferramentas modernas como *Prometheus*, *Grafana*, e *Elastic Stack* estão sendo amplamente utilizadas para coletar, monitorar e analisar indicadores de desempenho de sistemas e aplicações (TARGET SOLUTIONS, 2024). A partir de sua utilização, tornou-se evidente que essas plataformas vão além de simples ferramentas destinadas a análises de um fato ocorrido no passado, mas sim ferramentas para guiar decisões ao prover *insights* sobre dados em tempo real (SOTTILARE, GRAESSER, *et al.*, 2017).

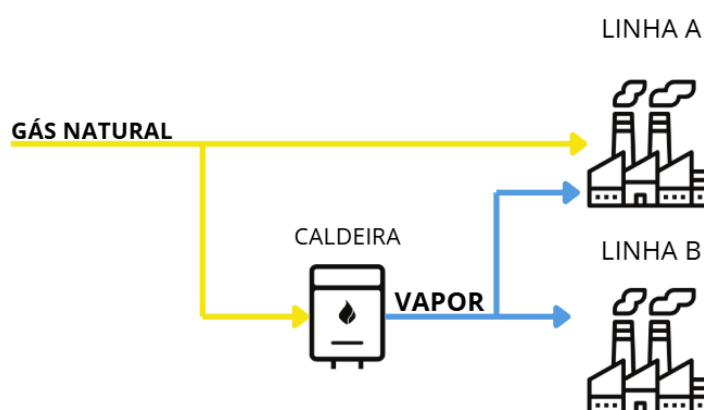
No contexto industrial, especialmente na planta foco deste trabalho, ainda são adotados procedimentos manuais para a coleta de dados provenientes de diversas fontes, com o objetivo de gerar indicadores de desempenho, e o *Grafana* é uma ferramenta que poderia estar sendo usado para mostrar esses indicadores. Para uma indicação de sua adoção pela indústria, a lista de usuários do *Grafana* inclui Intel, Ebay, Staples e Fermilab (SOTTILARE, GRAESSER, *et al.*, 2017). Estes procedimentos, muitas vezes realizados diariamente, envolvem a consulta a dados de fornecedores e ao próprio sistema de automação da fábrica, cuja consolidação é feita manualmente em planilhas. O intuito é analisar essas informações, identificar relações e extrair *insights* relevantes sobre o estado operacional e estratégico do negócio.

O desenvolvimento de uma solução, para automatizar a coleta, o armazenamento e a visualização de dados, é de grande relevância tanto para os colaboradores quanto para a empresa. Para os profissionais envolvidos, essa automação elimina a necessidade de executar tarefas repetitivas e complexas, que pouco contribuem para seu crescimento técnico e profissional. Para a organização, representa uma oportunidade de realocar o tempo e o esforço desses colaboradores para atividades mais estratégicas e produtivas, agregando maior valor ao negócio.

1.1 Limitação de Escopo

Na planta industrial objeto deste estudo, existe um processo de coleta de dados para destrinchar o uso de gás natural por duas linhas de produção, cuja finalidade é gerar um indicador de eficiência e ratear a conta de gás por centro de custo. Uma linha, chamada nesse estudo de linha “A”, usa gás natural diretamente no processo e indiretamente através da geração de vapor por uma caldeira, enquanto a outra, chamada de linha “B” neste estudo, usa apenas indiretamente através de vapor, compartilhando a mesma caldeira, como disposto na Figura 1.

Figura 1 : Fluxo de gás natural.



Fonte: Imagem do autor (2025).

A solução atual é consultar o site da Copergás para exportar um arquivo com o consumo total de gás do dia anterior, acessar uma aplicação Web, implementada por uma empresa terceira conceituada no ramo de automação, para baixar o consumo de gás pelo processo da linha “A”, exportar os dados de vapor pelos *dataloggers* conectados aos medidores de vapor e por fim adicionar essas informações numa planilha do *Excel*, pré-formatada com macros, que destrincha todas as informações para prover o consumo de gás por linha.

Em contrapartida a essa rotina manual e morosa, propõe-se criar uma aplicação que monitore, armazene e disponha de todos esses dados numa interface gráfica. A solução visa simplificar a consulta e análise dos dados, reduzindo significativamente o tempo gasto e minimizando erros humanos, contribuindo assim para uma gestão mais eficiente dos recursos energéticos da planta industrial.

Esse trabalho busca adaptar essas ferramentas para o cenário industrial usando os protocolos compatíveis com essa realidade.

1.2 Objetivos

Esse trabalho tem como objetivo aumentar a produtividade do setor da manutenção, automatizando a coleta de dados das diversas fontes necessárias para realizar o rateamento do consumo e gás natural internamente.

1.2.1 Geral

Desenvolver e implementar uma solução escalável e integrada para coleta, armazenamento e visualização de dados, visando facilitar o monitoramento do consumo de gás natural em indústrias.

1.2.2 Específicos

- Baixo custo de implementação.

- Fácil integração com dispositivos industriais já existentes.
- Visualização intuitiva dos dados coletados.
- Acessibilidade via rede corporativa.
- Flexibilidade para expansão futura.
- Automação completa da coleta e armazenamento dos dados.
- Organização dos dados em *dashboards* claros e interativos.
- Simplicidade.

1.3 Organização do Trabalho

O trabalho está dividido em quatro partes principais: fundamentação teórica, desenvolvimento da solução, apresentação dos resultados e conclusões e análises futuras.

No Capítulo 2, são apresentados os conceitos necessários para o entendimento do projeto, como instrumentação industrial, protocolos de comunicação utilizados, banco de dados de séries temporais e ferramentas de visualização de dados.

O Capítulo 3 foca na descrição do ambiente industrial analisado, nos dispositivos utilizados, na integração entre eles e nas ferramentas empregadas para a coleta, armazenamento e visualização dos dados. São abordados ainda os desafios práticos e as soluções adotadas durante a implementação do sistema.

No Capítulo 4 são apresentados os resultados obtidos com a integração de todas as ferramentas empregas na seção anterior, bem como a estrutura final da solução implementada.

No Capítulo 5, são discutidos os benefícios alcançados com a aplicação, como a automatização do processo de coleta de dados, a redução do tempo de trabalho manual e a melhoria na confiabilidade das informações, além de sugestões de melhorias para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Para melhor compreensão dos temas abordados neste trabalho, serão apresentados conceitos de instrumentação, séries temporais, análise de dados e automação de processos. Desta forma, é discutido sobre o fluxo dos dados, desde sua geração pelo sensor até seu consumo pelo usuário final.

2.1 Instrumentação

Um sistema de medição é um conjunto de dispositivos (sensores, circuitos, cabos, visores, equações, programas de computador etc.), cujo objetivo é fornecer informação sobre o valor da grandeza física que se deseja medir, o mesurando (AGUIRRE, 2013). O componente fundamental de um sistema de medição é o sensor. Uma configuração comumente encontrada para sensores constitui de um transdutor, que realiza conversão de energia, e um condicionador de sinal, responsável por manipular o sinal de saída do transdutor para valores padronizados de tensão ou corrente, ou interpretar diretamente para a grandeza desejada por meio de um circuito digital.

Com base nisso, os instrumentos podem ser categorizados como digitais ou analógicos. Num instrumento analógico, tanto o sensor quanto o condicionamento do sinal é realizado de maneira analógica, enquanto num instrumento digital o condicionamento é feito de forma digital. É comum o termo transmissor quando se trata de um instrumento completo, que disponibilize um sinal de saída que corresponda a grandeza mensurada pelo elemento sensor.

Entre as variáveis de processo, a mais comumente medida é temperatura, logo seguida por pressão (REZA LANGORI, 2016). Ambas são cruciais para o controle e a estabilidade dos processos. A vazão também se destaca como uma variável amplamente medida, sendo essencial para quantificar o volume de fluidos utilizados em diferentes etapas da produção. Um exemplo prático de sua aplicação ocorre no cotidiano, durante o abastecimento de um veículo, quando se mede precisamente a quantidade de combustível transferida. De forma semelhante, as indústrias utilizam

medidores de vazão para monitorar o consumo de fluidos em seus processos produtivos.

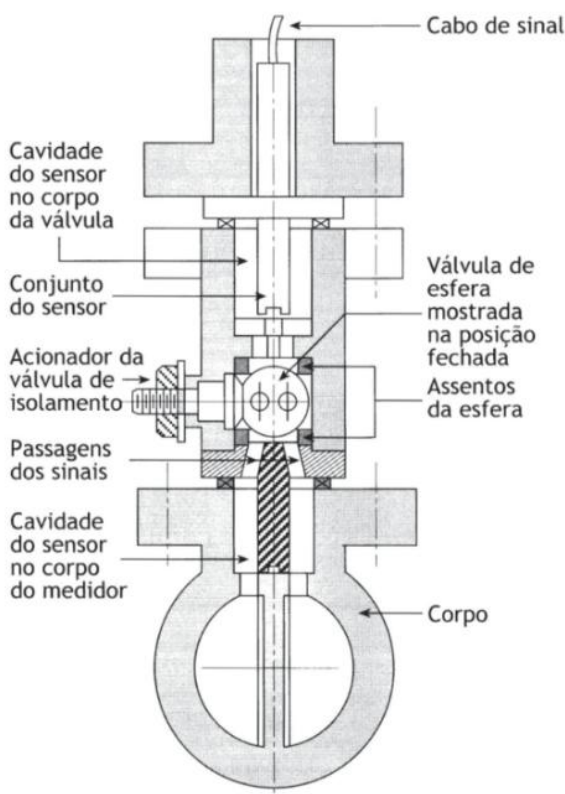
Uma das formas mais comuns de transmitir os dados do sensor é através de um sinal analógico de corrente, nela o range do sensor é transcrito num sinal de corrente que varia de 4 a 20 miliamperes. Porém, com a melhora da eletrônica digital foram desenvolvidos protocolos de comunicação para enviar dados através de barramento através de sinais digitais.

Com os avanços na eletrônica embarcada e nas redes sem fio, surgiram no mercado transmissores capazes de transmitir medições por meio de redes sem fio, esses dispositivos são usados para formar redes de sensores sem fio (WSN). Já que WSNs usam comunicação sem fio em vez de cabos, elas não necessitam de infraestrutura complexa e, devido a estrutura sem cabos, WSNs se tornam mais baratas (KOCAKULAK e BUTUN, 2017).

2.1.1 Transmissores de Vazão do Tipo Vórtex

Os medidores de vazão do tipo vórtex consistem em um elemento de tubulação com um corpo falso presente, como visto na Figura 2, que possibilita a criação de vórtices, a jusante, com frequência proporcional a velocidade do fluido. Com o aumento da velocidade, as linhas de fluidas deixam de acompanhar o contorno do obstáculo, ocorrendo a separação do escoamento em relação à sua superfície. Essa separação provoca o surgimento de velocidades locais mais elevadas, com zonas de baixa pressão, e a ruptura da camada-limite, como resultado, formam-se turbilhões ou vórtices (DELMÉE, 2003). A partir de um número de Reynolds mínimo, definido para cada modelo de medidor, existe um fator K de proporcionalidade entre frequência e a vazão atual. Esse fator considera as dimensões da tubulação onde a medição é realizada, sendo expresso em unidades de volume por número de pulsos, e serve como parâmetro de calibração do medidor (DELMÉE, 2003).

Figura 2 : Medidor de vazão do tipo vórtex.



Fonte: Adaptado de (DELMÉE, 2003).

Um instrumento que opera por este princípio se torna bastante versátil em suas aplicações, ao considerar que possui a capacidade de mensurar a vazão tanto de líquidos quanto de gases, e, pelo fato de contar o número de pulsos, não possui desvio do ponto zero.

2.1.2 Transmissores de Vazão do Tipo Turbina

Esse tipo de sensor consiste em uma turbina posicionada no interior de uma tubulação. A teoria das turbinas consiste em um rotor, provido de aletas e formando um ângulo α com as linhas de escoamento, que é posto a girar quando há vazão, devido à incidência das linhas de velocidade do fluxo contra os planos inclinados representadas pelas pás do rotor (DELMÉE, 2003).

A aferição da velocidade de rotação da turbina é geralmente feita por acoplamento ótico, através de fontes de luz e fototransistores, ou magnético, através de imãs e sensores de efeito Hall ou bobinas. Tais componentes são responsáveis por gerar um determinado número de pulsos de tensão à medida que as pás completam sua rotação ou um sinal, possibilitando um microcontrolador realizar a contagem do número de voltas e estimar a vazão de fluido através da tubulação.

O comportamento da velocidade angular é dado pela Equação (2.1), adaptada de (DELMÉE, 2003), onde v é a velocidade do fluido, α é o ângulo das pás com o eixo do motor e R é o raio das pás. O posicionamento da turbina pode ser visto na Figura 3.

$$\omega = \frac{v * \tan(\alpha)}{R} \quad (2.1)$$

Figura 3 : Turbina para líquidos, com retificador de fluxo incorporado.



Fonte: Adaptado de (DELMÉE, 2003).

2.2 Protocolos de Comunicação

O advento da microeletrônica trouxe benefícios significativos para diversas áreas. Posteriormente, a difusão de equipamentos microprocessados voltados à automação industrial gerou a necessidade de estabelecer meios eficazes de

comunicação entre esses dispositivos (AGUIRRE, 2007). Comunicação é uma tarefa que exige mais de uma entidade para ser realizada com sucesso, parte disso a necessidade de uma padronização de regras que gerem entendimento entre as partes envolvidas.

Como o início de tudo na eletrônica, os primeiros protocolos eram analógicos, destinados a troca de informação entre dois dispositivos, porém, o avanço da tecnologia trouxe os protocolos digitais. Esses protocolos são regras de comunicação entre dois ou mais dispositivos (AGUIRRE, 2007).

Em relação aos protocolos digitais, muitas definições devem ser feitas para conseguir efetivamente formalizar um protocolo e conseguir transferir a informação do ponto A ao ponto B. Alguns parâmetros constituem do meio físico, níveis dos sinais elétricos representando o nível lógico “1” e o nível lógico “0”, a velocidade de transmissão, entre outros.

Como esses parâmetros podem ser combinados de diferentes formas, surgiram vários protocolos digitais no mercado. Historicamente, os primeiros protocolos surgiram no formato proprietário, sendo usado somente pela empresa que o desenvolveu e mantidos como segredo empresarial (AGUIRRE, 2007). Nesse período não era possível comunicar equipamentos de diferentes fabricantes, obrigando os consumidores a escolher um ecossistema de equipamentos.

Num período posterior, diante dessa restrição, foram desenvolvidos os protocolos abertos. A ausência da barreira da comunicação possibilitou o surgimento e diferentes fabricantes competindo em qualidade, custo e serviços diferentes (AGUIRRE, 2007). Com o avanço dos protocolos abertos e a crescente necessidade de integração entre diferentes dispositivos e sistemas industriais, surgiram diversos protocolos de comunicação voltados para a transmissão de dados em redes locais e na internet, entre eles o HTTP, MQTT, Modbus TCP, entre outros.

2.2.1 HTTP

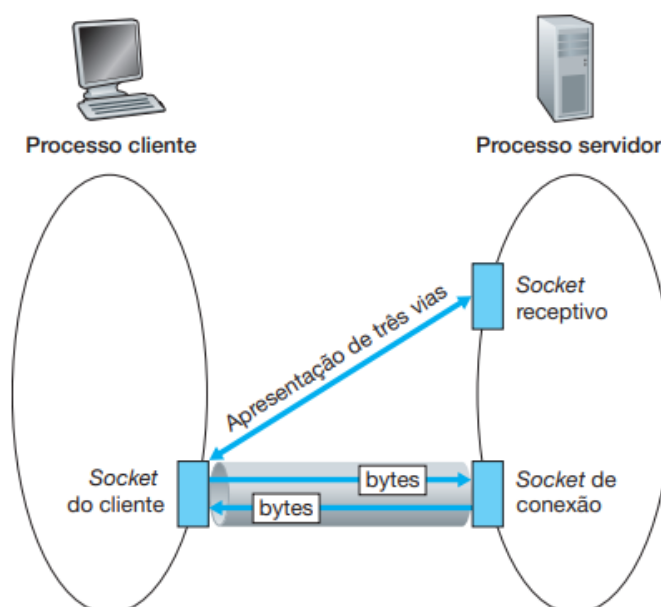
O Protocolo de transferência de Hypertexto (HTTP) define como os clientes requisitam páginas aos servidores e como eles as transferem aos clientes (KUROSE

e ROSS, 2013). Se trata de um protocolo de comunicação comum na internet, baseado no modelo cliente-servidor, e é amplamente usado para interação com páginas na internet.

O HTTP usa o TCP como seu protocolo de transporte subjacente (KUROSE e ROSS, 2013). O serviço de conexão possibilita a troca de informações de controle da camada de transporte, entre o cliente e o servidor, antes que as mensagens comecem a fluir na camada de aplicação, com o intuito de preparar ambas as partes para o envio de dados. O serviço de transferência garante que a cadeia de dados que um processo lê a partir de seu buffer de recebimento TCP não está corrompida, não tem lacunas, não tem duplicações e está em sequência (KUROSE e ROSS, 2013).

O cliente HTTP primeiro inicia a conexão *TCP* com o servidor. Esta conexão é estabelecida através de um processo conhecido como *3-way handshake*. Que se resume em o cliente enviar um pequeno segmento *TCP* ao servidor, este o reconhece e responde com um pequeno segmento ao cliente que, por fim, o reconhece novamente para o servidor (KUROSE e ROSS, 2013), como ilustrado na Figura 4.

Figura 4: Representação 3-way handshake.

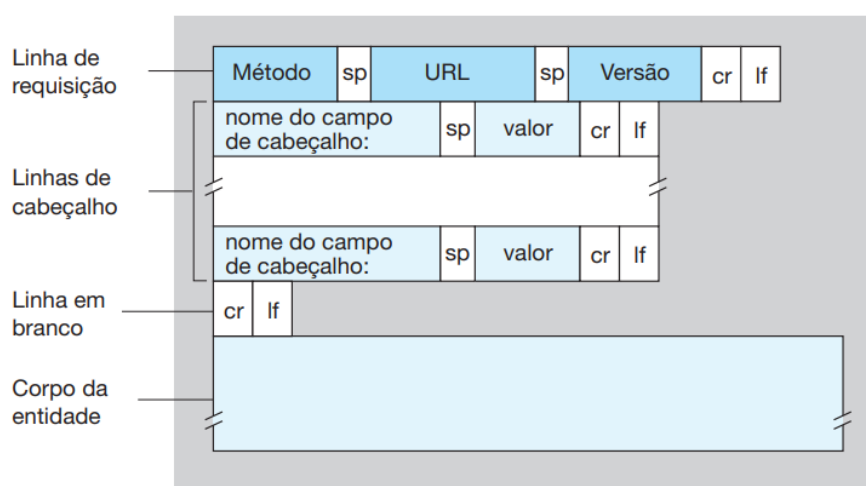


Fonte: Adaptado de (KUROSE e ROSS, 2013).

Uma requisição HTTP é geralmente constituída de três partes, como visível na Figura 5, sendo elas a linha de requisição, linhas de cabeçalho e o corpo da entidade. A linha de requisição tem três campos: o do método, o do *URL* e o da versão do HTTP. O campo do método pode assumir vários valores diferentes, entre eles *GET*, *POST*, *HEAD*, *PUT* e *DELETE* (KUROSE e ROSS, 2013). O método é o que condiz a ação desejada, os métodos *GET* e *HEAD* solicitam informações, o método *POST* envia informações, o *PUT* altera informações existentes e o *DELETE* é usado para deletar informações.

O cabeçalho normalmente inclui informações como o domínio do servidor que receberá a requisição, preferências de linguagem, data que a requisição foi gerada, tamanho em *bytes* da requisição entre outras informações. Essas informações ajudam o servidor a saber quem está solicitando, o que está sendo solicitado e como lidar com a resposta. Já o corpo da entidade carrega o objeto solicitado, representado por dados, ele é usado apenas em métodos usados para transferência de informação.

Figura 5: Formato geral de uma mensagem de requisição *HTTP*.

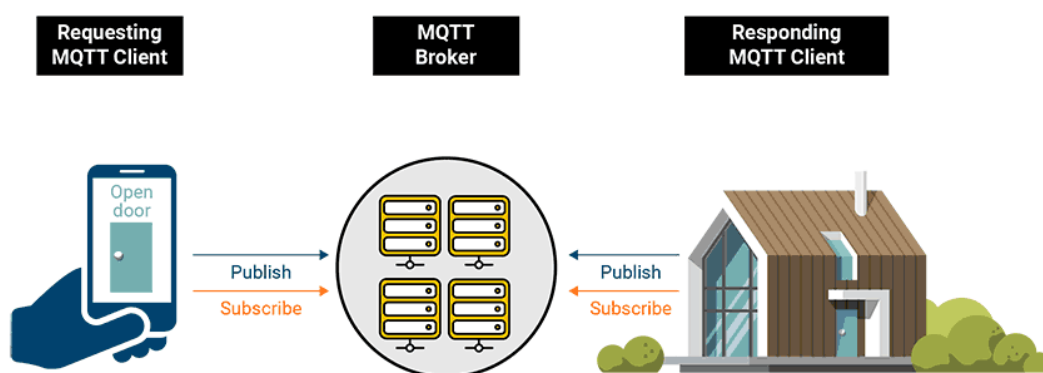


Fonte: Adaptado de (KUROSE e ROSS, 2013).

2.2.2 MQTT

O MQTT é o protocolo primário que dispositivos, *gateways* e aplicativos usam para se comunicar com ferramentas voltadas a internet das coisas (IoT) (IBM, 2025). Ele foi criado pela *IBM* em 1999 e construído em cima do protocolo TCP/IP, ele utiliza do modelo de publicador e assinante, onde as comunicações são intermediadas por um *broker*, como pode ser visto na Figura 6.

Figura 6 : Exemplo de aplicação com MQTT.

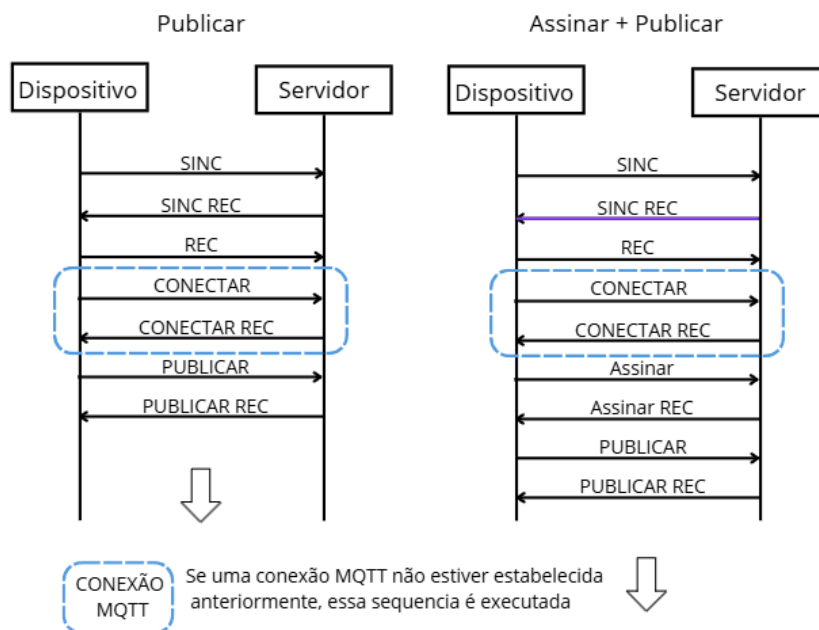


Fonte: Adaptado de (HIVEMQ, 2024).

O *broker* é um servidor que executa o protocolo MQTT, e é responsável por autenticar as partes envolvidas, receber as mensagens e distribuir a quem tem interesse. O principal papel deste servidor é receber todas as mensagens, filtrá-las, decidir quais clientes estão interessados nestas mensagens e encaminhar estas mensagens a todos os clientes assinantes.

Como o protocolo MQTT faz também faz uso do protocolo TCP, para publicar um dado é necessário iniciar a conexão com o broker pelo *3-way handshake*, então estabelecer a conexão MQTT, se inscrever em um tópico e após isso publicar o dado no tópico de interesse, como exemplificado na Figura 7.

Figura 7 : Sequência de comunicação do protocolo MQTT.



Fonte: Adaptado de (YOKOTANI e SASAKI, 2016).

Esse protocolo apresenta diversas vantagens para dispositivos de campo que reportam dados, uma vez que independentemente do número de usuários interessados num dado gerado por um dispositivo de campo, o dispositivo responsável pelo sensoriamento envia o dado apenas uma vez direcionado ao *broker*, e este fica responsável por levar a mensagem aos interessados.

Também é observado que a comunicação é mais rápida ao usar MQTT. Segundo a Tabela 1, que compara a velocidade de transmissão e recebimento de dados entre os protocolos MQTT e HTTP, levando em conta o estabelecimento de conexão para MQTT #1 e a conexão já estabelecida em MQTT #2, é possível observar que o tempo médio e máximo de transmissão e recebimento de dados é maior no protocolo HTTP.

Tabela 1 : Comparação entre envio e recebimento de dados sequenciais entre dispositivos e um servidor.

		MQTT #1	MQTT #2	HTTP
N. de viagens		4	1	5
Duração do início ao fim	Máxima	800ms	200ms	1s
	Média	400ms	100ms	500ms

Fonte: Adaptado de (YOKOTANI e SASAKI, 2016).

2.2.3 Modbus TCP

Um dos primeiros protocolos a se popularizar no meio industrial foi o *Modbus*. Ele foi desenvolvido pela *Modicon* em 1979, e em pouco tempo foi largamente usado para comunicação entre diferentes CLPs, tornando-se de domínio público (AGUIRRE, 2007). A evolução do protocolo fez com que surgissem 3 variantes: *Modbus RTU*, *Modbus ASCII* e *Modbus TCP*.

O *Modbus TCP* utiliza o protocolo *ASCII* encapsulado num frame *Ethernet*. O cabeçalho consiste no identificador de transação, do identificador de protocolo, do tamanho em *bytes* da mensagem e do endereço do escravo. O código a função indica o tipo de operação a ser realizada, e pode assumir valores dentro das opções vistas na Tabela 2. A mensagem refere-se ao dado a ser escrito no endereço do escravo pelo mestre, para o caso de uma operação de escrita, ou o dado de interesse do mestre, numa operação de leitura.

O endereço do escravo só é usado em aplicações com um *gateway* que converte *Modbus TCP* para *Modbus RTU*, uma vez que o protocolo *TCP* usa o endereço de IP para encontrar os dispositivos da rede.

Tabela 2: Códigos de função do protocolo *Modbus*.

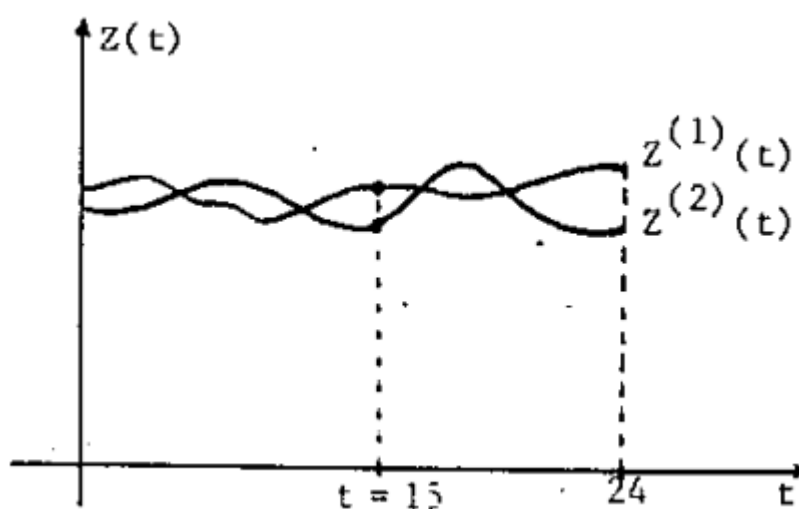
Código da função	Descrição
1	Leitura de bloco de bits do tipo <i>coil</i> (saída discreta).
2	Leitura de bloco de bits do tipo entradas discretas.
3	Leitura de bloco de registradores do tipo <i>holding</i> .
4	Leitura de bloco de registradores do tipo <i>input</i> .
5	Escrita em um único bit do tipo <i>coil</i> (saída discreta).
6	Escrita em um único registrador do tipo <i>holding</i> .
7	Ler o conteúdo de 8 estados de exceção.
8	Prover uma série de testes para verificação da comunicação e erro internos.
11	Modbus: Obter o contador de eventos.
12	Modbus: Obter um relatório de eventos.
15	Escrita em bloco de bits do tipo <i>Coil</i> (saída discreta).
16	Escrita em bloco de registradores do tipo <i>holding</i> .
17	Ler algumas informações do dispositivo.
20	Ler informações de um arquivo.
21	Escrever informações em um arquivo.
22	Modificar o conteúdo de registradores de espera através de operações lógicas.
23	Combina ler e escrever em registradores numa única transação.
24	Ler o conteúdo da fila FIFO de registradores.
43	Identificação do modelo do dispositivo.

Fonte: Adaptado de (FREITAS, 2014).

2.3 Series Temporais

Uma série temporal é qualquer conjunto de observações ordenadas no tempo (MORETTIN PA, 1985). Ao mensurar a temperatura de um dado local por dois dias, encontraremos curvas diferentes, como exemplificado na Figura 8, essas curvas são chamadas de trajetórias do processo físico observado.

Figura 8 : Duas trajetórias para a temperatura de um certo local num dia.



Fonte: Adaptado de (MORETTIN PA, 1985).

A análise de séries temporais é um processo que utiliza estatística e modelagem para realizar previsões que podem ser extremamente importantes na tomada de decisões estratégicas (SPADINI e ALENCAR, 2022). O estudo de series temporais se torna interessante quando se tem o desejo de aprofundar o entendimento de um processo físico, os interesses comuns nesse estudo podem ser vistos na Tabela 3, que tem a finalidade de entender o comportamento do fenômeno observado a fim de realizar previsões.

Tabela 3: Interesses comuns no estudo de series temporais.

Interesses comuns no estudo de series temporais
Investigar o mecanismo gerador da série temporal
Fazer previsões de valores futuros da serie
Descrever o comportamento da serie
Procurar periodicidades relevantes nos dados

Fonte: Adaptado de (MORETTIN PA, 1985).

Series temporais são descritas através de modelos de processos estocásticos, ou seja, processos controlados por leis probabilísticas (MORETTIN PA, 1985). O processo de construção de um modelo varia de acordo com vários fatores, que podem incluir o objetivo da análise, o conhecimento a priori que já temos do fenômeno observado e o próprio comportamento observado do fenômeno.

Ao lidar com um problema novo de series temporais é comum tomar como ponto de partida suposições sobre os dados a serem analisados para tentar começar a extrair informações. Uma das suposições mais comuns sobre séries temporais é a de que elas são estacionárias, ou seja, que se desenvolvem no tempo aleatoriamente ao redor de uma média constante (MORETTIN PA, 1985).

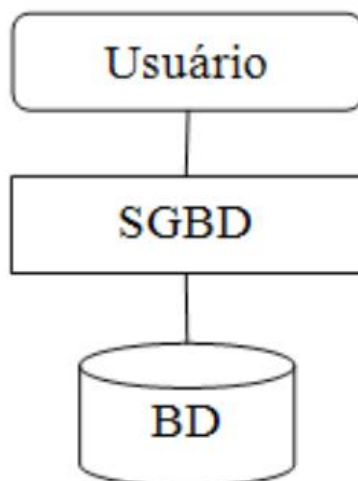
Contudo, a maior parte das series temporais tende a apresentar alguma forma de não-estacionariedade, indicando alguma forma de tendencia. Series econômicas e financeiras, por exemplo, apresentam em geral tendências, sendo o caso mais simples aquele em que a série flutua ao redor de uma reta, podendo possuir inclinação positiva ou negativa (MORETTIN PA, 1985).

2.4 Banco de Dados

Um sistema de banco de dados é um sistema que através do computador manipula e organiza registros, tornando-os disponíveis ao usuário (C.J.DATE, 2004).

Toda a interação do banco de dados com o usuário é intermediada por um sistema de gerenciamento de banco de dados (SGBD), como mostrado na Figura 9, que é responsável por estruturar e organizar os registros na memória.

Figura 9 : Visão superficial de um SGBD e seus relacionamentos.



Fonte: Adaptado de (C.J.DATE, 2004).

Um sistema gerenciador de banco de dados é uma coleção de programas que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações (KORTH e SUDARSHAN, 1995). O sistema gerenciador é responsável por simplificar todas as interações com o banco de dados.

Bancos de dados podem ser classificados entre bancos de dados relacionais e bancos de dados não relacionais, apesar de ambos terem os mesmos objetivos: armazenar, fornecer quando requisitado e coordenar eventuais mudanças. Os bancos de dados relacionais ganharam popularidade primeiro por possuírem um diferencial frente aos seus competidores a época, o formalismo relacional fornece uma estruturação conceitual que permite uma formulação mais abrangente, tornando-os geralmente mais fáceis de administrar (C.J.DATE, 2004).

Contudo, problemas modernos exigiram uma solução que entregasse mais performance para grandes volumes e dados, e nesse contexto surgiram os bancos de

dados não relacionais. A diferença para os bancos de dados relacionais é que os não relacionais abandonam algumas das funções dos bancos de dados relacionais com a finalidade de aumentar a escalabilidade (DUNNING e FRIEDMAN, 2014).

Um caso específico são os bancos de dados voltados a armazenar series temporais, eles são usados para os casos em que é necessário armazenar grandes volumes de dados que vão ser consultados por meio de períodos e tempo. A escolha de usar bancos de dados de series temporais abre a possibilidade de descobrimento de padrões em dados de series temporais, tendencias de logo prazo e correlações entre dados representando diferentes tipos de eventos (DUNNING e FRIEDMAN, 2014).

Por exemplo, o surgimento de bancos de dados confiáveis e de condigo aberto, como *InfluxDB* e *TimescaleDB*, os fez serem ferramentadas indispensáveis sobre as quais são construídos fluxos de trabalhos movido a dados (SOTTILARE, GRAESSER, *et al.*, 2017). Essas ferramentas reduzem o custo inicial de implementação de soluções voltadas ao armazenamento de séries temporais, o que contribui para sua crescente popularização.

2.4.1 InfluxDB

O *InfluxDB* é parecido com um banco de dados *SQL*, mas diferente em diversas maneiras. *InfluxDB* é projetado para lidar com dados de series temporais (INFUXDATA, 2024). Ele oferece uma versão de código aberto que pode ser facilmente configurada em uma máquina sem custos, bem como uma versão empresarial que utiliza servidores dedicados para garantir operação ininterrupta.

O *InfluxDB* é destinado a ser usada como um banco de dados de suporte para qualquer caso de uso envolvendo grandes quantidades de dados com marcas temporais, incluindo monitoramento de *DevOps*, métricas de aplicativos, dados de sensores e análises em tempo real (INFUXDATA, 2024). Seu modelo de dados consiste em medidas que compõem o banco de dados. As informações são inseridas com uma marcação temporal, incluindo a data da medição, acompanhado dos campos

de valores a serem armazenados e os campos de características dos dados para consultas futuras.

O *InfluxDB* suporta duas linguagens de consulta: a *InfluxQL*, que possui uma sintaxe semelhante às linguagens *SQL*, e a linguagem *Flux*, desenvolvida pela própria empresa. A simplicidade do modelo facilita a adaptação e, por ser projetado exclusivamente para lidar com séries temporais, implica um alto grau de otimização para lidar com grandes volumes de dados sendo constantemente armazenados.

Alguns compromissos foram feitos em relação a bancos de dados *SQL* para lidar melhor com series temporais, a dinâmica de substituir os identificadores por marcas temporais implica que não podem existir duplicidade nos dados num mesmo momento, podendo os dados antigos serem sobrescrevidos. Além disso, como foi pensando para sempre estar salvando os dados conforme são gerados em campo, salvar dados de momentos aleatórios ou mais antigos requer um tempo maior para ser realizado (INFUXDATA, 2024).

2.4.2 TimescaleDB

TimescaleDB é um banco de dados de condigo aberto projetado para fazer *SQL* escalável para dados em formato de series temporais (TIMESCALE, 2025). Esse banco de dados utiliza um modelo relacional para armazenar seus dados, por consequência de ser baseado no *PostgreSQL*, um dos bancos de dados relacionais mais populares. Porém com otimizações voltadas a lidar melhor com conjuntos de dados em formato de séries temporais.

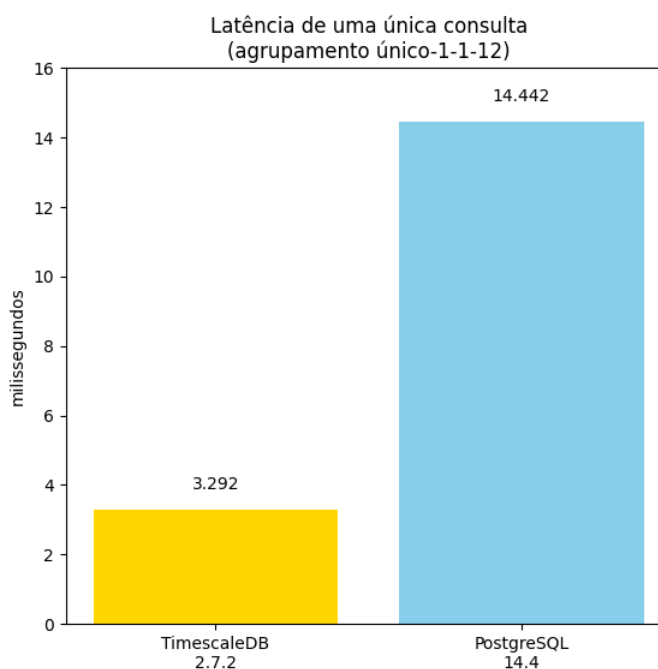
O fato de ser baseado no *PostgreSQL* traz uma série de vantagens, que derivam do compartilhamento de várias ferramentas voltadas ao gerenciamento dos dados, uma vez observado a ampla história desse banco de dados, que incluem extensões, drivers, ferramentas de administração e suporte total ao *SQL* padrão. Consequentemente, o *TimescaleDB* garante uma flexibilidade maior para lidar com séries temporais e todas as vantagens de um banco de dados *SQL*.

O principal conceito do *TimescaleDB* é a ideia de uma super tabela, fruto de um particionamento de dados que possibilita ao usuário a abstração de estar lidando com

uma única tabela (BOOZ, 2025). Esse particionamento é realizado de maneira a otimizar as consultas, permitindo que o banco de dados descarte automaticamente os períodos que não são relevantes para a busca, reduzindo o volume de dados processados. Desta forma, essa super tabela é manegada como uma tabela comum do *PostgreSQL*, porém muito mais eficiente para series temporais.

Quando as super tabelas são comprimidas, a quantidade de dados que as consultas precisam levar em consideração diminui drasticamente, possibilitando tempos de consulta muito menores comparados com o *PostgreSQL* (BOOZ, 2025). Podemos ver na Figura 10, referente ao tempo de latência de consulta entre o *TimescaleDB* e o *PostgreSQL*, que o *TimescaleDB* é capaz de entregar um desempenho superior.

Figura 10 : Comparação do tempo de latência de consultas entre *TimescaleDB* e *PostgreSQL* 14.4.



Fonte: Adaptado de (BOOZ, 2025).

2.5 Análise de Dados Industriais

Os sistemas de produção contemporâneos buscam incessantemente a otimização de processos com o intuito de reduzir custos. No entanto, é crucial conduzir estudos preliminares antes de implementar qualquer ação e definir indicadores relevantes, com a finalidade de identificar de maneira clara os principais desafios. Essa abordagem possibilita uma avaliação simplificada dos problemas existentes, facilitando a priorização de novos projetos dentro da empresa. Numa planta que almeja ser Lean, o propósito principal de dados de inspeção, os quais são compostos de dados de variáveis do processo em vez de simplesmente de dados de conformidade, é resolução de problemas (EILSON, 2009).

Os dados de inspeção inicialmente levavam em conta apenas atributos do produto, porém, eles representam apenas consequências dos parâmetros usados no processo de produção. Foi estabelecido que não seria possível alcançar um alto padrão de qualidade apenas com atributos do produto, por isso, tornou-se necessário correlacionar os defeitos identificados por atributos do produto com os parâmetros do processo (EILSON, 2009).

No cenário atual, a indústria 4.0 emerge com conceitos inovadores que facilitam a obtenção dos dados necessários para fundamentar as decisões dos profissionais encarregados do aprimoramento contínuo nos processos fabris. Segundo (SOTTILARE, GRAESSER, *et al.*, 2017), diferentes grupos de stakeholders desejam visualizar diferentes aspectos dos dados que são importantes para seus objetivos e tarefas distintos. Um dos cenários envolvendo cálculo de indicadores é o consumo de energia elétrica, gás e matéria prima com relação a produção, que necessita sempre estar sendo acompanhado para manter o orçamento sobre controle.

Em resumo, para o pleno funcionamento da fábrica como um negócio rentável, se faz necessário o cálculo e acompanhamento de indicadores chave, para reagir a qualquer condição adversa e manter a consistência.

2.6 Visualização de Dados

Visualização de dados é a apresentação de dados no formato de uma imagem ou gráfico. Essa abordagem facilita a análise de grandes volumes de dados, permitindo aos usuários compreenderem conceitos complexos e identificarem padrões emergentes de forma mais intuitiva (SOTTILARE, GRAESSER, *et al.*, 2017).

Visualização aproveita a incrível capacidade do sistema visual humano para mover uma grande quantidade de informação para o cérebro de maneira rápida (ILIINSKY e STEELE, 2011). Ferramentas de visualização de dados permitem transformar conjuntos de dados extensos e complexos em gráficos, como gráficos de linhas ou barras, com a finalidade de evidenciar algum aspecto dos dados, como *outliers*, máximos, mínimos entre outros, tornando mais simples e rápido o processo de extrair informações relevantes.

Toda apresentação de informações deve ser orientada por um propósito específico, cabendo ao desenvolvedor determinar esse objetivo e, a partir dele, estabelecer os requisitos de design adequados. Segundo (SOTTILARE, GRAESSER, *et al.*, 2017), tais objetivos podem incluir: introduzir informações sobre um determinado tema, facilitar a compreensão de um processo, transmitir uma noção de escala, narrar uma história, provocar respostas emocionais ou ampliar o conhecimento do público a respeito de um assunto.

No contexto da visualização de séries temporais, algumas ferramentas se destacam por oferecerem flexibilidade suficiente para atender aos diversos objetivos de apresentação e análise de dados. Dentre essas ferramentas, o *Grafana* se sobressai por permitir a criação de dashboards capazes de refletir, em tempo real, o conteúdo das séries temporais monitoradas (SOTTILARE, GRAESSER, *et al.*, 2017).

2.6.1 Grafana

O *Grafana* é um *software* de código aberto para visualização e análise. Ele permite consultas, visualização, alertas e exploração de dados, não importando onde eles estejam armazenados (GRAFANA LABS, 2024). No que diz respeito a consultas,

o *Grafana* disponibiliza uma interface gráfica que facilita a navegação para usuários menos experientes ou para consultas mais simples. Além disso, oferece um espaço dedicado para consultas por meio de linguagem escrita, permitindo a expressão de consultas de forma mais avançada e personalizada. Quando concluído, um *dashboard* se torna uma ferramenta que facilita a análise de dados, como pode ser visto na Figura 11.

Este *software* já possui integração com diversos bancos de dados, conforme apresentado na Tabela 4, proporcionando flexibilidade para uma ampla gama de aplicações. As consultas realizadas pelo *Grafana* são projetadas para conjuntos de dados extensos, razão pela qual o sistema inclui um mecanismo que, em vez de processar dados individualmente, agrupa automaticamente as informações em períodos de tempo, de acordo com o intervalo solicitado.

Tabela 4 - Bancos de dados integrados ao Grafana.

<i>Alertmanager</i>	<i>Microsoft SQL Server</i>
<i>AWS CloudWatch</i>	<i>MySQL</i>
<i>Azure Monitor</i>	<i>OpenTSDB</i>
<i>Elasticsearch</i>	<i>PostgreSQL</i>
<i>Google Cloud Monitoring</i>	<i>Prometheus</i>
<i>Graphite</i>	<i>Pyroscope</i>
<i>InfluxDB</i>	<i>Tempo</i>
<i>Jaeger</i>	<i>Testdata</i>
<i>Loki</i>	<i>Zipkin</i>

Fonte: Adaptado de (GRAFANA LABS, 2024).

Além de ser simples de usar para criar *dashboard* complexos, *Grafana* também inclui ferramentas que encorajam compartilhamento e colaboração na edição dos *dashboards* (SOTTILARE, GRAESSER, *et al.*, 2017). Essas ferramentas ajudam a

democratizar os dados nas organizações, fortalecendo a cultura das decisões baseadas em dados.

Figura 11: Exemplo de dashboard com o Grafana.



Fonte: Adaptado de (GRAFANA LABS, 2024).

2.6.2 Power BI

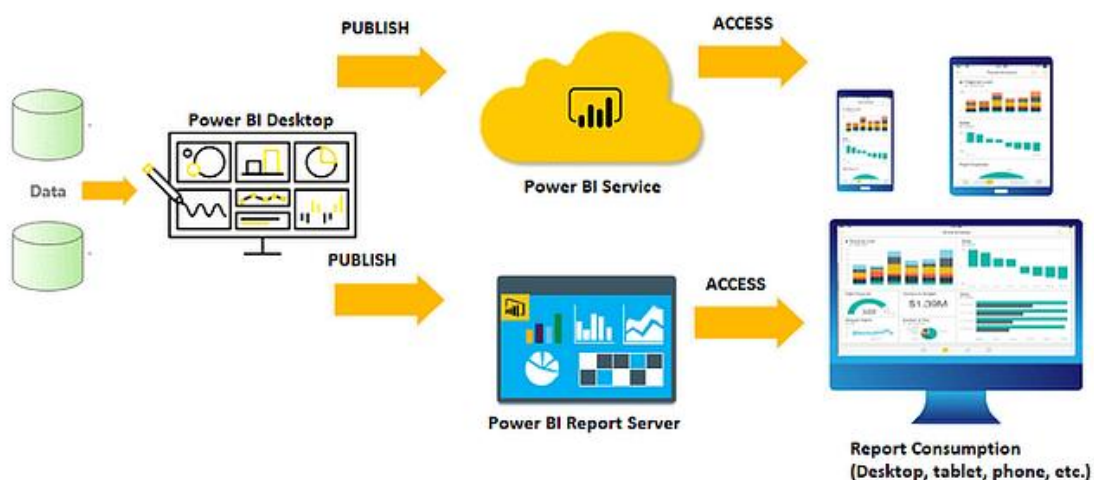
O *Power BI* foi lançado em 24 de julho de 2015 pela Microsoft com o intuito de fornecer uma ferramenta para revelar insights para a tomada de decisões estratégica em empresas. O *Power BI* é uma coleção de serviços de software, aplicativos e conectores que trabalham juntos para transformar suas fontes de dados não relacionadas em informações coerentes, visualmente envolventes e interativas. (MICROSOFT, 2024).

Ao conectar o *Power BI* com um banco de dados, é possível criar relatórios, gráficos e estatísticas de maneira simples e eficiente, possibilitando realizar análise de dados e escolher visualizações que evidenciem os aspectos de principais da mensagem que se está tentando transmitir.

O *Power BI* funciona principalmente através de dois componentes, o primeiro sendo o *Power BI Desktop*, onde o foco é na conexão com banco de dados e o

desenvolvimento de relatórios interativos. O segundo é o *Power BI Service*, que é a plataforma online para o compartilhamento de relatórios. Na Figura 12 está presente o fluxo normal de uso do *Power BI*, com a criação de relatórios no *Power BI Desktop* e o compartilhamento pelo *Power BI Service*.

Figura 12 Fluxo do Power BI.



Fonte: Adaptado de (AFTAB, 2024).

2.7 Node-RED

Node-RED é uma ferramenta de programação baseado em fluxos adequada para implementar controle de dados para aplicações *web* e IoT. Contudo, o *Node-RED* não é apenas uma ferramenta de programação, mas também uma plataforma de execução que encapsula o *runtime* do *Node.js* para aplicações desenvolvidas usando o *Node-RED* (HAGINO, 2021).

Foi criado pela empresa *IBM*, inicialmente era apenas uma prova de conceito para ajudar a entender e mapear tópicos do protocolo MQTT, mas em pouco tempo se tornou uma ferramenta difundida que poderia ser estendida para outras aplicações.

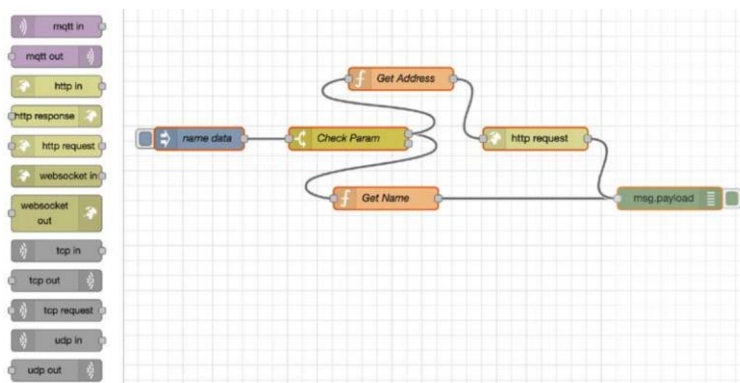
Segundo (HAGINO, 2021), os pontos positivos do *Node-RED* são sua simplicidade, eficiência, qualidade, capacidade de ser usado em várias plataformas,

ser de código aberto e ter uma ampla coleção de bibliotecas. E esses pontos são refletidos no seu uso, que se dá através de um *browser*, programando de maneira gráfica usando uma abordagem chamada programação baseada em fluxo.

Programação baseada em fluxo interpreta a aplicação como uma rede processos assíncronos que começam num determinado ponto, executam uma série de ações sequenciais, um por um, até finalizar todos os comandos (HAGINO, 2021). A organização em fluxos, apresentada na Figura 13, compostos por blocos conectados, simplifica a programação ao utilizar uma interface gráfica. Além disso, ao tratar cada fluxo como um processo assíncrono, a abordagem permite aproveitar múltiplos núcleos do processador, aumentando a eficiência da execução da aplicação.

Esta ferramenta é comumente usada como *gateway* de dados por possuir bibliotecas gratuitas que possibilitam a comunicação através de diversos protocolos comuns na indústria, essencialmente servindo como uma ponte entre o dispositivo que gera o dado e onde o dado é consumido.

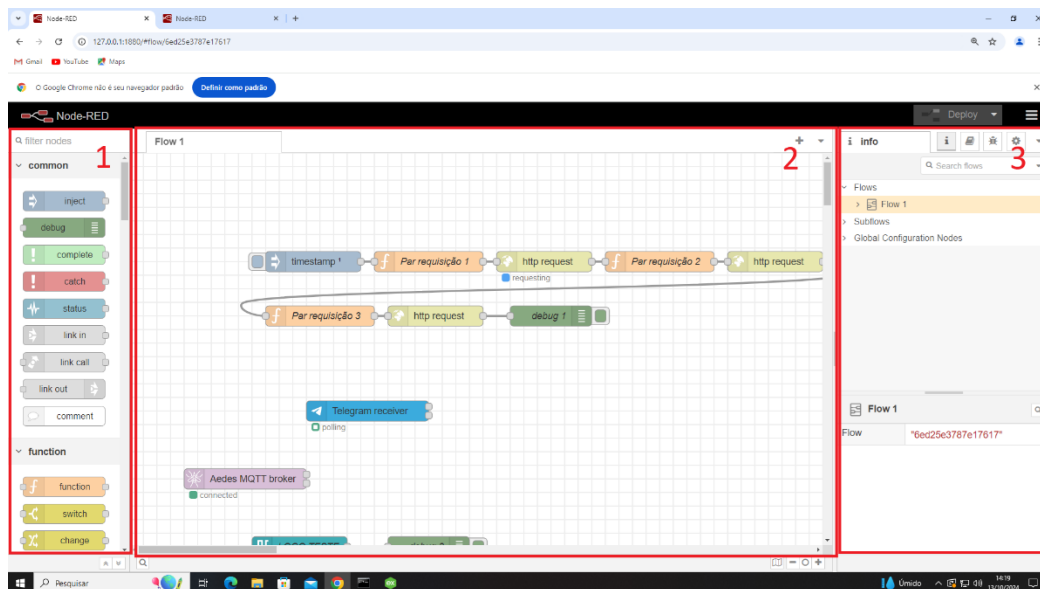
Figura 13: Editor do *Node-RED* como uma ferramenta de programação baseada em fluxos.



Fonte: Adaptado de (HAGINO, 2021).

Sua interface no navegador é dividida em 3 áreas, como visto na Figura 14 a primeira é a área com os nós, que consistem em blocos com códigos implementos em *Javascript*, a segunda área consiste no espaço de desenvolvimento e a terceira é a área de configuração. A terceira área também apresenta a aba de *debug*, indica as mensagens de erro e nós geradores das mesmas.

Figura 14: Interface do Node-RED



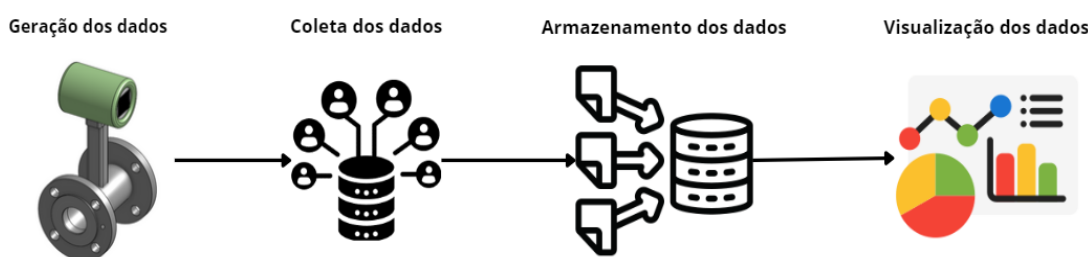
Fonte: Imagem do autor.

3 DESENVOLVIMENTO DO TRABALHO

Após a análise das informações relativas aos sensores, protocolos de comunicação, banco de dados e visualização de dados, a próxima etapa consiste na discussão sobre a integração dessas ferramentas e o fluxo de dados na aplicação proposta. O diagrama da solução pode ser visto na Figura 15, ilustra que a informação é gerada pelo sensor, recebida e armazenada em um banco de dados para futuras consultas e visualizações.

Para a construção da solução que entregará os dados do consumo de gás dos equipamentos é necessário trabalhar na automatização da coleta de dados, armazenamento de dados e criar painéis de gráficos e indicadores que entreguem todas as informações relevantes do processo.

Figura 15 : Diagrama do fluxo de dados.



Fonte: Imagem do autor (2025).

3.1 Especificações e Requisitos

O objetivo principal desse trabalho é trabalhar na construção de uma aplicação que torne mais eficiente o processo de obtenção de dados e informações relevantes ao consumo de gás, centralizando os dados relevantes, facilitando acesso por qualquer colaborador e mantendo os dados atualizados. Para garantir esses aspectos, os requisitos para essa aplicação são:

- Banco de dados centralizado com acesso por API.
- Plataforma para visualização de dados com interface gráfica interativa.

- Compatibilidade com infraestrutura existente.

Para o cumprimento dos objetivos gerais e específicos deste estudo, torna-se necessária a seleção dos *softwares* que comporão os blocos fundamentais da solução, bem como a devida configuração e integração entre eles.

3.2 Análise do Contexto

Na planta industrial analisada neste estudo, existe um processo de coleta de dados para destrinchar o uso de gás natural por duas linhas de produção, chamadas de A e B nesse estudo, ambas linhas partilham da mesma caldeira para gerar vapor para o processo, porém uma delas, a linha A, também usa gás natural diretamente no processo, como visto anteriormente na Figura 1.

Atualmente, a coleta de dados é realizada de forma manual. Um funcionário deve acessar diariamente o site da Copergás para obter informações sobre o consumo total de gás natural do dia anterior. Em seguida, ele utiliza uma aplicação *web* de uma empresa terceira para coletar os dados de consumo de gás da linha A e, por fim, realiza medições em campo para registrar o consumo de vapor das linhas A e B em um *datalogger*.

Na planta, existem quatro transmissores de vazão: três do tipo vórtex e um do tipo turbina. O primeiro é um medidor de gás tipo turbina instalado pela Copergás, circulado em vermelho na Figura 16, responsável por medir o consumo total da fábrica. O segundo medidor refere-se ao consumo de gás da linha A, enquanto os terceiros e quartos medidores monitoram o consumo de vapor das duas linhas de produção, ambos podem ser visualizados na Figura 17, circulados em vermelho.

Figura 16: Transmissão de vazão de gás da Copergás.



Fonte: Imagem do autor (2025).

Figura 17: Tubulação de distribuição de vapor.



Fonte: Imagem do autor (2025).

Como a aplicação será desenvolvida para operar em um servidor na rede corporativa, é fundamental garantir o acesso, por meio dessa rede, aos dados de medições dos instrumentos envolvidos no projeto. Para isso, é necessário considerar

a saída analógica desses equipamentos, conectando os medidores a um dispositivo que possa ler esses dados e permitir sua consulta através da infraestrutura de rede.

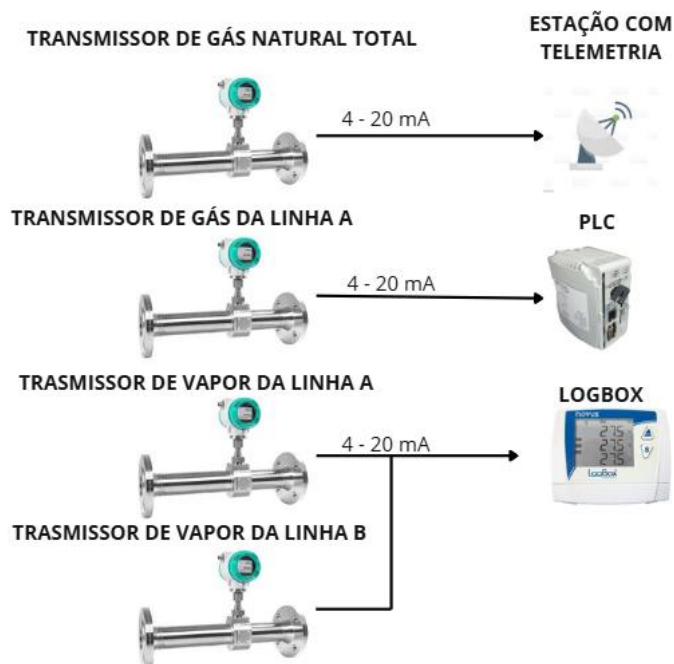
A Copergás já adota essa abordagem para o envio de dados aos seus servidores, utilizando interfaces de telemetria responsáveis pelo monitoramento contínuo do consumo de gás natural nos estabelecimentos atendidos. Como o projeto Gás online, a Copergás criou a possibilidade de comunicação com seus servidores através de APIs, disponibilizando aos seus clientes uma maneira simples de ter acesso em tempo real aos seus dados de consumo através de requisições HTTP.

Para acessar os dados dos outros dispositivos, é necessário utilizar uma interface semelhante. O transmissor de vazão usado para gás natural na linha A já está conectado a um CLP da Rockwell 1769-L36erm, que por sua vez está conectado na rede corporativa. Portanto, estabelecer uma conexão com esse CLP é essencial para obter esses dados.

Os transmissores de vapor, da série Yewflo, da Yokogawa, foram conectados a um *Logbox Wi-Fi* da Novus, visando facilitar futuras integrações. O *Logbox Wi-Fi* é um dispositivo *IoT* com *datalogger* integrado e conectividade *Wi-Fi*, que permite o monitoramento através dos protocolos MQTT ou Modbus TCP em instalações com infraestrutura de rede *Wi-Fi*. No entanto, como essa integração ainda não havia sido realizada, um colaborador precisava ir diariamente ao campo para fazer o *backup* desses dados.

Os sensores utilizados neste projeto, ilustrados na Figura 18, incluem quatro medidores de vazão. Além disso, estão presentes os dispositivos que possibilitam o acesso às medições: um CLP, um *datalogger* e o servidor da Copergás.

Figura 18: Equipamentos disponíveis na indústria.



Fonte: Imagem do autor (2025).

3.3 Escolha dos Protocolos

É fundamental utilizar protocolos de comunicação para coletar dados de diferentes equipamentos localizados em diversas áreas da indústria. Para obter informações sobre o consumo total de gás da Copergás, é necessário consultar a documentação da API que eles desenvolveram, a qual utiliza o protocolo HTTP para acessar os dados mensurados.

O CLP *CompactLogix* suporta três protocolos de comunicação via rede *Ethernet*: Modbus TCP, Ethernet/IP e MQTT. Por outro lado, o *Logbox Wi-Fi* da Novus é compatível apenas com Modbus TCP e MQTT. Para simplificar a rede, é preferível utilizar o mesmo protocolo para ambos os dispositivos, descartando, assim, a opção de utilizar EtherNET/IP.

Ao escolher entre Modbus TCP e MQTT, a escalabilidade foi um fator determinante. No caso do Modbus, é necessário interpretar os dados de cada

equipamento individualmente, através de endereços específicos. Em contrapartida, a estrutura de tópicos do MQTT simplifica tanto o comissionamento quanto a organização geral da rede.

3.4 Escolha da Ferramenta de Coleta de Dados

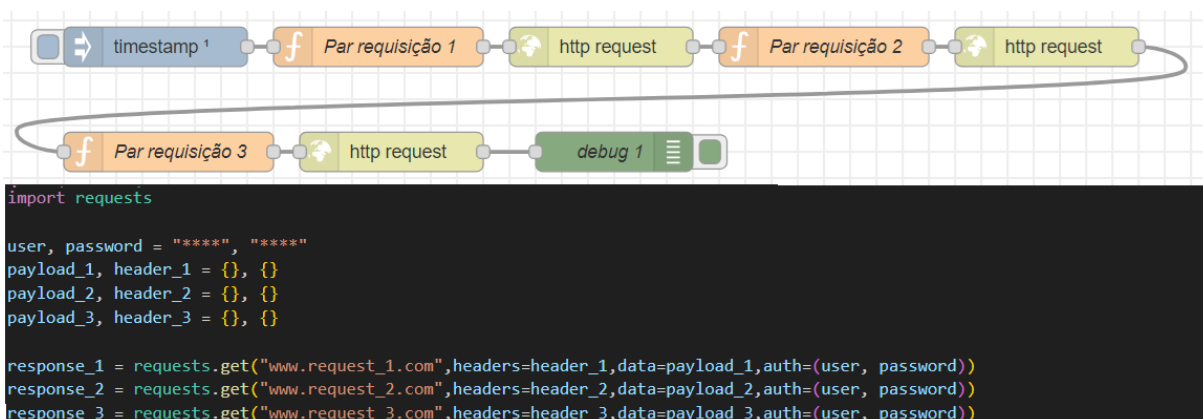
A Copergás disponibiliza seus dados de medição em tempo real por *API*. O medidor de gás do processo da linha A é conectado a um *CLP* da Rockwell com a possibilidade de fazer uso de 3 protocolos, sendo eles EtherNET/IP, Modbus TCP e MQTT, e os medidores de vazão de vapor são conectados a um *Logbox*, da Novus, que pode ser configurado para funcionar com MQTT ou Modbus TCP.

Para tal finalidade, faz-se necessária a escolha de uma ferramenta que permita a utilização dos referidos protocolos a fim de estabelecer a comunicação com os equipamentos. Dentre as opções disponíveis, destaca-se o *Node-RED*, amplamente utilizado na atualidade. Alternativamente, é comum a elaboração de scripts em *Python* com o propósito de automatizar processos e executar tarefas repetitivas de forma eficiente.

Ao analisar como seria o processo de elaborar uma aplicação em *Python* em relação a realizar algo comparável no *Node-RED*, é possível destacar alguns pontos chave. O *Node-RED* é programado a partir de uma interface gráfica pela junção de nós que ligam informação de um ponto ao outro ao mesmo tempo que permitem modificá-la através de funções, enquanto *Python* é uma linguagem de programação que exige a criação de *script* pelo uso de linhas de código.

Em atividades compostas por múltiplas etapas sequenciais, como por exemplo três requisições HTTP com *payload*, *header* e autenticação, é possível ter uma visão geral do código escrito em *Python*, por outro lado são necessários 6 nós no *Node-RED*, onde só é possível visualizar os parâmetros de 1 nó por vez. Essa situação pode ser observada na Figura 19, onde conseguimos visualizar todos os parâmetros nas linhas de código em *Python*, enquanto no *Node-RED* teríamos que clicar em cada nó individualmente para analisar os parâmetros.

Figura 19 : Realização de 3 requisições HTTP utilizando *Node-RED* e *Python*.



Fonte: Imagem do autor (2025).

No que se refere ao gerenciamento de erros e comportamentos inesperados, observou-se que falhas não tratadas em *scripts Python* podem interromper a execução da aplicação, ocasionando possíveis indisponibilidades. Por outro lado, no ambiente *Node-RED*, tais erros são apenas reportados na janela de depuração, permitindo que a aplicação mantenha sua operação normalmente. Um exemplo ilustrativo ocorre quando um site se torna temporariamente indisponível: se a requisição correspondente não for devidamente tratada em *Python*, o programa pode ser interrompido, enquanto no *Node-RED*, esse tipo de falha não compromete a continuidade do fluxo.

Foi concluído que para tarefas com poucos nós, usar o *Node-RED* simplifica a aplicação e diminui o tempo de desenvolvimento, porém para tarefas que precisem de um número alto de ações sequenciais a serem realizadas para atingir um fim, fazer em *Python* pode ser mais simples. Como a aplicação nesse trabalho irá receber dados para salvar num banco de dados, o *Node-RED* apresenta a maior compatibilidade.

3.5 Escolha do Banco de Dados

Segundo (HABBEMA), dois bancos de dados populares para o armazenamento de séries temporais são o *InfluxDB* e o *TimescaleDB*, ambos desenvolvidos com o objetivo de armazenar e consultar dados desse tipo. O *InfluxDB* foi especificamente

criado para séries temporais, utilizando uma modelagem de dados simples e direta, baseada em medidas, *tags* e valores, o que facilita sua aplicação em sistemas de monitoramento e IoT.

O *TimescaleDB*, por outro lado, é uma extensão do *PostgreSQL*, adaptando seu modelo relacional para lidar com séries temporais. Sua principal vantagem sobre o *InfluxDB* é a compatibilidade com a infraestrutura do *PostgreSQL*, permitindo o uso de dados relacionais e séries temporais em conjunto, além de oferecer consultas avançadas e suporte ao *SQL* padrão. No entanto, essa flexibilidade traz uma desvantagem: a maior complexidade ao lidar com a interface e gerenciamento do *PostgreSQL*, especialmente para quem busca uma solução mais simples.

Como o armazenamento de dados relacionais não é uma necessidade neste trabalho, as vantagens oferecidas pelo *TimescaleDB* não se aplicam. O *InfluxDB*, por outro lado, é especificamente projetado para otimizar o armazenamento e a consulta de séries temporais, tornando-se uma solução mais simples e prática para o objetivo de monitorar o consumo de gás e fornecer uma visualização clara do perfil de consumo.

3.6 Escolha de Ferramenta de Visualização

Ambos *Grafana* e *Power BI* são usados atualmente pelo setor de tecnologia da informação (TI), pertencente a empresa alvo desse trabalho, para visualização de dados. Dada a consolidação do *Power BI* como solução corporativa, justifica-se a realização de uma comparação entre essas plataformas no contexto da presente aplicação.

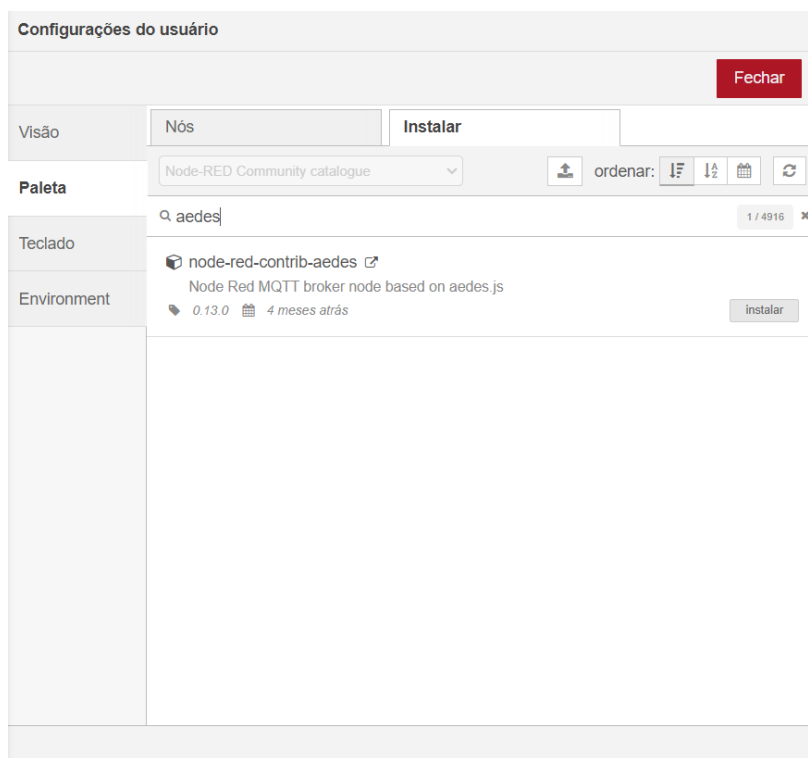
Sob a perspectiva da compatibilidade com bancos de dados, o *Power BI* apresenta suporte nativo ao *PostgreSQL* e outros bancos de dados relacionais. Isso demonstra que ele é otimizado para análise por lotes de dados, o que o torna adequado para análises periódicas. Em contrapartida, o *Grafana* é compatível com o *InfluxDB* e o *Postgre*, foi concebido com ênfase em atualizações contínuas e em tempo real, o que o torna particularmente adequado para aplicações que exigem monitoramento com atualizações contantes.

No caso do *Grafana*, quando executado em um servidor local, o compartilhamento dos *dashboards* ocorre por meio do navegador, de forma gratuita, desde que haja acesso à rede em que o servidor está hospedado. Por outro lado, os *dashboards* desenvolvidos no *Power BI* necessitam ser publicados no *Power BI Service* para possibilitar o compartilhamento, o qual está restrito a usuários que possuam licenças compatíveis com os recursos de colaboração da plataforma.

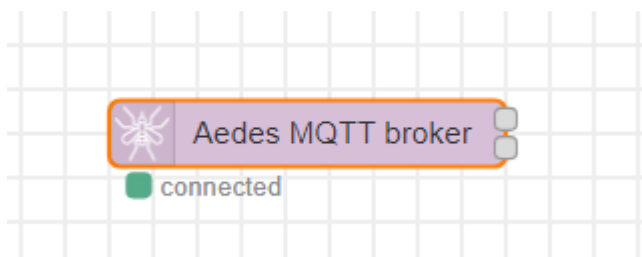
Considerando as necessidades da aplicação proposta, que envolvem monitoramento contínuo de dados e atualização em tempo real, o *Grafana* demonstra-se mais alinhado aos requisitos estabelecidos neste trabalho. Sua compatibilidade nativa com bancos de dados de séries temporais, como o *InfluxDB*, somada à possibilidade de compartilhamento local sem custos adicionais, confere maior flexibilidade, escalabilidade e economia à solução.

3.7 Comunicação MQTT com o Node-RED

Para realizar a comunicação usando o protocolo MQTT, é necessário ter um *broker* para intermediar a comunicação. Utilizando a biblioteca *node-red-contrib-aedes*, que pode ser baixada diretamente do *Node-RED*, como visto pela Figura 20, é possível criar esse agente intermediário de mensagens. Esta biblioteca é baseada no *broker* de código aberto chamado *Mosquitto*. A Figura 21 mostra o bloco responsável pelo *broker*.

Figura 20: Instalação do *broker* no *Node-RED*.

Fonte: Imagem do autor (2025).

Figura 21 : Bloco de *broker MQTT* no *Node-RED*.

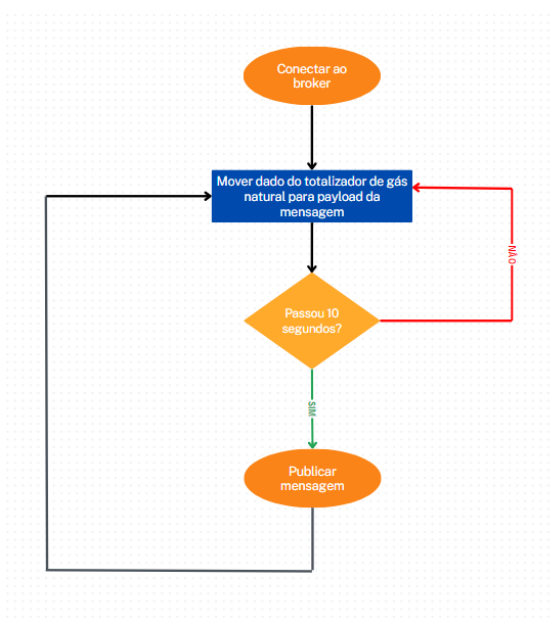
Fonte: Imagem do autor (2025).

3.8 Comunicação com CLP *CompactLogix*

A programação dos CLPs *CompactLogix* mais recentes é realizada por meio do *software Studio 5000* da Rockwell. Utilizando a linguagem de programação *Ladder*, as funções de comunicação são implementadas por meio de blocos específicos, previamente disponibilizados pela própria Rockwell.

Para publicar o valor do totalizador de gás a cada 10 segundos é usado o código em *Ladder* disponível na Figura 45, presente no Apêndice A, ele se comporta como no fluxograma visualizado na Figura 22. Além das configurações no código, foi feito a configuração do *IP* do *broker* e da identificação de cliente, como presente na Figura 23.

Figura 22: Fluxograma de publicação de dados CLP da linha A.



Fonte: Imagem do autor (2025).

Figura 23 : Configuração do IP do *broker* e da identificação do CLP.

▼ Hndl	Public	{...}
▼ Hndl.IP_address		{...}
▶ Hndl.IP_address[0]		192
▶ Hndl.IP_address[1]		168
▶ Hndl.IP_address[2]		0
▶ Hndl.IP_address[3]		103
▶ Hndl.ClientId		'LogixMqttClient_linha2'

Fonte: Imagem do autor (2025).

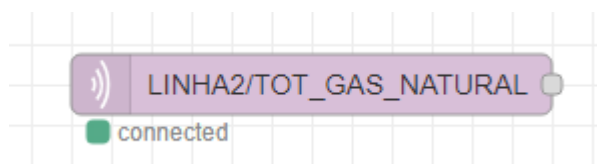
Para receber o dado no *Node-RED*, é necessário usar o bloco de comunicação MQTT in presente na Figura 24, precisando configurar o *broker* e o tópico, como presente na Figura 25, com os parâmetros presentes na Tabela 5.

Tabela 5: Parâmetros para envio de dados do CLP para o *Node-RED*.

IP do broker	192.168.0.103
Tópico	LINHA2/TOT_GAS_NATURAL

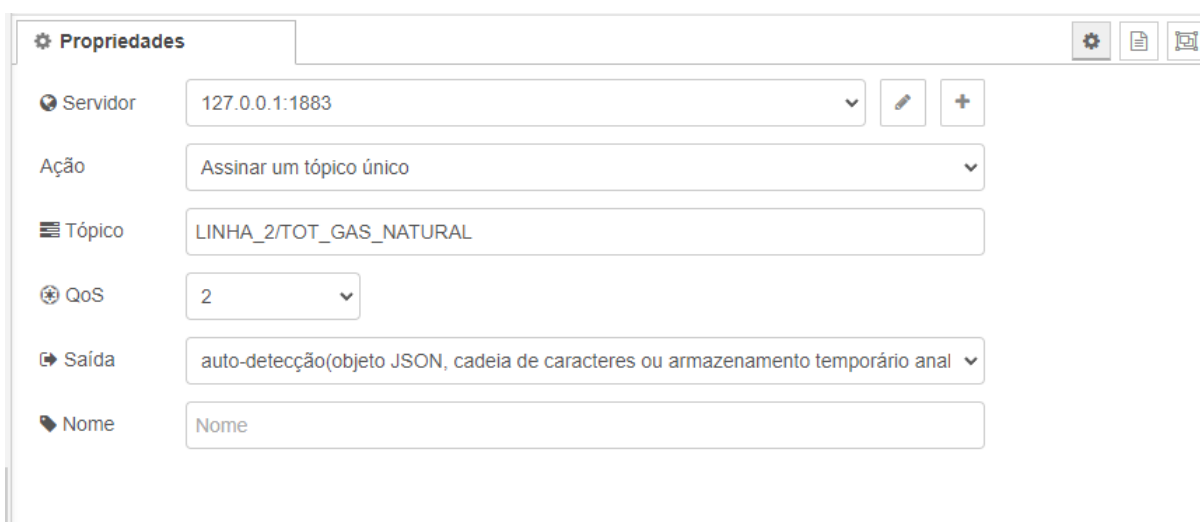
Fonte: Tabela do autor (2025).

Figura 24 : Nó MQTT inscrito no tópico do totalizador de gás natural do processo da linha 2.



Fonte: Imagem do autor (2025).

Figura 25 : Configuração do nó de entrada MQTT.



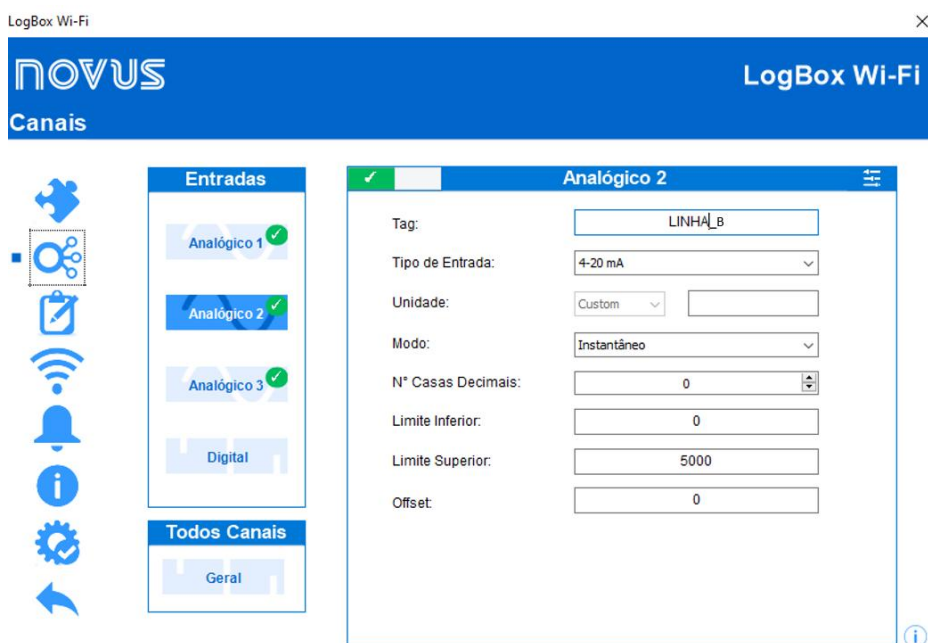
Fonte: Imagem do autor (2025).

3.9 Comunicação MQTT com Logbox Wi-Fi

O *Logbox* é um dispositivo que, nesta aplicação, recebe dois sinais analógicos de 4 a 20 miliamperes provenientes de medidores de vazão, que monitoram o fluxo de vapor em kg/h em ambas as linhas. Esses dados são então publicados em formato JSON via protocolo MQTT. Após realizar as conexões elétricas, é necessário configurar o equipamento para reportar as informações por MQTT, processo feito através do *software NXperience*, fornecido pela Novus.

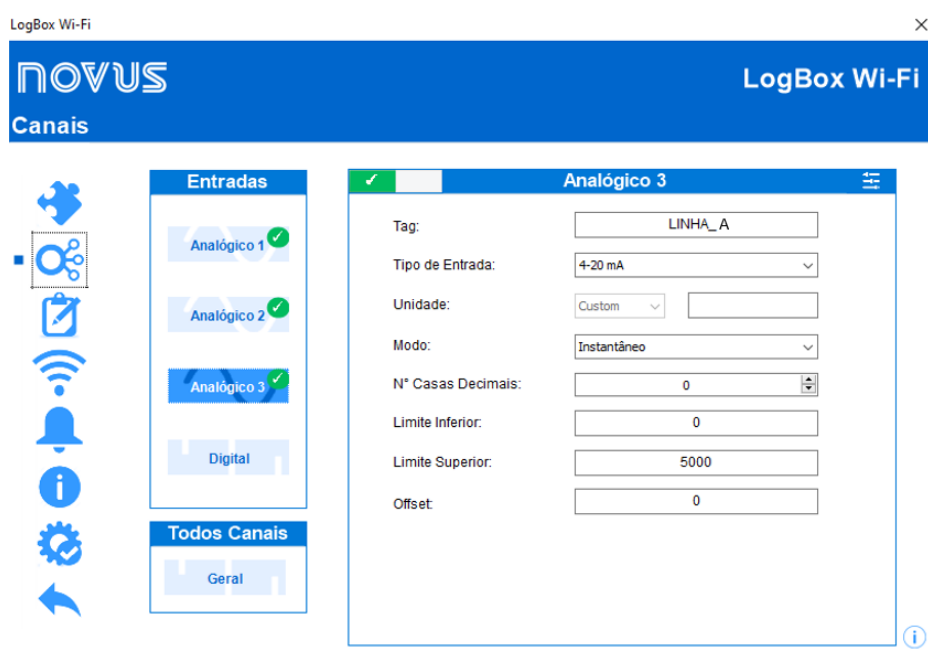
A configuração requer o cumprimento de duas etapas principais. A primeira consiste no estabelecimento dos parâmetros necessários para converter a leitura de corrente elétrica nas respectivas grandezas físicas aferidas pelos sensores de campo. A segunda etapa envolve a configuração da comunicação via protocolo MQTT. As configurações dos dois canais utilizados podem ser observadas na Figura 26 e na Figura 27, sendo necessário informar o *range* medido pelo sensor. Já as configurações relativas à comunicação estão apresentadas na Figura 28, onde foi apenas necessário informar o *broker* e a porta. Nessa mesma figura, é possível visualizar o tópico no qual o dispositivo publica as informações, conforme destacado em vermelho. Para receber os dados no *Node-RED*, é usado o nó MQTT como configurado na Figura 29.

Figura 26: Configuração da entrada analógica para receber sinal do transmissor de vazão de vapor da linha B.



Fonte: Imagem do autor (2025).

Figura 27: Configuração da entrada analógica para receber sinal do transmissor de vazão de vapor da linha A.



Fonte: Imagem do autor (2025).

Figura 28: Configuração de comunicação MQTT do *Logbox*.

The screenshot shows the 'LogBox Wi-Fi' configuration window with the 'MQTT' tab selected. The settings are as follows:

- Habilitar MQTT:** **Habilitado**
- Tópico de Publicação:** novus/19247104/[...] (highlighted with a red box)
- Usuário do Broker:** [Empty field]
- Senha do Broker:** [Empty field]
- URL ou IP do Broker:** 192.168.0.103
- Porta de Serviço:** 1883
- QoS:** 2
- Formato de Tempo:** DateTime
- Formato JSON:** Array estático
- Formato de booleana:** Numérico
- Segurança:** Nenhuma
- Modo Clean:** **Habilitado**
- Permissões de Escrita:**
 - Buzzer
 - Relógio Interno:
 - Limites de alarmes
 - Ajuste de Download

Fonte: Imagem do autor (2025).

Figura 29: Configuração do nó de recebimento no *Node-RED*.

The screenshot shows the 'Editar mqtt in nó' configuration window in Node-RED. The properties are as follows:

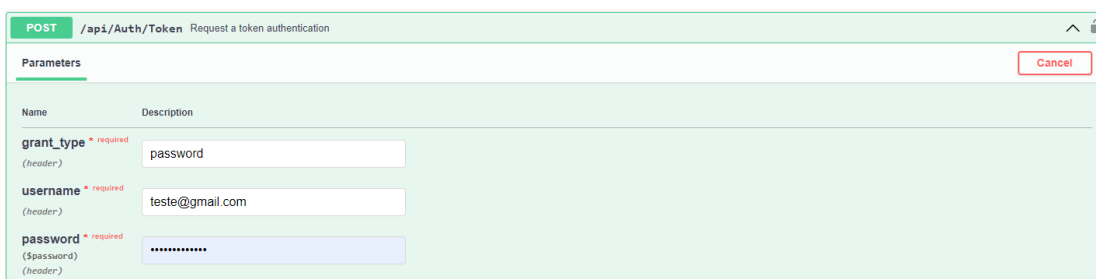
- Servidor:** 127.0.0.1:1883
- Ação:** Assinar um tópico único
- Tópico:** novus/19247104
- QoS:** 2
- Saída:** auto-deteção(objeto JSON, cadeia de caracteres ou armazenamento temporário ana)
- Nome:** Nome

Fonte: Imagem do autor (2025).

3.10 Comunicação HTTP com *Node-RED*

O *Node-RED* já disponibiliza, por padrão, os nós necessários para a implementação deste protocolo de comunicação. Para sua correta utilização, é suficiente consultar a documentação da Copergás, presente em (COPERGÁS, 2017), na qual são descritos os procedimentos exigidos, incluindo a informação de que o intervalo de atualização dos dados é de dois minutos. A implementação requer duas etapas: a primeira consiste na autenticação e obtenção do *token* de acesso, conforme ilustrado na Figura 30, enquanto a segunda refere-se à consulta propriamente dita, direcionada ao totalizador convertido, conforme descrito na Figura 31.

Figura 30: *Url* para obtenção de token e autenticação na documentação da Copergás.

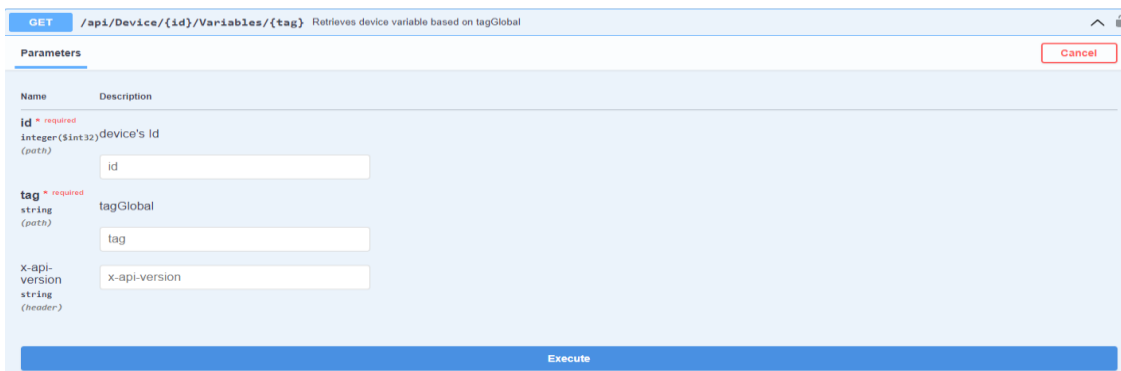


The screenshot shows a Node-RED HTTP Request node configuration. The method is POST and the URL is /api/Auth/Token. The description is 'Request a token authentication'. The parameters section is expanded, showing a table with columns 'Name' and 'Description'. There are three required parameters: 'grant_type' (header) with value 'password', 'username' (header) with value 'teste@gmail.com', and 'password' (header) with a masked value '*****'. A 'Cancel' button is visible in the top right corner.

Name	Description
grant_type * required (header)	password
username * required (header)	teste@gmail.com
password * required (\$password) (header)	*****

Fonte: Adaptado de (COPERGÁS, 2017).

Figura 31: *Url* para consulta de variáveis mensuradas pela Copergás.



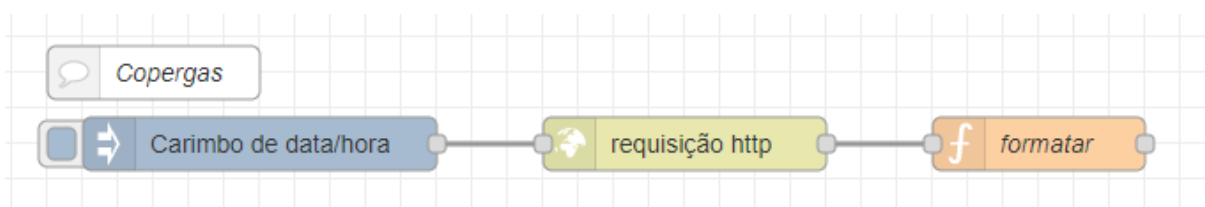
The screenshot shows a Node-RED HTTP Request node configuration. The method is GET and the URL is /api/Device/{id}/Variables/{tag}. The description is 'Retrieves device variable based on tagGlobal'. The parameters section is expanded, showing a table with columns 'Name' and 'Description'. There are three required parameters: 'id' (path) with value 'id', 'tag' (path) with value 'tag', and 'x-api-version' (header) with value 'x-api-version'. A 'Cancel' button is visible in the top right corner. An 'Execute' button is visible at the bottom.

Name	Description
id * required integer (\$int32) (path)	device's Id
tag * required string (path)	tagGlobal
x-api-version string (header)	x-api-version

Fonte: Adaptado de (COPERGÁS, 2017).

Após obter o *token* de autenticação, basta realizar uma requisição do tipo *GET* para recuperar o valor do totalizador de gás natural, conforme ilustrado na Figura 32, com as configurações como disposto na Figura 33. No *Node-RED*, essa operação pode ser executada usando o nó *HTTP Request*. Para isso, é necessário configurar o nó adequadamente, inserindo a *URL* do *endpoint* e os parâmetros de autenticação no formato especificado. A integração do nó *HTTP* com o fluxo de dados permitirá que as respostas sejam recebidas e processadas diretamente no *Node-RED*, simplificando o acesso e o uso dos dados no sistema.

Figura 32: Nós para leitura de variáveis da Copergás.



Fonte: Imagem do autor (2025).

Figura 33: Configuração da requisição do tipo *GET* para obter medida da Copergás.

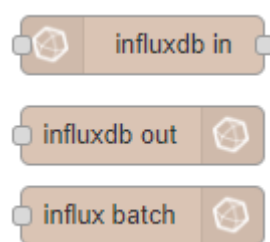
Fonte: Imagem do autor (2025).

3.11 Integração do *InfluxDB*

A biblioteca “*node-red-contrib-influxdb*” do *Node-RED* oferece nós específicos para facilitar a inserção e consulta de dados no *InfluxDB*, como ilustrado na Figura 34. Essa biblioteca permite diversas operações, como salvar uma única medida, realizar consultas ou inserir várias medidas simultaneamente. Para esta aplicação específica, onde há quatro variáveis a serem armazenadas, os dois totalizadores de gás natural e a vazão de vapor das duas linhas, foi decidido consolidar esses dados em uma única medida chamada "GAS". Essa estratégia visa centralizar todas as informações relacionadas em um único ponto, simplificando o gerenciamento e a consulta posterior.

Além da estruturação das medidas, é essencial configurar o banco de dados e definir uma política de retenção de dados apropriada para otimizar o uso de recursos de armazenamento. Para este projeto, foi estabelecida uma política de retenção de 90 dias. Essa decisão foi tomada para evitar o acúmulo excessivo de dados, prevenindo o uso desnecessário de memória e garantindo que o sistema mantenha dados recentes e relevantes para a análise, enquanto elimina automaticamente os dados antigos.

Figura 34: Nós para comunicação com servidor com *InfluxDB*.

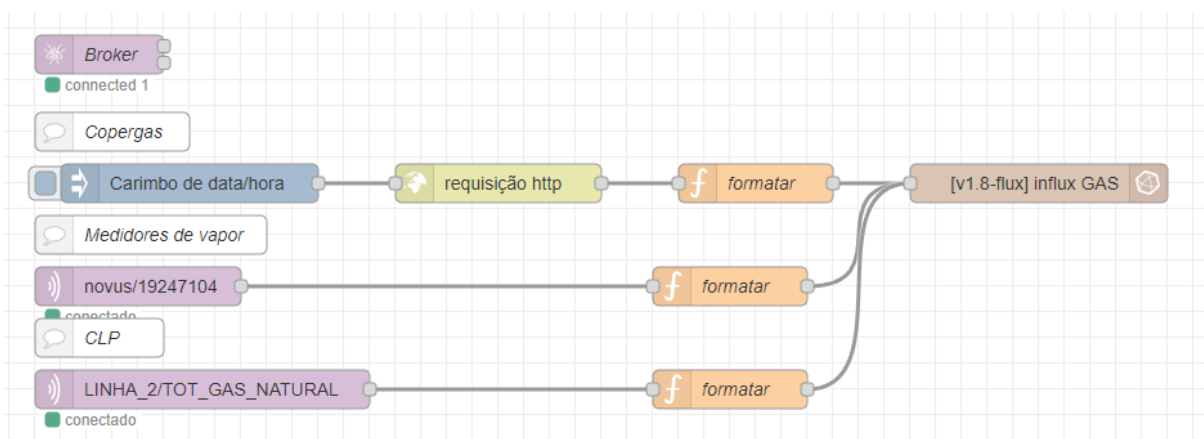


Fonte: Imagem do autor (2025).

Ao escolher essa abordagem, a aplicação garante uma organização eficiente dos dados dentro do *InfluxDB*, proporcionando uma forma clara e acessível de acessar métricas históricas sem comprometer a performance do banco de dados ao longo do tempo. A política de retenção adequada também é fundamental para manter

um equilíbrio entre a quantidade de dados armazenados e o desempenho geral do sistema. Por fim, o nó do *InfluxDB* é adicionado como nó final dos fluxos de coleta de dados, resultando na Figura 35.

Figura 35: Integração entre fonte de dados e o banco de dados.

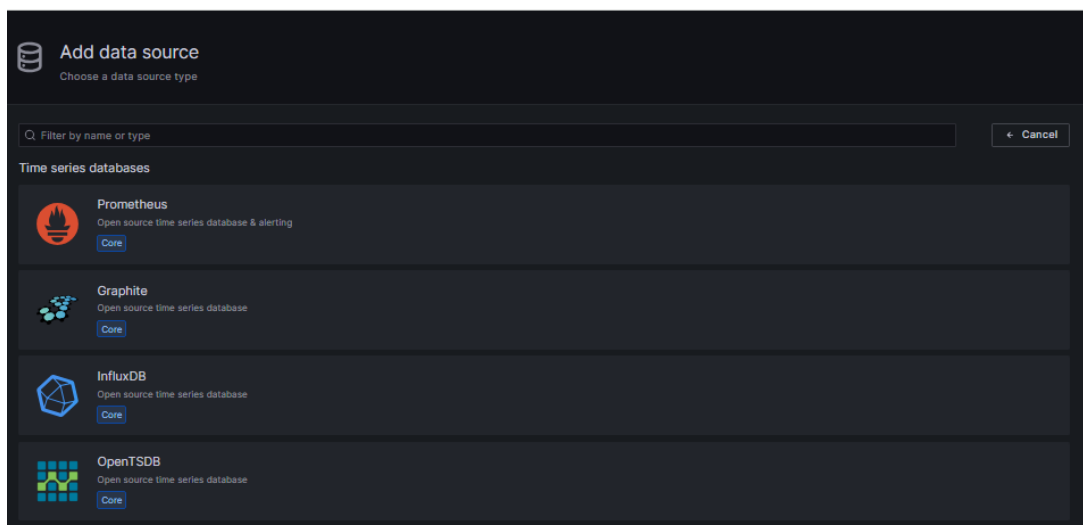


Fonte: Imagem do autor (2025).

3.12 Visualização de Séries Temporais com o Grafana

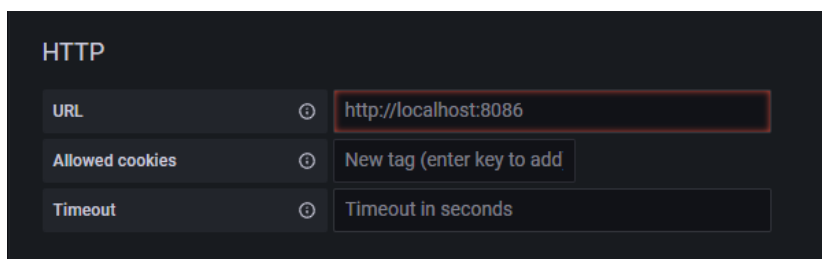
O *Grafana*, por padrão, oferece integração nativa com o *InfluxDB*. Para configurá-lo, basta selecionar o *InfluxDB* como o banco de dados desejado, conforme ilustrado na Figura 36. Em seguida, na seção de configurações, é necessário inserir o endereço IP e a porta da máquina que hospeda o banco de dados, como o banco de dados e a plataforma de visualização funcionarão na mesma máquina, pode ser usado o padrão já mostrado no campo a ser preenchido, como visto na Figura 37. Por fim, basta apenas adicionar a base de dados criada para armazenar as medições desse trabalho, no campo de *Database*, presente na Figura 38.

Os demais campos nesta etapa de configuração são referentes a parâmetros de segurança e parâmetros mais específicos de comunicação, que são opcionais e fogem do foco deste trabalho.

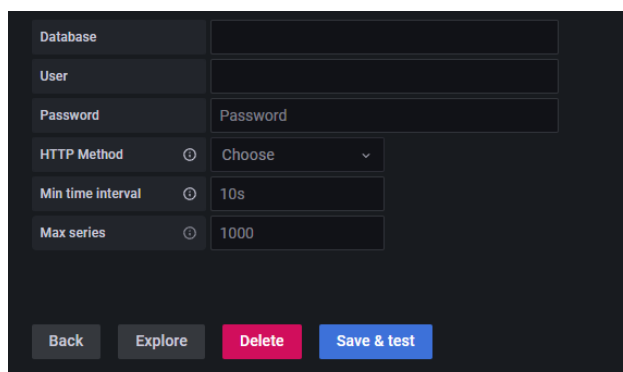
Figura 36: Opções para fontes de dados no *Grafana*.

Fonte: Imagem do autor (2025).

Figura 37: Adicionando endereço do banco de dados.



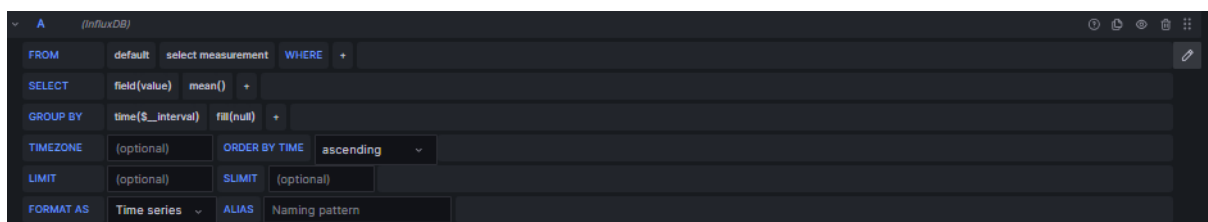
Fonte: Imagem do autor (2025).

Figura 38: Adicionando a base de dados no *Grafana*.

Fonte: Imagem do autor (2025).

Com o nome da base de dados inserido corretamente, o *Grafana* realiza automaticamente uma varredura de todas as medidas armazenadas, permitindo que sejam selecionadas diretamente por meio de sua interface gráfica, presente na Figura 39. Esse recurso simplifica significativamente a criação de *dashboards*, reduzindo o processo a apenas selecionar as medições desejadas com o uso do mouse.

Figura 39: Consulta gráfica no *Grafana*.



Fonte: Imagem do autor (2025).

O *Grafana* fornece algumas opções de gráfico, disponíveis na Tabela 6. O primeiro gráfico a ser feito é o de medição de vapor, levando em que são recebidos os dados de vazão instantânea, foi escolhido fazer o gráfico de linha para cada transmissor de vapor e colocar um *Stat* para a última medição.

Tabela 6: Tipos de gráfico no Grafana.

Tipos de Gráfico no Grafana
<i>Time series</i>
<i>Bar chart</i>
<i>Stat</i>
<i>Gauge</i>
<i>Bar gauge</i>
<i>Table</i>
<i>Pie chart</i>
<i>State timeline</i>
<i>Heatmap</i>
<i>Status history</i>
<i>Histogram</i>
<i>Candlestick</i>
<i>Node Graph</i>
<i>Traces</i>
<i>XY Chart</i>

Fonte: Tabela do autor (2025).

O indicador mais importante é o cálculo de percentual de uso de vapor por linha, que levará em conta a média das leituras das duas variáveis de vapor vistas nos gráficos de linha, no *Grafana* é apenas selecionando a média dessas variáveis, como na Equação (3.1) para a linha A e (3.2) para a linha B.

$$Vapor LA = \frac{Media\ do\ vapor\ da\ linha\ A}{Media\ do\ vapor\ da\ linha\ A + Media\ do\ vapor\ da\ linha\ B} \quad (3.1)$$

$$Vapor LB = \frac{Media\ do\ vapor\ da\ linha\ B}{Media\ do\ vapor\ da\ linha\ A + Media\ do\ vapor\ da\ linha\ B} \quad (3.2)$$

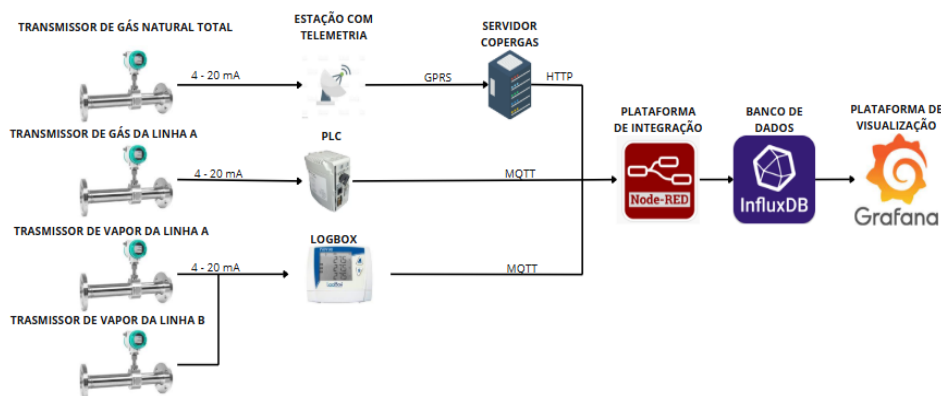
Para os gráficos de consumo de gás é necessário colocar o consumo de gás aferido pelo medidor da Copergás e o da linha A, feito subtraindo o último ponto do

totalizador menos o primeiro na janela de tempo escolhida, e o histórico de vazão, feito subtraindo os pontos individuais do totalizador.

4 RESULTADOS E ANÁLISES

Foi criado uma solução completa para coleta, armazenamento e visualização de dados. Usando o *Node-RED* para coleta de dados, o *InfluxDB* para armazenamento e o *Grafana* para visualização, representada pelo diagrama presente na Figura 40.

Figura 40 : Diagrama da solução final.



Fonte: Imagem do autor (2025).

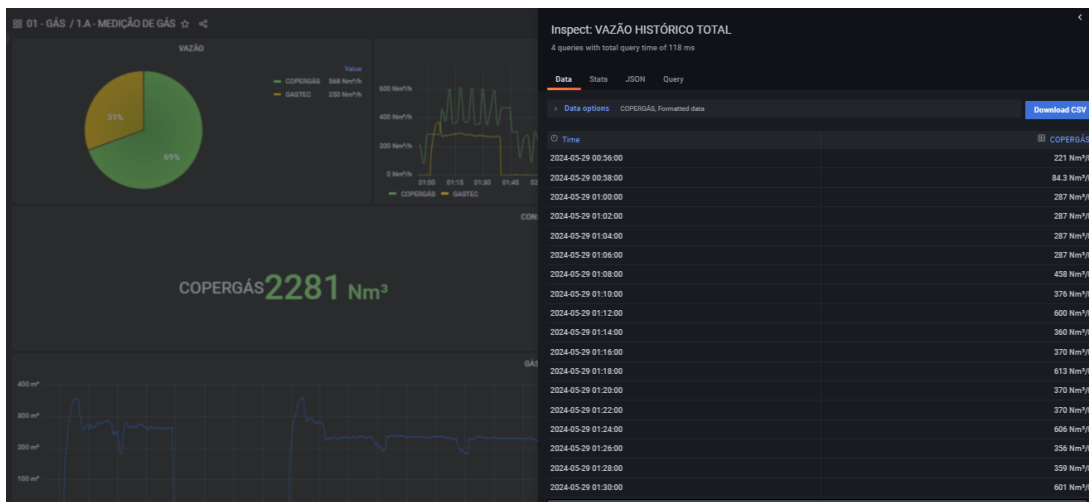
A utilização do *Grafana* como interface de monitoramento proporcionou uma visualização clara e eficiente dos dados, permitindo o acompanhamento em tempo real do consumo de vapor e gás natural na planta industrial, conforme demonstrado na Figura 41 e Figura 42. Além disso, através da plataforma é possível exportar os dados em formato CSV, como visto na Figura 43, facilitando análises externas por parte dos colaboradores, que podem utilizar ferramentas adicionais conforme suas preferências e necessidades.

Figura 41: Gráficos do consumo de vapor no *Grafana*.

Fonte: Imagem do autor (2025).

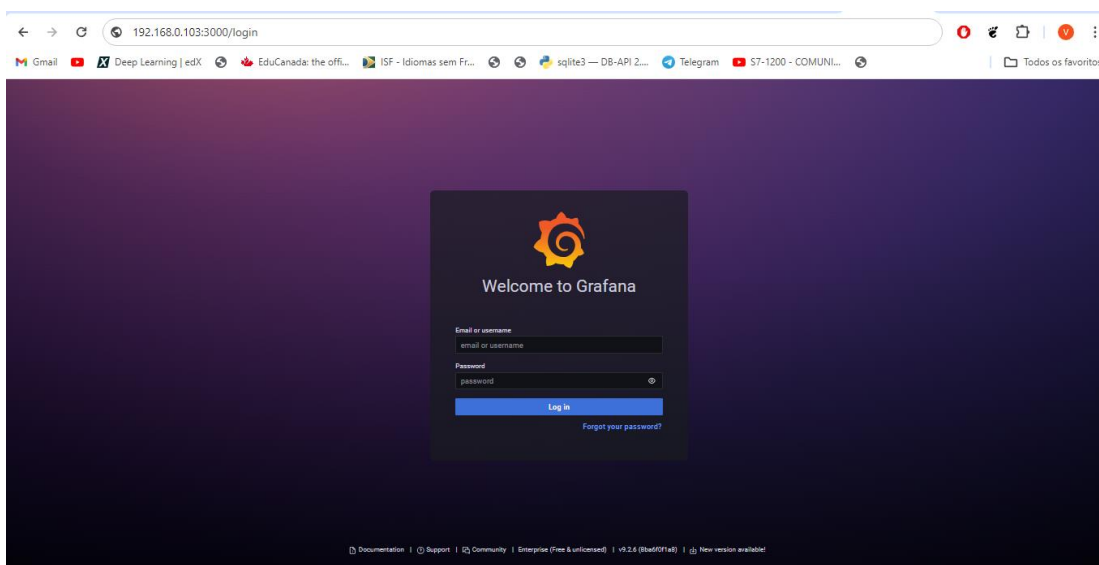
Figura 42: Gráfico do consumo de gás no *Grafana*.

Fonte: Imagem do autor (2025).

Figura 43: Exportando dados pelo *Grafana*.

Fonte: Imagem do autor (2025).

Essa implementação permitiu consultar os dados em tempo real e promoveu a democratização dos dados na rede da empresa, permitindo que qualquer pessoa autorizada tenha fácil acesso. O *login* pode ser realizado diretamente pelo navegador, como mostrado na Figura 44, com dados preestabelecidos pelo administrador do sistema.

Figura 44: Autenticação no *Grafana*.

Fonte: Imagem do autor (2025).

Essa centralização das informações proporcionou uma significativa redução no tempo necessário para a execução da tarefa, que passou de aproximadamente uma hora para apenas cinco minutos, uma redução de 91,7%. Esses ganhos foram obtidos graças à redução no número de etapas necessárias para realizar a atividade, que incluía a necessidade de se deslocar na área com um notebook para coletar dados dos *dataloggers*.

Os dados obtidos foram coletados sob condições normais de operação da planta, e a aplicação prática da solução evidenciou ganhos em termos de eficiência operacional, gestão de análises e acessibilidade à informação. Além disso, observou-se que o processo de treinamento de novos colaboradores foi agilizado, permitindo que dominassem a tarefa menor tempo.

5 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

É possível concluir que, conforme descrito nos objetivos propostos neste trabalho, foi realizada com êxito a implementação de uma solução para coleta, armazenamento e visualização de dados industriais em tempo real. A rotina de alimentação do banco de dados com os dados provenientes dos transmissores, conforme ilustrado na Figura 18, foi desenvolvida com sucesso.

Durante o trabalho foi analisado o uso de três protocolos de rede diferentes, sendo eles MQTT, EtherNET/IP e Modbus, para conectar dispositivos industriais. O protocolo MQTT se provou o mais adequado em compatibilidade, facilidade de organização e escalabilidade para esta aplicação.

A ferramenta escolhida para integrar os dispositivos ao banco de dados foi o *Node-RED*, pela agilidade de sua interface gráfica, que permite programar de forma simples e intuitiva, executar atividades de forma independente e identificar erros visualmente, facilitando a resolução.

Comparando a experiência do usuário com o *Grafana* em relação ao *Excel*, pode ser destacado o dinamismo do *Grafana*, sendo possível interagir com vários gráficos ao mesmo tempo. Assim como o próprio *Power BI*, mas com a adição de ser próprio para series temporais e gratuito para compartilhamento.

A integração de ferramentas de código aberto possibilitou o desenvolvimento de uma solução completa e econômica para resolver um problema relevante para a rotina da empresa. A implementação da proposta permitiu atingir os objetivos gerais e específicos deste estudo, construindo uma solução eficaz que aprimora o monitoramento do consumo de gás em ambiente industrial.

Uma tarefa que anteriormente levava cerca de uma hora, devido à necessidade de consultar diferentes fontes e equipamentos no campo, foi reduzida a aproximadamente 5 minutos, representando uma economia de 91,7% no tempo de execução. Além disso, foram observados ganhos na gestão de análises, uma vez que o procedimento para realizar essa foi drasticamente simplificado. A possibilidade de erro humano na coleta dos dados foi eliminada, assegurando, assim, uma maior confiabilidade nas informações obtidas.

Pela solução projetada trabalhar com séries temporais, sua capacidade de fornecer relatórios com valores de momentos específicos, como fechamento de turno ou final do dia, é limitada. Para gerar esses relatórios é necessário exportar os dados para análises externas, ou desenvolver um código que consulte o banco de dados nesses momentos específicos e gere o relatório desejado.

Pesquisas futuras poderiam explorar métodos de análise de dados com técnicas estatísticas clássicas ou de inteligência artificial diretamente do banco de dados, automatizando tarefas dos analistas de processo, visando extrair informações relevantes em tempo real. Também existe a possibilidade de inserir novos dispositivos que mensurem consumo de energia e matérias primas, reaproveitando a mesma arquitetura para gerar mais *dashboards* atualizados em tempo real.

REFERÊNCIAS

- AFTAB, Mohammad. The Complete Power BI Terminologies Should Be Aware by Every Data Analyst, 2024. Disponível em: <<https://medium.com/@aftab4092/the-complete-power-bi-terminologies-should-be-aware-by-every-data-analyst-5c396c402ec9>>. Acesso em: 22 set. 2024.
- AGUIRRE, L. A. **Enciclopédia de Automática**. 1ª. ed.
- AGUIRRE, L. A. **Fundamentos de instrumentação**. 1ª. ed.
- BOOZ, Ryan. PostgreSQL + TimescaleDB: 1,000x Faster Queries, 90 % Data Compression, and Much More, 2025. Disponível em: <<https://www.timescale.com/blog/postgresql-timescaledb-1000x-faster-queries-90-data-compression-and-much-more>>. Acesso em: 04 mar. 2025.
- C.J.DATE. **Introdução a Sistemas de Bancos de Dados**. 8ª. ed.
- COPERGÁS. **ScadaFlex 6 API - 1.0**, 2017. Disponível em: <<https://gasonline.copergas.com.br/api/doc/index.html>>. Acesso em: 5 Jun 2024.
- DELMÉE, Gérard J. **Manual de medição de vazão**. 3ª. ed.
- DUNNING, Ted; FRIEDMAN, Ellen. **Time Series Databases: New Ways to Store and Access Data**. 1ª. ed.
- EILSON, Lonnie. **How to Implement lean manufacturing**. 1ª. ed.
- FREITAS, Carlos M. Protocolo Modbus: Fundamentos e Aplicações, 2014. Disponível em: <<https://embarcados.com.br/protocolo-modbus/>>. Acesso em: 05 out. 2024.
- GRAFANA LABS. **Grafana Documentaion**, 2024. Disponível em: <<https://grafana.com/docs/>>. Acesso em: 13 mar 2025.
- GRAFANA LABS. Grafana data sources, 2024. Disponível em: <<https://grafana.com/docs/grafana/latest/datasources/>>. Acesso em: 09 mar. 2025.
- HABBEMA, Hugo. **Bancos de Dados de Séries Temporais**. Disponível em: <<https://medium.com/@habbema/bancos-de-dados-de-s%C3%A9ries-temporais-243ab07d52a3>>. Acesso em: 16 out. 2024.
- HAGINO, Taiji. **Practical Node-RED Programming: Learn powerful visual programming techniques and best practices for the web and IoT**. 1ª. ed.
- HIVEMQ. **MQTT Essentials**, 2024. Disponível em: <<https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>>. Acesso em: 21 jan 2024.
- IBM. Comunicação com aplicativos (MQTT), 2025. Disponível em: <<https://www.ibm.com/docs/pt-br/masv-and-l/maximo-monitor/cd?topic=reference-communicating-applications-mqtt>>. Acesso em: 12 mar. 2025.
- ILIINSKY, N.; STEELE, J. **Designing data visualizations**. 1ª. ed.
- INFUXDATA. **InfluxDB v1 Documentation**, 2024. Disponível em: <<https://docs.influxdata.com/influxdb/v1/>>. Acesso em: 13 mar. 2025.
- KOCAKULAK, Mustafa; BUTUN, Ismail. An overview of Wireless Sensor Networks towards internet of things. **IEEE Annual Computing and Communication Workshop and Conference (CCWC)**, 2017.
- KORTH, Henry F. . S. A.; SUDARSHAN, S. **Sistema de Banco de Dados**. 1ª. ed.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet uma abordagem top-down**. 6^a. ed.

MICROSOFT. O que é Power BI?, 2024. Disponível em: <<https://learn.microsoft.com/pt-br/power-bi/fundamentals/power-bi-overview>>. Acesso em: 15 jan. 2025.

MORETTIN PA, Toloi C. **Previsão de séries temporais**. 2^a. ed.

REZA LANGORI, Alan S. M. **Measurement and Instrumentation Theory and Application**. 2^a. ed.

SOTTILARE, Robert et al. **Design Recommendations for Intelligent Tutoring Systems**. 1^a. ed.

SPADINI, Allan; ALENCAR, Valquíria. **Séries temporais com Prophet: Análise e previsão de dados com Python**. 1^a. ed.

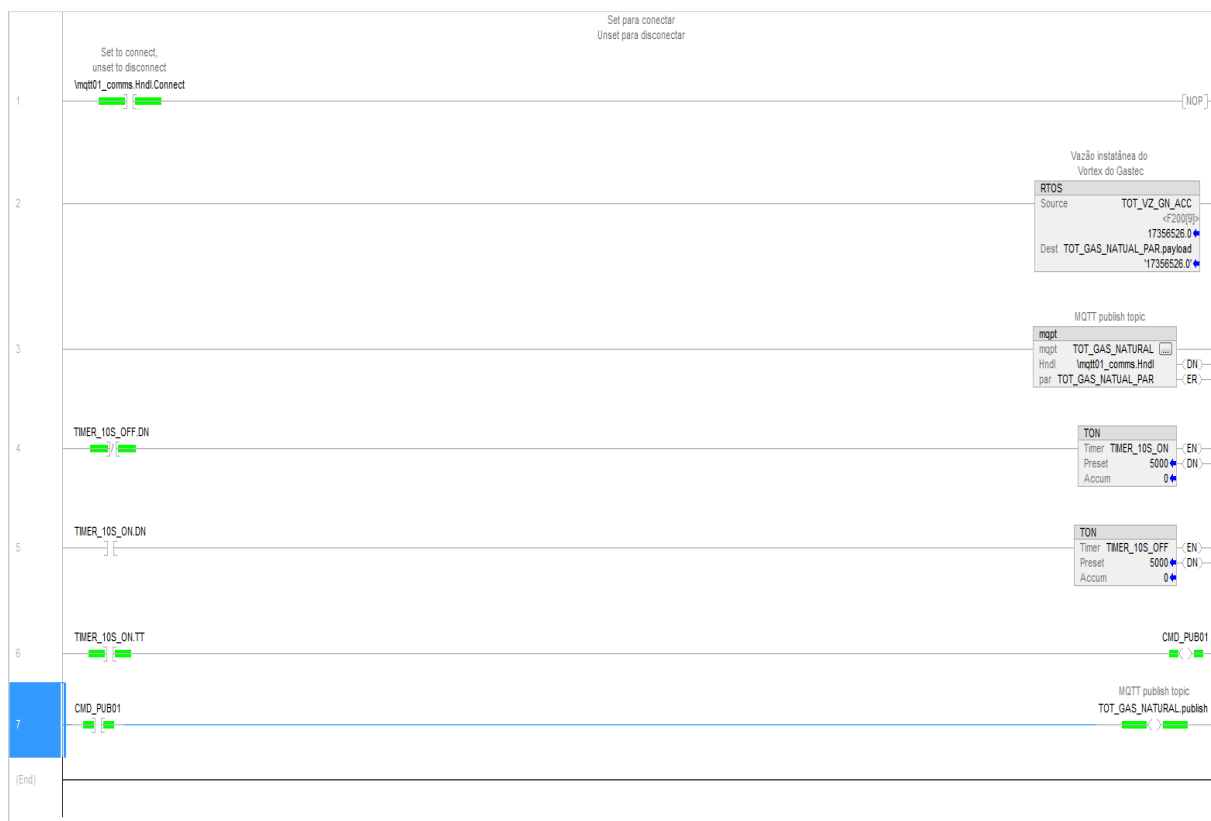
TARGET SOLUTIONS. DevOps: Novas Tendências em Ferramentas de Monitoramento, 2024. Disponível em: <https://www.targetso.com/2024/03/01/devops-ferramentas-de-monitoramento/?utm_source=chatgpt.com>. Acesso em: 2025 jan. 12.

TIMESCALE. TimescaleDB, 2025. Disponível em: <<https://github.com/timescale/timescaledb>>. Acesso em: 15 fev. 2025.

YOKOTANI, Tetsuya; SASAKI, Yuya. Comparison with HTTP and MQTT on Required Network Resources for IoT, 2016.

APÊNDICE A – CÓDIGO CLP ROCKWELL

Figura 45 : Código em Ladder desenvolvido para publicação de mensagem MQTT referente ao totalizador de gás natural usado no processo da linha A.



Fonte: Imagem do autor (2025).