

SISTEMA IOT INDUSTRIAL PARA MONITORAMENTO DE VIBRAÇÕES DE MÁQUINAS ROTATIVAS

[\[ver artigo online\]](#)

Jhone Fontenele¹
Daniel Barbutto Rossato²

RESUMO

Sistemas IoT industriais tem sido utilizado para coleta e análise de dados de processos industriais. Neste trabalho, buscamos desenvolver um sistema com tecnologias IoT para realizar o monitoramento de vibração de máquinas rotativas para utilização em sistemas de manutenção preditiva. O projeto utiliza um sensor de vibração de dois eixos a ser fixado em um motor. O sinal é enviado para um controlador lógico programável usando protocolo Modbus RTU. O histórico de medições é armazenado em um banco de dados relacional MySQL e apresentado por meio de um painel dashboard usando Grafana. A conexão entre o controlador e o banco de dados é feito por intermédio de fluxo desenvolvido em Node-RED usando protocolo OPC-UA. A análise inicial é realizada usando faixas de valores conforme normas técnicas para estabelecer níveis toleráveis de vibração. Foram realizados testes em bancada de ensaio comprovando o funcionamento do sistema; a metodologia e os resultados estão descritos neste artigo.

Palavras-chave: manutenção preditiva; IoT industrial; monitoramento de vibrações; OPC-UA; Node-RED.

INDUSTRIAL IOT SYSTEM FOR MONITORING ROTARY MACHINERY VIBRATIONS

ABSTRACT

Industrial IoT systems have been used to collect and analyze data from industrial processes. In this paper, we describe the development of a system containing IoT technologies to perform vibration monitoring of rotating machines for use in predictive maintenance systems. A two-axis vibration attached to a motor is used to generate the signal which is sent to a programmable logic controller using Modbus RTU protocol. Measurement history is stored in a MySQL relational database and presented through a dashboard using Grafana. Connection between controller and database is done by means a flow developed in Node-RED using OPC-UA protocol. Initial analysis were carried out using ranges of values in accordance with technical standards to establish tolerable levels of vibration. The system was successfully deployed and tested in a workbench; the methodology and results are reported in this paper.

Keywords: predictive maintenance; Industrial IoT; vibration monitoring; OPC-UA; Node-RED.

1 Especialista em Automação e Controle, Faculdade de Tecnologia SENAI "Mariano Ferraz", São Paulo, Jhone.fontenele@turck.com.

2 Professor orientador, Mestre em Engenharia Elétrica, Faculdade de Tecnologia SENAI "Mariano Ferraz", São Paulo, daniel.rossato@sp.senai.br.



INTRODUÇÃO

Este artigo se refere a um estudo de comportamento de máquinas rotativas, com base em manutenção preditiva, através de monitoramento de dados de vibrações. Este método de manutenção é descrito como: *“manutenção que permite garantir uma qualidade de serviço desejada, com base na aplicação sistemática de técnicas de análise, utilizando-se de meios de supervisão centralizados ou de amostragem, para reduzir ao mínimo a manutenção preventiva e diminuir a manutenção corretiva”* (NBR 5462, 1994). A importância para a indústria, se deve à exponencial redução de custos com materiais sobressalentes, gerenciamento de agenda para manutenção, bem como maior tempo de utilidade do ativo. *“Seu objetivo é elevar e garantir os índices de disponibilidade de um equipamento”* (OPTIMUS, 2019).

Apesar de existirem outras fontes de análise, como ultrassom, óleo e termografia, a análise de vibrações continua a ser a tecnologia mais popular principalmente em máquinas rotativas (PÉREZ; MÁRQUEZ, 2015). A análise de vibrações permite a detecção de diversas falhas neste tipo de equipamento, tais como: desbalanceamento, desalinhamento, problemas elétricos, problemas de lubrificação, desgaste de rolamentos, folga mecânica em engrenagem, polia, correia, entre outros (GEITNER; BLOCH, 2012).

As tecnologias de Internet das Coisas Industrial (IIoT) viabilizam a formação de uma rede de dispositivos inteligentes conectados para constituir sistemas que monitoram, coletam, trocam e analisam dados (POSEY; ROSENCRANCE; SHEA, 2021). Desta forma, possibilitam a implementação de um sistema de monitoramento de vibrações em tempo real, permitindo rápida reação por parte do time de manutenção, possibilitando o planejamento e intervenção inclusive de forma remota.

Em outras palavras a manutenção preditiva baseada em IoT utiliza dados históricos de várias fontes, incluindo dispositivos e sensores para fazer previsões precisas sobre a integridade, utilização e possibilidade de falha de ativos, permitindo tomar medidas com base nessas informações (IT CONVERGENCE, 2019).

Com a alta demanda de coleta de dados e processamento de informações para tomada de decisões, tecnologias de comunicação e integração de dispositivos como OPC-UA, controladores de borda e computação em nuvem, tem se tornado indispensáveis para sistemas IIoT. Assim como, interfaces que possibilitem a conversão de diversos tipos de protocolos,

como por exemplo, Node-RED e gateway inteligentes. Desta forma, sistemas que utilizam diferentes protocolos podem interagir por meio da internet, e disponibilizar diagnósticos ao usuário, inclusive em linguagem natural, para que decisões possam ser tomadas de modo mais fácil e rápido.

A linha de pesquisa foi definida de acordo com a alta necessidade e demanda observada na indústria, onde o fator disponibilidade do ativo é fundamental para redução de custos e melhoria do processo. A proposta deste projeto é desenvolver um sistema de monitoramento de vibrações em tempo real utilizando conceitos e tecnologias de Internet das Coisas industrial. Este sistema terá a funcionalidade de diagnosticar níveis de vibrações em regiões descritas como: ideal; satisfatório; insatisfatório; inaceitável. Essas informações poderão ser visualizadas em dispositivos móveis e computadores de forma simples e intuitiva, com o objetivo de reduzir o tempo de ação. Nas próximas seções, serão apresentados conceitos relacionados à análise de vibrações e tecnologias IIoT, seguida pela apresentação da arquitetura proposta. Logo após, serão descritas as etapas do desenvolvimento do sistema, incluindo: coleta de dados do sensor via Modbus em Codesys; programação de alarmes; telas de visualização em Codesys; configuração para envio de dados por OPC-UA; programação de um fluxo em Node-RED para envio dos dados do servidor OPC-UA para um banco de dados; configuração do banco de dados relacional; telas de visualização tipo dashboard em Grafana. Por fim, são mostrados os resultados obtidos por meio de uma bancada de testes com três sensores em diferentes posições com medição de vibração nos sentidos radial e axial. A bancada permite inserir falhas mecânicas como desbalanceamento, folgas e desalinhamento para geração de vibrações e teste do sistema implementado.

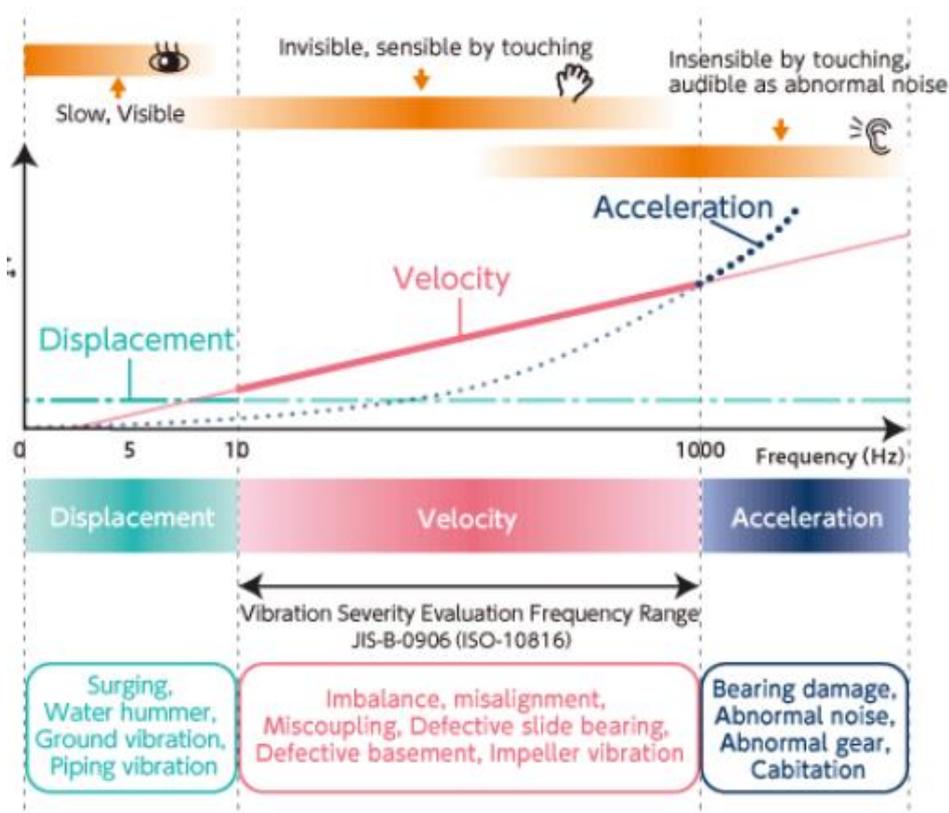
1 ANÁLISE DE AMPLITUDE DE VIBRAÇÃO

A análise de vibração é uma maneira eficiente de detectar a saúde de máquinas rotativas sendo utilizada como técnica para antecipar falhas. Pode ser realizada através de monitoramento contínuo onde a amostragem é realizada por segundo, ou periódico em que a coleta de dados é definida de acordo com o período de tempo determinado pelo usuário. A frequência da vibração é um indicador importante para o diagnóstico do tipo de falha que a máquina apresenta. Em

conjunto com a frequência, a amplitude do sinal deve ser observada a fim de detectar a severidade de uma determinada falha (MAIS; BRADY, 2002).

As frequências das vibrações podem ser divididas em três tipos de acordo com a percepção humana, conforme pode ser observado na figura 1. As vibrações visíveis estão em baixas frequências e são medidas pelo deslocamento (micrômetros). As invisíveis, porém, perceptíveis ao toque estão em médias frequências, medidas pela velocidade da vibração (mm/s). E as audíveis, em frequências maiores, e medidas pela aceleração. A avaliação de severidade, por meio da amplitude da velocidade da vibração é regulamentada pela norma ISO 10816. Nesta faixa de frequência que vai de 10 a 1000 Hz aproximadamente, podem ser detectados: desalinhamento, desbalanceamento, problemas de acoplamento, rolamento linear defeituoso, vibração do rotor.

Figura 1 – Tipos de vibrações



Fonte: IMV CORPORATION, 2020.

A amplitude por velocidade é utilizada para identificar a severidade de uma falha. A norma ISO sugere faixas com limites aceitáveis da amplitude da velocidade da vibração para cada classe de máquinas, de acordo com a figura 2.

Figura 2 – Classificação de níveis de vibrações conforme a ISO 10816-1

	Machine		Class I Small Machines	Class II Medium Machines	Class III Large Rigid Foundation	Class IV Large Soft Foundation
	in/s	mm/s				
Vibration Velocity Vrms	0.01	0.28				
	0.02	0.45				
	0.03	0.71		good		
	0.04	1.12				
	0.07	1.80				
	0.11	2.80		satisfactory		
	0.18	4.50				
	0.28	7.10		unsatisfactory		
	0.44	11.2				
	0.70	18.0				
	1.10	28.0		unacceptable		
	1.77	45.9				

Fonte: BANNER ENGINEERING, 2020.

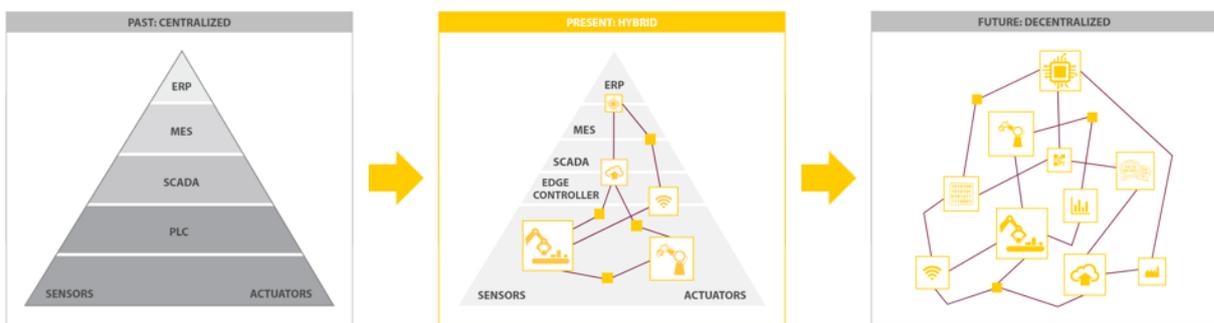
O software desenvolvido possibilita utilizar as quatro classes de máquinas. Para ensaios em bancada foi utilizado a classe de máquinas 1, para potências menores que 20 CV. Os níveis de vibração sugeridos pela ISO10816-1, serão os devidos limites de alarmes separados por regiões descritas como: Zona A - ideal; Zona B - satisfatório; Zona C - insatisfatório; Zona D - inaceitável.

2 TOPOLOGIA E COMPONENTES DO SISTEMA

O sistema é composto por tecnologias de IoT aplicadas à automação industrial com ênfase em manutenção preditiva. O objetivo principal é proporcionar considerável redução de tempo de máquina parada, por meio de monitoramento das condições da máquina rotativa fornecendo facilidades para análise e tomada de decisão para resposta de problemas de forma ágil.

O conceito de IoT na indústria envolve a troca de dados de forma descentralizada. Esses dados passam por um pré-processamento e, por meio de análises, são convertidos em informações valiosas para tomadas de decisão. Desta forma, a hierarquia de vários níveis representado pela pirâmide de automação está se tornando cada vez mais permeável (observe a figura 3), estabelecendo a base para uma nova era de automação (TURCK, 2020).

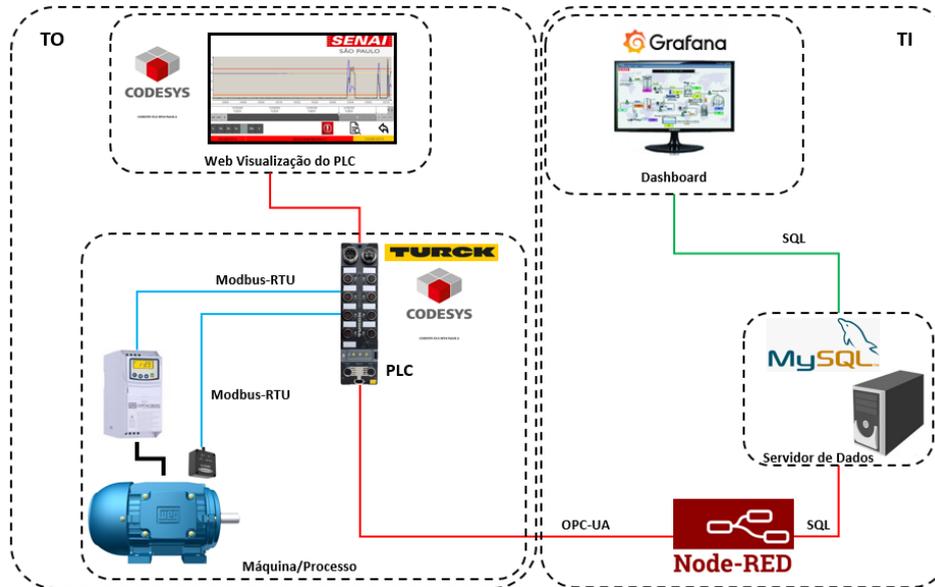
Figura 3 – Pirâmide da automação: passado, presente e futuro



Fonte: TURCK, 2020.

Na figura 4, podemos observar a arquitetura proposta para o sistema IIoT de monitoramento de vibração. Ela está dividida em dois tipos de tecnologias: Tecnologia Operacional (TO), que envolve os componentes de automação industrial para instrumentação e controle do processo; e Tecnologia da Informação (TI), que inclui os componentes de hardware e software para armazenamento, visualização e análise de dados. A comunicação dos dados na camada de TO é realizada por meio de protocolo Modbus *Remote Terminal Unit* (Modbus-RTU) por ser um dos mais antigos com ampla utilização industrial e principalmente por ser um protocolo aberto e compatível com diversos dispositivos de acionamento de motores (inversores de frequência) e sensores. Entre as camadas de TO e TI é utilizado o padrão OPC *Unified Architecture* (OPC-UA) como protocolo devido à sua facilidade de integração entre dispositivos por meio de rede Ethernet, independente de plataformas e fabricantes, e integração com mecanismos de segurança. É considerado atualmente o padrão de protocolo de comunicação para viabilidade da IoT industrial. Da perspectiva de TI, o OPC-UA é a interface de programação da planta e fábrica conectada e um habilitador crítico para aplicações de Internet das Coisas Industriais (OPC FOUNDATION, 2020).

Figura 4 – Topologia do sistema IIoT para monitoramento de vibração



Fonte: Autoria própria, 2021.

Para instrumentação, foi selecionado um sensor de vibração modelo QM30VT2 fabricado pela empresa Banner. Conforme análise das características técnicas na figura 5, possui comunicação Modbus-RTU e frequência de medição de 10 a 4k Hz, atendendo a região de interesse (10 a 1k Hz). Além disso, possui uma eletrônica embutida para processamento e filtragem do sinal de vibração, disponibilizando os valores em análise espectral: frequência e amplitude RMS.

Figura 5 – Características técnicas

Specifications

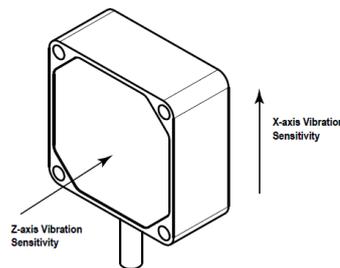
<p>Supply Voltage 10 V DC to 30 V DC</p> <p>Current Active comms: 9 mA at 30 V DC</p> <p>Communication Interface: RS-485 serial Baud rates: 9.6k, 19.2k (default), or 38.4k Data format: 8 data bits, no parity (default), 1 stop bit (even or odd parity available) Protocol: Modbus RTU</p> <p>Mounting Options The sensor can be mounted using a variety of methods, including M4 x 0.7 hex screw, epoxy, thermal tape, or magnetic mount.</p> <p>Mechanical Shock MIL-STD-202G, Method 213B, Condition I (100G 6x along X, Y, and Z axes, 18 shocks), with device operating</p> <p>Certifications</p>	<p>Vibration Sensor Measuring Range: 0 to 46 mm/sec or 0 to 1.8 in/sec RMS Frequency Range: 10 Hz to 4 kHz Accuracy: ±10% at 25 °C Sampling Frequency: 20 kHz (default) Record Length: 8192 points (default) Sample Duration: 0.4 s (default)</p> <p>Temperature Sensor Measuring Range: -40 °C to +105 °C (-40 °F to +221 °F) Resolution: 1 °C Accuracy: ± 3 °C Operating the sensor at higher voltages can induce internal heating that can reduce the accuracy.</p> <p>Environmental Rating Stainless steel model: IP69K per DIN 40050-9 Aluminium model: IEC IP67</p> <p>Operating Temperature -40 °C to +105 °C (-40 °F to +221 °F) ¶</p>
---	--



Fonte: BANNER ENGINEERING, 2020.

Este sensor é capaz de realizar medições em duas direções: axial e radial, ou eixos X e Z, conforme figura 6. Para obter melhores resultados, é necessário instalar o sensor o mais próximo possível do rolamento do motor, ou instalar em uma superfície que esteja em conexão rígida com as características de vibração do motor.

Figura 6 – Orientações para medições



Fonte: BANNER ENGINEERING, 2020.

Foi escolhido o controlador lógico programável (CLP) Turck IIoT modelo TBEN-L5-PLC-11, por possuir diferentes protocolos de comunicação disponíveis como: Modbus-RTU; Modbus-TCP; Ethernet/IP; ProfiNET; CANopen; SAE J1939; OPC-UA. Desta forma, possibilita acessar diferentes camadas da pirâmide de automação e comunicação máquina-a-máquina (M2M), tornando possível a descentralização da automação. Além disso, pode ser programado utilizando o ambiente de desenvolvimento Codesys, gratuito, por possuir o Codesys *Runtime* instalado em seu sistema operacional Linux, compatível com a norma IEC 61131-1, independente de fabricante de hardware e amplamente utilizado em controladores industriais. Também permite desenvolver uma interface de visualização Web para interação com usuário do sistema de automação.

A parte descrita como tecnologias de TI permitem que os dados sejam acessados via Internet, possibilitando, se necessário, a utilização de serviços em nuvem. Foi implementado localmente (poderia ser na nuvem também) um banco de dados relacional MySQL para o armazenamento do histórico de dados. O interfaceamento entre o protocolo OPC-UA e o banco de dados é realizado por meio de um fluxo programado em Node-RED, uma ferramenta de programação visual criada especificamente para conectar dispositivos de hardware à API's, serviços de nuvem, banco de dados, como parte de um sistema IoT.

O software Grafana foi escolhido para visualização Web do histórico de dados utilizando um painel de indicadores (*dashboard*). As informações são acessadas por meio de consulta SQL ao banco de dados, com o objetivo de analisar o comportamento da máquina rotativa com amostragem de longo prazo.

Foram selecionados softwares gratuitos, proporcionando uma solução de baixo custo, com investimento aproximado de hardware de R\$10.000,00, que inclui o CLP Turck, o sensor de vibração, cabos e conectores.

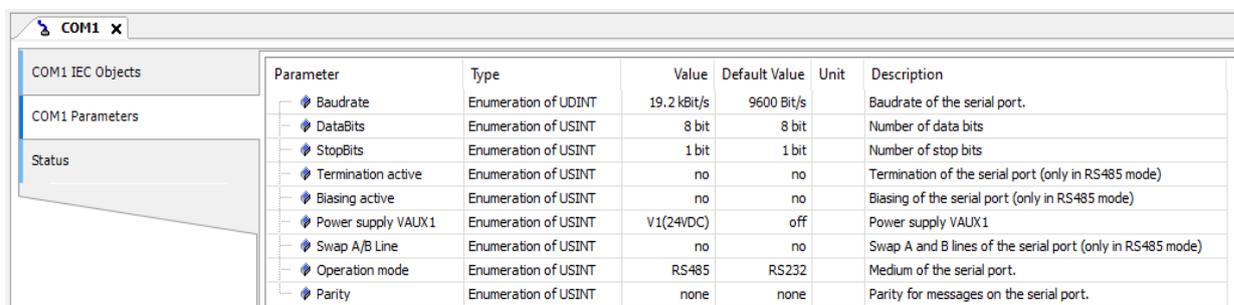
3 MONITORAMENTO EM CLP COM CODESYS

A aplicação desenvolvida no CLP Turck, será descrita em tópicos, uma vez que nela estão algumas das principais características do projeto: coleta e filtragem dos dados, e disponibilização para o sistema de TI.

3.1 Registros do Sensor

O CLP Turck possui dois canais de comunicação serial, estes podem funcionar com os padrões RS-232 ou RS-485. Na plataforma Codesys V3.5 podemos configurar os parâmetros de comunicação do protocolo Modbus-RTU, tais como: *baudrate*, *databits*, entre outros, conforme pode ser visto na figura 7. A porta COM1 foi definida como o canal responsável onde todos os três sensores foram conectados em rede.

Figura 7 – Canal de comunicação COM1

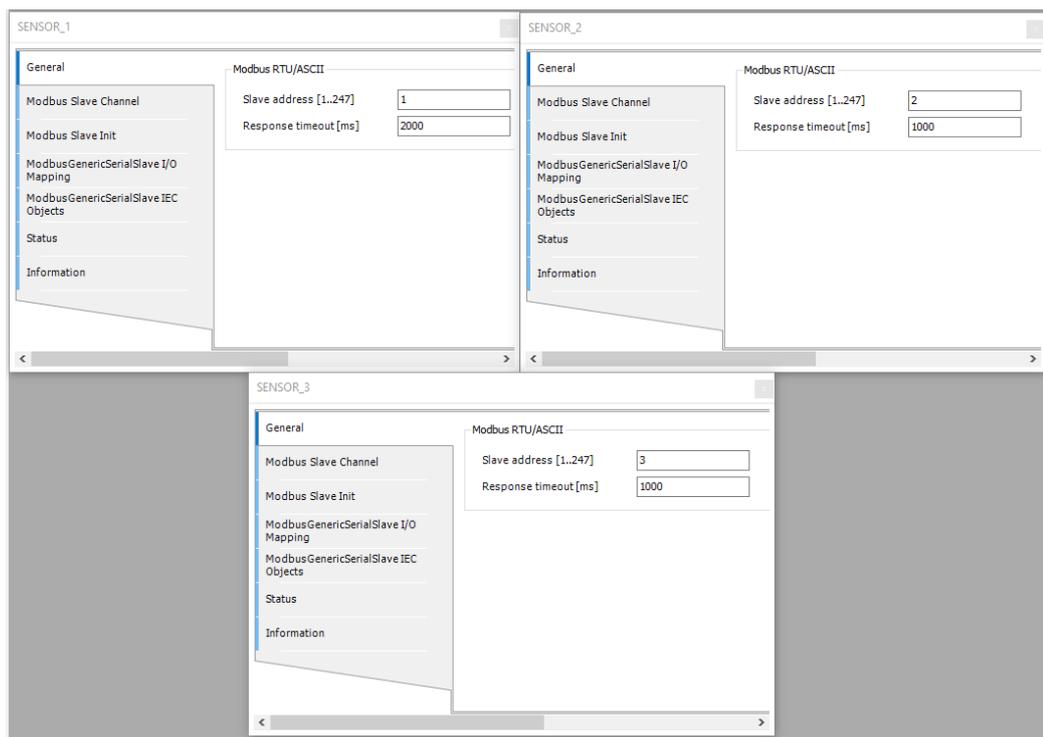


Parameter	Type	Value	Default Value	Unit	Description
Baudrate	Enumeration of UDINT	19.2 kBit/s	9600 Bit/s		Baudrate of the serial port.
DataBits	Enumeration of USINT	8 bit	8 bit		Number of data bits
StopBits	Enumeration of USINT	1 bit	1 bit		Number of stop bits
Termination active	Enumeration of USINT	no	no		Termination of the serial port (only in RS485 mode)
Biasing active	Enumeration of USINT	no	no		Biasing of the serial port (only in RS485 mode)
Power supply VAUX1	Enumeration of USINT	V1(24VDC)	off		Power supply VAUX1
Swap A/B Line	Enumeration of USINT	no	no		Swap A and B lines of the serial port (only in RS485 mode)
Operation mode	Enumeration of USINT	RS485	RS232		Medium of the serial port.
Parity	Enumeration of USINT	none	none		Parity for messages on the serial port.

Fonte: Autoria própria, 2021.

No driver mestre Modbus-RTU foram inseridos os respectivos escravos, neste caso, os três sensores de vibração QM30VT2. Foi definido para cada sensor um endereço, *Slave address* 1, 2 e 3, como pode ser visto na figura 8.

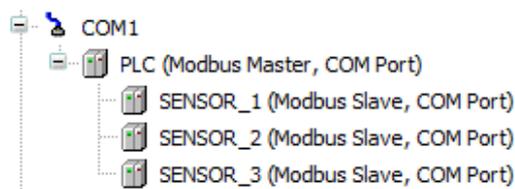
Figura 8 – Endereços de comunicação dos sensores



Fonte: Autoria própria, 2021.

Após as parametrizações da comunicação, é possível ser visto na figura 9, de forma hierárquica a estrutura da rede.

Figura 9 – Driver Modbus-RTU mestre e escravo para os sensores



Fonte: Autoria própria, 2021.

Os registros foram configurados conforme os respectivos endereços *Holding Register* de cada sensor. Todos os registros dos sensores podem ser acessados na área de *IO-Mapping* do respectivo escravo, conforme figura 10.

Figura 10 – Mapa de registros dos sensores

Variable	Mapping	Channel	Address	Type	Description
Z_AxisRMSVelocity_in_sec		Z_AxisRMSVelocity_in_sec	%IW4	ARRAY [0..0] OF WORD	Read Holding Registers
Z_AxisRMSVelocity_in_sec		Z_AxisRMSVelocity_in_sec[0]	%IW4	WORD	0x0961
Z_Axis_RMS_Velocity_mm_sec		Z_Axis_RMS_Velocity_mm_sec	%IW5	ARRAY [0..0] OF WORD	Read Holding Registers
Z_Axis_RMS_Velocity_mm_sec		Z_Axis_RMS_Velocity_mm_sec[0]	%IW5	WORD	0x0963
TemperatureF		Temperature_F	%IW6	ARRAY [0..0] OF WORD	Read Holding Registers
TemperatureF		Temperature_F[0]	%IW6	WORD	0x0031
TemperatureC		Temperature_C	%IW7	ARRAY [0..0] OF WORD	Read Holding Registers
TemperatureC		Temperature_C[0]	%IW7	WORD	0x002B
X_AxisRMSVelocity_in_sec		X_AxisRMSVelocity_in_sec	%IW8	ARRAY [0..0] OF WORD	Read Holding Registers
X_AxisRMSVelocity_in_sec		X_AxisRMSVelocity_in_sec[0]	%IW8	WORD	0x0993
X_AxisRMSVelocity_mm_sec		X_AxisRMSVelocity_mm_sec	%IW9	ARRAY [0..0] OF WORD	Read Holding Registers
X_AxisRMSVelocity_mm_sec		X_AxisRMSVelocity_mm_sec[0]	%IW9	WORD	0x0995

Fonte: Autoria própria, 2021.

A figura 11 mostra a rotina chamada *Sensor_1*, onde cada registro individual é transferido para um array. Além dos dados serem indexados, foram submetidos aos cálculos de conversão para suas devidas unidades de engenharia. Para os três sensores utilizados no projeto foi realizado o mesmo padrão de cálculo e tratamento.

Figura 11 – Rotina *Sensor_1*

```

1
2 // Registros padrões
3 //GVL.sensor_1[1] := WORD_TO_REAL(Z_AxisRMSVelocity_insec)/10000;
4 GVL.sensor_1[2] := WORD_TO_REAL(Z_Axis_RMS_Velocity_mmsec)/1000;
5 //GVL.sensor_1[3] := WORD_TO_REAL(TemperatureF)/100;
6 GVL.sensor_1[4] := WORD_TO_REAL(TemperatureC)/1000;
7 //GVL.sensor_1[5] := WORD_TO_REAL(X_AxisRMSVelocity_insec)/1000;
8 GVL.sensor_1[6] := WORD_TO_REAL(X_AxisRMSVelocity_mmsec)/1000;
9 GVL.sensor_1[7] := WORD_TO_REAL(Z_AxisPeakAccelerationG)/10;
10 GVL.sensor_1[8] := WORD_TO_REAL(X_AxisPeakAccelerationG)/10;
11 GVL.sensor_1[9] := WORD_TO_REAL(Z_AxisPeakVelocityComponentFrequencyHz)/1000;
12 GVL.sensor_1[10] := WORD_TO_REAL(X_AxisPeakVelocityComponentFrequencyHz)/1000;
13 GVL.sensor_1[11] := WORD_TO_REAL(Z_Axis_RMSAccelerationG)/1000;
14 GVL.sensor_1[12] := WORD_TO_REAL(X_AxisRMSAccelerationG)/1000;
15 GVL.sensor_1[13] := WORD_TO_REAL(Z_Axis_Kurtosis)/1000;
16 GVL.sensor_1[14] := WORD_TO_REAL(X_Axis_Kurtosis)/1000;
17 GVL.sensor_1[15] := WORD_TO_REAL(Z_AxisCrest_Factor)/1000;
18 GVL.sensor_1[16] := WORD_TO_REAL(X_AxisCrest_Factor)/1000;

```

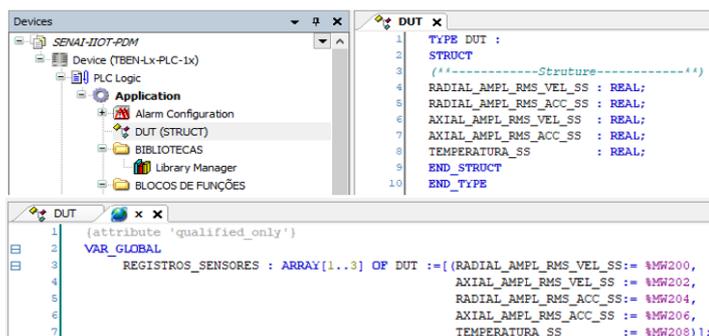
Fonte: Autoria própria, 2021.

Os registros monitorados nesta aplicação, são:

- Vibração no sentido Axial em velocidade RMS (mm/s)
- Vibração no sentido Radial em velocidade RMS (mm/s)
- Temperatura em °C
- Vibração no sentido Axial em aceleração RMS (g)
- Vibração no sentido Radial em aceleração RMS (g)

Foi utilizado a abordagem de estruturas de dados *Structure* para tornar possível a criação de grupos de registros para cada sensor e utilizá-los de forma indexada, com o objetivo de diminuir o tamanho de código e agilizar o desenvolvimento da programação. A estrutura de dados criada pode ser vista na figura 12.

Figura 12 – Bloco de registros em *Structure*



Fonte: Autoria própria, 2021.

3.2 Níveis de Alarme em Amplitude

Após a atribuição dos registros à estrutura de dados, será possível coletar um grupo de registros por índice onde cada índice representa um sensor. Todos os sensores possuirão as mesmas características. Além dos registros recebidos do sensor, é inserido à estrutura de cada sensor os limites de avisos e alarmes para cada condição, que serão utilizados posteriormente na interface. Portanto sempre que os valores medidos pelos sensores ultrapassarem os limites de alarmes, teremos a informação sobre o comportamento crítico do sistema. Para monitoramento dos níveis de vibrações na região de frequência de 10 a 1000 Hz é utilizado como refe-

rência a tabela fornecida pela ISO10816-1(ver figura 2), que define os limites para quatro classes de máquinas. Por exemplo, para classe de máquinas 1, temos os seguintes valores de amplitude de vibração em velocidade para serem usados como setpoints dos alarmes:

- Limite de vibração ideal – 0,28 a 0,71 mm/s
- Limite de vibração satisfatória – 0,72 a 1,80 mm/s
- Limite de vibração insatisfatória – 1,81 a 4,50 mm/s
- Limite de vibração inaceitável – maior que 4,51mm/s

Na figura 13 podemos ver como é realizado o monitoramento de vibração radial e axial para a região de vibração ideal, definido pela norma como Zona A.

Figura 13 – Limites de vibração para região ideal – Zona A

```

10 (* Limites para Região de Alarmes de Amplitude em Velocidade para Frequências de 10 a 1KHz *)
11 //Se o valor atual for menor que o SP Maximo RADIAL, sinaliza que a Amplitude esta na regioao Ideal
12 IF ((x.REGISTROS_SENSORES[i].RADIAL_AMPL_RMS_VEL_SS < x.MaxVib_RADIAL_IDEAL[i]))
13 THEN x.Alarme_RADIAL_AMPL_RMS_VEL_IDEAL[i] := TRUE;
14 ELSE x.Alarme_RADIAL_AMPL_RMS_VEL_IDEAL[i] := FALSE ;END_IF
15
16 //Se o valor atual for menor que o SP Maximo AXIAL, sinaliza que a Amplitude esta na regioao Ideal
17 IF ((x.REGISTROS_SENSORES[i].AXIAL_AMPL_RMS_VEL_SS < x.MaxVib_AXIAL_IDEAL[i]))
18 THEN x.Alarme_AXIAL_AMPL_RMS_VEL_IDEAL[i] := TRUE;
19 ELSE x.Alarme_AXIAL_AMPL_RMS_VEL_IDEAL[i] := FALSE ;END_IF

```

Fonte: Autoria própria, 2021.

A mesma estratégia de monitoramento é utilizada para a verificação de alarmes para os níveis satisfatório, insatisfatório e inaceitável. Cada registro do sensor, ou seja, o valor atual de medição é comparado com o valor máximo e mínimo de cada região e em função do valor, os níveis de alarmes serão acionados e diagnosticados na tela de visualização e acompanhamento do sistema. Para definição dos limites de cada classe, foi utilizado o recurso da plataforma Codesys chamado *Text List*, conforme é possível ver na figura 14. O *Text List* informa os valores de seu *ID* a uma variável de 16 bits, conforme a seleção de uma das classes. Vinculado a um *combo box* torna simples selecionar qual classe será utilizada.

Figura 14 – Text List para classes de máquinas

ID	Default
0	CLASSE I - 0...20CV
1	CLASSE II - 20...100CV
2	CLASSE III - >100CV com Base Rígida
3	CLASSE IV - >100CV com Base Flexível

Fonte: Autoria própria, 2021.

Desta forma, na rotina que inclui a categoria de máquinas, é habilitada a condição conforme o valor do *ID* do *Text List*. Podemos entender a lógica observando figura 15.

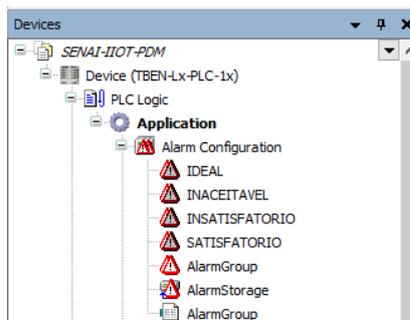
Figura 15 – Definição de Setpoints em função da classe da máquina

```
1  
2 (* Indice *)  
3 i := i+1;  
4  
5 (* Limite de Contagem *)  
6 IF i>3 THEN i:=0; END_IF  
7  
8 IF (cat = 0) THEN  
9  
10 x.MaxVib_AXIAL_IDEAL[i] := 0.71;  
11 x.MaxVib_RADIAL_IDEAL[i] := 0.71;  
12  
13 x.MaxVib_AXIAL_SATISFATORIO[i] := 1.80;  
14 x.MaxVib_RADIAL_SATISFATORIO[i] := 1.80;  
15  
16 x.MaxVib_AXIAL_INSATISFATORIO[i] := 4.50;  
17 x.MaxVib_RADIAL_INSATISFATORIO[i] := 4.50;  
18  
19 x.MaxVib_AXIAL_INACEITAVEL[i] := 8.00;  
20 x.MaxVib_RADIAL_INACEITAVEL[i] := 8.00;  
21  
22 END_IF  
23  
24 IF (cat = 1) THEN  
25  
26 x.MaxVib_AXIAL_IDEAL[i] := 1.12;  
27 x.MaxVib_RADIAL_IDEAL[i] := 1.12;  
28  
29 x.MaxVib_AXIAL_SATISFATORIO[i] := 2.80;  
30 x.MaxVib_RADIAL_SATISFATORIO[i] := 2.80;  
31  
32 x.MaxVib_AXIAL_INSATISFATORIO[i] := 7.10;  
33 x.MaxVib_RADIAL_INSATISFATORIO[i] := 7.10;  
34  
35 x.MaxVib_AXIAL_INACEITAVEL[i] := 8.00;  
36 x.MaxVib_RADIAL_INACEITAVEL[i] := 8.00;  
37  
38 END_IF  
39
```

Fonte: Autoria própria, 2021.

Os servidores de alarmes foram definidos conforme o grupo de valores para cada região de vibração. Na tela de visualização é possível monitorar os alarmes de tempo real e disponíveis em históricos. Na figura 16 é possível entender a arquitetura de configuração de alarmes.

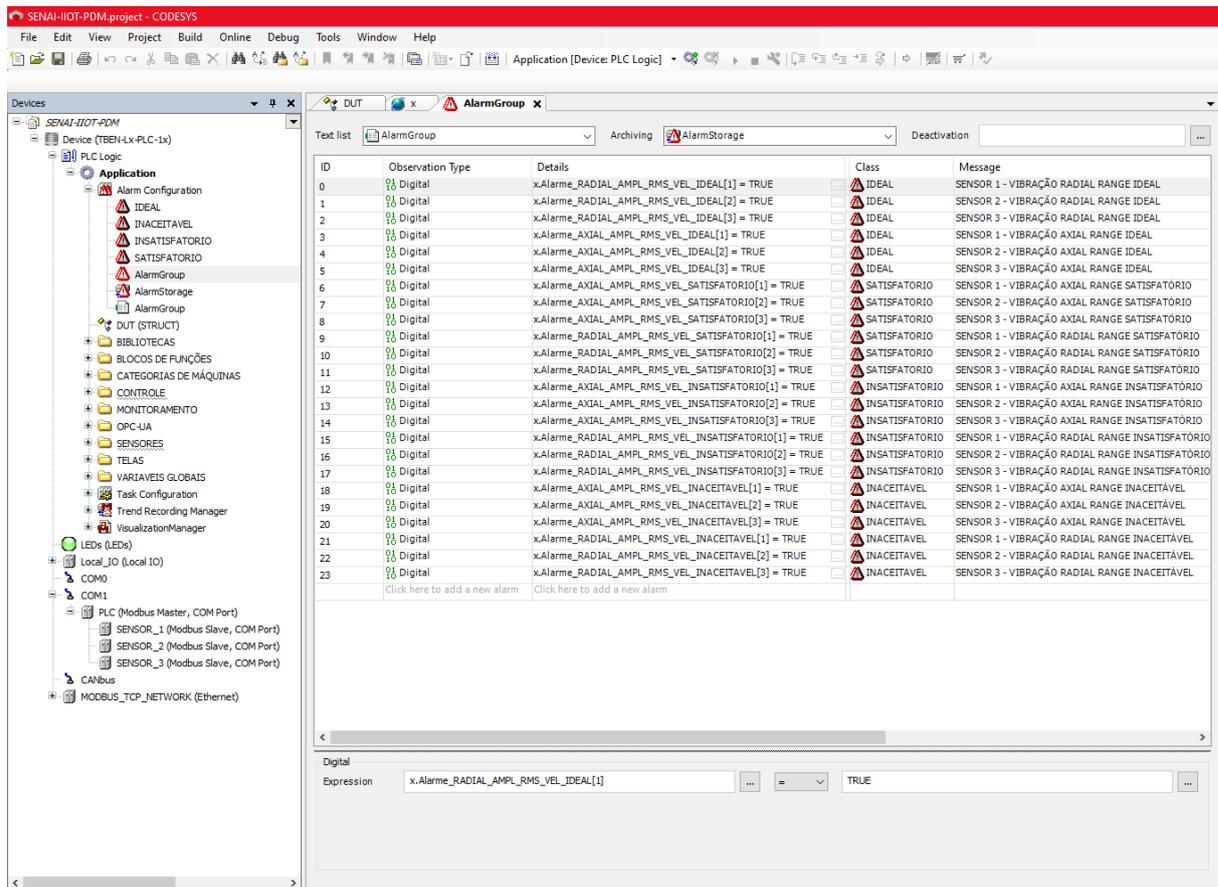
Figura 16 – Servidores de alarmes



Fonte: Autoria própria, 2021.

Para ser possível o monitoramento é necessário que todas as variáveis responsáveis por habilitar a condição de alarme sejam vinculadas ao servidor de alarme, bem como o grupo que ele pertence e a mensagem que será disponibilizada ao usuário. A configuração pode ser verificada na figura 17.

Figura 17 – Definição de alarmes



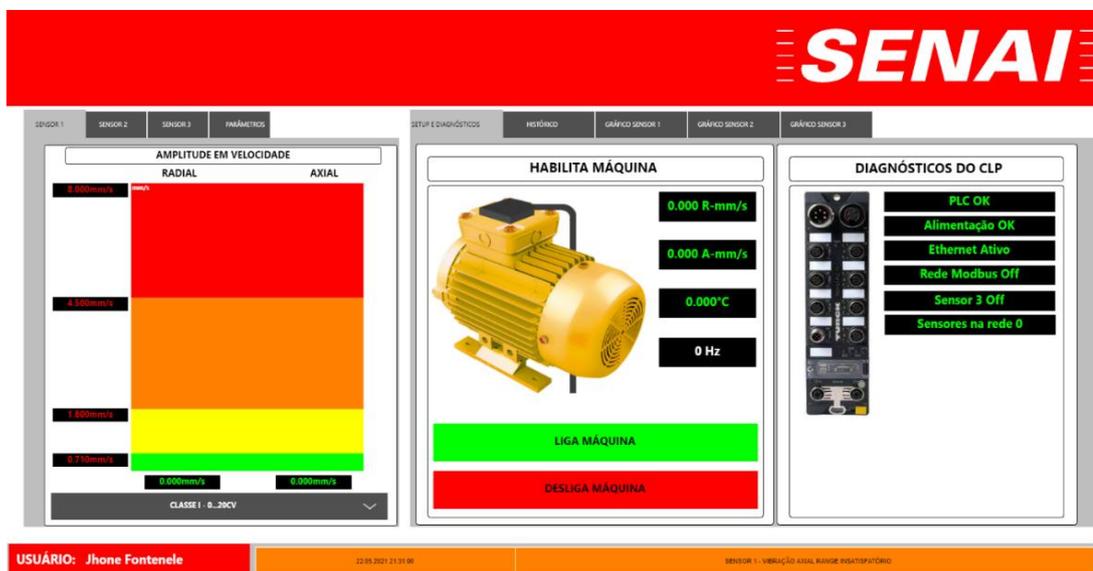
Fonte: Autoria própria, 2021.

4 TELA DE PARAMETRIZAÇÃO E VISUALIZAÇÃO EM CODESYS

A plataforma Codesys V3.5 proporciona recursos de visualização baseado em páginas web, sendo possível acessar via navegadores. Para este projeto o caminho de acesso da página de configuração e monitoramento foi definido conforme o IP e porta de rede do CLP Turck, neste caso: 192.168.0.44: 8080/vt.htm.

O sistema de visualização em Codesys foi desenvolvido com o objetivo de facilitar o acesso ao monitoramento de alarmes e configurações da máquina. Esta visualização pode ser acessada tanto por computadores que estiverem na mesma rede como por celulares e tablets. A aplicação conta com controle de acesso, onde um grupo tem acesso aos parâmetros e monitoramento e outro grupo somente ao monitoramento. A tela principal pode ser vista na figura 18.

Figura 18 – Tela principal do sistema de monitoramento via visualização em Codesys

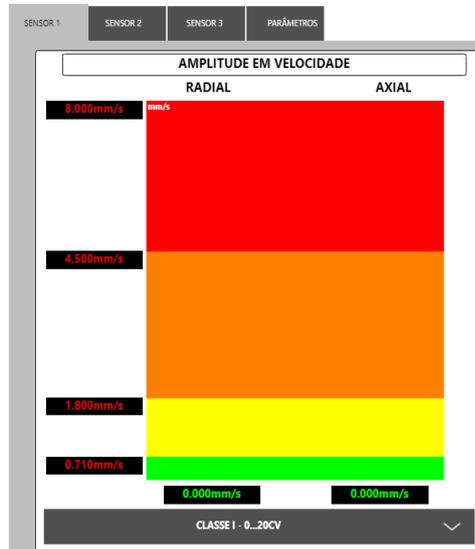


Fonte: Autoria própria, 2021.

A interface de visualização foi desenvolvida considerando ter o máximo de informações úteis em apenas uma tela, porém possuindo abas de navegação entre sensores, parâmetros e histórico. Na interface, além de monitoramento e parâmetros do comportamento do sistema, é possível também executar o controle da máquina e monitoramento de tendências.

O sistema de visualização baseado em web foi parametrizado e programado utilizando conceitos presentes em sistemas supervisórios industriais. Para o monitoramento em tempo real da região de vibração em amplitude foi utilizado histograma para cada sensor, onde é possível selecionar qual a classe do sistema a ser monitorado. De forma automática conforme anteriormente definido, o grupo de Setpoints assumem seus valores. Para cada região foi definido uma cor para visualização de forma intuitiva, conforme mostrado na figura 19.

Figura 19 – Histograma de amplitude dos sensores



Fonte: Autoria própria, 2021.

O controle do motor é realizado por meio de inversor de frequência que recebe os comandos do CLP. Estes comandos são realizados por meio da aba de Setup e Diagnósticos, onde é possível ligar e desligar manualmente assim como definir a frequência de rotação (figura 20).

Figura 20 – Habilita máquina



Fonte: Autoria própria, 2021.

Para cada limite de vibração definidos pelos Setpoints, a imagem do motor é definida com uma cor. Estas cores seguem o padrão para toda a aplicação, tendo as mesmas características nas abas de tendências, histograma e histórico de alarmes, para todos os sensores. O histórico de alarmes é responsável por armazenar dados de vibrações, temperatura e diagnósticos do CLP. A figura 21 mostra um exemplo.

Figura 21 – Histórico de alarmes

SETUP E DIAGNÓSTICOS	HISTÓRICO	GRÁFICO SENSOR 1	GRÁFICO SENSOR 2	GRÁFICO SENSOR 3
	Timestamp	Message		
0	22.05.2021 21:31:00	SENSOR 1 - VIBRAÇÃO AXIAL RANGE INSATISFATÓRIO		
1	22.05.2021 21:25:18	SENSOR 1 - VIBRAÇÃO RADIAL RANGE INSATISFATÓRIO		
2	22.05.2021 21:20:10	SENSOR 1 - VIBRAÇÃO RADIAL RANGE SATISFATÓRIO		
3	22.05.2021 21:18:00	SENSOR 3 - VIBRAÇÃO AXIAL RANGE IDEAL		
4	22.05.2021 21:18:00	SENSOR 2 - VIBRAÇÃO AXIAL RANGE IDEAL		
5	22.05.2021 21:18:00	SENSOR 1 - VIBRAÇÃO AXIAL RANGE IDEAL		
6	22.05.2021 21:18:00	SENSOR 3 - VIBRAÇÃO RADIAL RANGE IDEAL		
7	22.05.2021 21:18:00	SENSOR 2 - VIBRAÇÃO RADIAL RANGE IDEAL		
8	22.05.2021 21:18:00	SENSOR 1 - VIBRAÇÃO RADIAL RANGE IDEAL		

Fonte: Autoria própria, 2021.

O monitoramento dos dados de processo é muito importante para a aplicação, porém tão importante quanto os dados do processo é o diagnóstico de saúde do controlador do processo. Por este motivo foi configurado na aba de Setup e Diagnósticos as principais informações sobre as condições atuais do controlador, conforme figura 22.

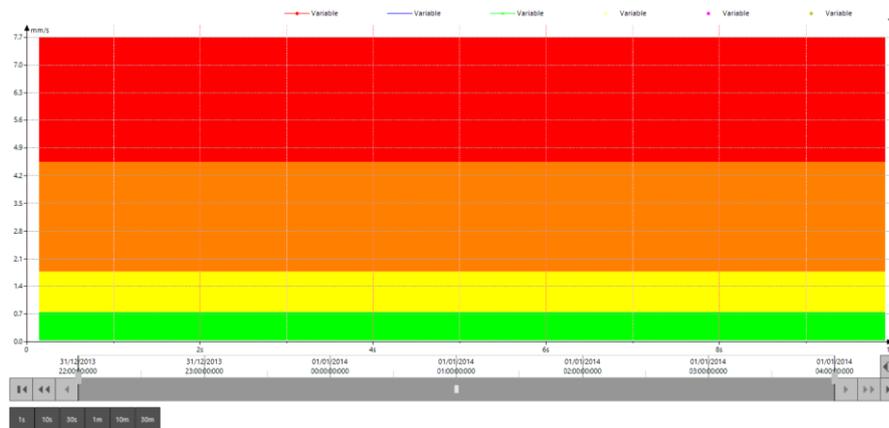
Figura 22 – Habilita máquina



Fonte: Autoria própria, 2021.

A análise de tendências é uma das técnicas mais utilizadas para entender o comportamento de um sistema. Na aplicação foi estabelecido um gráfico para cada sensor, no qual o setpoint para o nível de vibração é simbolizado em cores (figura 23).

Figura 23 – Gráfico de tendência web visualização em Codesys



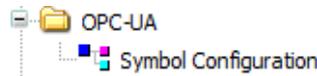
Fonte: Autoria própria, 2021.

O sistema de visualização web armazena os dados coletados na memória interna do CLP. Por este motivo, é possível a historização de dados de até 40 dias. Para períodos maiores de coleta de dados, foi utilizado um sistema de banco de dados e visualização no Grafana.

5 COMUNICAÇÃO EM OPC-UA

O CLP Turck possui servidor OPC-UA integrado, que comunica pelo padrão *Client-Server*. Este padrão utiliza transmissão de dados em UA *Binary format* via UA TCP, baseado na arquitetura SOA *service-oriented architecture* (OPC FOUNDATION, 2020). Para a aplicação foi utilizado o *endpoint url* `opc.tcp://192.168.0.44:4840`, onde a porta 4840 é padrão para o servidor. Esta comunicação não utilizará certificado de validação, que será implementado no futuro. Para realizar a parametrização da comunicação são necessárias algumas etapas. Após a criação de todas as variáveis da aplicação é necessário na plataforma Codesys, criar uma configuração de símbolos, que é responsável por listar todas as variáveis a serem disponibilizadas pelo servidor, conforme figura 24.

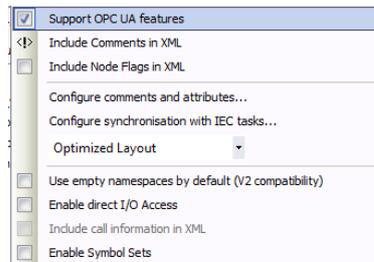
Figura 24 – Configuração de símbolos



Fonte: Autoria própria, 2021.

Ao criar a lista de símbolos, é necessário confirmar o suporte aos recursos de comunicação em OPC-UA. De forma automática, o Codesys gera uma lista de dados e configura o servidor, conforme figura 25.

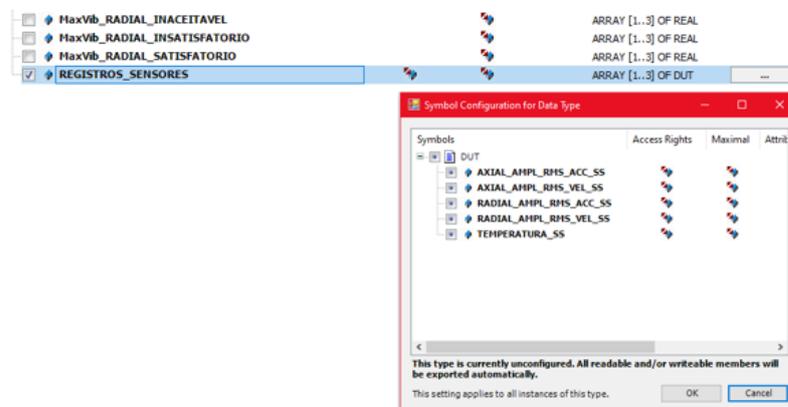
Figura 25 – Configuração de recursos OPC-UA



Fonte: Autoria própria, 2021.

Com a lista de símbolos configurada, é possível acessar todas as variáveis da aplicação em Codesys e definir quais estarão disponíveis pelo servidor aos clientes conforme figura 26.

Figura 26 – Lista de variáveis para comunicação OPC-UA

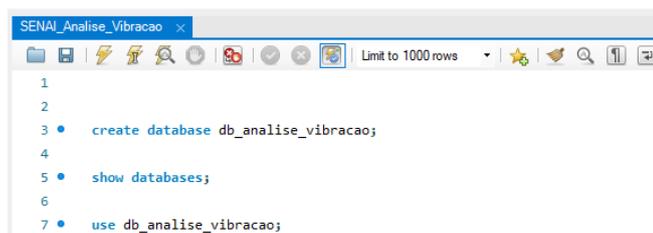


Fonte: Autoria própria, 2021.

6 BANCO DE DADOS

O banco de dados tem papel fundamental na aplicação, uma vez que para conhecer um determinado equipamento ou sistema, é necessário possuir histórico de dados. Para realizar a conexão entre MySQL e Node-RED, MySQL e Grafana, foi utilizado o servidor XAMPP, responsável por gerenciar a porta de comunicação 3306 para o serviço. Após a estrutura de comunicação ser estabelecida, foram utilizadas as ferramentas de desenvolvimento da plataforma MySQL Workbench para escrever os *scripts* SQL para cada funcionalidade. Inicialmente foi criado o banco de dados, conferido sua existência e posteriormente definido para uso, conforme sequência ilustrada na figura 27.

Figura 27 – Comandos SQL para database

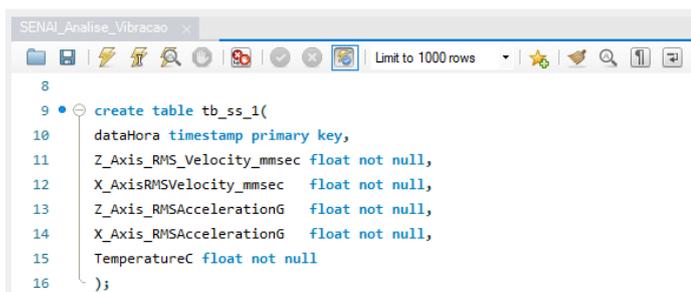


```
1  
2  
3 • create database db_analise_vibracao;  
4  
5 • show databases;  
6  
7 • use db_analise_vibracao;
```

Fonte: Autoria própria, 2021.

Com o banco de dados já disponível, foi realizado o comando de criação de tabela, com a definição de colunas conforme os registros dos sensores a serem armazenados. Para cada sensor foi criado uma tabela, conforme pode ser visto na figura 28. O *timestamp* é gerenciado pelo MySQL enquanto os demais registros são enviados pela interface Node-RED.

Figura 28 – Comandos SQL para tabela



```
8  
9 • create table tb_ss_1(  
10   dataHora timestamp primary key,  
11   Z_Axis_RMS_Velocity_mmsec float not null,  
12   X_AxisRMSVelocity_mmsec float not null,  
13   Z_Axis_RMSAccelerationG float not null,  
14   X_Axis_RMSAccelerationG float not null,  
15   TemperatureC float not null  
16 );
```

Fonte: Autoria própria, 2021.

Após a criação da tabela, é possível verificar sua estrutura, assim como realizar leitura dos registros armazenados com o comando SQL *select*, conforme figura 29.

Figura 29 – Comandos SQL registros

```
35  
36 • describe tb_ss_1;  
37 • select * from tb_ss_1;
```

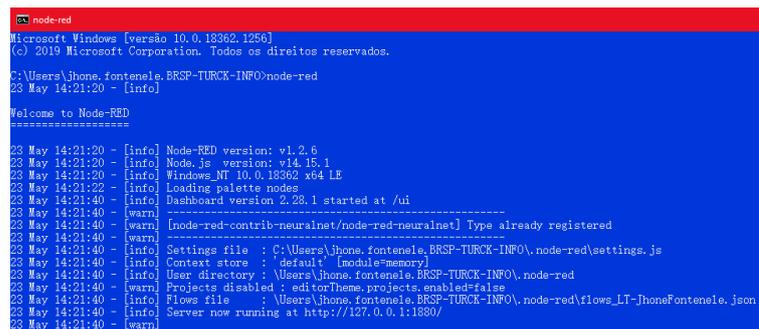
Fonte: Autoria própria, 2021.

7 INTERFACE EM NODE-RED

Node-RED é uma ferramenta de programação baseada em fluxo, originalmente desenvolvida pela equipe de serviços de tecnologia emergentes da IBM e agora uma parte da Fundação JS (NODE-RED, 2019).

Após a instalação do servidor, para iniciá-lo é necessário acessar o prompt de comando e inserir o comando `node-red`. Com o software devidamente instalado o servidor deverá retornar o IP e porta de rede disponível para acessar a interface, conforme figura 30.

Figura 30 – Servidor Node-RED

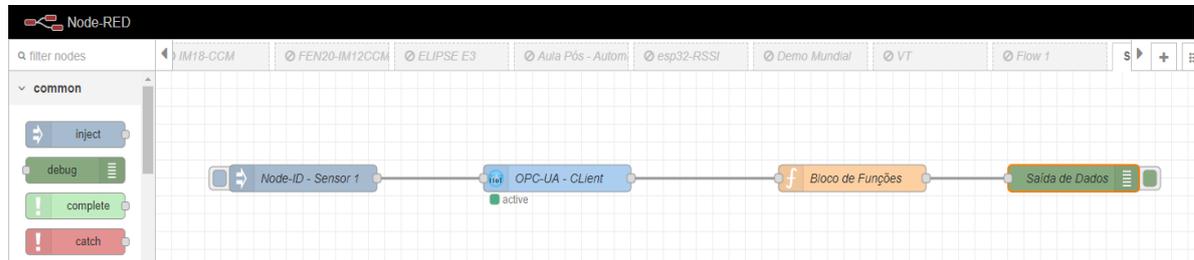


```
node-red  
Microsoft Windows [versão 10.0.18362.1256]  
(c) 2019 Microsoft Corporation. Todos os direitos reservados.  
C:\Users\jhone.fontenele.BRSP-TURCK-INFO\node-red  
23 May 14:21:20 - [info]  
Welcome to Node-RED  
-----  
23 May 14:21:20 - [info] Node-RED version: v1.2.6  
23 May 14:21:20 - [info] Node.js version: v14.15.1  
23 May 14:21:20 - [info] Windows_NT 10.0.18362 x64 LE  
23 May 14:21:22 - [info] Loading palette nodes  
23 May 14:21:40 - [info] Dashboard version 2.28.1 started at /ui  
23 May 14:21:40 - [warn]  
23 May 14:21:40 - [warn] [node-red-contrib-neuralnet/node-red-neuralnet] Type already registered  
-----  
23 May 14:21:40 - [info] Settings file : C:\Users\jhone.fontenele.BRSP-TURCK-INFO\node-red\settings.js  
23 May 14:21:40 - [info] Context store : default (in-memory)  
23 May 14:21:40 - [info] User directory : \Users\jhone.fontenele.BRSP-TURCK-INFO\node-red  
23 May 14:21:40 - [warn] Projects disabled : editorTheme.projects.enabled=false  
23 May 14:21:40 - [info] Flows file : \Users\jhone.fontenele.BRSP-TURCK-INFO\node-red\flows_LT-jhonefontenele.json  
23 May 14:21:40 - [info] Server now running at http://127.0.0.1:1880/  
23 May 14:21:40 - [warn]
```

Fonte: Autoria própria, 2021.

Na interface Node-RED foi instalado a biblioteca *node-red-contrib-lativ* e utilizado o *node* OPCUA-IIOT-READ, *Client* responsável por estabelecer a comunicação com o servidor OPC-UA. Com a interface rodando, conforme ilustrado na figura 31, podemos criar um fluxo de recebimento de informações enviadas pelo CLP Turck, por meio de protocolo OPC-UA.

Figura 31 – Fluxo Node-RED



Fonte: Autoria própria, 2021.

A Configuração para o OPC-UA *Client*, é definido por meio do *node* OPC-UA Read, onde é informado o *Endpoint*, conforme imagem 32.

Figura 32 – Definição de Endpoint

The image shows the configuration settings for an OPC-UA Client. There are two tabs: 'Settings' (selected) and 'Security'. Under 'Settings', there are three fields: 'Endpoint' with the value 'opc.tcp://192.168.1.44:4840', 'Keep Session Alive' with a checked checkbox, and 'Name' with the value 'TBEN-L5-PLC-11'.

Fonte: Autoria própria, 2021.

Com o *Client* configurado é possível realizar a conexão com o CLP que é o servidor, com isto também é possível acessar o *Node-ID*. A definição do *Node-ID* do registro possui um formato específico que pode ser visualizado na figura 33.

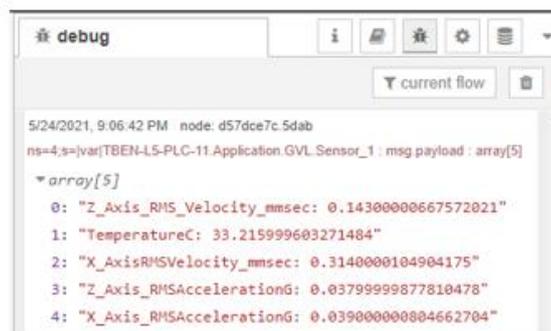
Figura 33 – Definição de Node-ID

The image shows the configuration for a Node-ID. The 'Name' field is set to 'Node-ID - Sensor 1'. Below, there are two rows of configuration for message properties: 'msg.payload' is set to 'timestamp' and 'msg.topic' is set to 'ns=4;s=|var|TBEN-L5-PLC-11.Application.GVL.Sensor_1'. Each row has a small 'x' icon to the right.

Fonte: Autoria própria, 2021.

No *node* bloco de funções foi desenvolvido um filtro para receber os dados pertinentes do sensor em OPC-UA. Desta forma, são transmitidos somente os registros úteis para a aplicação, neste caso os dados do sensor. Ao enviar os registros para saída do fluxo é possível visualizar o resultado, conforme figura 34.

Figura 34 – Leitura de dados do sensor no Node-RED



Fonte: Autoria própria, 2021.

Para armazenar os dados de monitoramento de vibração além de ter o banco de dados criado é necessário também ter as devidas tabelas que cada sensor irá historiar. Ainda no Node-RED é necessário estabelecer a conexão com o servidor de banco de dados e também configurar o comando SQL com a função de inserir os dados na tabela, conforme figura 35. Para esta função foi utilizado o comando *Insert Into*, este comando é responsável por incrementar dados à tabela.

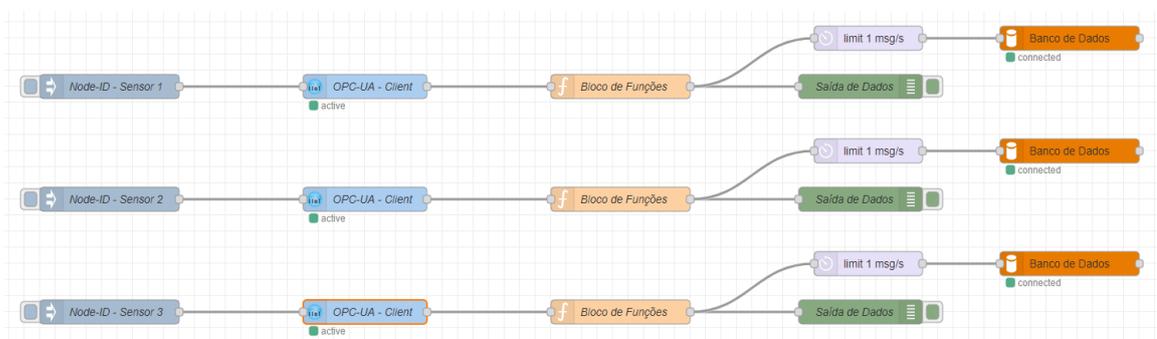
Figura 35 – Comando SQL insert into



Fonte: Autoria própria, 2021.

A comunicação com o banco de dados MySQL foi estabelecida utilizando a biblioteca *node-red-node-mysql* e o *node mysql*. Neste *node* foi configurado o acesso fornecendo o mesmo usuário e senha definido na criação do banco de dados no *software* MySQL Workbench. Também foi configurado o *host* do computador que o servidor foi instalado, a porta 3306 de comunicação, e a devida tabela *db_analise_vibracao*, onde os dados serão armazenados. A inserção de informações na tabela só é possível uma vez que o *node* Banco de dados esteja devidamente configurado, conforme figura 36.

Figura 36 – Node banco de dados on-line



Fonte: Autoria própria, 2021.

Cada sensor possui sua tabela dentro do mesmo banco de dados, onde cada linha de fluxo representa um sensor, conforme figura 36. Desta forma, com o servidor de banco de dados rodando, o Node-RED envia os comandos a cada um segundo com o valor atual dos sensores para serem armazenados e posteriormente serem utilizados no Grafana. O resultado dos comandos realizados manualmente pode ser visto na figura 37.

Figura 37 – Resultado do comando SQL insert into na plataforma MySQL workbench

dataHora	Z_Axis_RMS_Velocity_mmsec	X_AxisRMSVelocity_mmsec	Z_Axis_RMSAccelerationG	X_Axis_RMSAccelerationG	TemperatureC
2021-05-24 23:35:44	0	0	0	0	0
2021-05-24 23:38:55	0.155	0.198	0.034	0.034	33.952
2021-05-24 23:45:21	0.164	0.143	0.038	0.034	34

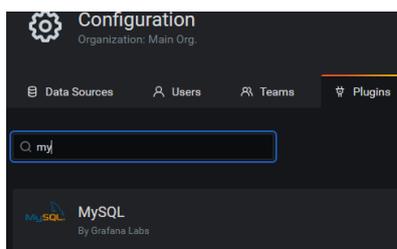
Fonte: Autoria própria, 2021.

Ponto importante de observação é que o Node-RED executa o comando de *insert* e o Grafana o comando *select*.

8 GRAFANA

O Grafana é uma solução *open source* para gestão de métricas e indicadores. Amplamente utilizado para análise de *big data*, principalmente no universo de software livre, possuindo integração com diversas soluções, como MySQL, PostgreSQL, InfluxDB e outros serviços. Para o projeto estamos utilizando o MySQL como fonte de dados, onde por meio comandos SQL é possível coletar dados e disponibilizá-los em diversos tipos de gráficos. Para configuração da fonte de dados foi necessário adicionar o *plug-in* do banco de dados conforme ilustrado na figura 38.

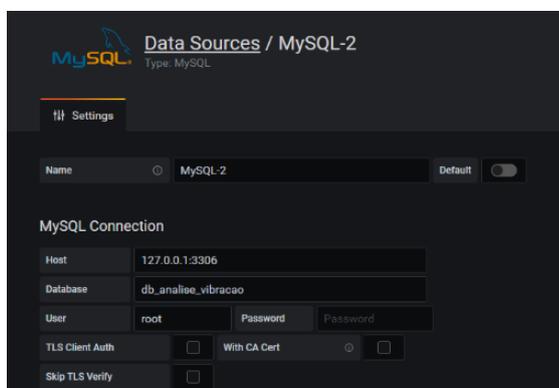
Figura 38 – Plug-in MySQL



Fonte: Autoria própria, 2021.

Com o MySQL já configurado como fonte de dados, se faz necessário a definição dos dados do servidor. Todas as informações necessárias para a comunicação do banco de dados estão descritas conforme a figura 39.

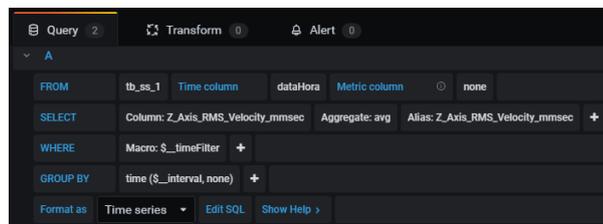
Figura 39 – Fonte de dados MySQL



Fonte: Autoria própria, 2021.

Para criar o monitoramento de uma tendência, alguns pontos devem ser observados. Levando em consideração que são coletados cinco registros de cada sensor e cada sensor possui sua tabela, é importante observar que para disponibilizar estes dados em tendência, primeiro é necessário definir qual tabela será utilizada, bem como quais colunas e em função do que será a amostragem. Para a aplicação, foi definido que as colunas de vibração em velocidade radial e axial estão dispostas no mesmo gráfico, assim como a aceleração da vibração radial e axial. Apenas a temperatura estará sozinha em um único gráfico. Todos os registros possuem amostragem gráfica em função do tempo. Detalhes da configuração das penas conforme figura 40.

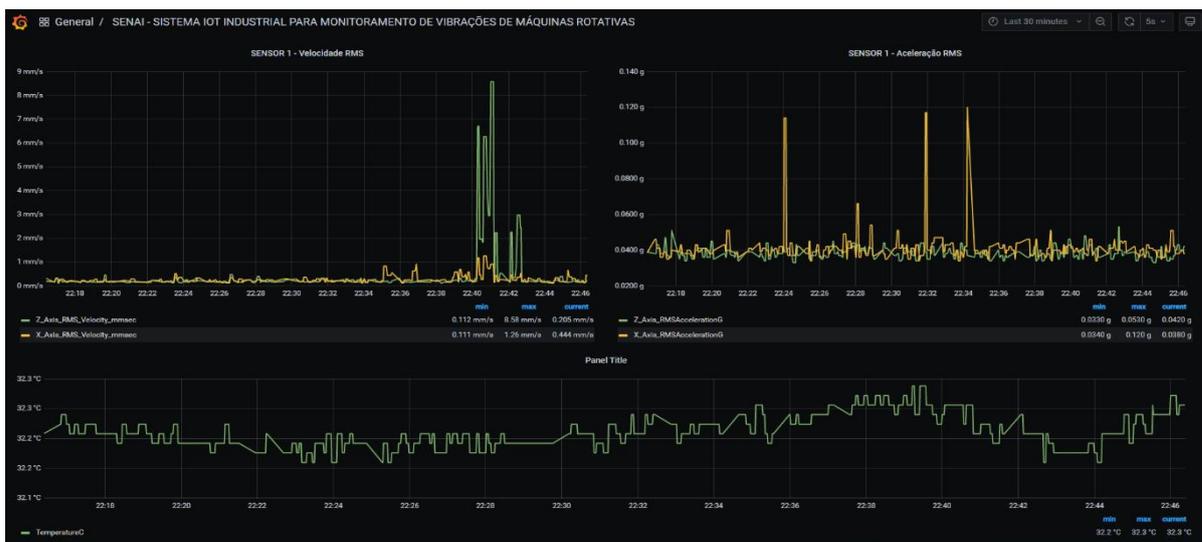
Figura 40 – Parâmetros de tendências



Fonte: Autoria própria, 2021.

Ao fim das configurações é possível ver todas as penas com o histórico existente no banco de dados, conforme a figura 41.

Figura 41 – Grafana tendências para análise de vibrações



Fonte: Autoria própria, 2021.

9 RESULTADOS

O sistema IoT Industrial para monitoramento de vibrações de máquinas rotativas, cujo desenvolvimento foi demonstrado nas seções anteriores, foi testado com o objetivo de validar o funcionamento e viabilidade. Foi utilizado um sistema CPS *ciber physical system* chamado *live demo*, disponibilizado para os ensaios, pela empresa TURCK do Brasil.

O *live demo* é composto por uma estrutura com um motor de 0,5 CV instalado de forma fixa acoplado a um conjunto de dois mancais com um flange simétrico em seu eixo. O sistema possui também componentes de controle, como: Inversor de Frequência; CLP Turck; Gateways IO-Link, como pode ser visto na figura 42.

Figura 42 – **Live demo**



Fonte: Autoria própria, 2021.

Os sensores de vibração biaxiais são fixados com base magnética nos pontos onde se deseja fazer as medições de vibração. Portanto, conforme pode ser visto na figura 43, o *live demo* possui 4 sensores que possibilitam medições diretamente no motor e também nos mancais.

Figura 43 – Disposição dos sensores



Fonte: Autoria própria, 2021.

Com a disposição dos sensores mostrado na figura 43, é possível realizar o monitoramento de diagnósticos da máquina rotativa, onde cada tipo de falha pode ser analisado considerando qual é o sentido de vibração predominante, radial ou axial. Para diagnosticar exatamente quais tipos de falhas poderão surgir, é necessário também avaliar em quais níveis de frequências existem valores críticos de amplitude. Neste projeto, os recursos do *live demo* foram utilizados com o objetivo de analisar as amplitudes de vibrações, não fazendo parte desta versão a análise espectral em função da frequência, mesmo que o equipamento possua estas características.

O *live demo* possibilita inserir distúrbios conhecidos propositais, sendo eles:

- Desalinhamento Angular
- Desalinhamento Paralelo
- Desbalanceamento estático
- Folgas Mecânicas
- Rolamentos danificados

Entre todos os tipos possíveis de ensaios, foi escolhido o desbalanceamento estático. Por possuir maior flexibilidade para ajustar a mecânica do sistema, não precisa de instrumentos adicionais como laser para alinhamento. Somente um ensaio foi realizado, uma vez que os procedimentos são os mesmos e o comportamento do sistema é semelhante, alterando somente em qual direção (radial ou axial) haverá maior intensidade de vibrações.

O desbalanceamento estático surge quando a máquina rotativa possui distribuição de cargas assimétricas em seu eixo, que resulta na alteração do centro de gravidade do disco acoplado ao motor. Desta forma, gera excesso de vibrações com mais intensidade no sentido radial. O disco possui furações simétricas para possibilitar a inserção de cargas, conforme pode ser visto na figura 44.

Figura 44 – Disco simétrico para geração de desbalanceamento estático



Fonte: Autoria própria, 2021.

Com o sistema de desbalanceamento ainda desacoplado, podemos habilitar o motor por meio do botão “Liga Máquina” no supervisor web desenvolvido em Codesys. Vemos na figura 45 que, mesmo rodando em vazio, o sistema apresenta níveis de vibrações, porém baixos. Por questões de segurança, a frequência de rotação do *live demo* é limitado a 20 Hz.

Figura 45 – Monitoramento on-line motor rodando em vazio



Fonte: Autoria própria, 2021.

Em seguida podemos ver na figura 46, o resultado com o sistema de desbalanceamento com duas cargas de simulação inseridas ao disco, desta vez visualizando em página web via dispositivo móvel.

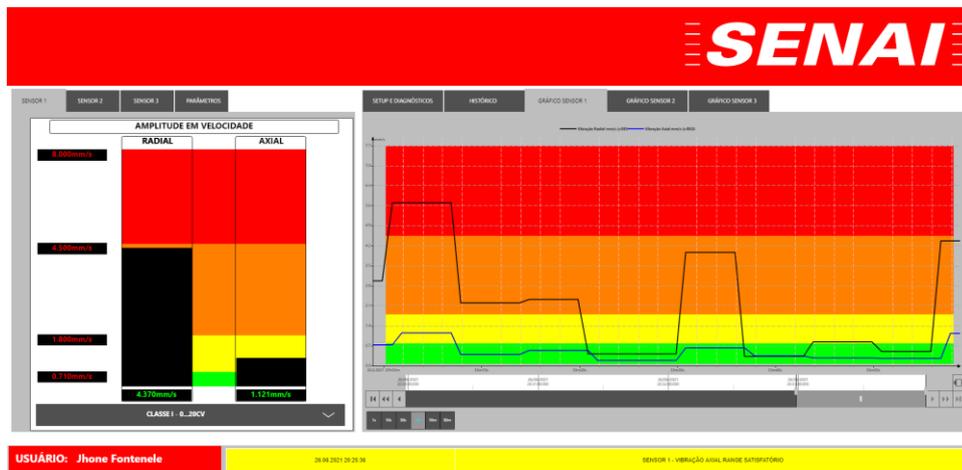
Figura 46 – Monitoramento on-line motor com carga desbalanceada



Fonte: Autoria própria, 2021.

No próximo passo, veremos como o sistema se comporta quando é inserido somente uma carga no disco. Esta configuração de carga altera drasticamente o centro de massa do eixo acoplado ao motor. Os níveis de vibração podem ser observados graficamente no visualizador web, conforme figura 47.

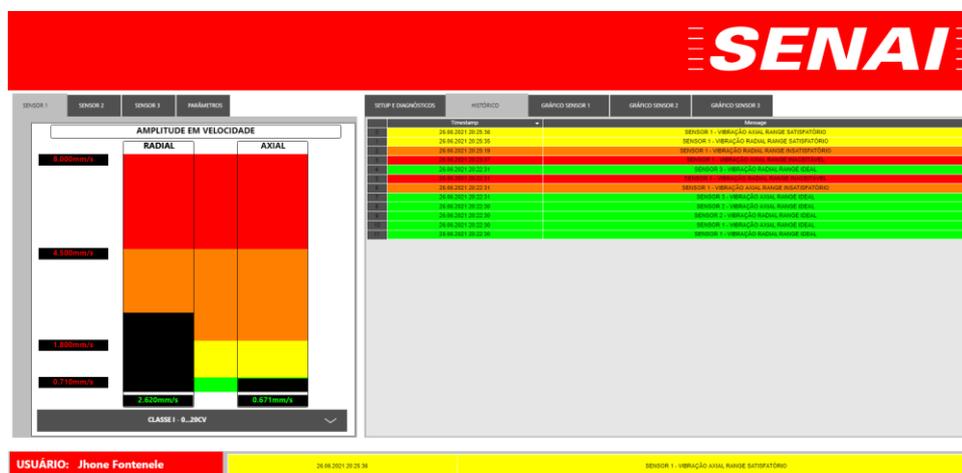
Figura 47 – Monitoramento on-line com tendência



Fonte: Autoria própria, 2021.

Na figura 48 vemos o histórico de alarmes em função dos níveis de vibrações, para este mesmo ensaio.

Figura 48 – Monitoramento on-line com histórico de alarmes



Fonte: Autoria própria, 2021.

Durante todo o ensaio, os registros foram armazenados no banco de dados. O sistema *Cloud Grafana* consulta a respectiva tabela de dados dos sensores e então disponibiliza todas as informações existentes em gráficos, possibilitando visualizar a tendência em períodos de amostragem maiores. Na figura 49, pode ser visto o momento de transição em forma de picos de vibração, mostrando o comportamento do sistema com desbalanceamento e posteriormente novamente o motor rodando em vazio.

Figura 49 – Transição entre sistema desbalanceado e rodando em vazio



Fonte: Autoria própria, 2021.

Após diversos comandos executados, a fim de estressar o sistema, foi possível entender o comportamento da máquina rotativa em função dos níveis de vibrações, assim como os benefícios de possuir um sistema de monitoramento *IIoT on-line*.

Na prática, apenas com o monitoramento da amplitude de vibração foi possível detectar as características de desbalanceamento. Pois, conforme visto na figura 47, os níveis de vibração radial tendem a ser cerca de três vezes maior do que os níveis na direção axial.

Sabendo que uma falha pode surgir de outra, é possível que haja vários tipos de falhas ao mesmo tempo prejudicando a tradicional forma de análise espectral, então o monitoramento *on-line* se mostra mais eficiente, informando à manutenção que o equipamento possui um comportamento que precisa de atenção.

CONSIDERAÇÕES FINAIS

O objetivo deste artigo foi realizar um projeto envolvendo hardware e software, resultando em um sistema IoT Industrial para manutenção preditiva utilizando monitoramento de vibrações. Foi realizado um estudo abrangente, sobre como este tema é abordado na indústria, onde o primeiro passo foi entender o porquê o monitoramento é viável para manutenção, e segundo o que poderia tornar este monitoramento realmente vantajoso em termos de investimento, assim como as características que podem ser consideradas relevantes para utilização deste tipo de solução.

Por meio de teste na bancada *live demo*, foi possível detectar os níveis de vibrações críticos conforme a norma ISO10816-1 e mostrou com os resultados obtidos, a viabilidade de utilizar um sistema IoT Industrial para manutenção preditiva, disponibilizando dados valiosos em tempo real, em qualquer dispositivo conectado à Web.

Pretende-se em breve, acrescentar funcionalidades a esta aplicação, tais como detecção de falhas utilizando análise espectral, estimação de tempo de vida útil por meio de técnicas de *machine learning* e armazenamento de dados em serviços de nuvem.

AGRADECIMENTOS

Acima de tudo, agradecemos a Deus por proporcionar saúde e sabedoria.

Agradecemos à empresa Turck do Brasil por tornar este projeto possível, proporcionando total acesso à tecnologia e componentes.

Agradecemos à Faculdade de Tecnologia SENAI “Mariano Ferraz” e a todos os professores pela elevada qualidade do ensino oferecido.

REFERÊNCIAS

BANNER ENGINEERING. **QM30VT2 | QM30VT Vibration Sensor**. Disponível em: <<https://info.bannerengineering.com/cs/groups/public/documents/literature/210732.pdf>>. Acesso em: 29 maio. 2021.

GEITNER; BLOCH. **Machinery Failure Analysis and Troubleshooting**. [s.l.] Elsevier, 2012.

IT CONVERGENCE. **predictive maintenance**. Disponível em: <<https://www.itconvergence.com/blog/5-benefits-iot-based-predictive-maintenance/>>. Acesso em: 6 jun. 2021.

MAIS, J.; BRADY, S. Introduction Guide to Vibration Monitoring. p. 30, 2002.

NBR 5462, A. **NBR 5462 Confiabilidade e manutenibilidade** ABNT-Associação Brasileira de Normas Técnicas, , 1994. Disponível em: <<https://www.abntcatalogo.com.br/norma.aspx?ID=004086>>. Acesso em: 6 jun. 2021

NODE-RED. **About : Node-RED**. Disponível em: <<https://nodered.org/about/>>. Acesso em: 30 maio. 2021.

OPC FOUNDATION. **OPC Unified Architecture**, 2020. Disponível em: <<https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN.pdf>>. Acesso em: 30 maio. 2021

OPTIMUS, E. **O que é manutenção preventiva, preditiva e corretiva?** **Optimus Prime**, 10 set. 2019. Disponível em: <<https://www.optimusprime.com.br/blog/o-que-e-manutencao-preventiva-preditiva-e-corretiva/>>. Acesso em: 6 jun. 2021

PÉREZ; MÁRQUEZ. **Condition monitoring and fault diagnosis in wind energy systems**. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9781782420101000112>>. Acesso em: 6 jun. 2021.

POSEY; ROSENCRANCE; SHEA. **What is IIoT? Industrial Internet of Things Explained**. Disponível em: <<https://internetofthingsagenda.techtarget.com/definition/Industrial-Internet-of-Things-IIoT>>. Acesso em: 6 jun. 2021.

TURCK. **Decentralized Automation**. Disponível em: <<https://www.turck.de/en/LandingPages/decentralized-automation-37277.php>>. Acesso em: 29 maio. 2021.