

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

FELIPE GOMES BEZERRA

**SISTEMA DE MANUTENÇÃO PREDITIVA PARA EIXOS
MECÂNICOS EM AMBIENTES INDUSTRIAIS**

SÃO CARLOS
2024

FELIPE GOMES BEZERRA

**SISTEMA DE MANUTENÇÃO PREDITIVA PARA EIXOS
MECÂNICOS EM AMBIENTES INDUSTRIAIS**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Mecânica da Universidade Federal de São Carlos, para obtenção do título de Bacharel em Engenharia Mecânica.

Orientador: Prof. Dr. João Vitor de Carvalho Fontes

SÃO CARLOS
2024



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS
COORDENAÇÃO DO CURSO DE ENGENHARIA MECÂNICA - CCEMec/CCET
Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905
Telefone: (16) 33519703 - <http://www.ufscar.br>

DP-TCC-FA nº 25/2024/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso
Folha Aprovação (GDP-TCC-FA)
FOLHA DE APROVAÇÃO

FELIPE GOMES BEZERRA

SISTEMA DE MANUTENÇÃO PREDITIVA PARA EIXOS MECÂNICOS EM AMBIENTES INDUSTRIAIS

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 29 de julho de 2024

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	João Vitor de Carvalho Fontes
Membro da Banca 1	Gustavo Franco Barbosa
Membro da Banca 2	Alexandre Tácito Malavolta



Documento assinado eletronicamente por **João Vitor de Carvalho Fontes, Docente**, em 29/07/2024, às 15:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Alexandre Tácito Malavolta, Docente**, em 29/07/2024, às 15:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Gustavo Franco Barbosa, Professor(a) do Ensino Superior**, em 29/07/2024, às 15:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **1519512** e o código CRC **1F1F3D45**.

Referência: Caso responda a este documento, indicar expressamente o Processo nº 23112.021671/2024-28

SEI nº 1519512

Modelo de Documento: Grad: Defesa TCC: Folha Aprovação, versão de 02/Agosto/2019

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me conceder força, saúde e perseverança para chegar até aqui. Sem Sua orientação e bênçãos, este trabalho não seria possível.

Expresso minha profunda gratidão à minha família, que sempre esteve ao meu lado, oferecendo suporte e incentivo. A minha mãe, Eliete, e ao meu pai, Roberto, por acreditarem em meu potencial. À minha irmã, Giovanna, pela parceria e palavras de incentivo.

À minha namorada, Ana Carolina, agradeço pelo apoio constante e encorajamento. Sua presença foi importante para que eu pudesse superar os desafios encontrados ao longo desta trajetória.

Agradeço à Universidade Federal de São Carlos (UFSCar) e aos professores do curso de Engenharia Mecânica pela oportunidade de desenvolver este trabalho e pela estrutura oferecida durante minha formação.

Agradeço ao meu orientador, Prof. Dr. João Vitor de Carvalho Fontes, pela orientação e suporte ao longo deste projeto.

A todos, meu mais sincero agradecimento.

O sucesso é a soma de pequenos esforços repetidos dia após dia.

RESUMO

Este trabalho apresenta uma metodologia para calcular o ciclo de vida de eixos mecânicos utilizando o diagrama SN (Stress-Number), com foco na implementação de manutenção preditiva eficaz. Tradicionalmente, a manutenção industrial era reativa, abordando falhas apenas após sua ocorrência, o que resultava em altos custos e paradas inesperadas. A manutenção preditiva, no entanto, permite a previsão de falhas por fadiga com base em análises estatísticas e de desempenho passado, facilitando intervenções oportunas e aumentando a vida útil dos componentes. A metodologia desenvolvida envolve a coleta de dados geométricos e do material dos eixos mecânicos, a aplicação de cargas e a realização de cálculos de tensões e resistências. Utilizando o diagrama SN, o sistema prevê a durabilidade dos eixos e armazena essas informações em um banco de dados continuamente atualizado. Esse banco de dados permite uma análise precisa e preditiva do estado dos componentes, auxiliando na identificação de pontos críticos e no planejamento de intervenções de manutenção antes que ocorram falhas catastróficas por fadiga. O sistema desenvolvido não só contribui para a redução de custos operacionais, mas também melhora a eficiência e a segurança das operações industriais. Ao implementar uma solução robusta para a manutenção preditiva de eixos mecânicos, este trabalho estabelece uma base sólida para futuras pesquisas e avanços na área de manutenção industrial.

Palavras-chave: Manutenção Preditiva. Eixos Mecânicos. Diagrama SN. Fadiga.

ABSTRACT

This work presents a methodology for calculating the lifecycle of mechanical shafts using the SN (Stress-Number) diagram, with a focus on implementing effective predictive maintenance. Traditionally, industrial maintenance was reactive, addressing failures only after their occurrence, which resulted in high costs and unexpected downtimes. Predictive maintenance, however, allows for the prediction of fatigue failures based on statistical analyses and past performance, facilitating timely interventions and extending the lifespan of components. The developed methodology involves the collection of geometric and material data of the shafts, application of loads, and performing stress and strength calculations. Using the SN diagram, the system predicts the durability of the shafts and stores this information in a continuously updated database. This database allows for accurate and predictive analysis of the components' condition, aiding in the identification of critical points and in planning maintenance interventions before catastrophic fatigue failures occur. The developed system not only contributes to the reduction of operational costs but also improves the efficiency and safety of industrial operations. By implementing a robust solution for the predictive maintenance of mechanical shafts, this work establishes a solid foundation for future research and advancements in the field of industrial maintenance.

Keywords: Predictive Maintenance. Mechanical Shafts. SN Diagram. Fatigue.

LISTA DE FIGURAS

Figura 1 – Componente passível de manutenção.	15
Figura 2 – Representação dos 3 estágios da falha por fadiga	23
Figura 3 – Representação de linhas de fluxo em chapa com concentrador de tensão	24
Figura 4 – Exemplo de Diagrama S-N	25
Figura 5 – Fluxo de dados da etapa de cadastro do eixo	28
Figura 6 – Fluxo de dados da etapa de atualização de ciclos já realizados	28
Figura 7 – Exemplo de Diagram de Esforço Normal Gerado pelo Sistema	33
Figura 8 – Exemplo de Diagram de Esforço Cortante Gerado pelo Sistema	33
Figura 9 – Exemplo de Diagrama de Momento Fletor Gerado pelo Sistema	33
Figura 10 – Exemplo Estrutura do Banco de Dados	40

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
PIB	Produto Interno Bruto
EUA	Estados Unidos da América
S-N	Tensão - Vida
TMEF	Tempo Médio Entre Falhas
FCT	Fator de Concentração de Tensão
MEF	Método dos Elementos Finitos

LISTA DE SÍMBOLOS

σ	Tensão Máxima
σ_n	Tensão Nominal
σ_m	Tensão Média
σ_{alt}	Tensão Alternada Nominal de Flexão
τ_{alt}	Tensão Alternada Nominal de Torção
σ_{alt}	Tensão Alternada Equivalente de Flexão
τ_{alt}	Tensão Alternada Equivalente de Torção
σ_{axial}	Tensão Alternada Equivalente Normal
σ'_{vM}	Tensão Equivalente de von Mises
S_{ut}	Resistência à Tração Máxima
S'_e	Limite de Resistência à Fadiga
S_e	Limite de Resistência à Fadiga Ajustada
K_t	Fator de Concentração de Tensão para Flexão, Tração ou Compressão
K_{ts}	Fator de Concentração de Tensão para Torção
K_f	Fator de Concentração de Tensão Ajustado para Flexão
K_{fa}	Fator de Concentração de Tensão Ajustado para Tração ou Compressão
K_{fs}	Fator de Concentração de Tensão Ajustado para Torção
$\Delta\sigma$	Faixa de Tensão
E	Módulo de Elasticidade
c	Distância do Centróide à Superfície
I	Momento de Inércia da Seção Transversal
J	Momento Polar de Inércia da Seção Transversal
T	Torque
r	Raio do Entalhe no Ponto Crítico

N	Esforço Normal
V	Esforço Cortante
M	Momento Fletor
A	Área da Seção Transversal do Eixo
Δ	Letra Grega Delta
c_{load}	Coefficiente de Carga
c_{size}	Coefficiente de Tamanho
c_{surf}	Coefficiente de Superfície
c_{temp}	Coefficiente de Temperatura
c_{reliab}	Coefficiente de Confiabilidade
c_{other}	Coefficiente para Outros Efeitos

SUMÁRIO

1 – INTRODUÇÃO	15
1.1 Objetivos	17
1.2 Estrutura do Texto	18
2 – FUNDAMENTAÇÃO TEÓRICA	19
2.1 Métodos de Manutenção	19
2.1.1 Manutenção Corretiva	19
2.1.2 Manutenção Preventiva	20
2.1.3 Manutenção Preditiva	20
2.2 Processo e Falha por Fadiga	21
2.2.1 Contexto Histórico	21
2.2.2 Carregamento Cíclico	22
2.2.3 Mecanismo de Fadiga	22
2.2.4 Métodos de Análise e Monitoramento	23
2.3 Concentradores de Tensão	24
2.3.1 Fator de Concentração de Tensão	24
2.4 Diagrama S-N	25
3 – DESENVOLVIMENTO	27
3.1 Cálculo dos Esforços Solicitantes	28
3.1.1 Coleta do Comprimento do Eixo	29
3.1.2 Identificação dos Apoios	29
3.1.3 Escolha do Plano de Análise	29
3.1.4 Obtenção das Cargas Aplicadas	29
3.1.5 Verificação da Determinação Estática do Eixo	30
3.1.6 Cálculo das Forças de Reação nos Apoios	30
3.1.7 Definição dos Esforços Solicitantes	31
3.1.8 Análise Biplanar	32
3.1.9 Diagrama de Esforços Solicitantes	32
3.2 Coleta de Dados do Material	33
3.3 Coleta e Estruturação dos Pontos Críticos	34
3.3.1 Processo de Identificação dos Pontos Críticos	35
3.4 Cálculo das Tensões Alternadas e do Fator de Concentração de Tensões Ajustado	35
3.4.1 Cálculo do Limite Teórico de Resistência à Fadiga	35
3.4.2 Cálculo das Tensões Alternadas	36

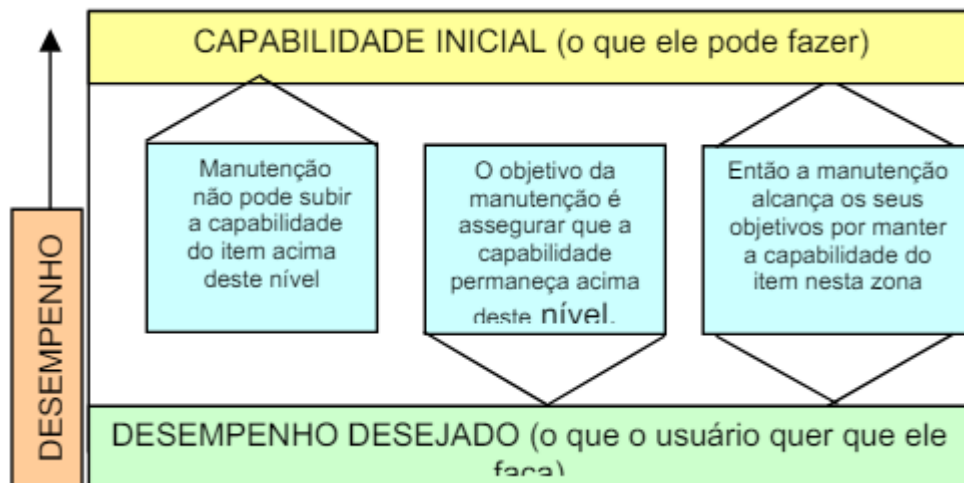
3.4.3	Tensão Equivalente de von Mises	37
3.4.4	Cálculo do Número de Ciclos até a Falha (Método do Diagrama S-N)	37
3.4.5	Identificação do Menor Número de Ciclos até a Falha	38
3.5	Atualização do Banco de Dados	38
4	– RESULTADOS	40
4.1	Cadastro dos Eixos	40
4.2	Atualização dos Ciclos Realizados	40
4.3	Resultados Obtidos	41
5	– CONCLUSÃO	42
	REFERÊNCIAS	43
	Apêndices	44
	APÊNDICE A – Código de Cadastro de Eixo Mecânico no Sistema	45
	APÊNDICE B – Atualização de Ciclos Realizados de Eixos Cadastrados	70
	APÊNDICE C – Entrada de Dados pelo Usuário	72

1 INTRODUÇÃO

A indústria moderna tem se beneficiado significativamente da aplicação de tecnologias avançadas e da digitalização dos processos de produção. Em particular, a manutenção preditiva tem se tornado uma abordagem vital para garantir a eficiência operacional e a longevidade dos equipamentos.

A Associação Brasileira de Normas Técnicas – ABNT, através da Norma Brasileira Regulamentadora - NBR 5462 (ABNT 1994), define a manutenção como: "a combinação de ações técnicas e administrativas, incluindo as de supervisão, destinadas a manter ou recolocar um item em um estado no qual possa desempenhar uma função desejada". Definição ilustrada na Figura 1.

Figura 1 – Componente passível de manutenção.



Fonte: (Moubray 2000)

Segundo (Pinto e Xavier 2001), A história da manutenção industrial pode ser dividida em três gerações, cada uma das quais foi marcada por transformações substanciais em métodos e tecnologias.

Na primeira geração, antes da Segunda Guerra Mundial, a indústria apresentava baixa mecanização, e a manutenção focava principalmente em reparos corretivos. Durante esse período, as intervenções ocorriam apenas quando surgiam problemas. Como a produtividade não era uma prioridade, a manutenção planejada não se fazia necessária.

O período entre a Segunda Guerra Mundial e os anos 60, representa a segunda geração, na qual houve o aumento na mecanização e na complexidade das instalações industriais. Este avanço trouxe novas técnicas de manutenção, como o uso de computadores grandes e lentos, sistemas manuais de planejamento e controle do trabalho, e monitoração por tempo. Surgiu então o conceito de manutenção preventiva, motivado pela necessidade

de maior disponibilidade e confiabilidade das máquinas para alcançar uma produtividade mais elevada. A indústria passou a depender fortemente do bom funcionamento dos equipamentos, resultando em uma vida útil mais longa.

A partir dos anos 70, na terceira geração, o aumento da automação e da mecanização tornou evidente que a confiabilidade e a disponibilidade eram essenciais em vários setores, como transporte, serviços de saúde, processamento de dados, telecomunicações e gerenciamento de edificações. A frequência de falhas aumentou com a automação, o que prejudicou a capacidade de manter padrões de qualidade estabelecidos. Este cenário reforçou a importância da manutenção preditiva, que é baseada na previsão de falhas por meio de análises de desempenho e estatísticas, o que permite intervenções rápidas e a prevenção de falhas catastróficas.

Essa evolução histórica mostra como a manutenção industrial passou de uma abordagem reativa para estratégias mais preventivas e preditivas. Isso evidencia como a manutenção preditiva está se tornando cada vez mais importante para garantir a eficiência e segurança das operações industriais.

Os eixos mecânicos são componentes essenciais em diversas máquinas, sendo as falhas por fadiga representantes de um problema significativo. Esses componentes estão frequentemente sujeitos a cargas cíclicas que podem levar ao desenvolvimento de fissuras que podem acarretar, eventualmente, à fratura. A aplicação de manutenções preditivas em eixos mecânicos é especialmente importante para evitar essas falhas, permitindo intervenções antes que as condições se tornem críticas. A identificação precoce de sinais de fadiga pode prolongar a vida útil dos eixos e melhorar a segurança e a confiabilidade das máquinas.

No contexto da engenharia mecânica, o livro "Projeto de Máquinas: Uma abordagem integrada" (Norton 2010) aborda o diagrama S-N como uma ferramenta fundamental para avaliar a resistência à fadiga de materiais metálicos em máquinas. O diagrama S-N é uma representação gráfica que relaciona a amplitude da tensão cíclica com o número de ciclos até a falha. Isso é especialmente importante no contexto da manutenção preditiva, pois permite prever a vida útil de componentes e identificar potenciais pontos de falha antes que ocorram.

Quando se considera o aspecto econômico das falhas por fadiga, percebe-se que há um custo financeiro significativo em relação à fadiga para a economia de um país. Nos Estados Unidos, estimou-se que o impacto econômico da fratura de componentes e estruturas, em sua maioria causada por fadiga, foi da ordem de 119 bilhões de dólares em 1982, quantia que equivale a, aproximadamente, 4% do PIB dos EUA no ano de referência (Reed e Smith 1983). Parte desse impacto econômico poderia ter sido reduzido caso fossem realizadas manutenções preditivas que impediriam falhas catastróficas ou a parada do maquinário em horários de produção.

Neste contexto, este trabalho apresenta uma solução baseada na integração de ferramentas de cálculo, estruturação de dados e análise de tensões para determinar o ciclo

de vida dos eixos mecânicos. Utilizando o diagrama SN, que relaciona a amplitude da tensão aplicada ao número de ciclos até a falha, o sistema desenvolvido calcula a durabilidade dos eixos e armazena essas informações de forma organizada. Essas informações são continuamente atualizadas com o número de ciclos já realizados, permitindo uma análise preditiva do estado dos componentes.

A metodologia proposta envolve a coleta de dados geométricos e de material dos eixos, a aplicação de cargas e a realização de cálculos detalhados de tensões e resistências. Os resultados são então utilizados para prever o número de ciclos até a falha, ajudando a identificar pontos críticos e planejar intervenções de manutenção antes que ocorram falhas catastróficas. Além disso, o sistema oferece uma interface intuitiva para a entrada de dados e geração de relatórios, tornando-se uma ferramenta prática e acessível para engenheiros de manutenção.

Este trabalho não só contribui para a redução de custos e melhoria da eficiência operacional, mas também promove a segurança e a sustentabilidade nas operações industriais. Ao implementar uma solução robusta para a manutenção preditiva de eixos mecânicos, espera-se fornecer uma base sólida para futuras pesquisas e desenvolvimentos na área de manutenção industrial e engenharia mecânica.

1.1 Objetivos

O objetivo geral desse trabalho desenvolver e implementar uma metodologia integrada baseada no diagrama SN (*Stress-Number*) para calcular o ciclo de vida de eixos mecânicos, com o intuito de facilitar a manutenção preditiva, aumentar a eficiência operacional, reduzir custos de manutenção, e prolongar a vida útil dos equipamentos. Este objetivo inclui a criação de um sistema robusto que permita a coleta precisa de dados geométricos e materiais dos eixos, a aplicação de cargas e realização de cálculos detalhados de tensões e resistências, bem como a estruturação de um banco de dados simplificado que deverá ser continuamente atualizado com informações dos ciclos de vida já realizados pelos eixos cadastrados, com o intuito que auxilie na análise preditiva do estado dos componentes.

Para que esse objetivo seja atingido os seguintes objetivos específicos foram propostos:

- Capturar informações geométricas e sobre o material que compõem o eixo mecânico em análise;
- Coletar informações sobre os apoios e cargas aplicadas ao componente;
- Obter informações sobre os esforços solicitantes de força cortante e momento fletor ao longo do comprimento do eixo;

- Utilizar critérios de falha por fadiga e o diagrama S-N para estimar a vida do eixo mecânico;
- Armazenar as informações obtidas em um banco de dados simplificado para identificar os eixos que necessitam de manutenção;

1.2 Estrutura do Texto

Esta monografia está organizada da seguinte forma:

- O capítulo 2 apresenta a fundamentação teórica que define o método da obtenção do número de ciclos da vida do eixo mecânico, com os conceitos dos tipos de manutenção existentes na indústria.
- O capítulo 3 descreve a o modelo proposto para esse trabalho e os métodos realizados;
- O capítulo 4 apresenta os resultados obtidos e as discussões relacionadas;
- O capítulo 5 contém as conclusões do trabalho;

2 FUNDAMENTAÇÃO TEÓRICA

Para realizar o projeto de um sistema para manutenção preditiva de eixos mecânicos em ambientes industriais é necessário compreender diversos conceitos fundamentais. Primeiramente, é preciso entender os métodos de manutenção, principalmente os mais utilizados historicamente, a manutenção corretiva, preventiva e preditiva, para escolher a abordagem mais adequada. A teoria da fadiga dos materiais, que envolve a formação e crescimento de trincas sob carregamentos cíclicos, o diagrama S-N, que relaciona a amplitude da tensão cíclica ao número de ciclos até a falha, fornece uma base para estimar a vida útil dos eixos mecânicos. Além disso, o conhecimento sobre concentradores de tensão e o fator de concentração de tensão vitais para identificar pontos críticos nos componentes onde o surgimento de trincas é mais propício.

No cenário industrial contemporâneo, a manutenção preditiva tem ganhado destaque por sua capacidade de prever falhas antes que elas ocorram, permitindo intervenções planejadas e evitando paradas inesperadas. Este tipo de manutenção depende da identificação de padrões que indicam a necessidade de manutenção, monitorando continuamente as condições operacionais dos equipamentos e analisando dados históricos (Lafraia 2001). Para criar um sistema eficiente e confiável, é necessário conhecimento das técnicas de manutenção, bem como o comportamento dos materiais sob fadiga, os limites teóricos de resistência e a utilização do diagrama S-N.

2.1 Métodos de Manutenção

2.1.1 Manutenção Corretiva

De acordo com (Pinto e Xavier 2001), este método consiste em uma intervenção não planejada que ocorre somente quando o equipamento perde sua função. Este método é aplicado apenas quando a falha do equipamento não traz consequências significativas para o processo produtivo. Exemplos típicos incluem motores de pequena potência, sistemas de ar condicionado e exaustores de restaurantes.

Quando a manutenção corretiva é aplicada inadequadamente, podem surgir várias consequências negativas, tais como: perda de produção, destruição catastrófica de equipamentos, planejamento ineficiente da mão de obra, baixa disponibilidade dos equipamentos, riscos de segurança e queda na qualidade dos produtos.

A manutenção corretiva pode ser dividida em duas categorias: planejada e não planejada.

- **Manutenção Corretiva Planejada:** Ocorre quando a correção é feita com base em um acompanhamento preditivo ou por decisão gerencial, ou seja, intervém-se com base

em monitoramento contínuo ou pela decisão consciente de operar o equipamento até sua falha.

- **Manutenção Corretiva Não Planejada:** Acontece quando se percebe que o equipamento não está produzindo conforme o esperado. Nesta situação, a manutenção é realizada em resposta a um problema já ocorrido, geralmente resultando em altos custos para a empresa devido à interrupção inesperada da produção e aos reparos emergenciais necessários

2.1.2 Manutenção Preventiva

A manutenção preventiva permite a realização de manutenção seguindo um cronograma pré-estabelecido pela empresa, com o objetivo de reduzir falhas, custos e queda no desempenho (Siqueira 2009). Seu principal foco é minimizar a probabilidade de falha de um equipamento.

Esse tipo de manutenção inclui a realização de rotineiras inspeções, ajustes, limpezas, lubrificação, substituição de peças, calibração e reparo de equipamentos e componentes. Como este método é aplicado em intervalos regulares sem levar em consideração as condições atuais do equipamento, ele é conhecido como manutenção baseada no tempo.

A realização regular de inspeções e manutenções em intervalos regulares pode aumentar a disponibilidade dos equipamentos e reduzir a probabilidade de falhas imprevistas.

O TMEF (Tempo Médio Entre Falhas) é uma ferramenta útil para calcular os períodos de manutenção. Entretanto, como muitos elementos podem ter falhas aleatórias ou períodos de utilização variáveis, esse critério nem sempre dá resultados precisos.

Embora a manutenção preventiva regular possa ser útil, ela também pode resultar em custos excessivos devido a paradas desnecessárias do equipamento, gastos excessivos com peças e riscos de danos ao equipamento por montagem inadequada.

2.1.3 Manutenção Preditiva

A manutenção preventiva busca intervir somente quando as instalações realmente necessitam (12). Esse método identifica problemas incipientes antes que se tornem críticos, permitindo um planejamento mais preciso e eficiente.

Também conhecida como manutenção baseada na condição, a manutenção preventiva utiliza técnicas de inspeção para monitorar a evolução do estado dos equipamentos, possibilitando intervenções no momento mais apropriado. Esse tipo de manutenção é viável quando o componente apresenta sinais que indicam seu processo de falha. Os principais sinais de uma falha iminente nos componentes incluem: mudanças no nível de vibração, aumento da temperatura, variação na espessura, presença de trincas e desgaste.

Diversas tecnologias foram desenvolvidas para avaliar o estado dos equipamentos. As principais incluem: análise de vibração, emissão acústica, análise do óleo, termogra-

fia, testes não destrutivos, medições de fluxo, análise de motores elétricos, detecção de vazamentos, monitoramento da corrosão e análises visuais e auditivas.

A implementação adequada de um programa de manutenção preventiva pode trazer diversos benefícios, como: máxima disponibilidade das máquinas; planejamento eficiente da mão-de-obra; gestão eficaz das peças sobressalentes; aumento da segurança operacional; melhoria na qualidade da manutenção; e gerenciamento global dos recursos

No entanto, a aplicação da manutenção preditiva é limitada pela disponibilidade de técnicas efetivas de monitoramento e pela relação custo/benefício de sua implantação. Este trabalho desenvolve técnicas efetivas e com boa relação custo/benefício para a manutenção preditiva de eixos mecânicos em ambientes industriais. Ao implementar um método robusto de monitoramento e análise, espera-se não só aumentar a eficiência operacional e reduzir custos, mas também garantir a segurança e a longevidade dos equipamentos, contribuindo significativamente para a sustentabilidade e a produtividade industrial.

2.2 Processo e Falha por Fadiga

A fadiga dos materiais é um fenômeno que provoca a degradação gradual das propriedades mecânicas de um material devido à aplicação repetida de cargas cíclicas, resultando em falhas mecânicas. Esse processo é caracterizado pela formação e crescimento de trincas em componentes sujeitos a carregamentos variáveis, que eventualmente levam à fratura completa após um determinado número de ciclos de carga.

2.2.1 Contexto Histórico

Os primeiros registros de falhas por fadiga remontam ao início dos anos 1800, quando eixos de vagões ferroviários começaram a apresentar fraturas após um curto período de serviço. Estes eixos, fabricados em aço dúctil, mostravam fraturas que pareciam frágeis e abruptas. Em 1843, Rankine sugeriu que as tensões flutuantes haviam "cristalizado" o material, tornando-o frágil. Naquela época, os projetos dos eixos eram baseados em experiências com estruturas sujeitas a cargas estáticas, sendo as cargas dinâmicas uma novidade trazida pelas máquinas a vapor.

Os eixos dos vagões, girando juntamente com as rodas, submetiam-se a tensões de flexão alternadas entre valores positivos e negativos, caracterizando um carregamento alternado. O engenheiro alemão August Wohler conduziu a primeira investigação científica das falhas por fadiga, testando eixos em laboratório até a falha sob carregamento alternado. Suas descobertas, publicadas em 1870, mostraram que o número de ciclos de tensão variável era responsável pelo colapso e que existia um limite de resistência à fadiga para os aços, um nível de tensão que suportava milhões de ciclos de tensão alternada.

O termo "fadiga" foi cunhado por Poncelet em 1839 para descrever esse fenômeno. Inicialmente, a causa das falhas por fadiga não era compreendida, e a aparência frágil

das fraturas em materiais dúcteis levou à especulação de que o material “cansava” devido às cargas oscilantes. Wohler mais tarde demonstrou que as metades dos eixos quebrados ainda mantinham a mesma resistência e ductilidade em testes de tração quanto o material original, consolidando o entendimento de que a fadiga era causada pelas cargas cíclicas. Apesar disso, o termo “falha por fadiga” continuou a ser utilizado para descrever falhas devido a cargas variáveis no tempo (Norton 2010).

2.2.2 Carregamento Cíclico

O carregamento cíclico consiste em variar o carregamento entre os limites mínimo (σ_{min}) e máximo (σ_{max}) de tensão, mantendo esses limites constantes enquanto o carregamento varia, como em uma função senoidal. Esse tipo de carregamento, chamado de Carregamento de Amplitude Constante, é comumente utilizado em testes de fadiga de materiais.

A faixa de tensão é dada pela diferença entre os valores máximos e mínimos de tensões:

$$\Delta\sigma = \sigma_{max} - \sigma_{min} \quad (1)$$

A tensão média (σ_m) é a média aritmética dos valores máximo e mínimo de tensão:

$$\sigma_m = \frac{\sigma_{max} + \sigma_{min}}{2} \quad (2)$$

A metade da faixa de tensão é chamada de amplitude de tensões (σ_a), que é a variação em relação à média:

$$\sigma_a = \frac{\Delta\sigma}{2} = \frac{\sigma_{max} - \sigma_{min}}{2} \quad (3)$$

Carregamentos cíclicos com média nula podem ser especificados pela amplitude de tensões (σ_a) ou pelo valor máximo de tensão (σ_{max}), que é igual, numericamente, à amplitude de tensões. Se a tensão média não é zero, é necessário utilizar duas variáveis independentes para especificar o carregamento, como σ_a e σ_m ou σ_{max} .

2.2.3 Mecanismo de Fadiga

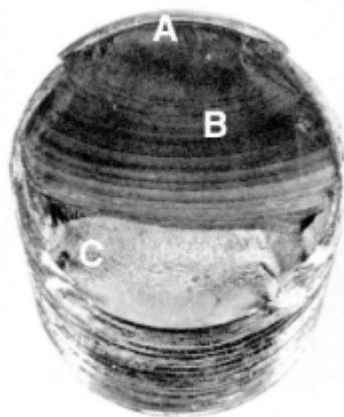
As falhas por fadiga geralmente começam com pequenas trincas, que podem estar presentes desde a fabricação do material ou se desenvolver ao longo do tempo devido a deformações cíclicas em torno de concentrações de tensões. Entender esse fenômeno é essencial na engenharia mecânica para prever a vida útil dos componentes e implementar estratégias de manutenção eficazes. Identificar precocemente sinais de fadiga pode aumentar significativamente a vida útil dos eixos mecânicos, melhorando a segurança e a confiabilidade das máquinas industriais.

O processo de fadiga é geralmente dividido em três estágios:

1. **Estágio I - Iniciação de Trinca:** Microtrincas se formam devido à deformação plástica cíclica, geralmente em locais de concentração de tensões, como superfícies rugosas, entalhes ou inclusões internas. Essas trincas são microscópicas e não visíveis a olho nu.
2. **Estágio II - Propagação de Trinca:** As microtrincas evoluem para macrotrincas, formando superfícies de fratura com platôs paralelos e sulcos, conhecidos como marcas de praia ou marcas de concha de ostra. Durante o carregamento cíclico, essas superfícies abrem e fecham repetidamente.
3. **Estágio III - Fratura Final:** A trinca atinge um tamanho crítico, resultando em uma fratura súbita e completa do componente. A aparência da fratura pode ser frágil, dúctil ou uma combinação de ambas, com possíveis marcas de praia indicando as origens das trincas iniciais (Budynas e Nisbett 2016).

Os estágios I, II e III são representados na Figura 2.

Figura 2 – Representação dos 3 estágios da falha por fadiga



Fonte: (ASM 1990)

2.2.4 Métodos de Análise e Monitoramento

A análise da fadiga envolve diversas metodologias e ferramentas, como o diagrama S-N, que relaciona a amplitude da tensão cíclica ao número de ciclos até a falha, e técnicas de monitoramento como análise de vibração, termografia e ensaios não destrutivos. A aplicação dessas técnicas em programas de manutenção preditiva permite identificar e mitigar os riscos de falha por fadiga, assegurando a integridade estrutural dos componentes e a continuidade das operações industriais.

No entanto, esses métodos apresentam algumas desvantagens. O custo dos equipamentos de monitoramento e análise pode ser elevado, representando um investimento significativo para as empresas. Além disso, muitos desses métodos requerem a parada das

máquinas para a realização de inspeções, o que pode interromper a produção e resultar em perdas econômicas. A necessidade de técnicos especializados para operar e interpretar os resultados também aumenta os custos operacionais, tornando o processo mais complexo e oneroso.

2.3 Concentradores de Tensão

Entalhes são descontinuidades geométricas, como furos, filetes, sulcos de anéis elásticos e rasgos de chaveta, essas estruturas são inevitavelmente encontradas na maioria dos componentes mecânicos das máquinas e equipamentos industriais. Apesar de serem necessários para a funcionalidade dos componentes, os detalhes afetam severamente a distribuição de tensão local, aumentando as tensões nominais em torno de suas raízes. Esse efeito é quantificado pelo fator de concentração de tensão (FCT) K_t , definido como:

$$K_t = \frac{\sigma}{\sigma_n} \quad (4)$$

Em que σ é a tensão máxima na raiz do entalhe e σ_n é a tensão nominal que atuaria ali caso o entalhe não tivesse efeito sobre a distribuição de tensão. Altos valores de K_t podem ter um efeito muito prejudicial em mecanismos de falhas, pois servem como iniciadores de trincas. A presença de um entalhe em um componente pode aumentar significativamente a tensão local em relação à tensão aplicada nominalmente. O efeito de como um furo em uma placa concentra linhas de tensão representativas, indicando a redistribuição de tensões ao redor do entalhe é mostrado na Figura 3.

Figura 3 – Representação de linhas de fluxo em chapa com concentrador de tensão



Fonte: (Simões 2012)

2.3.1 Fator de Concentração de Tensão

O fator de concentração de tensão K_t é uma medida adimensional que descreve quanto a tensão é aumentada em torno de um entalhe em comparação com a tensão nominal aplicada. A determinação de K_t pode ser feita experimentalmente, através de ensaios mecânicos, ou analiticamente, utilizando-se fórmulas empíricas e métodos numéricos como o Método dos Elementos Finitos (MEF).

Na prática, K_t é usado na estimativa do fator de amplificação de fadiga, K_f , esse fator busca ajustar a tensão nominal em cálculos de fadiga e fratura, fornecendo uma estimativa mais precisa da vida útil do componente. Para materiais dúcteis, onde a plasticidade local pode redistribuir as tensões, o fator de concentração de tensão efetivo pode ser menor do que o valor teórico K_t . No entanto, para materiais frágeis, onde a redistribuição de tensão é limitada, o valor de K_t teórico é normalmente uma boa aproximação.

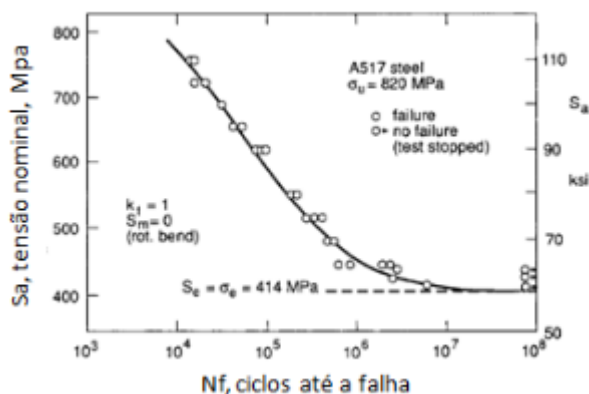
2.4 Diagrama S-N

A análise da fadiga dos materiais é fundamental na engenharia, pois muitos componentes estruturais são submetidos a carregamentos cíclicos ao longo de sua vida útil. A resistência à fadiga é determinada pela aplicação de diferentes níveis de tensão cíclica em corpos de prova e pela medição do número de ciclos até a falha. A representação gráfica dessa relação é conhecida como diagrama S-N, que ilustra a amplitude da tensão cíclica (S) em relação ao número de ciclos até a falha (N).

O diagrama S-N, também denominado curva de Wohler, é uma ferramenta crucial na caracterização do comportamento de fadiga dos materiais. Este gráfico mostra como diferentes níveis de tensão influenciam a durabilidade de um material. A curva típica de S-N apresenta uma queda exponencial, onde, em tensões mais altas, o número de ciclos até a falha é relativamente baixo, enquanto em tensões mais baixas, o material pode suportar um número maior de ciclos.

Uma curva S-N típica é ilustrada na Figura 4, em que a resistência à fadiga é definida como a tensão na qual a falha por fadiga ocorre após um determinado número de ciclos. Esta informação é vital para engenheiros e projetistas durante o processo de design, permitindo a previsão da vida útil dos componentes e a implementação de medidas preventivas para evitar falhas prematuras.

Figura 4 – Exemplo de Diagrama S-N



Fonte: (Dowling 2013)

Ao integrar a análise do diagrama S-N com a compreensão dos efeitos dos entalhes, tema abordado na Seção 2.3, é possível ajustar as tensões nominais para refletir a realidade das tensões aumentadas na presença de entalhes. Isso resulta em previsões mais precisas da vida útil dos componentes e permite a implementação de estratégias que minimizem os efeitos adversos dos entalhes, como a utilização de raios de curvatura maiores e tratamentos superficiais para reduzir as concentrações de tensão.

O diagrama S-N é essencial na definição de estratégias de manutenção preditiva e preventiva. Conhecendo a resistência à fadiga de um material, é possível estabelecer intervalos de inspeção apropriados e identificar componentes que estão em risco de falha antes que ocorram.

3 DESENVOLVIMENTO

Partindo-se do conhecimento teórico levantado na seção anterior, seguiu-se para o desenvolvimento do sistema de manutenção preditiva. Para que possa atender às demandas de grandes indústrias, o sistema basei-se em um banco de dados em que os eixos mecânicos devem ser cadastrados. Esse cadastro conta com uma identificação para cada eixo e com o número de ciclos que o eixo é capaz de realizar até a falha. Além disso, esse banco de dados é periodicamente alimentado de maneira com que o número de ciclos já realizados por cada eixo cadastrado seja armazenado.

Para serem cadastrados no sistema de manutenção preditiva, os eixos mecânicos devem atender a determinadas características. Primeiramente, é necessário que sejam fabricados a partir de materiais ferrosos, alumínio ou cobre garantindo a resistência e durabilidade exigidas por aplicações industriais. Além disso, os eixos devem possuir uma seção circular. Essas especificações atendem a grande parte dos eixos utilizados nas indústrias e, por isso, foram priorizadas no desenvolvimento deste trabalho.

O sistema é formulado em duas grades etapas, a primeira etapa consiste no cadastro do eixo mecânico no banco de dados, tendo como maior foco o cálculo do número de ciclos que o eixo pode realizar até a falha. Já a segunda etapa tem como foco a somatória recorrente dos ciclos já realizados por cada um dos eixos, de maneira que os valores do número de ciclos totais até a falha e o número de ciclos já realizados por cada eixo possam ser comparados. A dinâmica de funcionamento da primeira e segunda etapa do sistema é ilustrada pelas Figuras 5 e 6.

Embora o desenvolvimento deste trabalho foque na metodologia de manutenção preditiva para eixos mecânicos e no cálculo da sua vida útil, é importante ressaltar que a captura de dados dos ciclos realizados pelos eixos e a geração do relatório de atualização do banco de dados com os ciclos realizados não estão incluídas no escopo deste estudo. A atualização das informações no banco de dados, entretanto, faz parte do escopo deste trabalho, sendo essencial para a eficácia do sistema de manutenção preditiva proposto. A implementação de um sistema de monitoramento para capturar e registrar os ciclos realizados pelos eixos, representa uma oportunidade para pesquisas futuras. Investigar formas eficazes de integrar sensores e sistemas de armazenamento de dados em tempo real é um passo importante para aprimorar o método desenvolvido.

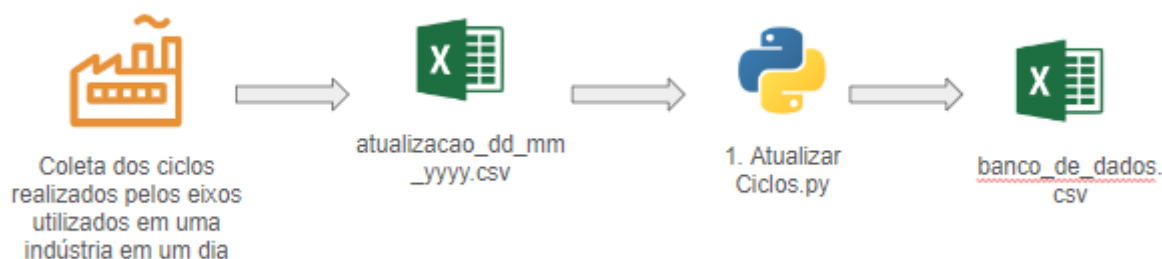
Para determinar o número de ciclos de vida até a falha na etapa de cadastro do eixo mecânico, é necessário primeiramente obter os esforços solicitantes aplicados ao eixo. Em seguida, deve-se identificar os pontos críticos e suas propriedades específicas, sendo que esses pontos críticos normalmente são aqueles em que o eixo possui algum tipo de entalhe. Para esses pontos críticos, calcula-se o fator de concentração de tensões, a fim de

Figura 5 – Fluxo de dados da etapa de cadastro do eixo



Fonte: Autor (2024)

Figura 6 – Fluxo de dados da etapa de atualização de ciclos já realizados



Fonte: Autor (2024)

determinar a tensão alternada nominal e a tensão alternada equivalente. Assim, pode-se calcular o número máximo de ciclos de vida do eixo mecânico, considerando o menor valor entre os ciclos de vida de cada um dos pontos críticos analisados.

Em suma, foram desenvolvidos dois conjuntos de códigos para a implementação da metodologia proposta. O primeiro conjunto, responsável pelo cadastro dos eixos e pelo cálculo do número de ciclos até a falha, está detalhado no Apêndice A. O segundo conjunto de códigos, que trata da atualização dos ciclos já realizados pelos eixos cadastrados, está descrito no Apêndice B. Ambos os apêndices fornecem uma visão abrangente dos algoritmos utilizados, facilitando a reprodução e compreensão dos processos envolvidos.

3.1 Cálculo dos Esforços Solicitantes

Para calcular os esforços solicitantes aplicados ao eixo, realizou-se um processo que envolve a coleta de informações sobre o eixo, apoios, cargas aplicadas e a definição dos esforços solicitantes. Este processo é detalhadamente apresentado nessa seção.

3.1.1 Coleta do Comprimento do Eixo

Primeiramente, o comprimento do eixo é definido pelo usuário, que insere o valor manualmente em milímetros. Este valor é utilizado diretamente nos cálculos subsequentes, sendo um dado fundamental para a caracterização geométrica do eixo mecânico.

3.1.2 Identificação dos Apoios

Em seguida, é necessário identificar os apoios presentes no sistema. Os apoios podem ser classificados como simples ou móveis, dependendo da configuração do sistema. A identificação correta dos apoios é essencial para a definição das reações de apoio que o eixo irá sofrer, influenciadas pela rigidez e pelo tipo de ligação entre o eixo e os apoios. Este processo inclui a determinação do número de apoios e a posição de cada apoio ao longo do eixo, esses dados são inseridos manualmente pelo usuário e então essas informações são estruturadas de maneira organizada.

3.1.3 Escolha do Plano de Análise

Após a identificação dos apoios, procede-se à escolha do plano de análise. A análise dos esforços solicitantes pode ser realizada em um plano ou em dois planos, dependendo da complexidade do sistema e da distribuição das forças. Para sistemas mais simples, onde as forças atuam predominantemente em um plano, a análise uniplanar é suficiente. No entanto, para sistemas mais complexos, onde as forças atuam em múltiplos planos, a análise biplanar é necessária. Esta escolha é realizada pelo usuário e deve ser feita com base nas condições de carregamento e na distribuição das forças sobre o eixo.

3.1.4 Obtenção das Cargas Aplicadas

Com base nas informações coletadas sobre o eixo e os apoios, define-se as cargas aplicadas. O sistema solicita ao usuário o número total de cargas externas aplicadas no eixo. Se o número inserido for menor ou igual a zero, o sistema acusa um erro, pois deve haver pelo menos uma carga aplicada. Para cada carga, o usuário deve fornecer o tipo de carga (radial, momento, distribuída ou axial), o valor e a posição da carga ao longo do eixo.

Para cargas radiais, axiais e momentos, o usuário insere diretamente o valor da carga e a posição em milímetros. Já para cargas distribuídas, o usuário deve fornecer o valor inicial e final da carga em N/mm, bem como as posições inicial e final ao longo do eixo. O sistema então estrutura essas cargas distribuídas, calculando a força total e a posição média da carga. No caso de cargas distribuídas que mudam de sentido, a carga é dividida em duas partes: uma representando um triângulo à esquerda e outra representando um triângulo à direita, cada uma com suas respectivas forças totais e posições médias.

3.1.5 Verificação da Determinação Estática do Eixo

Antes de prosseguir com os cálculos de esforços, é necessário verificar se o eixo é estaticamente determinado ou indeterminado. Esta verificação é feita comparando o número de equações de equilíbrio disponíveis com o número de incógnitas (forças de reação nos apoios). Para um sistema ser estaticamente determinado, o número de incógnitas deve ser igual ao número de equações de equilíbrio disponíveis. Se o número de incógnitas for maior que o número de equações, o sistema é estaticamente indeterminado. Se o número de incógnitas for menor, o sistema é subapoiado.

Para determinar o número de incógnitas, contabiliza-se o número de reações nos apoios. Cada tipo de apoio tem um número específico de reações: apoios móveis têm 0.5 reações, apoios fixos têm 1.5 reações e engastes têm 2 reações. A soma dessas reações é comparada ao número de equações de equilíbrio disponíveis, que são duas em um sistema estático em um plano (equilíbrio de forças verticais e equilíbrio de momentos).

3.1.6 Cálculo das Forças de Reação nos Apoios

Uma vez verificadas as condições de determinação estática, procede-se ao cálculo das forças de reação nos apoios utilizando as equações de equilíbrio estático. O primeiro passo é formar a matriz dos coeficientes, A , e a matriz dos termos independentes, B . A matriz A representa as posições relativas dos apoios e a matriz B representa as forças externas aplicadas.

A matriz A é configurada de modo que a primeira linha garanta o equilíbrio das forças verticais, e as linhas subsequentes garantem o equilíbrio dos momentos em torno de um ponto de referência (geralmente o primeiro apoio). A matriz B é preenchida com os valores das forças externas aplicadas, ajustadas pelas posições relativas ao ponto de referência.

A matriz dos coeficientes A pode ser representada da seguinte forma para um sistema com dois apoios:

$$A = \begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix} \quad (5)$$

onde x_1 e x_2 são as posições dos apoios ao longo do eixo.

A matriz dos termos independentes B é formada considerando as forças externas F_i e suas posições x_i :

$$B = \begin{bmatrix} \sum F_i \\ \sum F_i \cdot x_i \end{bmatrix} \quad (6)$$

A seguir, resolve-se o sistema de equações lineares representado por $A \cdot X = B$, em que X é o vetor das forças de reação nos apoios. Este sistema é resolvido numericamente para encontrar os valores das reações nos apoios:

$$X = A^{-1} \cdot B \quad (7)$$

Com os valores das reações calculados, essas reações são atribuídas aos respectivos apoios.

3.1.7 Definição dos Esforços Solicitantes

Com base nas informações coletadas anteriormente, define-se as forças cortantes e momentos fletores que atuam no eixo. As forças cortantes ($V(x)$) e os momentos fletores (M_x) são calculados ao longo do comprimento do eixo, levando em consideração as cargas aplicadas e as reações nos apoios.

Para calcular a força cortante em um ponto ao longo do eixo, utiliza-se a soma das forças aplicadas até aquele ponto. A fórmula utilizada é:

$$V(x) = \sum F_i \quad (8)$$

Em que F_i são as forças aplicadas, contando com as forças de reações nos apoios calculadas anteriormente, até a posição x , ao longo do eixo. Esta fórmula permite determinar o valor da força cortante em qualquer ponto ao longo do eixo.

O momento fletor em um ponto é definido pela integral da força cortante em relação à posição ao longo do eixo. A fórmula utilizada é:

$$M(x) = \int V(x) dx \quad (9)$$

Esta integral permite calcular o momento fletor em qualquer ponto ao longo do eixo, considerando a distribuição das forças cortantes.

Para calcular os esforços normais ($N(x)$), que são resultantes de cargas axiais ao longo do eixo, utiliza-se a soma das cargas axiais aplicadas até aquele ponto. A fórmula utilizada é:

$$N(x) = \sum F_{axial} \quad (10)$$

Em que F_{axial} são as forças axiais aplicadas ao eixo até a posição x .

Para realizar os cálculos de força cortante, momento fletor e esforços normais de forma programática, o sistema inicializa vetores correspondentes ao longo do comprimento do eixo com zeros. Em seguida, itera sobre as cargas aplicadas ao eixo. Para cargas radiais, a posição da carga é identificada, e a força é adicionada ao vetor de força cortante a partir dessa posição até o final do eixo. Para cargas distribuídas, a posição inicial e final da carga são identificadas, e a força total distribuída é adicionada proporcionalmente ao longo do intervalo. Para cargas axiais, a força axial é adicionada ao vetor de esforços normais na posição correspondente.

Para o cálculo do momento fletor, o sistema inicializa um vetor de momento fletor ao longo do comprimento do eixo com zeros. A força cortante ao longo do eixo é calculada previamente. Em seguida, o sistema itera ao longo do comprimento do eixo, calculando o momento fletor como a integral da força cortante.

3.1.8 Análise Biplanar

Em casos de análise biplanar, é necessário consolidar os diagramas de momento fletor e força cortante obtidos em cada plano. A análise biplanar é essencial quando as forças aplicadas ao eixo têm componentes em mais de um plano, o que exige uma avaliação mais complexa para garantir a integridade estrutural do eixo.

Primeiramente, realiza-se a análise no plano XY, coletando-se e definindo-se as cargas externas nesse plano para obter os diagramas de força cortante (V_{xy}) e momento fletor (M_{xy}). Em seguida, repete-se o processo para o plano XZ, obtendo-se os diagramas de força cortante (V_{xz}) e momento fletor (M_{xz}).

A consolidação dos resultados obtidos nos planos XY e XZ é feita através da combinação vetorial dos valores de força cortante e momento fletor. Esta combinação permite obter valores consolidados de força cortante e momento fletor em qualquer seção do eixo, levando em consideração as contribuições de ambos os planos.

As equações para a combinação vetorial são as seguintes:

Para a força cortante resultante:

$$V_{total} = \sqrt{V_{XY}^2 + V_{XZ}^2} \quad (11)$$

Para o momento fletor resultante:

$$M_{total} = \sqrt{M_{XY}^2 + M_{XZ}^2} \quad (12)$$

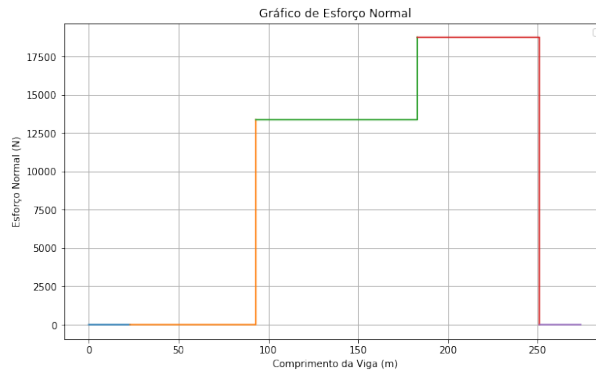
Estas equações garantem que as forças cortantes e os momentos fletores em qualquer seção do eixo sejam avaliados considerando a influência combinada das cargas em ambos os planos.

As forças axiais atuam simultaneamente em ambos os planos. Portanto, não precisam ser submetidas à consolidação na análise biplanar

3.1.9 Diagrama de Esforços Solicitantes

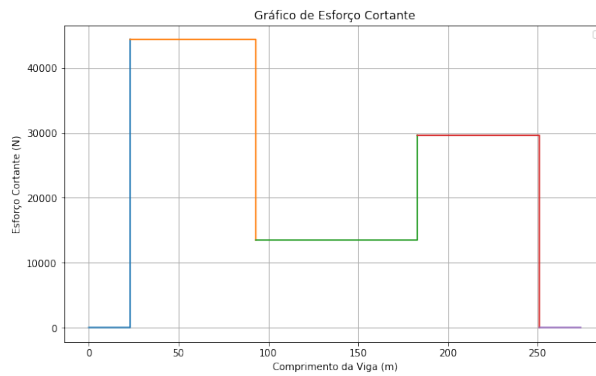
Para finalizar, o sistema plota os diagramas de esforços solicitantes de força normal, força cortante e momento fletor, respectivamente representados pelas Figuras 7, 8 e 9.

Figura 7 – Exemplo de Diagram de Esforço Normal Gerado pelo Sistema



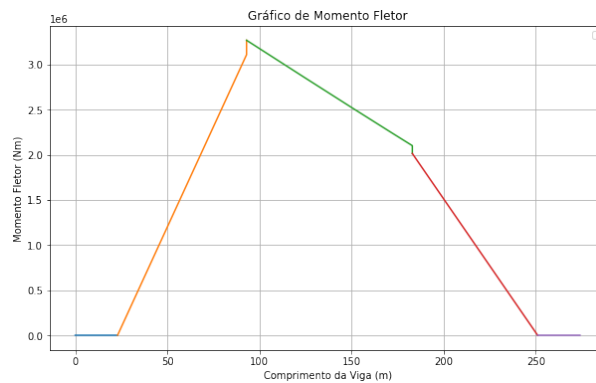
Fonte: Autor (2024)

Figura 8 – Exemplo de Diagram de Esforço Cortante Gerado pelo Sistema



Fonte: Autor (2024)

Figura 9 – Exemplo de Diagrama de Momento Fletor Gerado pelo Sistema



Fonte: Autor (2024)

3.2 Coleta de Dados do Material

A coleta de dados do material é uma etapa essencial no desenvolvimento do sistema de manutenção preditiva, pois as propriedades do material determinam a resistência e a durabilidade do eixo. O sistema permite que o usuário insira informações sobre o material, como o tipo de material, o módulo de elasticidade e a resistência à tração máxima (S_{ut}).

O tipo de material é a primeira informação solicitada ao usuário. Os materiais disponíveis para escolha no sistema são aço, ferro, alumínio e cobre. O tipo de material influencia diretamente no cálculo do limite de resistência à fadiga (Se'). Em seguida é solicitado a inserção do módulo de elasticidade, ou módulo de Young, que é uma medida da rigidez de um material. O usuário insere o módulo de elasticidade em gigapascal (GPa).

Em seguida, o usuário fornece a resistência à tração máxima (S_{ut}) do material em megapascal (MPa), que é a tensão máxima que o material pode suportar antes de falhar sob um carregamento de tração.

Com esses dados, o sistema procede ao cálculo do limite de resistência à fadiga (Se'), que é a tensão máxima que o material pode suportar sob carregamento cíclico antes de falhar devido à fadiga. Este cálculo é ajustado de acordo com o tipo de material e a resistência à tração máxima fornecida.

Em eixos mecânicos fabricados em aço, se S_{ut} for menor que 1400 MPa:

$$Se' = 0,5 \cdot S_{ut} \quad (13)$$

Caso contrário, Se' é limitado a 700 MPa.

Para materiais de ferro, se S_{ut} for menor que 400 MPa, para materiais de alumínio, se S_{ut} for menor que 330 MPa ou, então, para materiais de cobre, se S_{ut} for menor que 280 MPa:

$$Se' = 0,4 \cdot S_{ut} \quad (14)$$

Senão, Se' é limitado, respectivamente à 160, 130 ou 100 MPa.

O sistema também calcula a resistência do material em 10^3 ciclos (S_m), utilizando a seguinte expressão:

$$S_m = 0,9 \cdot S_{ut} \quad (15)$$

Após a inserção dos dados, o sistema verifica se os valores fornecidos são positivos e maiores que zero. Se qualquer valor inserido for inválido, o sistema exibe uma mensagem de erro e solicita a entrada de um valor válido. Uma vez validados, os valores de módulo de elasticidade, resistência à tração máxima, limite de resistência à fadiga e limite de resistência à fadiga em carregamento reverso são armazenados em um dicionário. Estes valores são utilizados nas etapas subsequentes do cálculo de vida útil do eixo.

3.3 Coleta e Estruturação dos Pontos Críticos

Para identificar os pontos críticos ao longo do eixo, o sistema desenvolvido segue um processo que combina análise geométrica e de carregamento. Os pontos críticos são regiões em que, normalmente, há um concentrador de tensões, frequentemente devido a entalhes ou mudanças na seção transversal, que aumentam a probabilidade de falha por fadiga.

3.3.1 Processo de Identificação dos Pontos Críticos

Primeiramente, o usuário deve fornecer a posição exata, em milímetros, dos pontos críticos ao longo do eixo, assim como as características geométricas e de carregamento desses pontos. Especificamente, o sistema solicita os seguintes dados para cada ponto crítico:

- **Posição do ponto crítico:** Inserida em milímetros, é a localização exata ao longo do comprimento do eixo onde o ponto crítico se encontra;
- **Torque atuante:** Valor do torque aplicado no ponto crítico, medido em N.mm;
- **Diâmetro na posição:** O diâmetro do eixo no ponto crítico, em milímetros. Este é tipicamente o menor diâmetro em caso de mudanças na seção transversal;
- **Raio do entalhe:** O raio do entalhe presente no ponto crítico, em milímetros.

Além disso, para que se faça uma correlação com componentes em operação devem ser considerados diversos aspectos de correção. Deste modo, são solicitados diversos coeficientes, como:

- Coeficiente de carga (c_{load});
- Coeficiente de carga (c_{load});
- Coeficiente de tamanho (c_{size});
- Coeficiente de superfície (c_{surf});
- Coeficiente de temperatura (c_{temp});
- Coeficiente de confiabilidade (c_{reliab});
- Coeficiente para outros efeitos (c_{others}).

Todos esses coeficientes devem ser calculados previamente pelo usuário, utilizando os métodos descritos por (Norton 2010), para então serem inseridos no sistema.

Durante a coleta de dados dos pontos críticos, o sistema também solicita que o usuário forneça os fatores de concentração de tensões K_t para flexão, tração e compressão e K_{ts} para torção. Esses fatores são utilizados para calcular os fatores de concentração de tensões ajustados K_f e K_{fs} , que consideram a sensibilidade ao entalhe do material.

3.4 Cálculo das Tensões Alternadas e do Fator de Concentração de Tensões Ajustado

3.4.1 Cálculo do Limite Teórico de Resistência à Fadiga

O limite teórico de resistência à fadiga (S'_e) é calculado com base na resistência à tração máxima (S_{ut}) do material, ajustada conforme o tipo de material. Para aços, é utilizado o valor de $0,5 S_{ut}$ para resistências menores que 1400 MPa, enquanto para valores superiores, o limite é fixado em 700 MPa. Para outros materiais como ferro, alumínio e ligas de cobre, os limites são ajustados para $0,4 S_{ut}$, com seus próprios valores máximos

específicos.

$$S'_e = \begin{cases} 0,5S_{ut} & \text{para } S_{ut} < 1400 \text{ MPa} \\ 700 \text{ MPa} & \text{para } S_{ut} \geq 1400 \text{ MPa} \end{cases} \quad (16)$$

Os valores ajustados são calculados considerando os coeficientes de correção, que incluem os efeitos de carga, tamanho, acabamento superficial, temperatura de operação, confiabilidade e outros fatores.

$$S_e = c_{\text{load}} \cdot c_{\text{size}} \cdot c_{\text{surf}} \cdot c_{\text{temp}} \cdot c_{\text{reliab}} \cdot c_{\text{other}} \cdot S'_e \quad (17)$$

3.4.2 Cálculo das Tensões Alternadas

O sistema realiza o cálculo das tensões alternadas para a análise de fadiga. As tensões alternadas são calculadas a partir dos momentos fletores, torques e forças normais aplicados no eixo. O processo é detalhado a seguir.

Primeiramente, as tensões alternadas nominais são determinadas a partir dos momentos fletores (M), torques (T) e forças normais (N), usando as seguintes equações:

Para tensões alternadas de flexão (σ_{alt}):

$$\sigma_{\text{alt}} = \frac{M \cdot c}{I} \quad (18)$$

onde:

- M é o momento fletor,
- c é a distância do centroide à superfície,
- I é o momento de inércia da seção transversal.

Para tensões alternadas de torção (τ_{alt}):

$$\tau_{\text{alt}} = \frac{T \cdot r}{J} \quad (19)$$

Em que:

- T é o torque aplicado,
- r é o raio do eixo,
- J é o momento polar de inércia da seção transversal.

Para tensões alternadas normais (σ_{axial}):

$$\sigma_{\text{axial}} = \frac{N}{A} \quad (20)$$

Em que:

- N é a força normal aplicada,
- A é a área da seção transversal do eixo.

As tensões alternadas equivalentes são então ajustadas pelos fatores de concentração de tensões ajustados (K_f , K_{fs} e K_{fa}):

Para tensões alternadas de flexão equivalentes (σ'_{alt}):

$$\sigma'_{\text{alt}} = K_f \cdot \sigma_{\text{alt}} \quad (21)$$

Para tensões alternadas de torção equivalentes (τ'_{alt}):

$$\tau'_{\text{alt}} = K_{fs} \cdot \tau_{\text{alt}} \quad (22)$$

Para tensões alternadas normais equivalentes (σ'_{axial}):

$$\sigma'_{\text{axial}} = K_{fa} \cdot \sigma_{\text{axial}} \quad (23)$$

3.4.3 Tensão Equivalente de von Mises

A tensão equivalente de von Mises é usada para combinar os efeitos das tensões de flexão, normais e de cisalhamento, fornecendo um único valor que representa a intensidade de todas as tensões atuantes. A tensão de von Mises é calculada utilizando a seguinte fórmula:

$$\sigma'_{vM} = \sqrt{(\sigma'_{\text{alt}})^2 + (\sigma'_{\text{axial}})^2 + 3(\tau'_{\text{alt}})^2} \quad (24)$$

Em que:

- $\sigma_{\text{flexão}}$ é a tensão de flexão,
- σ_{axial} é a tensão normal,
- $\tau_{\text{torção}}$ é a tensão de torção.

Esse racional é aplicado em todos os pontos críticos listados pelo usuário. Essas tensões equivalentes são usadas para avaliar a vida em fadiga do componente, comparando-as com os limites de resistência à fadiga e utilizando o diagramas S-N.

3.4.4 Cálculo do Número de Ciclos até a Falha (Método do Diagrama S-N)

O número de ciclos até a falha é determinado utilizando o diagrama S-N, que relaciona a tensão alternada (σ_a) com o número de ciclos de vida do componente. O diagrama S-N é particularmente útil para materiais com limite de fadiga, como aços, onde é possível definir uma vida infinita para tensões abaixo do limite de fadiga (S_e).

Para tensões alternadas (σ_a) menores ou iguais a S_e , considera-se que o componente possui vida infinita:

$$\text{Se } \sigma_a \leq S_e, \text{ a vida é infinita}$$

Para tensões alternadas entre S_e e o limite de resistência (S_m), o número de ciclos (N) é calculado utilizando a relação log-log:

$$\log(N) = \frac{6 - 3 \log(\sigma_a/S_e)}{\log(S_m/S_e)} \quad (25)$$

Onde S_m é a resistência do material em 10^3 ciclos:

$$S_m = 0,9S_{ut} \quad (26)$$

Com isso, a expressão para N se torna:

$$N = 10^{\left(\frac{6-3 \log(\sigma_a/S_e)}{\log(S_m/S_e)}\right)} \quad (27)$$

Com isso, é possível calcular o número de ciclos até a falha de cada ponto crítico do eixo mecânico individualmente com base nas tensões alternadas atuantes e nas propriedades do material.

3.4.5 Identificação do Menor Número de Ciclos até a Falha

Após calcular o número de ciclos para cada um dos pontos críticos, o sistema identifica o menor deles para determinar a vida útil do eixo. Esta etapa consiste em o sistema percorrer todos os pontos críticos e comparar os valores de número de ciclos calculados.

O valor mínimo é então identificado como o número de ciclos máximo permitido antes da falha do eixo. Este valor é registrado no banco de dados, juntamente com a identificação do eixo e a data de inclusão.

Este processo assegura que a análise de fadiga leve em consideração o ponto mais crítico do eixo, fornecendo uma estimativa conservadora e segura da vida útil do componente.

3.5 Atualização do Banco de Dados

Para manter o banco de dados de manutenção preditiva atualizado com o número de ciclos realizados por cada eixo, o sistema implementa uma função específica que realiza a atualização dos dados a partir de um arquivo de entrada. Este processo é essencial para garantir que as informações sobre o estado dos eixos sejam precisas e atualizadas, permitindo uma manutenção eficiente e a prevenção de falhas.

O sistema inicia lendo o banco de dados existente, que contém as informações sobre os eixos e o número de ciclos realizados até a última atualização. Esse banco de dados é armazenado em um arquivo CSV.

Em seguida, o sistema identifica o arquivo de entrada mais recente na pasta de entradas, que contém os dados dos ciclos realizados pelos eixos desde a última atualização. A identificação do arquivo mais recente é feita com base na data de criação dos arquivos na pasta de entradas.

O sistema realiza a junção do banco de dados existente com a tabela de entradas mais recente usando o nome do eixo como chave. A junção é feita de forma que todas as entradas do banco de dados original sejam mantidas, adicionando os novos dados dos ciclos realizados. Essa junção é essencial para garantir que todas as informações pertinentes aos eixos sejam consideradas durante a atualização.

Após a junção, o sistema atualiza o número de ciclos realizados por cada eixo. Este processo consiste em somar os ciclos realizados anteriormente com os novos ciclos informados na tabela de entradas. Caso algum eixo não tenha dados de ciclos realizados na nova entrada, o sistema assume que não houve alteração nos ciclos realizados para aquele eixo específico. Esse procedimento garante que os dados de ciclos realizados estejam sempre atualizados.

A data de alteração é então atualizada para refletir a data da última modificação, utilizando a data atual do sistema. Essa atualização é realizada para manter um registro de quando os dados foram modificados pela última vez, facilitando o acompanhamento histórico das atualizações e intervenções de manutenção.

Finalmente, a tabela atualizada é salva de volta no banco de dados, substituindo o arquivo anterior com os dados mais recentes sobre os ciclos realizados pelos eixos.

Este processo automatizado de atualização garante que o banco de dados de manutenção preditiva reflita com o estado atual dos eixos, permitindo uma gestão eficiente e a previsão precisa da vida útil dos componentes.

4 RESULTADOS

O desenvolvimento do sistema de manutenção preditiva para eixos mecânicos resultaram em diversas etapas e cálculos programados e executados conforme os arquivos fornecidos. As principais funções implementadas incluem o cadastro dos eixos, a atualização do número de ciclos realizados e a estruturação dos dados para análise preditiva. Os resultados foram validados e demonstram a eficácia do sistema desenvolvido. A seguir, são descritas as etapas principais e os resultados obtidos:

4.1 Cadastro dos Eixos

O processo de cadastro dos eixos no banco de dados inclui a coleta de informações geométricas e de material dos eixos, a definição das cargas aplicadas, a análise dos esforços solicitantes e a identificação dos pontos críticos. O usuário insere manualmente as informações geométricas e de material do eixo, incluindo o comprimento, os apoios, as cargas aplicadas e os coeficientes de correção.

São calculados os esforços cortantes, momentos fletores e esforços axiais ao longo do eixo. A análise é realizada tanto em um plano quanto em dois planos, dependendo da configuração do sistema. Os pontos críticos são identificados com base na geometria e nas cargas aplicadas, calculando-se os fatores de concentração de tensão e as tensões alternadas nominais e equivalentes. Utilizando o diagrama S-N, são calculados os ciclos de vida até a falha para cada ponto crítico, sendo considerado o menor valor entre os ciclos de vida dos pontos críticos como o ciclo de vida total do eixo.

O processo de entrada dos dados pelo usuário é exemplificado no apêndice C, enquanto que a estrutura do banco de dados preenchido com alguns eixos mecânicos já cadastrados é ilustrado na Figura 10.

Figura 10 – Exemplo Estrutura do Banco de Dados

EIXO	CICLOS_REALIZADOS	CICLOS_MÁXIMOS	DATA_ALTERAÇÃO	DATA_INCLUSÃO
eixo_2024-07-13_15-54	25000	309360.0	2024-07-18	2024-07-13
eixo_2024-07-10_18-32	15500	463059.0	2024-07-18	2024-07-18
eixo_2024-07-07_07-47	100000	237077.0	2024-07-18	2024-07-07
eixo_2024-06-02_14-12	65500	375249.0	2024-07-18	2024-06-02

Fonte: Autor (2024)

4.2 Atualização dos Ciclos Realizados

A atualização periódica do banco de dados com o número de ciclos realizados pelos eixos cadastrados envolve várias etapas. Inicialmente, o banco de dados existente é

lido, contendo as informações dos eixos cadastrados e seus ciclos realizados até o momento. Em seguida, são lidas as entradas mais recentes, contendo o número de ciclos adicionais realizados por cada eixo. O número de ciclos realizados é atualizado somando-se os ciclos adicionais às informações existentes no banco de dados. A data de alteração é atualizada para refletir a data da última atualização dos ciclos realizados, e a tabela atualizada é salva de volta no banco de dados, garantindo que as informações estejam sempre atualizadas para a análise preditiva.

4.3 Resultados Obtidos

Os resultados obtidos com a implementação do sistema são significativos. O cálculo preciso dos ciclos de vida até a falha, considerando os pontos críticos e os fatores de concentração de tensão, permite uma previsão da vida útil dos eixos. A atualização periódica dos ciclos realizados garante que o banco de dados esteja sempre atualizado, permitindo uma análise preditiva contínua e eficiente. A implementação de manutenção preditiva baseada em dados precisos e atualizados contribui para a redução de custos operacionais, evitando falhas catastróficas e paradas inesperadas. Além disso, o sistema melhora a eficiência e a segurança das operações industriais, garantindo a integridade estrutural dos eixos e máquinas, permitindo intervenções oportunas.

5 CONCLUSÃO

O desenvolvimento deste sistema de manutenção preditiva para eixos mecânicos demonstrou potencial em ser uma solução eficaz para a previsão de falhas por fadiga e para a otimização da manutenção industrial. A combinação de métodos de cálculo com a atualização contínua do banco de dados fornece uma base para futuras pesquisas e avanços na área de manutenção preditiva e engenharia mecânica, que hoje conta com métodos e soluções de elevado custo financeiro.

No entanto, existem oportunidades para melhorias e futuras pesquisas que podem aprimorar ainda mais o sistema desenvolvido. Uma das áreas de melhoria é a captação automática do número de ciclos realizados pelos eixos. Atualmente, a atualização dos ciclos é feita através da leitura de entradas recentes. A implementação de sensores de monitoramento contínuo que capturam os ciclos realizados em tempo real permitiria a elaboração automática dos relatórios de atualização, reduzindo a intervenção manual e aumentando a precisão dos dados.

Outra área de pesquisa é a automatização da obtenção dos coeficientes de correção e dos fatores de concentração de tensões (K_t e K_{ts}). Atualmente, esses valores são calculados e inseridos pelo usuário, o que pode levar a erros e inconsistências. O desenvolvimento de métodos automáticos para calcular esses coeficientes com base em parâmetros geométricos e de materiais do eixo pode tornar o sistema mais eficiente. Algoritmos avançados de análise de imagens e gráficos, além de aprendizado de máquina podem ser utilizados para identificar características geométricas do eixo e analisar diagramas da literatura de falhas por fadiga, então calcular os fatores de concentração de tensão e coeficientes de correção automaticamente.

Implementar essas melhorias não só aumentará a eficácia do sistema de manutenção preditiva, mas também abrirá novas possibilidades para a pesquisa e desenvolvimento na área de manutenção industrial, garantindo a continuidade das operações e a segurança dos equipamentos, diminuindo os custos financeiros atualmente despendidos para realização desse tipo de manutenção.

REFERÊNCIAS

- ABNT. **NBR 5462: 1994. Confiabilidade e Manutenibilidade**. Rio de Janeiro: ABNT, 1994.
- ASM. **ASM Handbook Volume 2: Properties and Selection: Nonferrous Alloys and Special-Purpose Materials**. [S.l.]: ASM International, 1990.
- BUDYNAS, R. G.; NISBETT, J. K. **Elementos de Máquinas de Shigley**. [S.l.]: AMGH Editora LTDA, 2016.
- DOWLING, N. E. **Mechanical Behavior of Materials**. 4th. ed. [S.l.: s.n.], 2013.
- LAFRAIA, J. R. B. **Manual de Confiabilidade, Manutenibilidade e Disponibilidade**. Rio de Janeiro: Qualitymark: Petrobras, 2001.
- MOUBRAY, J. **Manutenção Centrada em Confiabilidade**. Lutterworth: Aladon Ltd., 2000.
- NORTON, R. L. **Projeto de Máquinas: Uma abordagem integrada**. Porto Alegre: Bookman Companhia Ed, 2010. ISBN 9788522109151.
- PINTO, A. K.; XAVIER, J. N. **Manutenção: função estratégica**. Rio de Janeiro: Qualitymark, 2001.
- REED, R.; SMITH, J. C. The economic effects of fracture in the united states. **National Buereau of Standards**, 1983.
- SIMÕES, D. d. A. **Entalhes melhorados e otimizados**. Rio de Janeiro: [s.n.], 2012. Acesso em: 07 abr. 2024. Disponível em: <https://www.maxwell.vrac.puc-rio.br/21361/21361_3.PDF>.
- SIQUEIRA, I. P. **Manutenção centrada na confiabilidade: manual de implementação**. Rio de Janeiro: Qualitymark, 2009.
- SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração de Produção**. São Paulo: Atlas, 2007.

Apêndices

APÊNDICE A – Código de Cadastro de Eixo Mecânico no Sistema

```
1 import numpy as np
2 from datetime import datetime
3 import os
4 import pandas as pd
5 import math
6 import matplotlib.pyplot as plt
7
8 def log_message(message):
9     return print(current_datetime() + " | " + message)
10
11 def log_error(error):
12     return print(current_datetime() + " > ERROR!!! | " +
13                 error)
14
15 def current_datetime():
16     return datetime.now().strftime("%d/%m/%Y %H:%M:%S")
17
18 def coletar_comprimento():
19     # Coleta e estrutura o do comprimento da viga
20     while True:
21         try:
22             comprimento_input = input("\tDigite o comprimento
23                                     do eixo [mm]: ")
24             comprimento = estruturar_comprimento(
25                 comprimento_input)
26             if comprimento <= 0:
27                 log_error("O comprimento do eixo deve ser
28                             maior que zero.")
29                 raise OSError("O comprimento do eixo deve ser
30                                 maior que zero.")
31             return comprimento
32             break
33         except OSError as err:
34             log_error("Erro ao adquirir comprimento do eixo")
35
36 def estruturar_comprimento(input_comprimento):
```

```
32     try:
33         comprimento = float(input_comprimento)
34         return comprimento
35     except ValueError:
36         raise ValueError("Defina um comprimento válido para
37             a viga.")
38
39 def coletar_apoios():
40     # Coleta e estrutura o dos dados dos apoios
41     while True:
42         try:
43             numero_apoios = int(input("\tDigite o número de
44                 apoios no eixo: "))
45             if numero_apoios <= 0:
46                 log_error("Deve haver pelo menos um apoio.")
47                 raise ValueError("Deve haver pelo menos um
48                     apoio.")
49             lista_apoios = []
50             for i in range(numero_apoios):
51                 log_message(f"Informe os do Apoio {i+1}:")
52                 tipo = input("\tTipo do apoio (fixo, móvel ou
53                     engaste): ")
54                 posicao = input("\tPosição do apoio no eixo
55                     [mm]: ")
56                 lista_apoios.append({'tipo': tipo, 'posicao':
57                     posicao})
58             apoios = estruturar_apoios(lista_apoios)
59             return apoios
60             break
61         except ValueError as e:
62             log_error("Erro ao adquirir informações sobre
63                 os apoios")
64
65 def estruturar_apoios(lista_apoios):
66     if not lista_apoios:
67         raise ValueError("Adicione apoios!")
68     apoios_estruturados = []
69     for apoio in lista_apoios:
70         tipo = apoio['tipo']
```

```
64     posicao = float(apoio['posicao'])
65     apoios_estruturados.append({'tipo': tipo, 'posicao':
66         posicao})
67
68 def definir_esforços_solicitantes(escolha_plano, comprimento,
69     apoios):
69     # Análise para o plano XY
70     log_message("Coletando cargas para o plano 1")
71     cargas_externas_XY = coletar_cargas_externas()
72     reacoes_XY, cortante_XY, fletor_XY, axial_XY =
73         realizar_analise(comprimento, apoios,
74             cargas_externas_XY)
73     # Consolida o da análise
74     if escolha_plano == "2":
75         # Análise para o plano XZ
76         log_message("Coletando cargas para o plano 2")
77         cargas_externas_XZ = coletar_cargas_externas()
78         reacoes_XZ, cortante_XZ, fletor_XZ, axial_XZ =
79             realizar_analise(comprimento, apoios,
80                 cargas_externas_XZ)
79     # Combinar Planos
80     cortante, fletor, axial = combinar_planos(cortante_XY,
81         cortante_XZ, fletor_XY, fletor_XZ, axial_XY,
82         axial_XZ)
81     else:
82         cortante_ajustada = []
83         for secao_XY in cortante_XY.values():
84             x_vals = secao_XY['x']
85             v_vals_XY = secao_XY['v']
86             cortante_ajustada.append({'x': x_vals, 'v':
87                 v_vals_XY})
87         cortante = cortante_ajustada
88         axial_ajustada = []
89         for secao_XY in axial_XY.values():
90             axial_vals = secao_XY['x']
91             axial_vals_XY = secao_XY['axial']
92             axial_ajustada.append({'x': axial_vals, 'axial':
93                 axial_vals_XY})
```



```
        valor_final': valor_final,
120        'posicao_inicial':
            posicao_inicial,
            'posicao_final':
            posicao_final})
121    cargas_externas = estruturar_cargas_externas(
        lista_cargas)
122    return cargas_externas
123    break
124    except ValueError as e:
125        log_error("Erro ao adquirir informa es sobre
            as cargas aplicadas")
126
127 def estruturar_cargas_externas(lista_cargas):
128     if not lista_cargas:
129         raise ValueError("Adicione alguma carga!")
130
131     cargas_estruturadas = []
132     for carga in lista_cargas:
133         tipo = carga['tipo']
134
135         if tipo == "radial" or tipo == "momento" or tipo == "
            axial":
136             valor = float(carga['valor'])
137             posicao = float(carga['posicao'])
138             cargas_estruturadas.append({'tipo': tipo, 'valor'
                : valor, 'posicao': posicao})
139
140         elif tipo == "distribuida":
141             valor_inicial = float(carga['valor_inicial'])
142             valor_final = float(carga['valor_final'])
143             posicao_inicial = float(carga['posicao_inicial'])
144             posicao_final = float(carga['posicao_final'])
145
146             if valor_inicial * valor_final < 0: # Carga
                distribu da muda de sentido
147                 largura = posicao_final - posicao_inicial
148                 largura_esquerda = largura * abs(
                    valor_inicial) / (abs(valor_inicial) + abs
```

```
(valor_final))

149
150 # Tri ngulo esquerdo
151 forca_total_esquerda = largura_esquerda *
    valor_inicial / 2
152 posicao_media_esquerda = posicao_inicial +
    largura_esquerda / 3 if valor_inicial > 0
    else posicao_inicial + 2 *
    largura_esquerda / 3
153 cargas_estruturadas.append({
154     'tipo': 'distribuida',
155     'valor_inicial': valor_inicial,
156     'valor_final': 0,
157     'posicao_inicial': posicao_inicial,
158     'posicao_final': posicao_inicial +
        largura_esquerda,
159     'forca_total': forca_total_esquerda,
160     'posicao_media': posicao_media_esquerda
161 })
162
163 # Tri ngulo direito
164 forca_total_direita = largura_esquerda *
    valor_final / 2
165 posicao_media_direita = posicao_inicial +
    largura_esquerda + largura_esquerda / 3 if
    valor_final > 0 else posicao_inicial +
    largura_esquerda + 2 * largura_esquerda /
    3
166 cargas_estruturadas.append({
167     'tipo': 'distribuida',
168     'valor_inicial': 0,
169     'valor_final': valor_final,
170     'posicao_inicial': posicao_inicial +
        largura_esquerda,
171     'posicao_final': posicao_final,
172     'forca_total': forca_total_direita,
173     'posicao_media': posicao_media_direita
174 })
175 else:
```

```
176         # Carga distribu da normal (retangular,
177             trapezoidal ou triangular)
178         largura = posicao_final - posicao_inicial
179         forca_total = largura * (valor_inicial +
180             valor_final) / 2
181         posicao_media = posicao_inicial + largura * (
182             valor_inicial / 3 + 2 * valor_final / 3) /
183             (valor_inicial + valor_final)
184
185         cargas_estruturadas.append({
186             'tipo': tipo,
187             'valor_inicial': valor_inicial,
188             'valor_final': valor_final,
189             'posicao_inicial': posicao_inicial,
190             'posicao_final': posicao_final,
191             'largura': largura,
192             'forca_total': forca_total,
193             'posicao_media': posicao_media
194         })
195
196     return cargas_estruturadas
197
198 def realizar_analise(comprimento, apoios, cargas_externas):
199     caso = verificar_caso(apoios)
200     reacoes = calcular_reacoes(cargas_externas, apoios) if
201         caso == "determinado" else []
202     cargas = estruturar_cargas(cargas_externas, reacoes)
203     cortante = calcular_cortante(cargas, comprimento)
204     fletor = calcular_fletor(cargas, comprimento)
205     normal = calcular_axial(cargas, comprimento)
206     return reacoes, cortante, fletor, normal
207
208 def verificar_caso(apoios):
209     num_rolete = sum(apoio['tipo'] == 'movel' for apoio in
210         apoios)
211     num_pino = sum(apoio['tipo'] == 'fixo' for apoio in
212         apoios)
213     num_engaste = sum(apoio['tipo'] == 'engaste' for apoio in
214         apoios)
```

```
207
208     num_reacoes = 0.5 * num_rolete + 1.5 * num_pino + 2 *
           num_engaste
209
210     # Lógica para verificar se o eixo está estaticamente
           determinada ou indeterminada
211     if num_reacoes > 2:
212         log_error("O eixo está estaticamente indeterminado")
213         raise ValueError("O eixo está estaticamente
           indeterminado")
214     elif num_reacoes < 2:
215         log_error("O eixo não está vel ou está
           subapoiada.")
216         raise ValueError("O eixo não está vel ou está
           subapoiada.")
217     else:
218         return "determinado"
219
220 def calcular_reacoes(cargas_externas, apoios):
221     num_apoios = len(apoios)
222     if num_apoios >= 2:
223         # Matriz dos coeficientes
224         A = np.zeros((num_apoios, num_apoios))
225         A[0, :] = 1 # Equilíbrio de forças verticais
226         for i in range(1, num_apoios):
227             for j in range(num_apoios):
228                 A[i, j] = apoios[j]['posicao'] - apoios[0][
           'posicao']
229
230         # Matriz dos termos independentes
231         B = np.zeros(num_apoios)
232         for carga in cargas_externas:
233             if carga['tipo'] == 'radial':
234                 B[0] -= carga['valor']
235                 for i in range(1, num_apoios):
236                     B[i] -= carga['valor'] * (carga['posicao']
           - apoios[0]['posicao'])
237             elif carga['tipo'] == 'momento':
238                 for i in range(1, num_apoios):
```

```
239         B[i] -= carga['valor']
240     elif carga['tipo'] == 'distribuida':
241         forca_total = carga['forca_total']
242         posicao_media = carga['posicaoMedia']
243         B[0] -= forca_total
244         for i in range(1, num_apoios):
245             B[i] -= forca_total * (posicao_media -
246                                     apoios[0]['posicao'])
247
248     # Resolvendo o sistema
249     R = np.linalg.solve(A, B)
250     reacoes = [{'tipo': 'reacao-radial', 'valor': R[i], '
251                 posicao': apoios[i]['posicao']} for i in range(
252                     num_apoios)]
253
254 elif num_apoios == 1:
255     # Reações para um único apoio (engaste)
256     forca_no_engaste = sum(-carga['valor'] for carga in
257                             cargas_externas if carga['tipo'] == 'radial')
258     momento_no_engaste = sum(-carga['valor']*(carga['
259                             posicao'] - apoios[0]['posicao']) for carga in
260                             cargas_externas if carga['tipo'] == 'radial')
261     momento_no_engaste += sum(carga['valor'] for carga in
262                             cargas_externas if carga['tipo'] == 'momento')
263     momento_no_engaste += sum(-carga['forca_total']*(
264                             carga['posicaoMedia'] - apoios[0]['posicao']) for
265                             carga in cargas_externas if carga['tipo'] == '
266                             distribuida')
267
268     return [
269         {'tipo': 'reacao-radial', 'valor':
270             forca_no_engaste, 'posicao': apoios[0]['
271             posicao']},
272         {'tipo': 'reacao-momento', 'valor':
273             momento_no_engaste, 'posicao': apoios[0]['
274             posicao']}]
275 ]
276
277 # Agora, calcular as reações axiais
```



```
                posicao_inicial'])
292         v_vals[j] += largura * intensidade
293         elif x >= carga['posicao_final']:
294             v_vals[j] += carga['forca_total']
295
296         cortante[(x_inicio, x_final)] = {'x': x_vals, 'v':
                v_vals}
297
298     return cortante
299
300 def calcular_fletor(cargas, comprimento):
301     # Escolha dos pontos importantes para construir as
        se es
302     pontos_importantes = [0, comprimento]
303
304     for carga in cargas:
305         if carga['tipo'] in ["radial", "momento", "reacao-
                radial", "reacao-momento"]:
306             pontos_importantes.append(carga['posicao'])
307         if carga['tipo'] == "distribuida":
308             pontos_importantes.append(carga['posicaoInicial'
                ])
309             pontos_importantes.append(carga['posicaoFinal'])
310
311     pontos_importantes = sorted(set(pontos_importantes))
312
313     # Cria o inicio e fim de cada seção
314     fletor = []
315     for i in range(len(pontos_importantes) - 1):
316         x_vals = np.linspace(pontos_importantes[i],
                pontos_importantes[i + 1], 100000)
317         m_vals = np.zeros_like(x_vals)
318         fletor.append({'x': x_vals, 'm': m_vals})
319
320     # Cálculo do momento fletor em cada seção
321     for secao in fletor:
322         for j, x in enumerate(secao['x']):
323             m = 0
324             for carga in cargas:
```

```
325         if carga['tipo'] in ["radial", "reacao-radial
326             "] and carga['posicao'] <= x:
327                 m -= carga['valor'] * (x - carga['posicao
328                     '])
329         elif carga['tipo'] == "distribuida":
330             xi, xf = carga['posicaoInicial'], carga['
331                 posicaoFinal']
332             if xi <= x < xf:
333                 # Calcula o momento devido parte
334                 da carga distribu da at x
335                 w_atual = carga['valorInicial'] + (
336                     carga['valorFinal'] - carga['
337                         valorInicial']) * (x - xi) / (xf -
338                             xi)
339                 forca_total_atual = (x - xi) * (carga
340                     ['valorInicial'] + w_atual) / 2
341                 posicao_media_atual = xi + (x - xi) *
342                     (carga['valorInicial'] / 3 + 2 *
343                         w_atual / 3) / (carga['
344                             valorInicial'] + w_atual)
345                 m -= forca_total_atual * (x -
346                     posicao_media_atual)
347             elif x >= xf:
348                 m -= carga['forcaTotal'] * (x - carga
349                     ['posicaoMedia'])
350         elif carga['tipo'] in ["momento", "reacao-
351             momento"] and carga['posicao'] <= x:
352             m += carga['valor']
353     secao['m'][j] = m
354
355     return fletor
356
357 def calcular_axial(cargas, comprimento):
358     # Identificar pontos importantes (in cio , fim, e pontos
359     onde as cargas s o aplicadas)
360     pontos_importantes = sorted(set([0, comprimento] + [carga
361         ['posicao'] for carga in cargas]))
362
363     # Inicializar o dicion rio para armazenar os esfor os
```

```
axiais
348     esforco_axial = {}
349
350     for i in range(len(pontos_importantes) - 1):
351         x_inicio, x_final = pontos_importantes[i],
352             pontos_importantes[i + 1]
353         x_vals = np.linspace(x_inicio, x_final, 100000)
354         axial_vals = np.zeros_like(x_vals)
355
356         for j, x in enumerate(x_vals):
357             for carga in cargas:
358                 if carga['tipo'] in ["axial", "reacao-axial"]
359                     and carga['posicao'] <= x:
360                     axial_vals[j] += carga['valor']
361
362     esforco_axial[(x_inicio, x_final)] = {'x': x_vals, '
363         axial': axial_vals}
364
365     return esforco_axial
366
367 def combinar_planos(cortantes_XY, cortantes_XZ, fletores_XY,
368     fletores_XZ, axial_XY, axial_XZ):
369     cortantes_combinados = []
370     fletores_combinados = []
371     axial_combinados = []
372
373     # Assumindo que as seções são em ambos os planos e são as
374     mesmas
375     for (secao_XY, secao_XZ) in zip(cortantes_XY.values(),
376         cortantes_XZ.values()):
377         x_vals = secao_XY['x']
378         v_vals_XY = secao_XY['v']
379         v_vals_XZ = secao_XZ['v']
380         v_combinado = np.sqrt(np.square(v_vals_XY) + np.
381             square(v_vals_XZ))
382         cortantes_combinados.append({'x': x_vals, 'v':
383             v_combinado})
384
385     for (secao_XY, secao_XZ) in zip(axial_XY.values(),
```

```
axial_XZ.values()):
378     x_vals = secao_XY['x']
379     axial_vals_XY = secao_XY['axial']
380     axial_vals_XZ = secao_XZ['axial']
381     axial_combinado = np.sqrt(np.square(axial_vals_XY) +
        np.square(axial_vals_XZ))
382     axial_combinados.append({'x': x_vals, 'axial':
        axial_combinado})
383
384     for (secao_XY, secao_XZ) in zip(fletores_XY, fletores_XZ)
        :
385         x_vals = secao_XY['x']
386         m_vals_XY = secao_XY['m']
387         m_vals_XZ = secao_XZ['m']
388         m_combinado = np.sqrt(np.square(m_vals_XY) + np.
            square(m_vals_XZ))
389         fletores_combinados.append({'x': x_vals, 'm':
            m_combinado})
390
391     return cortantes_combinados, fletores_combinados,
        axial_combinados
392
393 class EixoEstruturado:
394     def __init__(self, x_vals, v_vals, m_vals, axial_vals):
395         self.posicao = x_vals
396         self.cortante = v_vals
397         self.fletor = m_vals
398         self.normal = axial_vals
399         self.modulo_elasticidade = ""
400         self.sut = ""
401         self.se = ""
402         self.se_linha = ""
403         self.sm = ""
404         self.is_concentration_point = False
405         self.torque = ""
406         self.diametro = ""
407         self.raio_canto = ""
408         self.load_factor = ""
409         self.size_factor = ""
```

```
410     self.surface_factor = ""
411     self.temp_factor = ""
412     self.reliability_factor = ""
413     self.other_factor = ""
414     self.kt_flexao = ""
415     self.kt_torcaao = ""
416     self.kt_axial = ""
417     self.raiz_a_flexao = ""
418     self.raiz_a_torcaao = ""
419     self.q = ""
420     self.qs = ""
421     self.kf_flexao = ""
422     self.kfs_torcaao = ""
423     self.momento_area = ""
424     self.momento_polar = ""
425     self.tensao_flexao = ""
426     self.cisalhamento_cortante = ""
427     self.cisalhamento_torcional = ""
428     self.tensao_equivalente = ""
429     self.tensao_alternada_nominal_flexao = ""
430     self.tensao_alternada_nominal_torcaao = ""
431     self.tensao_alternada_equivalente_flexao = ""
432     self.tensao_alternada_equivalente_torcaao = ""
433     self.tensao_alternada_equivalente = ""
434     self.numero_ciclos = ""
435
436     def concentration_point(self, torque, diametro,
437                            raio_canto, load_factor, size_factor, surface_factor,
438                            temp_factor, reliability_factor, other_factor,
439                            kt_flexao, kt_axial, kt_torcaao):
440         self.is_concentration_point = True
441         self.torque = torque
442         self.diametro = diametro
443         self.raio_canto = raio_canto
444         self.load_factor = load_factor
445         self.size_factor = size_factor
446         self.surface_factor = surface_factor
447         self.temp_factor = temp_factor
448         self.reliability_factor = reliability_factor
```

```
446         self.other_factor = other_factor
447         self.kt_flexao = kt_flexao
448         self.kt_axial = kt_axial
449         self.kt_torca = kt_torca
450
451     def material_data(self, modulo_elasticidade, sut,
452                     se_linha, sm):
453         self.modulo_elasticidade = modulo_elasticidade
454         self.sut = sut
455         self.se_linha = se_linha
456         self.sm = sm
457
458     def structure_eixo_consolidado(cortante, fletor, normal):
459         # Organizando Classe
460         pontos_eixo = []
461         for secao_cortante, secao_fletor, secao_normal in zip(
462             cortante, fletor, normal):
463             for i, j, k in zip(range(len(secao_cortante['x'])),
464                             range(len(secao_fletor['x'])), range(len(
465             secao_normal['x']))):
466                 if secao_cortante['x'][i] == secao_fletor['x'][j]
467                     and secao_cortante['x'][i] == secao_normal['x']
468                       [k]:
469                     pontos_eixo.append(EixoEstruturado(
470                         secao_cortante['x'][i], secao_cortante['v']
471                         [i], secao_fletor['m'][j], secao_normal['v']
472                         axial'[k]))
473
474         return np.array(pontos_eixo)
475
476     def plotar_cortante(cortante):
477         plt.figure(figsize=(10, 6))
478         for secao in cortante:
479             plt.plot(secao['x'], secao['v'])
480
481         plt.title("Gráfico de Esforço Cortante")
482         plt.xlabel("Comprimento da Viga (m)")
483         plt.ylabel("Esforço Cortante (N)")
484         plt.legend()
485         plt.grid(True)
```

```
476     plt.show()
477
478 def plotar_fletor(fletor):
479     plt.figure(figsize=(10, 6))
480     for secao in fletor:
481         plt.plot(secao['x'], secao['m']) # Ajuste feito aqui
482
483     plt.title("Gráfico de Momento Fletor")
484     plt.xlabel("Comprimento da Viga (m)")
485     plt.ylabel("Momento Fletor (Nm)")
486     plt.legend()
487     plt.grid(True)
488     plt.show()
489
490 def plotar_normal(normal):
491     plt.figure(figsize=(10, 6))
492     for secao in normal:
493         plt.plot(secao['x'], secao['axial']) # Ajuste feito
494         aqui
495
496     plt.title("Gráfico de Esforço Normal")
497     plt.xlabel("Comprimento da Viga (m)")
498     plt.ylabel("Esforço Normal (N)")
499     plt.legend()
500     plt.grid(True)
501     plt.show()
502
503 def coletar_propriedades_material():
504     material = input("\tMaterial que o Eixo      fabricado (
505         aço ou ferro fundido): ")
506     elastic_modulus = float(input("\tMódulo de elasticidade
507         (GPa): "))
508     ultimate_tensile_strength = float(input("\tResistência
509         tração máxima (Sut) [MPa]: "))
510     sm = 0.9 * ultimate_tensile_strength
511     if material == "aço":
512         if ultimate_tensile_strength < 1400:
513             fatigue_resistance_limit = 0.5 *
514                 ultimate_tensile_strength
```

```
510         else:
511             fatigue_resistance_limit = 700
512     elif material == "ferro":
513         if ultimate_tensile_strength < 400:
514             fatigue_resistance_limit = 0.4 *
515                 ultimate_tensile_strength
516         else:
517             fatigue_resistance_limit = 160
518     elif material == "aluminio":
519         if ultimate_tensile_strength < 330:
520             fatigue_resistance_limit = 0.4 *
521                 ultimate_tensile_strength
522         else:
523             fatigue_resistance_limit = 130
524     elif material == "cobre":
525         if ultimate_tensile_strength < 280:
526             fatigue_resistance_limit = 0.4 *
527                 ultimate_tensile_strength
528         else:
529             fatigue_resistance_limit = 100
530     return elastic_modulus, ultimate_tensile_strength, sm,
531         fatigue_resistance_limit
532
533 def structure_material(elastic_modulus,
534     ultimate_tensile_strength, fatigue_resistance_limit, sm,
535     eixo_consolidado):
536     for ponto in eixo_consolidado:
537         ponto.material_data(elastic_modulus,
538             ultimate_tensile_strength,
539             fatigue_resistance_limit, sm)
540     return eixo_consolidado
541
542 def adquirir_pontos_criticos():
543     # solicita o de pontos de concentra o de tens es
544     posicoes_concentracao = [float(pos) for pos in input("\
545         tInforme as posi es dos pontos de concentra o de
546         tens es (Pontos Cr ticos) separados por v rgula: "
547         ).split(",")]
548     pontos_concentradores = []
```

```
538     for ponto in posicoes_concentracao:
539         posicao_concentrador = ponto
540         torque = float(input(f"\tInforme o torque [N.mm]
                    atuante na posi o {ponto}: "))
541         diametro = float(input(f"\tInforme o di metro [mm]
                    na posi o {ponto} (0 menor di metro em caso de
                    diminui o ou aumento da se o transversal): "
                    ))
542         raio_canto = float(input(f"\tInforme o raio do
                    entalhe [mm] presente na posi o {ponto}: "))
543         load_factor = float(input(f"\tCoeficiente de Carga na
                    posi o {ponto}: "))
544         size_factor = float(input(f"\tCoeficiente de Tamanho
                    na posi o {ponto}: "))
545         surface_factor = float(input(f"\tCoeficiente de
                    Superf cie na posi o {ponto}: "))
546         temp_factor = float(input(f"\tCoeficiente de
                    Temperatura na posi o {ponto}: "))
547         reliability_factor = float(input(f"\tCoeficiente de
                    Confiabilidade na posi o {ponto}: "))
548         other_factor = float(input(f"\tCoeficiente para
                    outros efeitos na posi o {ponto} (Igual 1
                    para desconsiderar): "))
549         kt_flexao = float(input(f"\tInforme o fator de
                    concentra o de tens es (Kt) sob flex o na
                    posi o {ponto}. zero (0) se n o aplic vel: "
                    )
                    )
550         kt_axial = float(input(f"\tInforme o fator de
                    concentra o de tens es (Kt) sob tra o ou
                    compress o na posi o {ponto}. zero (0) se n o
                    aplic vel: "))
551         kt_torca = float(input(f"\tInforme o fator de
                    concentra o de tens es (Kts) sob tor o na
                    posi o {ponto}. zero (0) se n o aplic vel: "
                    )
                    )
552
553         pontos_concentradores.append({'posicao':
                    posicao_concentrador, 'torque': torque, 'diametro'
                    : diametro, 'raio_canto': raio_canto, 'load_factor
```

```
        ': load_factor, 'size_factor': size_factor, '
        surface_factor': surface_factor, 'temp_factor':
        temp_factor, 'reliability_factor':
        reliability_factor, 'other_factor': other_factor,
        'kt_flexao': kt_flexao, 'kt_axial': kt_axial, '
        kt_torcaos': kt_torcaos})
554     return pontos_concentradores
555
556 def structure_concentradores(eixo_consolidado,
        pontos_concentradores):
557     tolerancia = 0.001
558     for ponto_concentrador in pontos_concentradores:
559         posicao_concentrador = ponto_concentrador['posicao']
560         torque = ponto_concentrador['torque']
561         diametro = ponto_concentrador['diametro']
562         raio_canto = ponto_concentrador['raio_canto']
563         load_factor = ponto_concentrador['load_factor']
564         size_factor = ponto_concentrador['size_factor']
565         surface_factor = ponto_concentrador['surface_factor']
566         temp_factor = ponto_concentrador['temp_factor']
567         reliability_factor = ponto_concentrador['
            reliability_factor']
568         other_factor = ponto_concentrador['other_factor']
569         kt_flexao = ponto_concentrador['kt_flexao']
570         kt_axial = ponto_concentrador['kt_axial']
571         kt_torcaos = ponto_concentrador['kt_torcaos']
572         ponto_associado = False # Flag para verificar se o
            ponto j foi associado
573     for ponto in eixo_consolidado:
574         if ponto_associado:
575             break # Se o ponto j foi associado, sair
                do loop interno
576         posicao_eixo = ponto.posicao
577         if abs(posicao_eixo - posicao_concentrador) <=
            tolerancia:
578             ponto.concentration_point(torque, diametro,
                raio_canto, load_factor, size_factor,
                surface_factor, temp_factor,
                reliability_factor, other_factor,
```

```
        kt_flexao, kt_axial, kt_torcao)
579     ponto_associado = True # Marcando o ponto
        como associado
580     return eixo_consolidado
581
582 def calculo_limite_teorico_resistencia_fadiga(
    eixo_consolidado):
583     for ponto in eixo_consolidado:
584         if ponto.is_concentration_point:
585             ponto.se = ponto.load_factor * ponto.size_factor
                * ponto.surface_factor * ponto.temp_factor *
                ponto.reliability_factor * ponto.other_factor
                * ponto.se_linha
586     return eixo_consolidado
587
588 def calculo_coeficiente_concentracao_tensoes_fadiga(
    eixo_consolidado):
589     for ponto in eixo_consolidado:
590         if ponto.is_concentration_point:
591             ponto.raiz_raio = math.sqrt(ponto.raio_canto /
                25.4)
592             ponto.raiz_a_flexao = 0.246 - (3.08 * 10**-3 * (
                ponto.sut * 0.145)) + (1.51 * 10**-5 * (ponto.
                sut * 0.145)**2) - (2.67 * 10**-8 * (ponto.sut
                * 0.145)**3)
593             ponto.raiz_a_torcao = 0.19 - (2.51 * 10**-3 * (
                ponto.sut * 0.145)) + (1.35 * 10**-5 * (ponto.
                sut * 0.145)**2) - (2.67 * 10**-8 * (ponto.sut
                * 0.145)**3)
594             ponto.q = 1/(1 + (ponto.raiz_a_flexao/ponto.
                raiz_raio))
595             ponto.qs = 1/(1 + (ponto.raiz_a_torcao/ponto.
                raiz_raio))
596             ponto.kf_flexao = 1 + (ponto.q * (ponto.kt_flexao
                - 1))
597             ponto.kfs_torcao = 1 + (ponto.qs * (ponto.
                kt_torcao - 1))
598             ponto.kf_axial = 1 + (ponto.q * (ponto.kt_axial -
                1))
```

```
599     return eixo_consolidado
600
601 def calculo_tensoes_ciclos(eixo_consolidado):
602     for ponto in eixo_consolidado:
603         if ponto.is_concentration_point:
604             ponto.momento_area = math.pi * ponto.diametro**4
605                 / 64
606             ponto.momento_polar = math.pi * ponto.diametro**4
607                 / 32
608             ponto.tensao_alternada_nominal_flexao = (ponto.
609                 fletor * (ponto.diametro/2)) / ponto.
610                 momento_area
611             ponto.tensao_alternada_nominal_torcao = (ponto.
612                 torque * (ponto.diametro / 2)) / ponto.
613                 momento_polar
614             ponto.tensao_alternada_nominal_normal = (ponto.
615                 normal) / ((math.pi * ponto.diametro) / 4)
616             # ponto.tensao_alternada_nominal_cisalhante =
617                 ponto.cortante / ((math.pi * ponto.diametro) /
618                 4)
619             ponto.tensao_alternada_equivalente_flexao = ponto
620                 .kf_flexao * ((ponto.fletor * (ponto.diametro
621                 /2)) / ponto.momento_area)
622             ponto.tensao_alternada_equivalente_torcao = ponto
623                 .kfs_torcao * ponto.
624                 tensao_alternada_nominal_torcao
625             ponto.tensao_alternada_equivalente_normal = ponto
626                 .kf_axial * ponto.
627                 tensao_alternada_nominal_normal
628             # ponto.tensao_alternada_equivalente_cisalhante =
629                 ponto.cortante / ((math.pi * ponto.diametro)
630                 / 4)
631             ponto.tensao_alternada_equivalente = math.sqrt(
632                 ponto.tensao_alternada_equivalente_flexao**2 +
633                 ponto.tensao_alternada_equivalente_normal**2
634                 + 3 * ponto.
635                 tensao_alternada_equivalente_torcao**2)
636             if ponto.tensao_alternada_equivalente <= ponto.se
637                 :
```

```
616         ponto.numero_ciclos = float('inf')
617     elif ponto.tensao_alternada_equivalente > ponto.
        se and ponto.tensao_alternada_equivalente <=
        ponto.sm:
618         ponto.numero_ciclos = math.floor(10**(6 - (3
            * (math.log(ponto.
                tensao_alternada_equivalente / ponto.se) /
                math.log(ponto.sm / ponto.se))))))
619     return eixo_consolidado
620
621 def obter_numero_ciclos(eixo_consolidado):
622     valor_minimo = float('inf')
623     nome_eixo = 'eixo_' + datetime_to_file_name()
624     data_inclusao = datetime_now()
625     for ponto in eixo_consolidado:
626         if ponto.is_concentration_point:
627             if ponto.numero_ciclos < valor_minimo:
628                 valor_minimo = ponto.numero_ciclos
629     return valor_minimo, nome_eixo, data_inclusao
630
631 def datetime_to_file_name():
632     return datetime.now().strftime("%Y-%m-%d_%H-%M")
633
634 def datetime_now():
635     return datetime.now().strftime("%Y-%m-%d")
636
637 def adicionar_banco_dados(numero_ciclos_maximos, nome_eixo,
        data_inclusao):
638     root_path = os.getcwd()[:-11]
639     data_base_path = root_path + r"\02_banco_de_dados"
640     data_base = pd.read_csv(data_base_path + '/banco_de_dados
        .csv', delimiter=';')
641     nova_linha = pd.DataFrame({
642         'EIXO': [nome_eixo],
643         'CICLOS_REALIZADOS': [0],
644         'CICLOS_MAXIMOS': [numero_ciclos_maximos],
645         'DATA_ALTERACAO': [data_inclusao],
646         'DATA_INCLUSAO': [data_inclusao]
647     })
```

```
648     # Concatenar o novo DataFrame com o original
649     df_concatenado = pd.concat([data_base, nova_linha],
        ignore_index=True)
650     # Escrever o DataFrame atualizado de volta para o arquivo
        CSV
651     df_concatenado.to_csv(data_base_path + '/banco_de_dados.
        csv', index=False, sep=';')
652     return log_message("Eixo adicionado     base de dados")
653
654 def main():
655     print("\n\n=====  

        =====\n\n")
656     log_message("Início")
657     log_message("Adquirindo comprimento do Eixo")
658     comprimento = coletar_comprimento()
659     log_message("Adquirindo informações sobre os apoios")
660     apoios = coletar_apoios()
661     # Escolha entre análise em um plano (1) ou dois planos
        (2)
662     log_message("Adquirindo informações sobre planos em análise")
663     escolha_plano = input("\tEscolha a análise em um plano
        (1) ou dois planos (2): ")
664     log_message("Adquirindo informações sobre cargas
        aplicadas")
665     cortante, fletor, normal = definir_esforços_solicitantes(
        escolha_plano, comprimento, apoios)
666     # Estruturar dados do eixo
667     eixo_consolidado = structure_eixo_consolidado(cortante,
        fletor, normal)
668     # Plotagem dos Diagramas de Esforços Solicitantes
669     plotar_cortante(cortante)
670     plotar_fletor(fletor)
671     plotar_normal(normal)
672
673     print("\n\n=====  

        =====\n\n")
674     log_message("Insira as propriedades do material do eixo:")
        )
```

```
675     # Coletar propriedades do material do eixo
676     elastic_modulus, ultimate_tensile_strength, sm,
        fatigue_resistance_limit =
        coletar_propriedades_material()
677     eixo_consolidado = structure_material(elastic_modulus,
        ultimate_tensile_strength, fatigue_resistance_limit,
        sm, eixo_consolidado)
678
679     print("\n\n==== C LCULO DE TENS ES =====\n\n")
680     log_message("Adquirindo informa es de Pontos Cr ticos
        ")
681     pontos_concentradores = adquirir_pontos_criticos()
682     eixo_consolidado = structure_concentradores(
        eixo_consolidado, pontos_concentradores)
683     eixo_consolidado =
        calculo_limite_teorico_resistencia_fadiga(
        eixo_consolidado)
684     eixo_consolidado =
        calculo_coeficiente_concentracao_tensoes_fadiga(
        eixo_consolidado)
685     eixo_consolidado = calculo_tensoes_ciclos(
        eixo_consolidado)
686
687     numero_ciclos_maximos, nome_eixo, data_inclusao =
        obter_numero_ciclos(eixo_consolidado)
688     adicionar_banco_dados(numero_ciclos_maximos, nome_eixo,
        data_inclusao)
689     log_message("Fim")
690     print("\n\nVoc  pode fechar este terminal agora")
691
692 # Chama a fun  o main se este script for o ponto de entrada
693 if __name__ == "__main__":
694     main()
```

Listing A.1 – Código de Cadastro de Eixo Mecânico no Sistema

APÊNDICE B – Atualização de Ciclos Realizados de Eixos Cadastrados

```

1 import os
2 import pandas as pd
3
4 def atualizar_ciclos_banco_dados():
5     print("\n\n===== ATUALIZANDO O N MERO DE CICLOS
6         REALIZADOS PELOS EIXOS CADASTRADOS =====\n\n")
7     root_path = os.getcwd()[:-11]
8     data_base_path = root_path + r"\02_banco_de_dados"
9
10    # L o banco de dados existente com delimitador ;
11    data_base = pd.read_csv(data_base_path + '/banco_de_dados
12        .csv', delimiter=';')
13
14    # Localiza o arquivo mais recente na pasta de entradas
15    entradas_path = root_path + r"\03_entradas"
16    arquivos_entrada = [f for f in os.listdir(entradas_path)
17        if os.path.isfile(os.path.join(entradas_path, f))]
18    arquivo_entrada_recente = max(arquivos_entrada, key=
19        lambda f: os.path.getctime(os.path.join(entradas_path,
20            f)))
21
22    # L a base de entradas com delimitador ;
23    entradas = pd.read_csv(os.path.join(entradas_path,
24        arquivo_entrada_recente), delimiter=';')
25
26    # Faz a jun o das tabelas pelo nome do eixo
27    tabela_atualizada = data_base.merge(entradas, on='EIXO',
28        how='left', suffixes=('', '_ENTRADA'))
29
30    # Atualiza os ciclos realizados
31    tabela_atualizada['CICLOS_REALIZADOS'] =
32        tabela_atualizada['CICLOS_REALIZADOS'].fillna(0) +
33        tabela_atualizada['CICLOS_REALIZADOS_ENTRADA'].fillna
34        (0)
35
36    # Remove a coluna de ciclos realizados da entrada

```

```
27     tabela_atualizada = tabela_atualizada.drop(columns=['
        CICLOS_REALIZADOS_ENTRADA'])
28
29     # Atualiza a data de alteração
30     tabela_atualizada['DATA_ALTERACAO'] = pd.to_datetime('
        today').strftime('%Y-%m-%d')
31
32     # Salva a tabela atualizada de volta no banco de dados
        com delimitador ;
33     tabela_atualizada.to_csv(data_base_path + '/
        banco_de_dados.csv', index=False, sep=';')
34
35     log_message("Número de Ciclos Realizados Atualizado com
        Sucesso")
36     print("\n\nVoc  pode fechar este terminal agora")
37     return
38
39 # Chama a fun o para atualizar o banco de dados
40 atualizar_ciclos_banco_dados()
```

Listing B.1 – C digo de Cadastro de Eixo Mec nico no Sistema

APÊNDICE C – Entrada de Dados pelo Usuário

===== CÁLCULO DE ESFORÇOS SOLICITANTES =====

13/07/2024 19:54:28 | Início

13/07/2024 19:54:28 | Adquirindo comprimento do Eixo

Digite o comprimento do eixo [mm]: 274

13/07/2024 19:54:34 | Adquirindo informações sobre os apoios

Digite o número de apoios no eixo: 2

13/07/2024 19:54:42 | Informações do Apoio 1:

Tipo do apoio (fixo, movel ou engaste): movel

Posição do apoio no eixo [mm]: 23

13/07/2024 19:54:45 | Informações do Apoio 2:

Tipo do apoio (fixo, movel ou engaste): fixo

Posição do apoio no eixo [mm]: 251

13/07/2024 19:54:48 | Adquirindo informações planos em análise

Escolha a análise em um plano (1) ou dois planos (2): 2

13/07/2024 19:54:50 | Adquirindo informações sobre cargas aplicadas

13/07/2024 19:54:50 | Coletando cargas para o plano 1

Digite o número de cargas externas aplicadas no eixo: 6

13/07/2024 19:54:52 | Informações para Carga 1:

Tipo da carga (axial, radial, momento, distribuida): axial

Valor da carga [N ou N.mm]: 13361.8302

Posição da carga no eixo [mm]: 93

13/07/2024 19:55:00 | Informações para Carga 2:

Tipo da carga (axial, radial, momento, distribuida): radial

Valor da carga [N ou N.mm]: -20102.8042

Posição da carga no eixo [mm]: 93

13/07/2024 19:55:09 | Informações para Carga 3:

Tipo da carga (axial, radial, momento, distribuida): momento

Valor da carga [N ou N.mm]: -470536.853

Posição da carga no eixo [mm]: 93

13/07/2024 19:55:20 | Informações para Carga 4:

Tipo da carga (axial, radial, momento, distribuida): axial

Valor da carga [N ou N.mm]: 5372.299

Posição da carga no eixo [mm]: 183

13/07/2024 19:55:30 | Informações para Carga 5:

Tipo da carga (axial, radial, momento, distribuida): radial

Valor da carga [N ou N.mm]: 7093.9477

Posição da carga no eixo [mm]: 183

13/07/2024 19:55:44 | Informações para Carga 6:

Tipo da carga (axial, radial, momento, distribuida): momento

Valor da carga [N ou N.mm]: 541151.695

Posição da carga no eixo [mm]: 183

13/07/2024 19:55:58 | Coletando cargas para o plano 2

Digite o número de cargas externas aplicadas no eixo: 2

13/07/2024 19:55:59 | Informações para Carga 1:

Tipo da carga (axial, radial, momento, distribuida): radial

Valor da carga [N ou N.mm]: 53591.3741

Posição da carga no eixo [mm]: 93

13/07/2024 19:56:08 | Informações para Carga 2:

Tipo da carga (axial, radial, momento, distribuida): radial

Valor da carga [N ou N.mm]: 18735.4337

Posição da carga no eixo [mm]: 183

===== COLETAR DADOS DO MATERIAL =====

13/07/2024 19:56:28 | Insira as propriedades do material do eixo:

Material que o Eixo é fabricado (aço ou ferro fundido): aço

Módulo de elasticidade (GPa): 210

Resistência à tração máxima (Sut) [MPa]: 1248

===== CÁLCULO DE TENSÕES =====

13/07/2024 19:59:06 | Adquirindo informações de Pontos Críticos

Informe as posições dos pontos de concentração de tensões (Pontos Críticos) separad

Informe o torque [N.mm] atuante na posição 53.0: 0

Informe o diâmetro [mm] na posição 53.0 (0 menor diâmetro em caso de diminuição ou

Informe o raio do entalhe [mm] presente na posição 53.0: 0.1

Coefficiente de Carga na posição 53.0: 1

Coefficiente de Tamanho na posição 53.0: 0.8019

Coefficiente de Superfície na posição 53.0: 0.8619

Coefficiente de Temperatura na posição 53.0: 1

Coefficiente de Confiabilidade na posição 53.0: 0.814

Coefficiente para outros efeitos na posição 53.0 (Igual à 1 para desconsiderar): 1

Informe o fator de concentração de tensões (Kt) sob flexão na posição 53.0. zero (0

Informe o fator de concentração de tensões (Kt) sob tração ou compressão na posição

Informe o fator de concentração de tensões (Kts) sob torção na posição 53.0. zero (

Informe o torque [N.mm] atuante na posição 93.0: 1887220.24

Informe o diâmetro [mm] na posição 93.0 (0 menor diâmetro em caso de diminuição ou

Informe o raio do entalhe [mm] presente na posição 93.0: 0.02

Coefficiente de Carga na posição 93.0: 1

Coefficiente de Tamanho na posição 93.0: 0.8019

Coeficiente de Superfície na posição 93.0: 0.8619

Coeficiente de Temperatura na posição 93.0: 1

Coeficiente de Confiabilidade na posição 93.0: 0.814

Coeficiente para outros efeitos na posição 93.0 (Igual à 1 para desconsiderar): 1

Informe o fator de concentração de tensões (Kt) sob flexão na posição 93.0. zero (0)

Informe o fator de concentração de tensões (Kt) sob tração ou compressão na posição

Informe o fator de concentração de tensões (Kts) sob torção na posição 93.0. zero (0)

13/07/2024 20:00:13 | Eixo adicionado à base de dados

13/07/2024 20:00:15 | Fim

Você pode fechar este terminal agora