

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

KEVIN VINÍCIUS GONÇALVES RODRIGUES

**APLICAÇÃO WEB EM NUVEM DESENVOLVIDA COM DJANGO PARA
AUTOMATIZAÇÃO DE ANÁLISE DE FADIGA EM EIXOS DE TRANSMISSÃO**

GUARAPUAVA

2022

KEVIN VINÍCIUS GONÇALVES RODRIGUES

**APLICAÇÃO WEB EM NUVEM DESENVOLVIDA COM DJANGO PARA
AUTOMATIZAÇÃO DE ANÁLISE DE FADIGA EM EIXOS DE TRANSMISSÃO**

*Cloud web application developed with Django to automate fatigue analysis on
transmission shafts*

Trabalho de conclusão de curso de graduação
apresentada como requisito para obtenção do título de
Bacharel em Engenharia Mecânica da Universidade
Tecnológica Federal do Paraná (UTFPR).
Orientador: Prof. Dr. David Lira Nunez.

GUARAPUAVA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

KEVIN VINÍCIUS GONÇALVES RODRIGUES

**APLICAÇÃO WEB EM NUVEM DESENVOLVIDA COM DJANGO PARA
AUTOMATIZAÇÃO DE ANÁLISE DE FADIGA EM EIXOS DE TRANSMISSÃO**

Trabalho de conclusão de curso de graduação
apresentada como requisito para obtenção do título de
Bacharel em Engenharia Mecânica da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 09/03/2022

Professor Dr. David Lira Nunez
Membro 1
Universidade Tecnológica Federal do Paraná

Professora Sílvia do Nascimento Rosa
Membro 2
Universidade Tecnológica Federal do Paraná

Professora Sediane Carmem Lunardi Hernandes
Membro 3
Universidade Tecnológica Federal do Paraná

GUARAPUAVA

2022

AGRADECIMENTOS

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Agradeço aos meus pais, Edmilza Rodrigues Gonçalves e Juvenal Rodrigues da Silva Filho, por todo o esforço investido na minha educação, apoio incondicional e incentivo durante toda a minha vida.

Ao meu orientador Prof. Dr. David Lira Nunez, pela sabedoria, dedicação e paciência com que me guiou nesta trajetória.

Aos meus colegas de curso, que estiveram comigo ao longo desta jornada e compartilharam experiências, descobertas e momentos inesquecíveis.

Aos meus amigos, que sempre estiveram ao meu lado, pela compreensão da minha ausência e apoio durante o desenvolvimento deste trabalho.

À Universidade Tecnológica Federal do Paraná e todo seu corpo docente, pelo aprendizado, ensinamentos e conselhos que possibilitaram o aperfeiçoamento de minha formação acadêmica e profissional.

Enfim, a todos aqueles que contribuíram, direta ou indiretamente, para a realização deste trabalho.

RESUMO

Eixo é um dos elementos de máquina mais amplamente usados em várias aplicações de engenharia mecânica. Algumas de suas principais atribuições são a transmissão de potência e sustentação de outros elementos mecânicos. Ao projetar um eixo, é necessário garantir que não ocorrerá falha por fadiga nesse elemento, utilizando cálculos e critérios que realizam sua verificação de acordo com a aplicação desejada. Contudo, existe uma demanda crescente relacionada à automatização de processos nos últimos anos. Essa necessidade está ligada à evolução constante da computação em nuvem e dos sistemas especialistas, relacionados à Indústria 4.0. Assim, surge uma oportunidade de pesquisa para incorporar essas tecnologias no dia-a-dia de um engenheiro mecânico, de forma a facilitar seu trabalho e auxiliar em suas tomadas de decisões. Dessa forma, a proposta deste trabalho é desenvolver uma aplicação web em nuvem para análise à fadiga de eixos mecânicos de transmissão. Para isso, será desenvolvida uma aplicação web amigável ao usuário utilizando o *framework web* Django, a fim de possibilitar a inserção dos dados de entrada no sistema através de um formulário. Em seguida, serão realizados dois estudos de caso, no qual eixos retirados da literatura acadêmica serão analisados para demonstrar a funcionalidade do *website*. Ao final, será possível observar que a aplicação web desenvolvida simplifica a alteração dos parâmetros de projeto, tais como o material do eixo, suas dimensões, acabamento superficial (relacionado ao seu processo de fabricação), entre outros. Assim, o sistema facilita a tomada de decisões e diminui a ocorrência de erros humanos que podem afetar o desempenho do projetista. Além disso, pode permitir a realização da análise de fadiga de maneira ubíqua por meio de uma simples aplicação web, apenas acessando um endereço eletrônico e inserindo os dados de entrada.

Palavras-chave: Eixo de Transmissão; Fratura por Fadiga; Aplicação Web; Django; Computação em nuvem.

ABSTRACT

Shaft is one of the most widely used machine elements in various mechanical engineering applications. Some of its main attributions are power transmission and support of other mechanical elements. When designing a shaft, it is necessary to ensure that there will be no fatigue failure in this element, using calculations and criteria that perform its verification according to the desired application. However, there is a growing demand related to process automation in recent years. This need is linked to the constant evolution of cloud computing and expert systems related to Industry 4.0. Thus, a research opportunity arises to incorporate these technologies into the daily life of a mechanical engineer, in order to facilitate their work and assist in their decision-making. Thus, the purpose of this work is to develop a cloud web application for fatigue analysis of mechanical transmission shafts. For this, a user-friendly web application will be developed using the Django web framework, in order to allow the insertion of input data into the system through a form. Then, two case studies will be carried out, in which shafts taken from academic literature will be analyzed to demonstrate the functionality of the website. At the end, it will be possible to observe that the web application developed simplifies the modification of design parameters, such as the shaft material, its dimensions, surface finish (related to its manufacturing process), among others. Thus, the system facilitates decision-making and reduces the occurrence of human errors that can affect the designer's performance. Furthermore, it can allow the performance of fatigue analysis ubiquitously through a simple web application, just by accessing an website and entering the input data.

Keywords: Transmission Shaft; Fatigue Fracture; Web Application; Django; Cloud Computing.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 - Eixo redondo com filetagem do ressalto em flexão..... | 19 |
| Figura 2 - Eixo redondo com filetagem do ressalto em torção | 19 |
| Figura 3 - Eixo redondo sulcado em flexão | 20 |
| Figura 4 - Eixo redondo sulcado em torção..... | 20 |
| Figura 5 - Eixo redondo com rasgo de chaveta de fundo plano em flexão..... | 21 |
| Figura 6 - Eixo redondo com rasgo de chaveta de fundo plano em torção | 21 |
| Figura 7 - Valores de q | 23 |
| Figura 8 - Valores de qs | 23 |
| Figura 9 - Tensão flutuante e suas components | 24 |
| Figura 10 - Diagrama de fadiga com diferentes critérios de falha | 27 |
| Figura 11 - Crescimento das principais linguagens de programação..... | 31 |
| Figura 12 - Comparação entre Python, Java e JavaScript..... | 32 |
| Figura 13 - Comparação entre Python, Java, JavaScript, Julia e R | 32 |
| Figura 14 - Funcionamento do Django | 38 |
| Figura 15 - Lógica para escolha do fator de segurança e diâmetro | 40 |
| Figura 16 - Funcionamento geral do programa | 41 |
| Figura 17 - Criação do projeto em Django | 42 |
| Figura 18 - Arquivos iniciais do projeto no VS Code | 43 |
| Figura 19 - urls.py da pasta “tcckevin” | 43 |
| Figura 20 - urls.py da pasta “main” | 44 |
| Figura 21 - views.py da pasta “main”..... | 44 |
| Figura 22 - Tratamento de exceção | 45 |
| Figura 23 - Arquivos do VS Code na elaboração da interface..... | 45 |
| Figura 24 - Início padrão do formulário | 46 |
| Figura 25 - Final padrão do formulário | 47 |
| Figura 26 - Página de resultados | 47 |
| Figura 27 - Página de erro | 47 |
| Figura 28 - Formulário da variável “ressalto” | 48 |
| Figura 29 - Atribuição dos dados do formulário a variáveis | 48 |
| Figura 30 - Chamada das funções e cálculo do resultado..... | 49 |
| Figura 31 - Primeiro estudo de caso..... | 50 |
| Figura 32 - Eixo modelado no CAD..... | 51 |
| Figura 33 - Desenhista de eixo do Solid Edge | 51 |
| Figura 34 - Diagrama de momento fletor | 52 |
| Figura 35 - Sugestões do programa para aumentar o fator de segurança..... | 53 |
| Figura 36 – Eixo modelado no CAD com novos valores de diâmetro | 54 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 - Parâmetros para cálculo do fator de superfície | 16 |
| Tabela 2 - Fator de carga | 16 |
| Tabela 3 - Efeito da temperatura na resistência à tração do aço | 17 |
| Tabela 4 - Fator de confiabilidade..... | 17 |
| Tabela 5 - Estimativas iniciais de K_t e K_{ts} | 22 |
| Tabela 6 - Valores de M_a nos pontos críticos | 52 |
| Tabela 7 - Fatores de segurança obtidos..... | 52 |
| Tabela 8 - Fator de segurança no ponto crítico para diferentes acabamentos . | 53 |
| Tabela 9 - Fator de segurança no ponto crítico para diferentes materiais e eixo laminado a frio..... | 53 |
| Tabela 10 - Fator de segurança no ponto crítico para diferentes materiais e eixo retificado | 54 |
| Tabela 11 - Fatores de segurança nos pontos críticos para novos valores de diâmetro | 55 |
| Tabela 12 - Diâmetros calculados para todos os critérios | 55 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-------|---|
| AI | Inteligência Artificial (do inglês – <i>Artificial Intelligence</i>) |
| ASME | Sociedade Americana de Engenheiros Mecânicos (do inglês – <i>American Society of Mechanical Engineers</i>) |
| BSD | Distribuição de Software Berkeley (do inglês – <i>Berkeley Software Distribution</i>) |
| CAD | Projeto Assistido por Computador (do inglês - <i>Computer-Aided Design</i>) |
| CAE | Engenharia Assistida por Computador (do inglês - <i>Computer-Aided Engineering</i>) |
| CAM | Fabricação Assistida por Computador (do inglês - <i>Computer-Aided Manufacturing</i>) |
| CSS | Folhas de Estilo em Cascatas (do inglês – <i>Cascading Style Sheets</i>) |
| DRY | Não Se Repita (do inglês – <i>Don't Repeat Yourself</i>) |
| ES | Sistemas Especialistas (do inglês – <i>Expert Systems</i>) |
| HTML | Linguagem de Marcação de HiperTexto (do inglês – <i>HyperText Markup Language</i>) |
| HTTP | Protocolo de Transferência de Hipertexto (do inglês – <i>Hypertext Transfer Protocol</i>) |
| HTTPS | Protocolo de Transferência de Hipertexto Seguro (do inglês – <i>Hypertext Transfer Protocol Secure</i>) |
| IaaS | Infraestrutura como Serviço (do inglês – <i>Infrastructure as a Service</i>) |
| IDE | Ambiente de Desenvolvimento Integrado (do inglês – <i>Integrated Development Environment</i>) |
| IoT | Internet das Coisas (do inglês – <i>Internet of Things</i>) |
| MVT | <i>Model-View-Template</i> |
| ORM | Mapeamento Objeto-Relacional (do inglês – <i>Object-Relational Mapping</i>) |
| PaaS | Plataforma como Serviço (do inglês – <i>Platform as a Service</i>) |
| PIB | Produto Interno Bruto |
| SaaS | Software como Serviço (do inglês – <i>Software as a Service</i>) |
| SOA | Arquitetura Orientada a Serviços (do inglês – <i>Service-Oriented Architecture</i>) |
| TI | Tecnologia da Informação |
| URL | Localizador Uniforme de Recursos (do inglês – <i>Uniform Resource Locator</i>) |
| UTFPR | Universidade Tecnológica Federal do Paraná |

SUMÁRIO

| | | |
|-------------|---|-----------|
| 1 | INTRODUÇÃO | 10 |
| 1.1 | Objetivos | 12 |
| 1.1.1 | Objetivo Geral..... | 12 |
| 1.1.2 | Objetivos Específicos | 12 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 13 |
| 2.1 | Eixos..... | 13 |
| 2.2 | Fadiga..... | 14 |
| 2.2.1 | Determinação dos fatores de correção..... | 15 |
| 2.3 | Concentradores de tensão | 18 |
| 2.3.1 | Fatores de concentração de tensão | 18 |
| 2.3.2 | Fatores de concentração de tensão para fadiga | 22 |
| 2.4 | Tensões flutuantes..... | 23 |
| 2.5 | Critérios de falha por fadiga..... | 25 |
| 2.5.1 | Critério de falha de <i>Soderberg</i> | 27 |
| 2.5.2 | Critério de falha de <i>Goodman</i> modificado | 28 |
| 2.5.3 | Critério de falha de <i>Gerber</i> | 28 |
| 2.5.4 | Critério de falha ASME-elíptico..... | 29 |
| 2.6 | Determinação dos esforços internos com CAD/CAE..... | 29 |
| 2.7 | Python | 30 |
| 2.8 | Django | 33 |
| 2.9 | Sistemas especialistas | 34 |
| 2.10 | Computação em nuvem | 35 |
| 3 | ASPECTOS METODOLÓGICOS..... | 37 |
| 4 | CONCEPÇÃO E DESENVOLVIMENTO DO TRABALHO | 42 |
| 4.1 | Instalação do Django | 42 |
| 4.2 | Elaboração da Interface Web com Django | 43 |
| 4.3 | Tratamento dos dados e exibição dos resultados | 48 |
| 4.4 | Hospedagem da aplicação web em nuvem..... | 49 |
| 4.5 | Primeiro estudo de caso..... | 50 |
| 4.6 | Segundo estudo de caso | 55 |
| 5 | CONCLUSÃO | 57 |
| | REFERÊNCIAS..... | 59 |
| | APÊNDICE A - Manual de uso | 62 |

1 INTRODUÇÃO

Eixos são elementos mecânicos usados para transmitir potência ou movimento. Ao projetar um eixo, deve-se prevenir os dois tipos de falha mecânica que podem ocorrer: falha estática e falha por fadiga. A falha estática é relativamente simples de ser prevenida, pois ela ocorre apenas quando o material atinge o seu limite de escoamento. Por outro lado, a falha por fadiga pode acontecer repentinamente a partir do surgimento de pequenas trincas imperceptíveis. Ela é súbita e total e, portanto, perigosa (BUDYNAS; NISBETT, 2016).

Segundo Callister (2002), a fadiga é a maior causa individual de falha em metais, representando cerca de 90% dos casos. Seu impacto na economia também é considerável. Um estudo econômico foi realizado nos Estados Unidos em 1978, para estimar o custo anual devido a ocorrência desse fenômeno no país. O estudo revelou que cerca de 119 bilhões de dólares anuais são desperdiçados com esse tipo de falha, representando 4% do PIB americano (soma de todos os bens e serviços finais produzidos na região). Esse mesmo estudo estima que 35 bilhões poderiam ter sido economizados anualmente caso as tecnologias disponíveis na época fossem aplicadas, e que pesquisas adicionais sobre a mecânica das fraturas poderiam ter reduzido esse número em mais 28 bilhões (ANDERSON, 2005). Além disso, muitos acidentes já ocorreram devido às falhas por fadiga. Alguns exemplos desse tipo de falha em eixos mecânicos podem ser encontrados em trabalhos como os de Chaudhary, Rajak e Ashish (2020), Azevedo et al. (2020) e Wang et al. (2021).

Dessa forma, para evitar a ocorrência de falhas por fadiga em eixos de transmissão, o cálculo ou a validação das dimensões de um eixo são realizados com base em critérios de falha. Contudo, para utilizar estes critérios, os projetistas ainda enfrentam os problemas de trabalhar com um grande número de fórmulas, cálculos e processos de iteração envolvidos. Isso tornou os procedimentos de projeto de eixos pesados e rigorosos, portanto demorados. Esse problema se torna ainda mais pronunciado se o projetista estiver interessado em ver os efeitos da variação de um ou mais parâmetros de projeto: isso significa que ele tem que começar novamente do zero (ADEKUNLE; ADEJUYIGBE; ARULOGUN, 2012).

Nesse contexto, a evolução da tecnologia e das linguagens de programação realizam um papel fundamental para a resolução de tais problemas. Com a ajuda de sistemas especialistas, capazes de reproduzir o conhecimento específico de um

humano especialista, esses processos tornam-se muito mais simples (FLORES, 2003).

Assim, é perceptível que existe uma demanda crescente relacionada à automatização de processos na última década, conforme trabalho apresentado por Adekunle, Adejuyigbe e Arulogun (2012). Aliando isso à evolução constante da computação em nuvem, estimulada pela indústria 4.0, surge uma oportunidade de pesquisa para incorporar essas novas tecnologias no dia a dia de um engenheiro mecânico. Dessa forma, é possível facilitar seu trabalho, reduzir erros e auxiliar em suas tomadas de decisões, ainda durante a fase de projeto de um eixo.

Portanto, a proposta deste trabalho é desenvolver uma aplicação web capaz de automatizar o processo de análise de fadiga de eixos de transmissão. Para isso, será utilizado o *framework web* Django, utilizando linguagens de programação interpretadas pelo navegador, como HTML, CSS e JavaScript, para criar uma interface gráfica amigável ao usuário. A lógica responsável por efetuar todos os cálculos será desenvolvida em Python, e os dados de entrada para o programa serão inseridos manualmente através de um formulário. Esse formulário será interativo, ou seja, exibirá apenas as informações que devem ser preenchidas obrigatoriamente pelo usuário, evitando redundâncias e otimizando o processo de inserção de dados.

Com isso, torna-se possível a automatização da análise de fadiga em eixos de forma remota, pública e em tempo real, apenas acessando a URL do *website* que estará hospedado em uma plataforma em nuvem e inserindo os dados de entrada desejados para análise. Todos os códigos desenvolvidos para o correto funcionamento do sistema serão disponibilizados no *GitHub*, bem como o *link* da aplicação web desenvolvida.

Neste trabalho, será utilizada a análise analítica focando nos pontos mais críticos do eixo, chamados de regiões de concentração de tensão, onde há maior acúmulo de tensões e maior probabilidade de ocorrência de falha por fadiga. Esse tipo de abordagem permite obter maior exatidão e precisão nos cálculos, sendo oposta à análise numérica por simulação, que por sua vez abrange o eixo como um todo, em troca da obtenção de resultados aproximados.

Vale destacar que os eixos, estudo de caso deste trabalho, são cilíndricos, maciços, constituídos de aço e possuem diâmetros entre 2,79 e 254 mm, já que este é o valor delimitado pela literatura clássica para o cálculo do fator de tamanho k_b . Além disso, não serão consideradas as cargas axiais presentes num eixo de transmissão,

já que estas normalmente estão relacionadas à dilatação térmica e são desprezadas durante a fase de projeto, visto que os mancais já suprem essa demanda (ADEKUNLE; ADEJUYIGBE; ARULOGUN, 2012).

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo deste trabalho é desenvolver uma aplicação web em nuvem para automatizar a análise de fadiga em eixos mecânicos de transmissão.

1.1.2 Objetivos Específicos

- Realizar um levantamento da literatura clássica sobre os principais conceitos e cálculos para uma análise de fadiga em eixos de transmissão;
- Pesquisar e detalhar termos usados no trabalho, tais como Sistemas especialistas, Computação em nuvem, Django;
- Desenvolver um código em Python para automatizar o processo de análise de fadiga em eixos de transmissão;
- Criar uma aplicação web com interface amigável ao usuário utilizando o *framework* Django, que possibilite a inserção dos dados de entrada que serão utilizados nos cálculos;
- Mesclar a interface gráfica com o código em Python, adaptando-o para o Django e permitindo a entrada de dados através do formulário e a exibição dos resultados;
- Hospedar a aplicação web em uma plataforma em nuvem, que possa ser acessada por qualquer usuário apenas utilizando uma URL;
- Demonstrar a aplicabilidade e funcionalidade da aplicação web criada através de dois estudos de caso retirados da literatura, discutindo e comparando os resultados.

2 FUNDAMENTAÇÃO TEÓRICA

Para desenvolver um sistema capaz de realizar a análise de fadiga em eixos de transmissão, faz-se necessário conhecer alguns conceitos básicos e revisar parte da literatura clássica referente à eixos, falhas por fadiga, concentradores de tensão, critérios de falha, etc. Também é preciso apresentar ao leitor a linguagem de programação que será utilizada neste trabalho, analisar o estado da arte dos sistemas especialistas, dissertar sobre o *framework* Django, computação em nuvem, entre outros tópicos.

Todas as equações, figuras e tabelas que serão apresentadas nesta seção encontram-se no livro Elementos de Máquinas de Shigley 10ª edição, de Budynas e Nisbett (2016).

2.1 Eixos

O termo eixo geralmente se refere a um elemento relativamente longo de seção transversal circular que gira e transmite potência. Um ou mais componentes, como engrenagens, rodas dentadas, polias, rotores e cames, são usualmente fixados aos eixos através de pinos, chavetas, cavilhas, anéis de pressão, cupilhas e outros elementos de fixação (JUVINALL; MARSHEK, 2016). O eixo é um dos elementos de máquina mais amplamente usados em várias aplicações de engenharia mecânica, especialmente na transmissão de potência. Eles são encontrados em aplicações como virabrequim, eixo de impulsor, eixos de hélice, eixos de comando de válvulas, caixa de engrenagens, ventiladores, etc., e a aplicação de um eixo geralmente determina sua nomenclatura (ADEKUNLE; ADEJUYIGBE; ARULOGUN, 2012).

Os eixos podem ser submetidos a diversas combinações de cargas torcionais, axiais e de flexão, e essas cargas podem ser estáticas ou flutuantes. Tipicamente, um eixo girante transmitindo potência fica submetido a um torque constante combinado com uma carga de flexão completamente alternada (JUVINALL; MARSHEK, 2016).

Tensões axiais em eixos, em virtude de componentes axiais transmitidas através de engrenagens helicoidais ou mancais de rolos cônicos, serão quase sempre desprezíveis comparadas às tensões de momento flexor e de torção. Elas são em geral também constantes, assim contribuem pouco à fadiga. Consequentemente é aceitável desprezar as tensões axiais induzidas por engrenagens e mancais quando a flexão está presente (BUDYNAS; NISBETT, 2016).

A fratura de um eixo pode ocorrer por falhas estáticas ou falhas por fadiga. As falhas estáticas são simples de serem identificadas, já que ocorrem quando o material atinge o seu limite de escoamento. Porém, as falhas por fadiga podem acontecer repentinamente a partir do surgimento de pequenas trincas imperceptíveis. Segundo Juvinall e Marshek (2016), para que ocorra uma falha por fadiga em um aço é necessário que haja antes uma deformação plástica repetida no material.

2.2 Fadiga

Uma falha por fadiga quase sempre começa em uma descontinuidade geométrica do eixo, tal como um entalhe, uma fissura, ou outra área de concentração de tensão (BUDYNAS; NISBETT, 2016). Em geral, a fratura final é principalmente “frágil” e ocorre devido a uma concentração de tensão e uma carga aplicada subitamente (JUVINALL; MARSHEK, 2016). Essas regiões mais críticas podem ser ressalto, canais de chaveta e canais de anéis de retenção, por exemplo, e fenômenos tais como o desgaste adesivo, abrasivo e por corrosão do material também podem diminuir a resistência à fadiga de um eixo.

Para garantir que não ocorra a falha por fadiga, é necessário que as tensões fiquem abaixo do limite à fadiga (que também é chamado de resistência à endurança). Para Budynas e Nisbett (2016), podemos estimar o valor da resistência à endurança teórica S'_e (para aços) de forma que:

$$\begin{aligned} S'_e &= 0,5 S_{ut} & S_{ut} &\leq 1400 \text{ MPa} \\ S'_e &= 700 \text{ MPa} & S_{ut} &> 1400 \text{ MPa} \end{aligned} \quad (1)$$

em que S_{ut} é a resistência última à tração do material em MPa .

Essa relação é uma estimativa obtida por dados experimentais através da máquina de testes de R. R. Moore, que é um modelo de máquina tradicional muito utilizado na indústria como referência para analisar o ensaio de fadiga por flexão rotativa, capaz de realizar simulações de carregamento alternado. Seu funcionamento consiste em submeter um corpo de prova à flexão alternada utilizando pesos enquanto o mesmo é rotacionado por um motor. Assim, determina-se o número de ciclos necessários para seu rompimento e gera-se um diagrama S-N (tensão & número de ciclos). Nesse diagrama é possível perceber que existe um comportamento assintótico para aços em geral, em que a tensão máxima admissível para suportar um número

de ciclos acima de um milhão é, em média, equivalente à metade da resistência última à tração, definindo-se assim o limite à endurance teórico.

No entanto, esse ensaio possui algumas especificações técnicas, entre elas a temperatura ambiente de 20 °C, o acabamento superficial polido do corpo de prova, sua geometria circular e a confiabilidade de 50%. Em situações reais, esse valor precisa ser corrigido devido à presença de fatores externos que influenciam em seu comportamento, tais como diferenças em relação ao material, manufatura, ambiente e projeto. Assim, são utilizados diversos fatores de correção que preveem a influência dessas adversidades sobre a resistência à fadiga do material na equação de Marin, de forma que a resistência à endurance torna-se:

$$S_e = k_a k_b k_c k_d k_e k_f S'_e \quad (2)$$

onde:

S_e = resistência à endurance

k_a = fator de superfície

k_b = fator de tamanho

k_c = fator de carga

k_d = fator de temperatura

k_e = fator de confiabilidade

k_f = fator de efeitos diversos

S'_e = resistência à endurance teórica

2.2.1 Determinação dos fatores de correção

O fator de superfície k_a depende da qualidade do acabamento da superfície da peça verdadeira e da resistência à tração do material da peça (BUDYNAS; NISBETT, 2016). Ele pode ser calculado por:

$$k_a = a S_{ut}^b \quad (3)$$

em que S_{ut} é a resistência última à tração do material, e os coeficientes a e b podem ser obtidos pela Tabela 1 de acordo com o processo de fabricação do eixo. Os valores de a listados na tabela são válidos somente se S_{ut} estiver em MPa.

Tabela 1 - Parâmetros para cálculo do fator de superfície

| Acabamento superficial | <i>a</i> | <i>b</i> |
|-------------------------------|-----------------|-----------------|
| Retificado | 1,58 | -0,085 |
| Usinado ou laminado a frio | 4,51 | -0,265 |
| Laminado a quente | 57,7 | -0,718 |
| Forjado | 272 | -0,995 |

Fonte: Adaptado de Budynas e Nisbett (2016)

O fator de tamanho k_b depende do diâmetro nominal do eixo, e para materiais dúcteis ele é calculado por:

$$\begin{aligned}
 k_b &= 1,24d^{-0,107} & 2,79 \leq d \leq 51 \text{ mm} \\
 k_b &= 1,51d^{-0,157} & 51 < d \leq 254 \text{ mm}
 \end{aligned}
 \tag{4}$$

em que d é o diâmetro nominal do eixo na seção que está sendo analisada. Para carregamentos axiais não há efeito de tamanho, portanto considera-se $k_b = 1$ (BUDYNAS; NISBETT, 2016).

O fator de carga k_c é definido em função do tipo de carga e da resistência última do material, porém ele varia muito pouco em relação à diferentes valores de S_{ut} . Assim, é mais apropriado especificar somente três valores médios de acordo com o tipo de carga atuante sobre o eixo, conforme mostrado na Tabela 2.

Tabela 2 - Fator de carga

| Tipo de carga | k_c |
|----------------------|-------------------------|
| Flexão | 1 |
| Axial | 0,85 |
| Torção pura | 0,59 |

Fonte: Adaptado de Budynas e Nisbett (2016)

Quando a torção está combinada com outro tipo de carregamento, como a flexão, assume-se $k_c = 1$.

O fator de temperatura k_d considera o efeito da temperatura sobre o material, e geralmente adota-se o valor $k_d = 1$ quando o eixo está trabalhando em temperatura ambiente, de aproximadamente 20 °C. Quando a temperatura não é considerada como ambiente, utiliza-se a Equação (5):

$$k_d = \frac{S_T}{S_{RT}} \tag{5}$$

onde

S_T = resistência à tração na temperatura de operação

S_{RT} = resistência à tração à temperatura ambiente

Os valores de S_T/S_{RT} estão disponíveis na Tabela 3, disponibilizada por Budynas e Nisbett (2016).

Tabela 3 - Efeito da temperatura na resistência à tração do aço

| Temperatura (°C) | S_T/S_{RT} |
|------------------|--------------|
| 20 | 1,000 |
| 50 | 1,010 |
| 100 | 1,020 |
| 150 | 1,025 |
| 200 | 1,020 |
| 250 | 1,000 |
| 300 | 0,975 |
| 350 | 0,943 |
| 400 | 0,900 |
| 450 | 0,843 |
| 500 | 0,768 |
| 550 | 0,672 |
| 600 | 0,549 |

Fonte: Adaptado de Budynas e Nisbett (2016)

O fator de confiabilidade k_e leva em consideração a incerteza dos dados utilizados, e geralmente considera-se que a confiabilidade é de 50% para aços com resistência última à tração inferior à 1400 MPa, o que resulta em $k_e = 1$. Este fator reconhece que uma estimativa mais confiável (acima de 50%) requer a utilização de um valor mais baixo do limite de fadiga (JUVINALL; MARSHEK, 2016). Para valores maiores de resistência à tração, a confiabilidade pode oscilar muito, portanto a estimativa de 50% é inválida nestes casos. O valor de k_e para outras confiabilidades pode ser obtido pela Tabela 4.

Tabela 4 - Fator de confiabilidade

| Confiabilidade (%) | k_e |
|--------------------|-------|
| 50 | 1,000 |
| 90 | 0,897 |
| 95 | 0,868 |
| 99 | 0,814 |
| 99,9 | 0,753 |
| 99,99 | 0,702 |
| 99,999 | 0,659 |
| 99,9999 | 0,620 |

Fonte: Adaptado de Budynas e Nisbett (2016)

O fator de efeitos variados k_f é apenas uma forma de lembrar ao projetista de que outros fatores externos, como a corrosão, também devem ser considerados na análise. Usualmente, adota-se o valor de $k_f = 1$ caso não haja nenhuma adversidade expressiva. Alguns exemplos de situações em que é preciso adotar um valor de k_f inferior à 1 são os casos onde há revestimento eletrolítico, pulverização de metal, frequência cíclica, corrosão de piezo-ciclofricção, entre outros, sendo o valor de k_f variável de acordo com cada caso e em muitos casos irá depender da experiência do projetista para assumir um valor diferente a 1. Uma análise mais detalhada sobre a influência desses fenômenos pode ser encontrada em Budynas e Nisbett (2016).

2.3 Concentradores de tensão

No desenvolvimento das equações fundamentais de tensão por flexão e torção, supôs-se que não ocorriam quaisquer irregularidades geométricas no elemento considerado. Porém, é muito difícil projetar uma máquina sem permitir algumas mudanças nas seções transversais dos elementos. Em uma peça existem irregularidades ou descontinuidades, tais como orifícios, sulcos ou entalhes, que aumentam as tensões teóricas significativamente nas proximidades imediatas da descontinuidade (BUDYNAS; NISBETT, 2016).

Essas descontinuidades são chamadas de concentradores de tensão, e as regiões em que elas ocorrem são chamadas de áreas de concentração de tensão. Para calcular a tensão máxima nessas regiões, utilizamos os fatores de concentração de tensão K_t , para tensões normais, ou K_{ts} , para tensões de cisalhamento, através das Equações (6) e (7):

$$\sigma_{m\acute{a}x} = K_t \sigma_0 \quad (6)$$

$$\tau_{m\acute{a}x} = K_{ts} \tau_0 \quad (7)$$

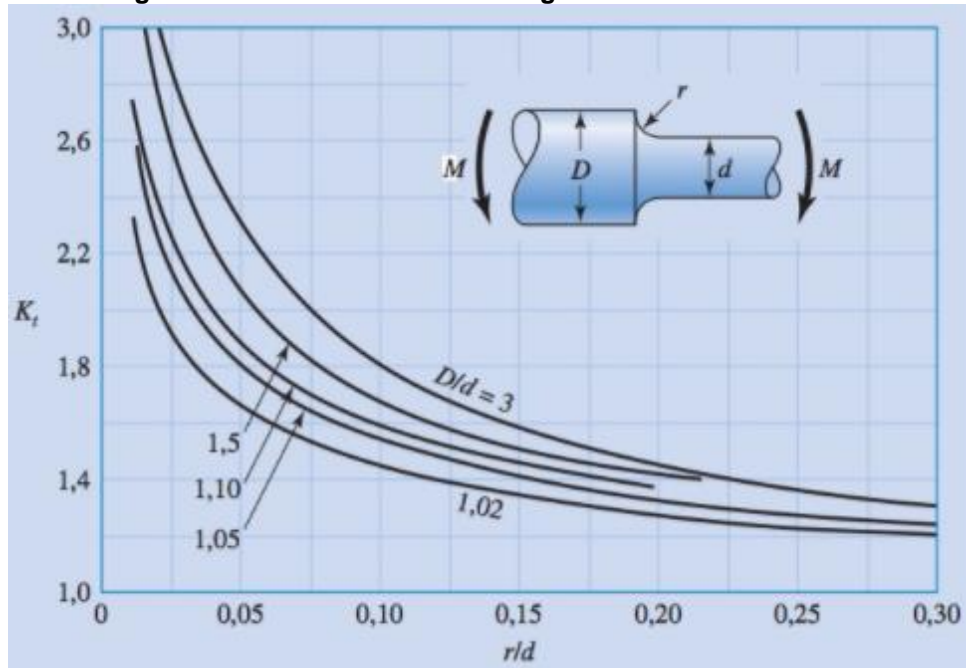
em que $\sigma_{m\acute{a}x}$ é a tensão normal máxima, σ_0 é a tensão nominal normal, $\tau_{m\acute{a}x}$ é a tensão de cisalhamento máxima e τ_0 é a tensão nominal de cisalhamento.

2.3.1 Fatores de concentração de tensão

Para a obtenção dos fatores de concentração de tensão K_t e K_{ts} , são utilizados gráficos que variam de acordo com a geometria do eixo e o tipo de carga que está sendo aplicada, que serão apresentados a seguir.

A Figura 1 mostra os valores de K_t para eixos cilíndricos com filetagem no ressalto submetidos a cargas de flexão.

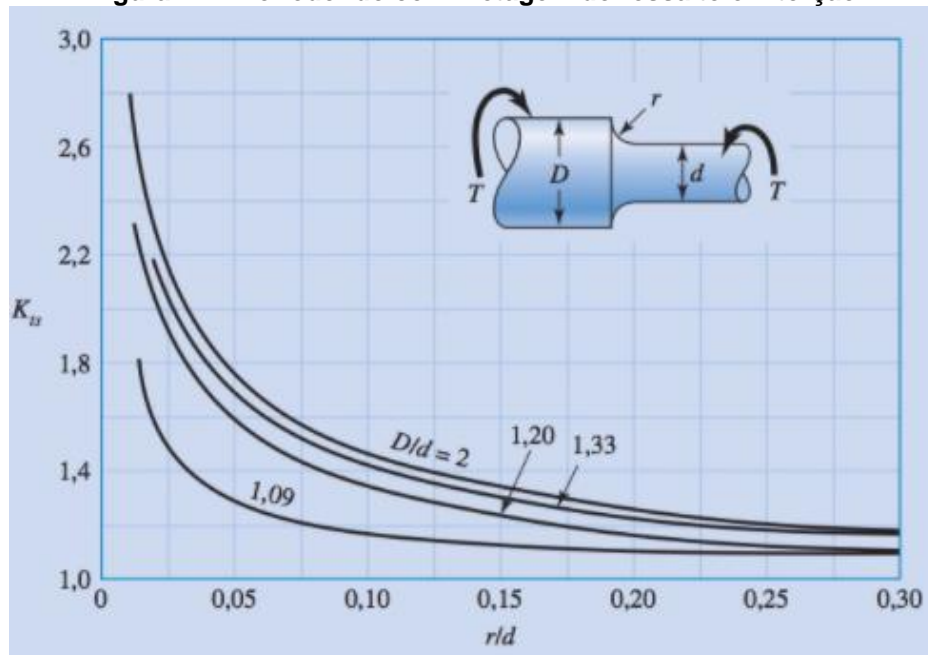
Figura 1 - Eixo redondo com filetagem do ressalto em flexão



Fonte: Budynas e Nisbett (2016)

A Figura 2 mostra os valores de K_{ts} para eixos cilíndricos com filetagem no ressalto submetidos a torção.

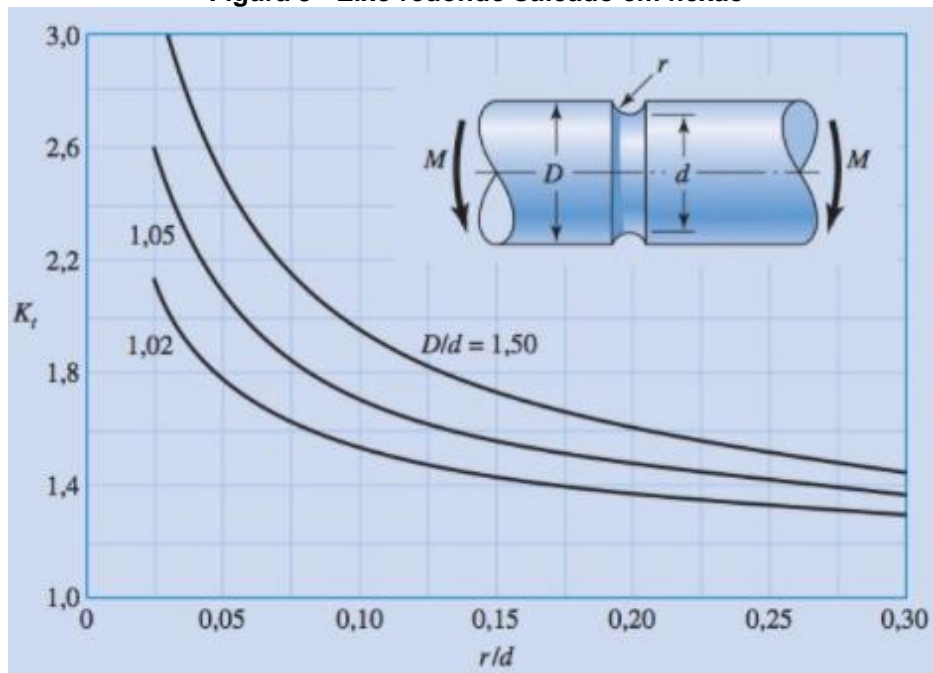
Figura 2 - Eixo redondo com filetagem do ressalto em torção



Fonte: Budynas e Nisbett (2016)

A Figura 3 mostra os valores de K_t para eixos cilíndricos em casos que existe um sulco para alívio de tensões, submetidos a cargas de flexão.

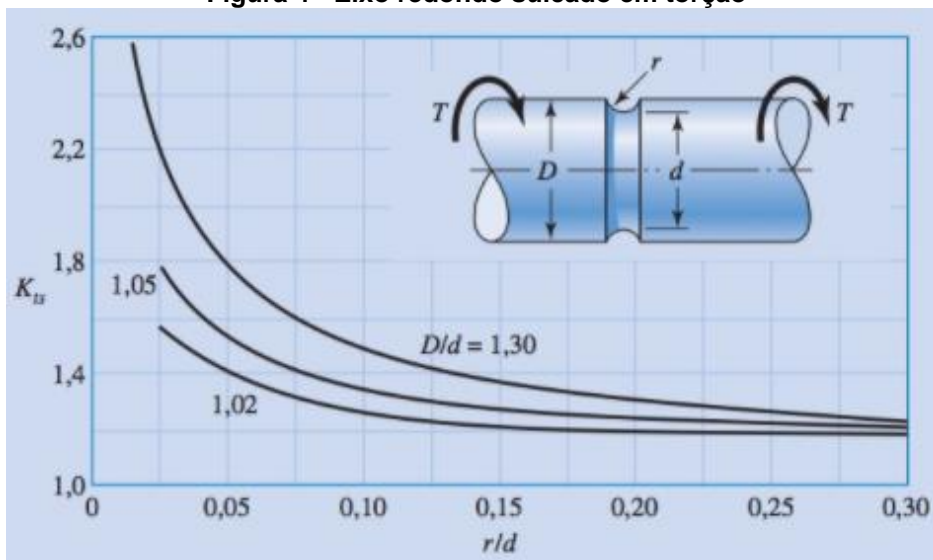
Figura 3 - Eixo redondo sulcado em flexão



Fonte: Budynas e Nisbett (2016)

A Figura 4 mostra os valores de K_{ts} para eixos cilíndricos sulcados e submetidos a cargas de torção.

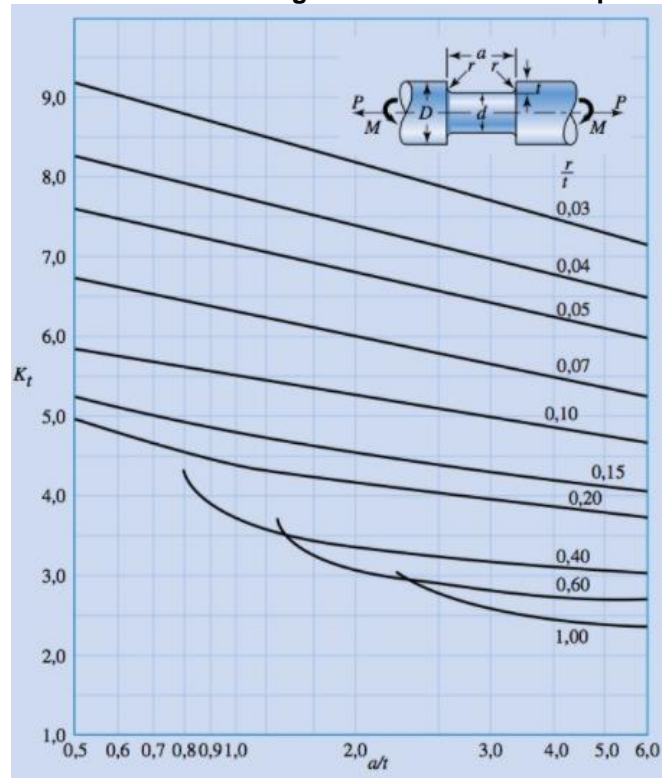
Figura 4 - Eixo redondo sulcado em torção



Fonte: Budynas e Nisbett (2016)

A Figura 5 mostra os valores de K_t para eixos cilíndricos com canais de fundo plano submetidos a cargas de flexão.

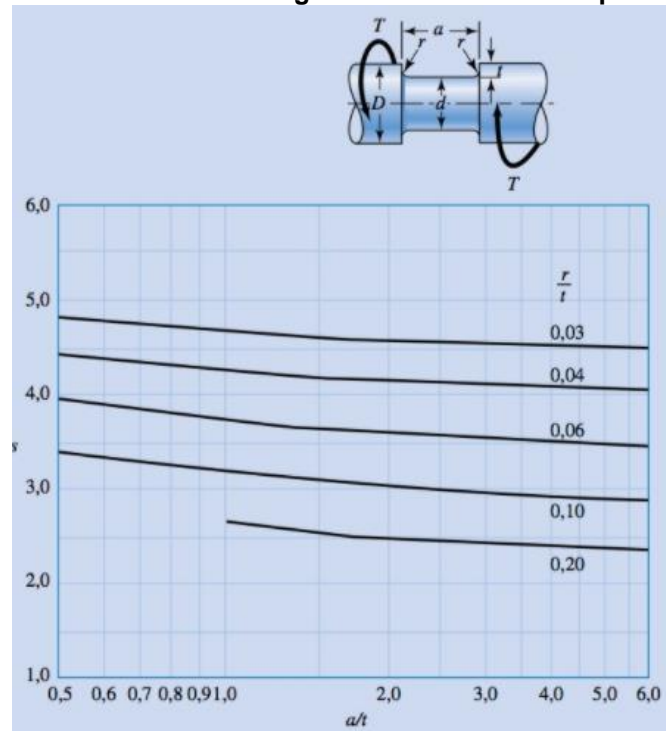
Figura 5 - Eixo redondo com rasgo de chave de fundo plano em flexão



Fonte: Budynas e Nisbett (2016)

A Figura 6 mostra os valores de K_{ts} para eixos cilíndricos com canais de fundo plano submetidos a cargas de torção.

Figura 6 - Eixo redondo com rasgo de chave de fundo plano em torção



Fonte: Budynas e Nisbett (2016)

Além disso, existem outros gráficos de fatores de concentração de tensão apresentando os valores de K_t e K_{ts} para geometrias mais específicas, que podem ser consultados no trabalho de Budynas e Nisbett (2016).

Contudo, quando as dimensões do eixo são desconhecidas, é possível estimar os valores iniciais de K_t e K_{ts} através da Tabela 5.

Tabela 5 - Estimativas iniciais de K_t e K_{ts}

| Situação | K_t | K_{ts} |
|--|-------|----------|
| Arredondamento de ressalto pontudo ($r/d = 0,02$) | 2,7 | 2,2 |
| Arredondamento de ressalto bem arredondado ($r/d = 0,1$) | 1,7 | 1,5 |
| Assento de chaveta de extremidade fresada ($r/d = 0,02$) | 2,14 | 3 |
| Assento de chaveta formato corredor de trenó | 1,7 | - |
| Sulco de anel retentor | 5 | 3 |

Fonte: Adaptado de Budynas e Nisbett (2016)

2.3.2 Fatores de concentração de tensão para fadiga

Alguns materiais não são totalmente sensíveis à presença de entalhes, logo, para esses, um valor reduzido de K_t e K_{ts} pode ser usado (BUDYNAS; NISBETT, 2016). Esses fatores reduzidos são chamados de fatores de concentração de tensão para fadiga, e podem ser calculados pelas Equações (8) e (9) para flexão e torção, respectivamente.

$$K_f = 1 + q(K_t - 1) \quad (8)$$

$$K_{fs} = 1 + q_s(K_{ts} - 1) \quad (9)$$

onde K_f é o fator de concentração de tensão de fadiga à flexão, K_{fs} é o fator de concentração de tensão de fadiga à torção, q é a sensibilidade de entalhe à flexão e q_s (ou $q_{\text{cisalhamento}}$) é a sensibilidade de entalhe à torção.

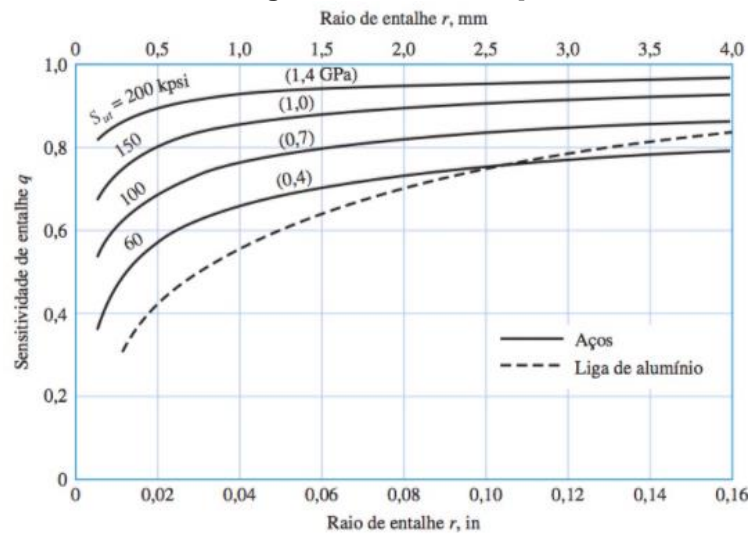
Os valores de q e q_s podem ser obtidos pelas Equações (10) e (11) respectivamente, utilizando S_{ut} em MPa e o raio de entalhe r em mm .

$$q = \frac{1}{1 + \frac{1}{\sqrt{r * 0,03937}} \times (0,246 - 3,08 \times 10^{-3}(S_{ut} * 0,145038) + 1,51 \times 10^{-5} * (S_{ut} * 0,145038)^2 - 2,67 \times 10^{-8} * (S_{ut} * 0,145038)^3)} \quad (10)$$

$$q_s = \frac{1}{1 + \frac{1}{\sqrt{r * 0,03937}} (0,190 - 2,51 \times 10^{-3}(S_{ut} * 0,145038) + 1,35 \times 10^{-5} * (S_{ut} * 0,145038)^2 - 2,67 \times 10^{-8} * (S_{ut} * 0,145038)^3)} \quad (11)$$

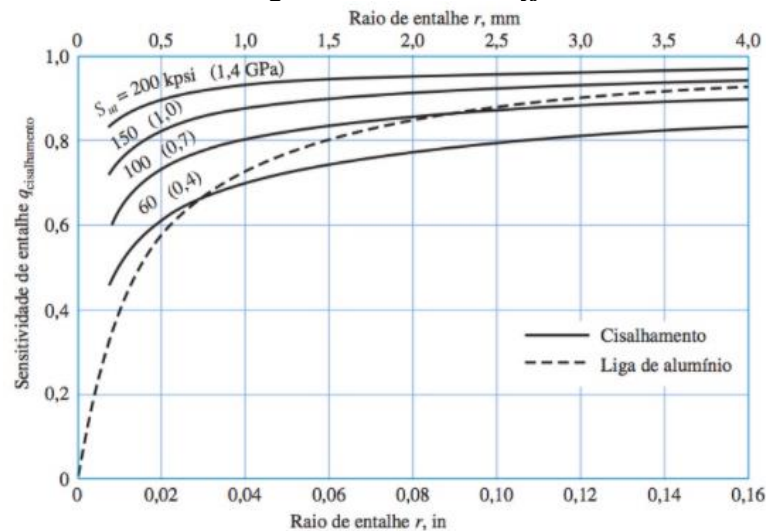
Além disso, os valores de q e q_s também podem ser obtidos pelas Figuras 7 e 8.

Figura 7 - Valores de q



Fonte: Budynas e Nisbett (2016)

Figura 8 - Valores de q_s



Fonte: Budynas e Nisbett (2016)

Além de se conhecer os fatores de correção e de concentração de tensão para fadiga, também devemos trabalhar em função de tensões flutuantes.

2.4 Tensões flutuantes

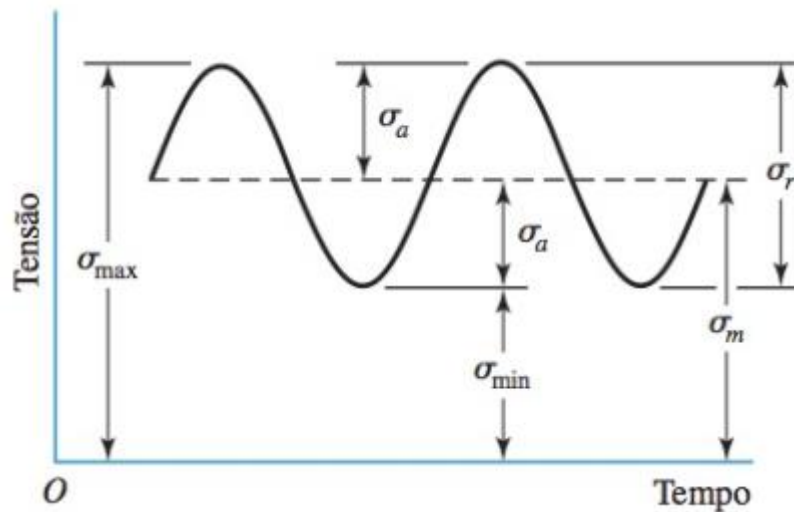
Tensões flutuantes frequentemente tomam a forma de um padrão senoidal por causa da natureza de algumas máquinas rotativas. Contudo, outros padrões, bastante irregulares, também podem ocorrer (BUDYNAS; NISBETT, 2016). Essas tensões podem ser decompostas em duas componentes distintas, sendo uma média (constante) e uma alternada (que oscila em torno da componente média), e essas componentes podem ser calculadas pelas Equações (12) e (13).

$$\sigma_m = \frac{(\sigma_{m\acute{a}x} + \sigma_{m\acute{i}n})}{2} \quad (12)$$

$$\sigma_a = \left| \frac{\sigma_{m\acute{a}x} - \sigma_{m\acute{i}n}}{2} \right| \quad (13)$$

As tensões flutuantes podem ser representadas graficamente de acordo com relações de tempo-tensão, onde a componente alternada oscila em torno da tensão média entre os pontos mínimos e máximos, conforme mostrado na Figura 9.

Figura 9 - Tensão flutuante e suas components



Fonte: Budynas e Nisbett (2016)

onde:

σ_m = tensão média

σ_a = tensão alternante

$\sigma_{m\acute{i}n}$ = tensão mínima

$\sigma_{m\acute{a}x}$ = tensão máxima

σ_r = variação de tensão

A divisão da tensão em componentes média e alternada também é válida para forças, momentos e tensões de cisalhamento que possam ser descritas dessa forma. Um exemplo de tensão flutuante em um eixo é a vibração a que o mesmo está submetido, possuindo uma componente alternante que varia o valor da tensão de acordo com a sua intensidade. Outro exemplo é a presença de um elemento mecânico que está sobre a superfície de um eixo rotativo e sujeito a cargas de flexão, como uma engrenagem. Isso faz com que ocorra o surgimento de um ponto do eixo com uma intensidade diferente, produzindo tensões flutuantes.

Toda essa teoria serve para obter critérios de segurança que possam estimar a possibilidade de ocorrência de falhas por fadiga numericamente, de acordo com os critérios de falha por fadiga que serão apresentados a seguir.

2.5 Critérios de falha por fadiga

Quando um engenheiro enfrenta o problema de executar um projeto utilizando um material específico, torna-se importante estabelecer um limite superior para o estado de tensão que define a falha do material (HIBBELER, 2010). Os critérios de falha são utilizados para determinar as tensões admissíveis máximas que o material pode suportar segundo um critério específico, e essas tensões são calculadas através do fator de segurança n , obtido por uma equação própria para cada critério. Ao obter a informação da tensão admissível máxima, é possível determinar o diâmetro do eixo de acordo com cada critério de falha para que não ocorra a sua falha por fadiga, sendo que diferentes critérios resultam em diferentes fatores de segurança e diâmetros.

Não existe nenhuma teoria de falha única que possa ser aplicada a um material específico todas as vezes. Quando usamos uma determinada teoria de falha, em primeiro lugar é necessário calcular as componentes da tensão normal e de cisalhamento em pontos do elemento estrutural onde essas tensões são maiores (HIBBELER, 2010). Segundo Budynas e Nisbett (2016) e também Juvinall e Marshek (2016), esses pontos onde as tensões são maiores são as regiões de concentração de tensão do eixo, que possuem maior momento fletor, torque e entalhes concentradores de tensão.

As cargas axiais, como já mencionado anteriormente, serão desconsideradas na análise de fadiga deste trabalho devido à sua pouca influência. Porém, elas devem ser levadas em consideração para aplicações onde são a principal forma de carga atuante, como em furadeiras. Relacionando os tópicos anteriores, ao combinarmos os fatores de concentração de tensão com as tensões flutuantes, temos que as componentes médias e alternadas das tensões normais e de cisalhamento, respectivamente, podem ser dadas pelas Equações (14) e (15):

$$\sigma_m = \frac{K_f M_m c}{I} \quad \sigma_a = \frac{K_f M_a c}{I} \quad (14)$$

$$\tau_m = \frac{K_{fs} T_m r}{J} \quad \tau_a = \frac{K_{fs} T_a r}{J} \quad (15)$$

onde M_m é a componente média do momento fletor, M_a é a componente alternada do momento fletor, T_m é a componente média do torque e T_a é a componente alternada do torque.

Sabe-se, da mecânica dos sólidos, que para um eixo maciço com seção transversal circular o centroide é dado por:

$$c = \frac{d}{2} \quad r = \frac{d}{2} \quad (16)$$

E os momentos de inércia e o momento polar de inércia são:

$$I = \frac{\pi d^4}{64} \quad J = \frac{\pi d^4}{32} \quad (17)$$

Ao substituírmos esses valores nas Equações (14) e (15), obtemos que as tensões médias e alternadas atuantes em um eixo cilíndrico maciço são:

$$\sigma_m = \frac{32K_f M_m}{\pi d^3} \quad \sigma_a = \frac{32K_f M_a}{\pi d^3} \quad (18)$$

$$\tau_m = \frac{16K_{fs} T_m}{\pi d^3} \quad \tau_a = \frac{16K_{fs} T_a}{\pi d^3} \quad (19)$$

Essas tensões também podem ser combinadas e escritas em termos da tensão equivalente de *von Mises*, que é utilizada para realizar uma comparação com o limite de resistência ao escoamento do material, a fim de saber se ocorrerá sua ruptura. A tensão equivalente de *von Mises* para um estado plano de tensão é dada por:

$$\sigma' = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 + 3\tau_{xy}^2} \quad (20)$$

E como a tensão normal neste trabalho estará presente apenas na direção do eixo, podemos simplificar a equação para as componentes média e alternada, de forma que:

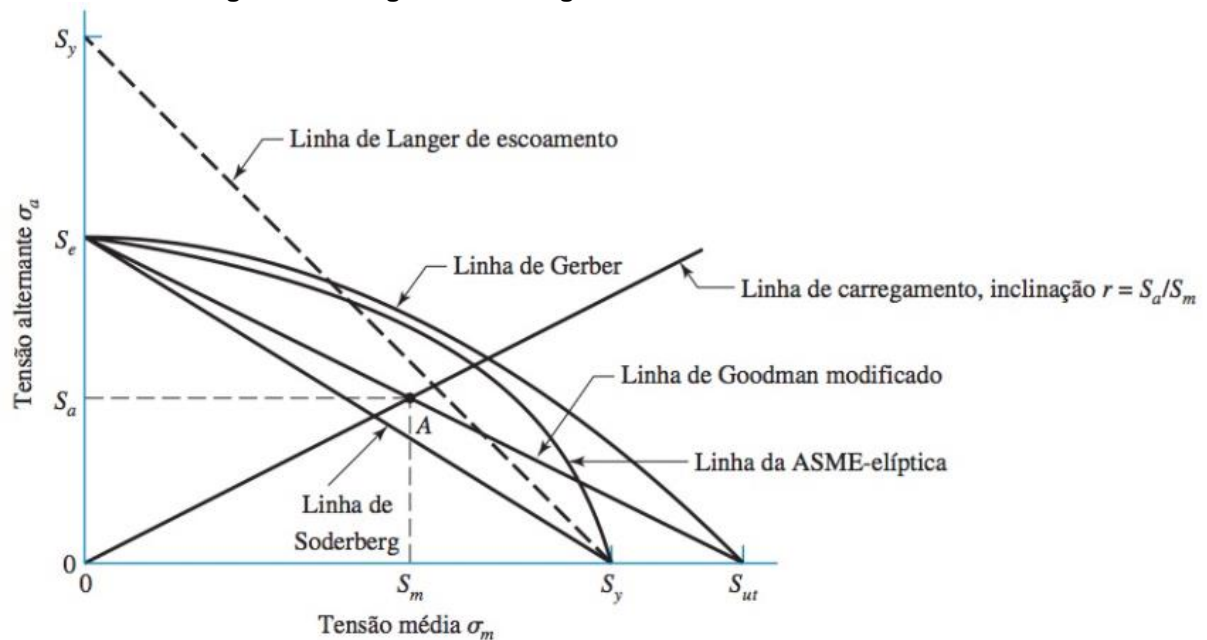
$$\sigma'_m = \sqrt{\sigma_m^2 + 3\tau_m^2} \quad (21)$$

$$\sigma'_a = \sqrt{\sigma_a^2 + 3\tau_a^2} \quad (22)$$

Utilizando as equações (21) e (22) juntamente com a resistência à endurância S_e , o limite de escoamento S_y e a resistência última à tração do material S_{ut} , podemos calcular o valor do fator de segurança, n , de acordo com diferentes critérios de falha. Segundo Juvinal e Marshek (2016), para eixos de transmissão pode-se considerar um fator de segurança entre 2 e 2,5, enquanto Mott (2015) recomenda a utilização de um fator entre 2,5 e 3.

Entre os critérios de falha, os mais difundidos e utilizados no cotidiano são o critério de *Goodman* modificado e o critério de *Soderberg*, além de existirem outros critérios como o de *Gerber* e o ASME-elíptico. Cada um desses critérios possui suas características próprias com vantagens e desvantagens, e uma comparação entre eles pode ser observada na Figura 10.

Figura 10 - Diagrama de fadiga com diferentes critérios de falha



Fonte: Budynas e Nisbett (2016)

2.5.1 Critério de falha de *Soderberg*

O critério de *Soderberg* prevê a falha com base na resistência à endurância S_e ao longo da linha de tensão alternada e na resistência ao escoamento S_y ao longo da linha de tensão média. Este critério previne a possibilidade de falha desde o primeiro ciclo e é muito seguro de se utilizar (ADEKUNLE; ADEJUYIGBE; ARULOGUN, 2012). O critério de *Soderberg* é o mais conservador entre todos aqui mencionados, o que implica na utilização de diâmetros maiores, e o seu fator de segurança é dado por:

$$n = \left(\frac{\sigma'_a}{S_e} + \frac{\sigma'_m}{S_y} \right)^{-1} \quad (23)$$

A análise de fadiga realizada pelo critério de *Soderberg* sempre respeitará a zona em que não ocorrerá falha por escoamento, conforme pode ser observado na Figura 2. Dessa forma, não é necessário realizar uma verificação de segurança para o primeiro ciclo ao utilizar este critério. O mesmo não acontece com o critério de *Goodman* modificado, que será visto a seguir.

2.5.2 Critério de falha de *Goodman* modificado

O critério de *Goodman* modificado considera a falha com base na resistência à endurança S_e e na resistência última à tração S_{ut} do material. No entanto, ele não aborda as possibilidades de falha por escoamento no primeiro ciclo e, como tal, deve ser verificado antes de ser utilizado. Esse critério possui ampla aplicação, em parte devido à sua simplicidade e também por ser menos conservador quando comparado com o critério de *Soderberg* (ADEKUNLE; ADEJUYIGBE; ARULOGUN, 2012). O fator de segurança de acordo com o critério de *Goodman* modificado é dado por:

$$n = \left(\frac{\sigma'_a}{S_e} + \frac{\sigma'_m}{S_{ut}} \right)^{-1} \quad (24)$$

Para realizar a verificação do critério de *Goodman* modificado no primeiro ciclo, como mencionado por Adekunle, Adejuyigbe e Arulogun (2012), podemos utilizar a tensão máxima de *von Mises*, que conforme Budynas e Nisbett (2016) é dada pela Equação (25) e deve ser utilizada para calcular o fator de segurança n_y através da Equação (26).

$$\sigma'_{m\acute{a}x} = \sqrt{(\sigma_m + \sigma_a)^2 + 3(\tau_m + \tau_a)^2} \quad (25)$$

$$n_y = \frac{S_y}{\sigma'_{m\acute{a}x}} \quad (26)$$

Em que n_y é o fator de segurança ao escoamento do material. Ou seja, para que não ocorra a fratura do eixo no primeiro ciclo de aplicação da carga, o valor de n_y deve ser maior do que o valor do fator de segurança n calculado pelo critério de *Goodman* modificado.

2.5.3 Critério de falha de *Gerber*

O critério de falha de *Gerber* é menos conservador do que os critérios de *Soderberg* e *Goodman* modificado, oferecendo maior risco de falha em troca da possível utilização de menos material no projeto do eixo, devido aos seus diâmetros menores.

Para Norton (2013), enquanto a curva de *Goodman* graficamente representa uma medida do comportamento mínimo dos dados experimentais, a curva de *Gerber* é uma medida do comportamento médio desses parâmetros para os materiais dúcteis. O fator de segurança proposto por *Gerber* é dado pela Equação (27).

$$n = \left\{ \frac{8A}{\pi d^3 S_e} \left[1 + \sqrt{1 + \left(\frac{2BS_e}{AS_{ut}} \right)^2} \right] \right\}^{-1} \quad (27)$$

onde:

$$A = \sqrt{4(K_f M_a)^2 + 3(K_{fs} T_a)^2} \quad (28)$$

$$B = \sqrt{4(K_f M_m)^2 + 3(K_{fs} T_m)^2} \quad (29)$$

2.5.4 Critério de falha ASME-elíptico

Em 1985, o Instituto Nacional Americano de Padrões e a Sociedade Norteamericana de Engenheiros Mecânicos (ANSI/ASME) publicou a norma B106.1M-1985 para projeto de eixos de transmissão (ASME, 1985), que posteriormente foi suspensa em 1994 e descontinuada, pois fornecia resultados não conservativos (JUVINALL; MARSHEK, 2016). Esse critério pressupõe que o carregamento é constituído de flexão alternada (componente de flexão média nula) e torque fixo (componente alternada nula do torque), em um nível que cria tensões abaixo da resistência ao escoamento por torção do material (NORTON, 2013). De acordo com Budynas e Nisbett (2016), o fator de segurança proposto pela ASME é dado pela Equação (30).

$$n = \left(\frac{n\sigma'_a}{S_e} + \frac{n\sigma'_m}{S_y} \right)^{-1} \quad (30)$$

Porém, realizando um levantamento da literatura mais recente, constatou-se que autores como Norton (2013), Juvinall e Marshek (2016) e Collins (2006) afirmam que para qualquer situação em que as componentes do carregamento descritas não são exatamente iguais a zero, é recomendado utilizar um procedimento mais geral que supõe apenas a ausência de carga axial no eixo, dado pela Equação (31).

$$n = \left\{ \frac{16}{\pi d^3} \left[\sqrt{4 \left(\frac{K_f M_a}{S_e} \right)^2 + 3 \left(\frac{K_{fs} T_a}{S_e} \right)^2} + 4 \left(\frac{K_f M_m}{S_y} \right)^2 + 3 \left(\frac{K_{fs} T_m}{S_y} \right)^2 \right] \right\}^{-1} \quad (31)$$

2.6 Determinação dos esforços internos com CAD/CAE

É possível observar que as equações de cálculo de tensões apresentadas até aqui necessitam dos valores de momento fletor e de torque que estão sendo aplicados no eixo. Para encontrar esses valores, serão utilizados diagramas de esforços internos

criados por um *software* CAD/CAE, a fim de evitar o trabalho manual desnecessário para este cálculo, como o método das seções utilizado na Mecânica dos Sólidos.

Os esforços internos em um elemento mecânico caracterizam as ligações internas de tensões, em outras palavras, são integrais de tensões ao longo da seção transversal de uma barra. Eles representam o efeito de forças e momentos entre duas porções de uma estrutura reticulada resultantes de um corte em uma seção transversal. Utilizando as ferramentas computacionais, a tarefa de projetar um eixo sujeito à diferentes cargas e gerar diagramas de esforços internos torna-se simples. Dessa forma, é possível saber exatamente o momento fletor e o torque atuantes em determinados pontos de interesse do eixo, que são os pontos de concentração de tensão.

Programas de Projeto Assistido por Computador (CAD - do inglês *Computer-Aided Design*) é o nome genérico de sistemas computacionais utilizados para facilitar o processo de modelagem de corpos sólidos e a sua representação em desenho técnico. Esse tipo de ferramenta computacional é o que possibilita, por exemplo, a modelagem de um eixo no computador. Contudo, quando alguma análise de forças, tensões, deflexões ou outros aspectos do comportamento físico do projeto é incluída, com ou sem os aspectos da geometria sólida, o processo é chamado de Engenharia Assistida por Computador (CAE - do inglês *Computer-Aided Engineering*) (NORTON, 2013).

Existem diversos programas que possibilitam realizar a modelagem e a análise de elementos mecânicos, tais como AutoCAD, Inventor, SolidWorks, Fusion 360, ANSYS, CATIA V5, entre outros. Para este trabalho, será utilizado o Solid Edge, que é um *software* completo para projeto, simulação, fabricação e muitas outras atividades relacionadas ao desenvolvimento de produtos, possuindo também uma ferramenta para criação automática de eixos na sua versão acadêmica (SIEMENS, 2020).

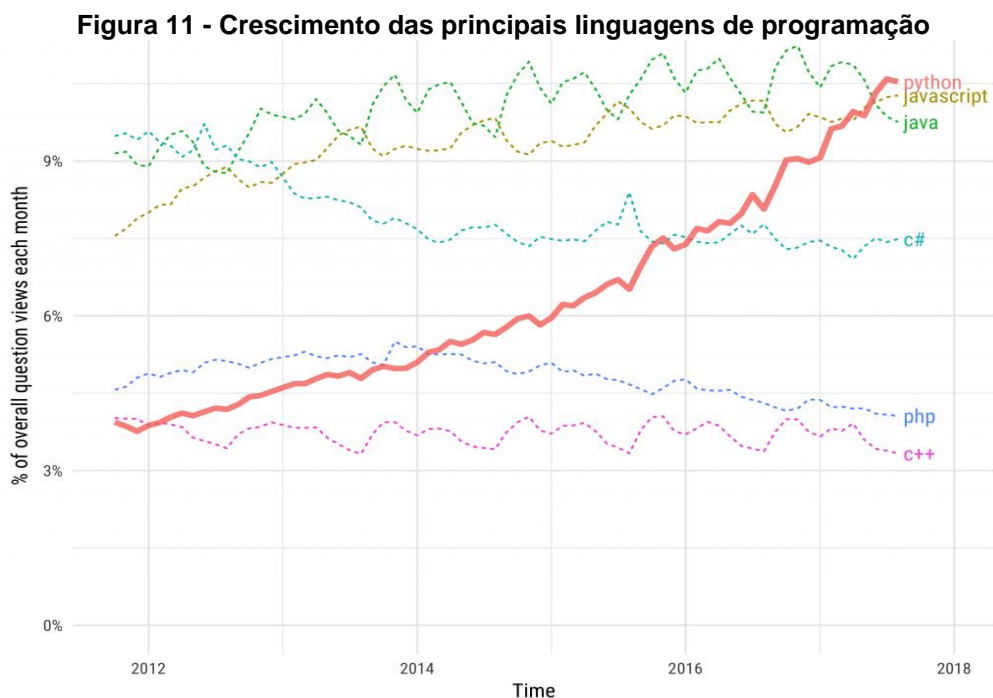
2.7 Python

Linguagens de programação são métodos padronizados de implementação de um código fonte, formados por um conjunto de regras sintáticas e semânticas, que podem ser compilados e transformados em um programa de computador (FISHER; GRODZINSKY, 1993). Segundo Fangohr (2019), para estudar um processo com modelagem computacional devemos distinguir duas etapas: a primeira delas é desenvolver um modelo capaz de descrever o sistema real através de equações, e a

segunda é resolver esse modelo de equações utilizando os métodos computacionais, i.e., uma linguagem de programação.

Python é uma linguagem de programação que possui um grande número de bibliotecas, funções de depuração e uma grande comunidade para oferecer suporte a problemas desafiadores. É uma linguagem com a qual grandes quantidades de dados podem ser gerenciados e processados (KÖRBER; FROMMEL, 2019). Algumas das principais vantagens de se trabalhar com Python são o fato de ser uma linguagem de programação gratuita, de alto nível e possuir sintaxe moderna e clara, o que favorece a escrita de códigos organizados, fáceis de compreender e manter, sem que haja perda de produtividade.

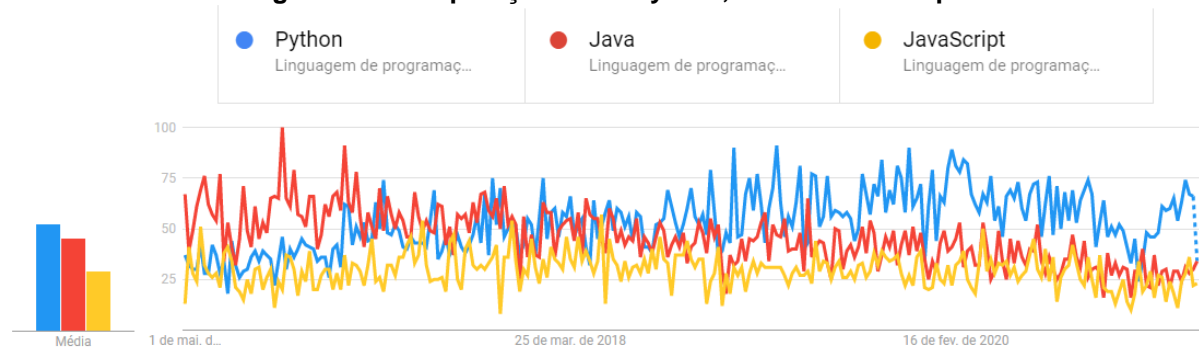
Sendo considerada a linguagem de mais fácil aprendizado, o Python continua sendo uma das linguagens de programação mais populares no mercado, mesmo tendo surgido em 1989, com o principal propósito de levar a programação para todos. Um levantamento feito pelo site *Stack Overflow* em 2017 mostra que Python foi a linguagem que mais cresceu na última década, conforme representado na Figura 11 (ROBINSON, 2017).



Fonte: Stack Overflow (2017)

Além disso, uma rápida pesquisa utilizando o *Google Trends* também evidencia esse crescimento, ao comparar a quantidade pesquisada dos termos “Python”, “Java” e “JavaScript” entre o período de abril de 2016 até abril de 2021. Essa comparação é mostrada na Figura 12.

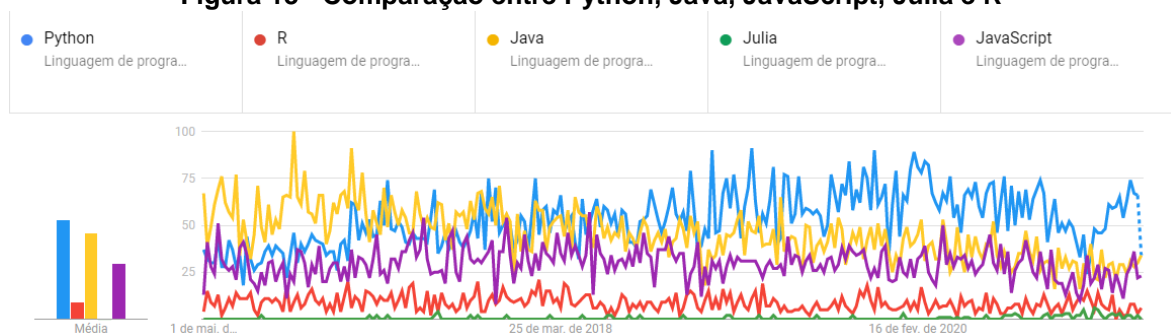
Figura 12 - Comparação entre Python, Java e JavaScript



Fonte: *Google Trends* (2021)

Apesar do Python possuir uma compilação mais lenta quando comparado à outras linguagens de programação, como a C++, Fangohr (2019) argumenta que não é apenas o tempo de execução que contribui para o custo de um projeto computacional: também deve-se considerar o custo do desenvolvimento e manutenção do programa. Existem algumas linguagens de programação concorrentes ao Python, como Julia e R. A linguagem Julia tem como uma de suas principais vantagens possuir uma compilação mais rápida e o intuito de suprir algumas deficiências do Python. Por outro lado, a linguagem R é mais voltada para programação funcional e possui uma série de pacotes embutidos para *Data Science* (em português – ciência de dados), enquanto o Python necessita de diversos pacotes de terceiros. Ao incluir essas linguagens de programação concorrentes na pesquisa do *Google Trends* realizada anteriormente, é perceptível que Julia e R são linguagens muito menos conhecidas e utilizadas quando comparadas ao Python, conforme observado na Figura 13.

Figura 13 - Comparação entre Python, Java, JavaScript, Julia e R



Fonte: *Google Trends* (2021)

Assim, observa-se que o Python possui uma grande comunidade ao redor do mundo e é mais experiente do que suas concorrentes, portanto será a principal linguagem utilizada neste trabalho.

Um ambiente de desenvolvimento integrado (IDE – do inglês *Integrated Development Environment*) é uma ferramenta de codificação que permite que o programador escreva, teste e depure seu código, visando maior produtividade na programação. Neste trabalho, o IDE e editor de código-fonte utilizado será o *Visual Studio Code*, que é customizável e inclui suporte para depuração, realce de sintaxe, complementação inteligente e refatoração de código, entre diversos outros recursos, utilizando a versão do Python 3.9.

2.8 Django

Para simplificar a criação de uma aplicação web com interface amigável ao usuário será utilizado o Django, que é um framework de desenvolvimento web de alto nível, gratuito e de código aberto escrito em Python. Um framework é um conjunto de bibliotecas que armazenam funções básicas para facilitar o desenvolvimento de aplicações web, tendo como principal objetivo resolver problemas recorrentes utilizando uma abordagem mais genérica. Assim, seu propósito é poupar tempo do programador e permiti-lo focar nos problemas da aplicação em si ao invés dos problemas de arquitetura e configurações (GUIMARÃES, 2019). Ou seja, o Django promete a construção de websites de forma muito mais rápida, completa e utilizando pouco código, se autointitulando um “framework web para perfeccionistas com prazos” (DJANGO, 2021).

Esse *framework* foi criado em 2005 por Adrian Holovaty e Simon Willison em Lawrence, no Kansas, sendo utilizado inicialmente como um sistema de gerenciamento de um site jornalístico para inserir notícias no website de maneira rápida e com alta frequência de atualização, e no mesmo ano o projeto se tornou um *software* de código aberto sob licença BSD. Atualmente, o Django é o principal *framework* Python para desenvolvimento web, sendo muito utilizado por grandes empresas como Instagram, Mozilla e Pinterest. Ele resolve problemas de autenticação de dados, roteamento, mapeamento objeto-relacional (ORM), migrações, prevenção de ataques web, entre outros.

O Django utiliza o padrão MVT, ou seja, *model-view-template*. A camada *model* é onde ocorre toda a interação com o banco de dados, enquanto a camada *view* é a responsável pela lógica sobre os dados, que processa uma requisição web e devolve uma resposta. Por fim, a camada *template* é onde ocorre a renderização dos dados para uma visualização mais intuitiva, sendo utilizada para gerar formatos baseados

em texto que o navegador web consegue entender, como HTML, CSS e JavaScript. Ele também possui o princípio DRY (*Don't Repeat Yourself*), ou seja, o princípio de evitar a repetição do código, fazendo com que o desenvolvedor aproveite ao máximo o código que já foi feito.

Quanto às suas aplicações, Hansen, Iftikhar e Bogh (2020) utilizam o Django para o desenvolvimento de uma interface web de uma *AI-Box*, i.e., uma caixa de inteligência artificial versátil e fácil de usar, voltada para empresas pequenas e médias lidarem com casos de alarmes falsos, erros audíveis e ocorrência de erros desconhecidos, de forma a possibilitar que o operador configure uma arquitetura de difícil aprendizagem corretamente, mesmo sem conhecê-la. Reiff et. al. (2020) desenvolvem uma plataforma web com interface intuitiva para análise e monitoramento de dados usando o Django, capaz de acessar e processar diferentes fontes de dados utilizando métodos modulares e que possui o objetivo de tornar os processos de análises complexos mais acessíveis para os operadores da máquina. Por fim, Zhou et. al. (2020) propõem um algoritmo de busca em área circular para o problema de controle do agendamento de serviços multi-robôs em diversas regiões, utilizando o Django como *framework* web na construção de uma plataforma em nuvem para arquitetura orientada a serviços (SOA – do inglês *Service-Oriented Architecture*).

2.9 Sistemas especialistas

Sistemas especialistas (ES – do inglês *Expert Systems*) são *softwares* capazes de reproduzir o raciocínio de um profissional em alguma área de conhecimento específica, através de técnicas de matemática tradicionais implementadas em uma linguagem de programação (VASCONCELOS, MARTINS, 2004). É importante ressaltar que o conceito de sistemas especialistas se difere da inteligência artificial (AI – do inglês *Artificial Intelligence*), visto que esta última envolve a tomada de decisões pelo próprio sistema.

Para Norton (2013), o surgimento dos computadores provocou uma verdadeira revolução no projeto e na análise de engenharia. Problemas cujos métodos de solução eram conhecidos por séculos, e que continuavam sendo praticamente insolúveis devido à alta demanda computacional, agora poderiam ser resolvidos em minutos por microcomputadores de baixo custo. Além disso, a utilização dos sistemas especialistas também contribui para a difusão do conhecimento de especialistas,

maior facilidade de tomada de decisões, maior produtividade e menor ocorrência de erros humanos.

A estrutura básica de um sistema especialista é constituída por: base do conhecimento, motor de inferência e interface com o usuário. Segundo Waterman (1986), os passos para o desenvolvimento de sistemas especialistas são: Seleção do problema; Aquisição de conhecimento; Representação do conhecimento; Codificação de conhecimento; Teste de conhecimento e avaliação; Implementação e manutenção.

A seleção do problema são os principais motivos que levam ao desenvolvimento do sistema especialista em questão, isto é, a necessidade da prevenção de falhas por fadiga de forma mais prática e eficiente, evitando possíveis erros humanos durante os cálculos. Aquisição de conhecimento é fazer o levantamento da fundamentação teórica necessária para desenvolver o sistema, realizando uma revisão da literatura clássica sobre eixos. Representação do conhecimento significa filtrar as informações mais relevantes, ou seja, quais são os principais critérios de falha por fadiga, fórmulas e gráficos que serão utilizados no sistema especialista. Codificação do conhecimento é transcrever as informações mais relevantes para uma linguagem de programação, como Python, adaptando as partes necessárias. Teste de conhecimento e avaliação consistem na realização de estudos de caso para verificar se o sistema desenvolvido é funcional, se está de acordo com a literatura clássica e se atende às expectativas e exigências necessárias para o seu funcionamento. Implementação é permitir que o sistema entre em funcionamento, neste caso utilizando o Django para desenvolver a aplicação web e uma plataforma em nuvem para possibilitar o acesso à mesma. Por fim, a manutenção consiste em conservar o sistema especialista desenvolvido e realizar alterações regulares quando necessário, a fim de mantê-lo sempre atualizado.

2.10 Computação em nuvem

Computação em nuvem é o nome utilizado para o fornecimento de serviços de TI (Tecnologia da Informação) pela internet, tais como o armazenamento de dados, poder de processamento, inteligência artificial, redes, análises e *softwares*. Ao terceirizar esses recursos, eliminam-se os custos para comprar e manter uma infraestrutura de TI física no local, além de aumentar a produtividade e velocidade de execução do serviço, que possui escala global (AZURE, 2021).

Os modelos de implantação de nuvem podem ser: nuvem pública, privada ou híbrida. Além disso, alguns tipos de serviços de computação em nuvem são: Infraestrutura como Serviço (IaaS – do inglês *Infrastructure as a Service*), Plataforma como Serviço (PaaS – do inglês *Platform as a Service*), Software como Serviço (SaaS – do inglês *Software as a Service*), entre outros (ZETTLER, 2021).

Neste trabalho, será utilizada uma nuvem pública através da plataforma Heroku, uma PaaS que suporta diversas linguagens de programação, capaz de realizar o *deploy* de várias aplicações *back-end* para hospedagem gratuita, possuindo integração com o *GitHub* (HEROKU, 2021). Assim, a aplicação desenvolvida será hospedada nessa plataforma diretamente do *GitHub*, podendo ser acessada por qualquer pessoa ao entrar com a URL correspondente em seu navegador.

Ao acessar a URL, uma solicitação do navegador é emitida para a plataforma através do protocolo de comunicação HTTPS (Protocolo de Transferência de Hipertexto Seguro – do inglês *Hypertext Transfer Protocol Secure*), que é utilizado para navegação em sites da internet, estabelecendo uma conexão segura entre o cliente (navegador) e o servidor (*site* ou domínio). Assim, o navegador envia um pedido para o acesso de uma página e o servidor retorna uma resposta junto com os arquivos da página que o usuário deseja acessar (GOMES, 2019). A partir desse ponto, o usuário já consegue enviar e receber informações da aplicação web desenvolvida.

3 ASPECTOS METODOLÓGICOS

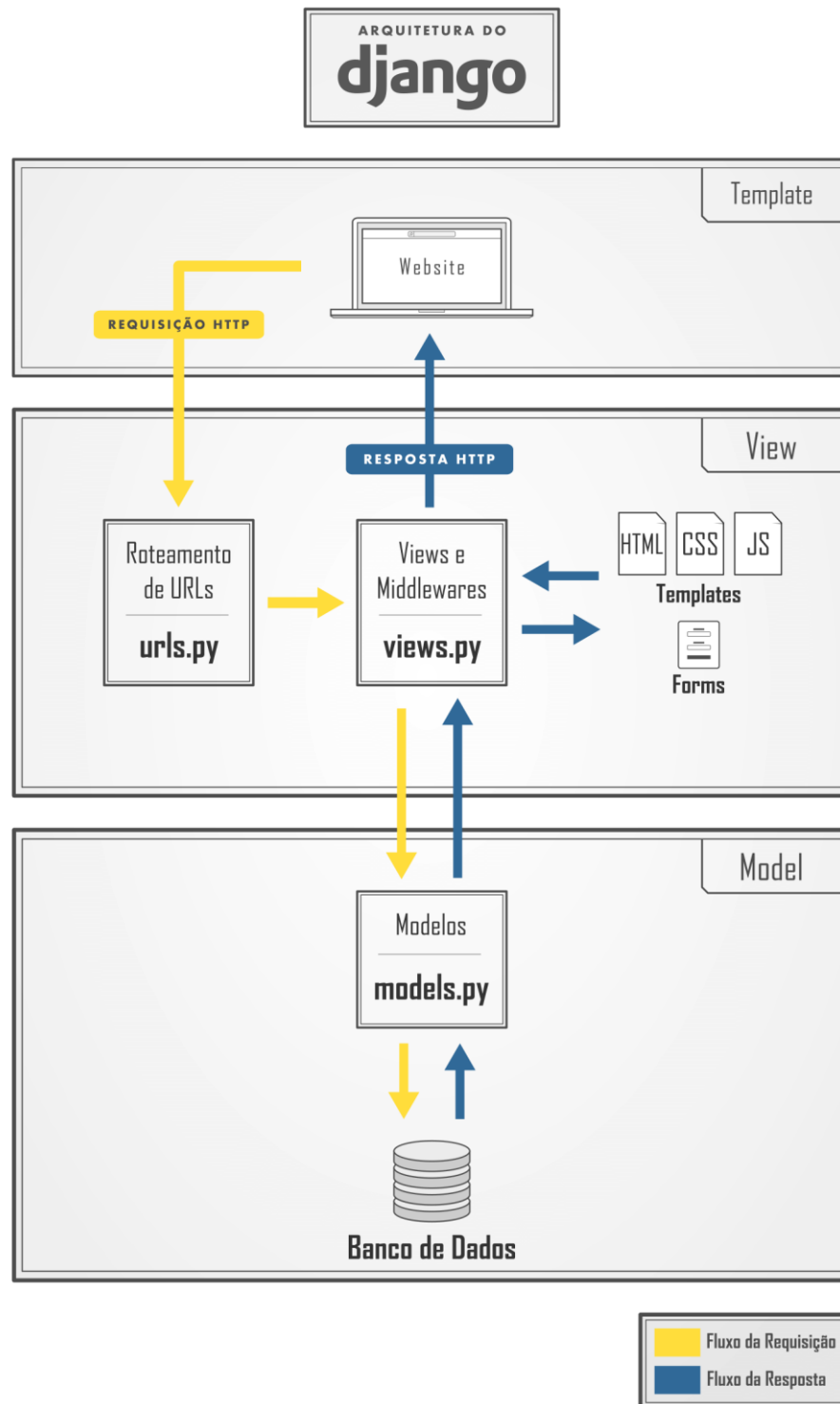
Serão realizados dois estudos de caso para demonstrar a funcionalidade da aplicação web desenvolvida, comparando e discutindo os resultados obtidos com os valores da literatura clássica. O eixo sujeito à análise em um dos estudos será modelado no *software* Solid Edge para se obter os valores dos esforços internos da configuração, selecionando somente seus pontos críticos para análise.

A aplicação web será desenvolvida com o *framework* Django, a fim de tornar o ambiente mais amigável para usuários leigos e facilitar o envio dos dados e o recebimento dos resultados na própria aplicação. Ela possuirá um formulário em sua página inicial, por meio do qual serão inseridos os dados de entrada necessários para realizar os cálculos, e a página dos resultados, que informa ao usuário quais foram os dados de entrada inseridos e retorna o resultado da análise, de acordo com a opção selecionada.

Todos os campos do formulário visíveis para o usuário devem ser preenchidos, visto que o mesmo é interativo e as informações desnecessárias serão ocultas, evitando o fornecimento de informações redundantes por parte do projetista. Caso um campo tenha sido preenchido incorretamente ou ocorra alguma exceção no código, o usuário será redirecionado para uma página de erro, explicando que algo deu errado e contendo um botão para retornar ao formulário e tentar novamente.

Os dados inseridos no formulário da página em HTML serão solicitados pela camada *view* do Django utilizando o método *post*, alocando-os em variáveis no arquivo em Python. Com todos os dados de entrada definidos como variáveis, são chamadas as funções do arquivo "*functions*" da camada *model* em ordem sequencial, para realizar o procedimento de análise por fadiga (processamento dos dados). Por fim, é chamada a função que retorna o resultado final de acordo com uma das seis opções disponíveis, alocando o resultado da análise em uma variável. Esse e outros valores de interesse são retornados para a página de resultados, da camada *template*, e exibidos para o usuário após o envio do formulário. A lógica de comunicação entre as diferentes camadas do Django e o caminho percorrido pela requisição HTTP do usuário pode ser visualizada de forma mais clara na Figura 14.

Figura 14 - Funcionamento do Django



Fonte: *Python Academy* (2021)

Para desenvolver o código em Python, será utilizado como base o trabalho de Carvalho Neto (2020), disponível publicamente no site *GitHub*. O código será modificado de forma a ser adaptado ao Django, simplificado e serão acrescentadas novas funcionalidades, como a validação do diâmetro e do fator de segurança

escolhidos pelo usuário. Além disso, incluirá novas recomendações do programa de acordo com os resultados que foram obtidos ou com os valores que foram utilizados para análise, como quando há a utilização de fatores de segurança muito baixos. Outra sugestão que o programa pode proporcionar é a alteração do tipo de acabamento superficial, material ou diâmetro do eixo, caso o fator de segurança definido pelo usuário não seja validado pela análise.

No formulário da página inicial também será adicionada a opção de selecionar um material pré-definido, com valores de tensão última e de escoamento já inclusas no banco de dados, baseados em valores retirados do *MatWeb* (2021) para evitar a utilização de valores fora da realidade. Assim, o usuário pode optar por escolher um material disponível ou digitar os valores de tensão exatos manualmente, caso já os possua.

Ao selecionar uma função desejada para o programa, será realizado um levantamento entre os diferentes critérios de falha por fadiga, comparando os resultados obtidos e selecionando o diâmetro e o fator de segurança mais indicados para a aplicação fornecida. Isso permitirá ao projetista conferir se os parâmetros de fabricação do eixo são aceitáveis ainda durante sua fase de projeto.

Entrando no aspecto operacional, primeiro é realizada a modelagem das curvas dos Gráficos 1 ao 6 no Microsoft Excel, utilizando dados que são retirados visualmente para obter as funções que podem descrevê-las, a fim de implementá-las na linguagem de programação. Com as funções das curvas obtidas, elas são implementadas no código para retornar o valor de K_t ou K_{ts} de acordo com o tipo de carga atuante, geometria e dimensões do eixo. Depois, são calculados q e q_s , e com isso é possível calcular os valores dos fatores de concentração de tensão a fadiga K_f e K_{fs} .

São definidas as componentes de tensão normal e de cisalhamento, que são necessárias para o cálculo das componentes média e alternada da tensão equivalente de *von Mises*. Também é calculada a tensão máxima de *von Mises*, utilizada para verificar a possibilidade de falha por escoamento no critério de *Goodman* modificado.

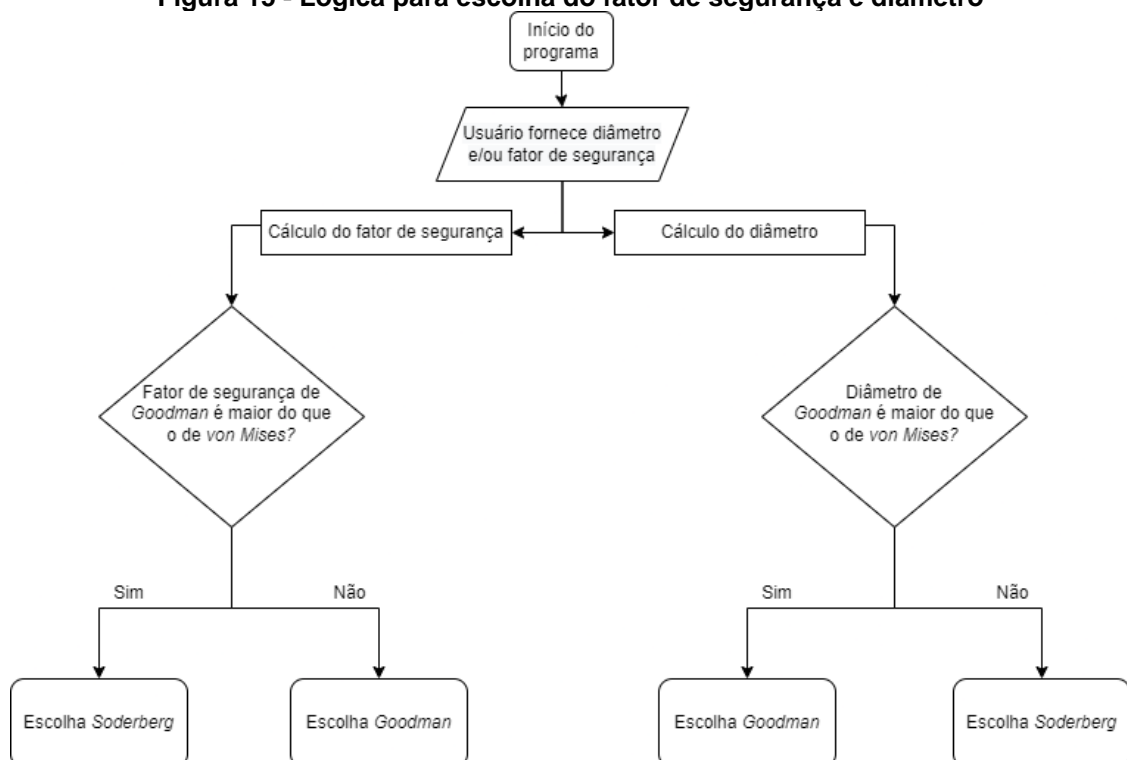
Feito isso, deve-se calcular a resistência à fadiga S_e . Portanto, é utilizado o fator de resistência à fadiga teórico S'_e e são calculados os fatores de correção k_a , k_b , k_c , k_d e k_e utilizando condicionais e funções que dependem dos parâmetros de entrada do programa e das tabelas disponíveis na seção 2.2.1. O valor de k_f será fixado como 1, pois este trabalho visa apenas a fase de projeto da fabricação de um eixo, não

sendo necessário considerar fatores diversos que influenciam na resistência à fadiga, como a corrosão. Com isso, é calculada a resistência à fadiga.

Dessa forma, é possível calcular os fatores de segurança de acordo com os critérios de *Soderberg*, *Goodman*, *von Mises*, *ASME* e *Gerber*. Para definir qual será o critério escolhido, é feita a seguinte comparação: se o fator de segurança de *Goodman* for maior do que o de *von Mises*, escolha *Soderberg*, pois há risco de falha por escoamento no primeiro ciclo; caso contrário, escolha *Goodman*. Em ambos os casos, o usuário recebe o resultado do critério que foi escolhido e a explicação dessa escolha. Os critérios de *ASME* e *Gerber* também serão calculados, porém apenas serão fornecidos ao usuário caso sejam solicitados.

Também é possível que o usuário forneça um valor para o fator de segurança desejado e receba o diâmetro para uma determinada seção do eixo, calculado de acordo com um dos critérios através das mesmas equações, porém dessa vez implementadas em função do diâmetro. O critério de escolha neste caso será semelhante: se o diâmetro de *Goodman* é maior que o de *von Mises*, escolha *Goodman*; caso contrário, escolha *Soderberg*, pois há risco de falha por escoamento. Os diâmetros de *ASME* e *Gerber* também são calculados e podem ser fornecidos ao usuário caso solicitados. Um fluxograma do processo de escolha de diâmetro e fator de segurança pode ser observado na Figura 15.

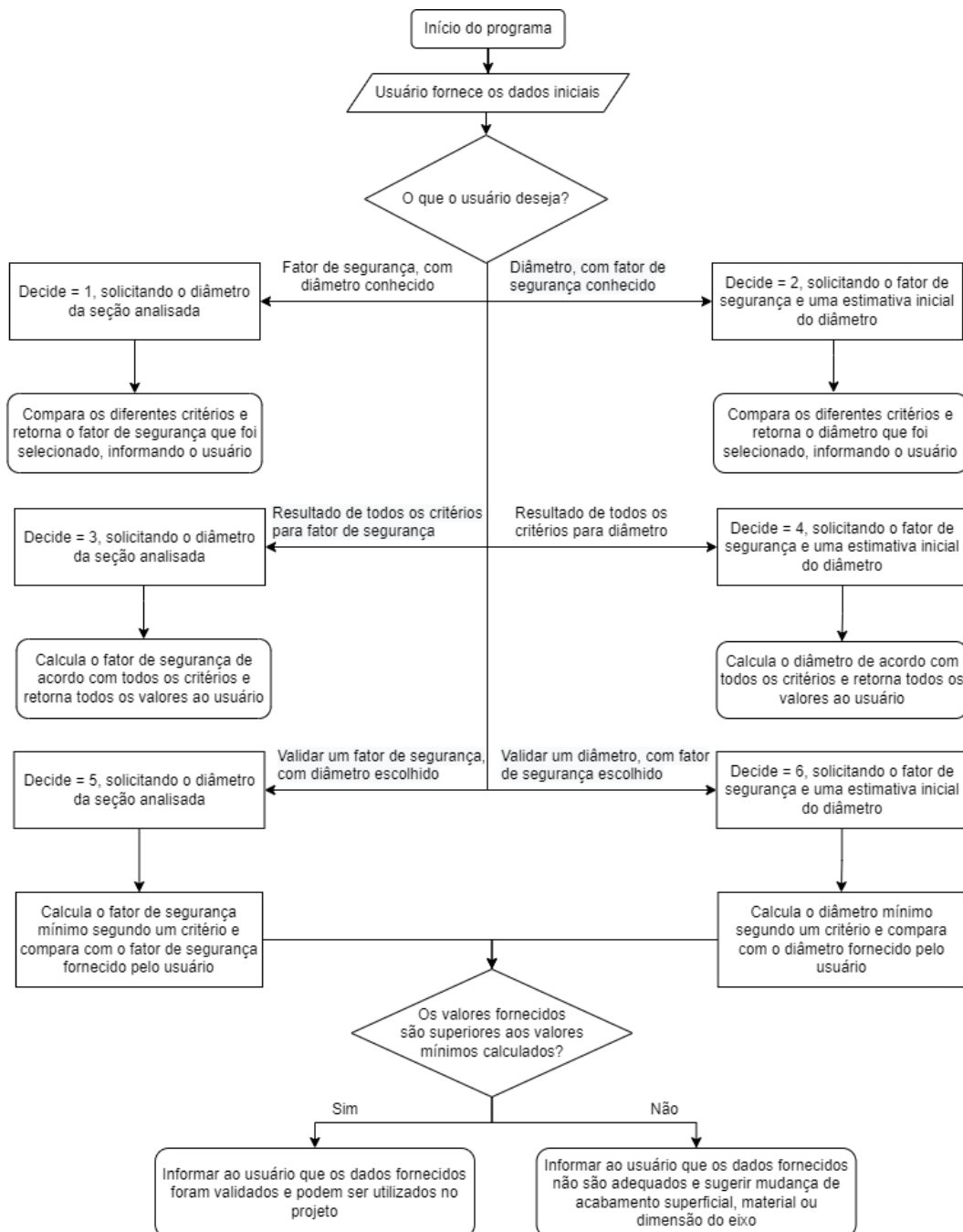
Figura 15 - Lógica para escolha do fator de segurança e diâmetro



Fonte: Autoria própria (2021)

Outra opção do usuário é fornecer um fator de segurança e validar um valor de diâmetro. Caso o diâmetro seja validado, o usuário será informado que o mesmo poderá ser utilizado para o fator de segurança escolhido. Caso contrário, o usuário recebe o diâmetro mínimo necessário e recomendações do programa para atingir o valor desejado. Esse processo ocorre de maneira análoga quando o usuário opta por fornecer um valor fixo de diâmetro e validar um fator de segurança desejado. Assim, existem um total de seis opções, e uma representação esquemática do funcionamento do programa de acordo com a seleção do usuário pode ser visualizada na Figura 16.

Figura 16 - Funcionamento geral do programa



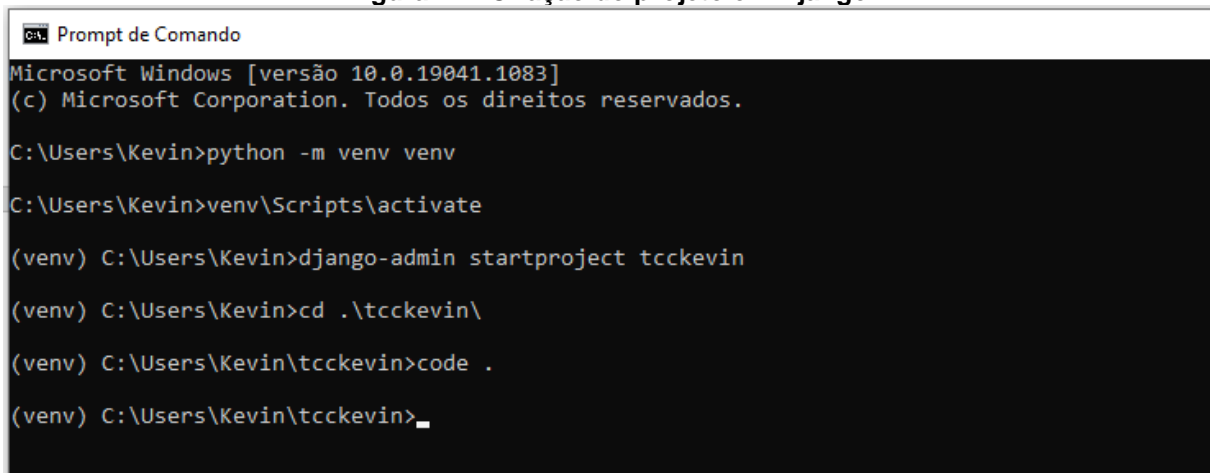
Fonte: Autoria própria (2021)

4 CONCEPÇÃO E DESENVOLVIMENTO DO TRABALHO

4.1 Instalação do Django

Primeiramente, foi criado um ambiente virtual com o Virtualenv, a fim de separar o projeto em um único lugar e não interferir em outros projetos. Para isso, basta digitar “python -m venv venv” no Prompt de Comando, seguido de “venv\Scripts\activate” para entrar no ambiente virtual. Para a elaboração da aplicação web, foi utilizado o *framework web* Django, em sua versão 3.2.5, em conjunto com o Python versão 3.9.6. Para instalar o Django com o Python já instalado no sistema operacional, basta digitar “pip install django” no Prompt de Comando do Windows. Para iniciar o projeto em Django, utilizou-se o comando “django-admin startproject tcckevin”, que cria uma pasta específica chamada “tcckevin” para o projeto que será desenvolvido em Django. Digita-se “cd .\tcckevin\” para entrar na pasta, e “code .” para inicializar o Visual Studio Code, que é o editor de código-fonte que será utilizado para desenvolver o projeto, conforme mostra a Figura 17.

Figura 17 - Criação do projeto em Django



```
Prompt de Comando
Microsoft Windows [versão 10.0.19041.1083]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Kevin>python -m venv venv

C:\Users\Kevin>venv\Scripts\activate

(venv) C:\Users\Kevin>django-admin startproject tcckevin

(venv) C:\Users\Kevin>cd .\tcckevin\

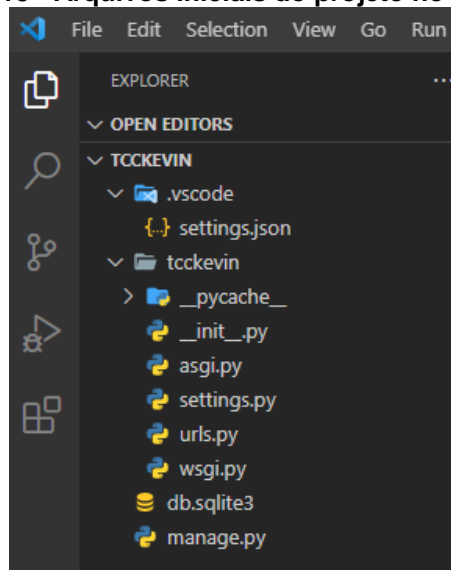
(venv) C:\Users\Kevin\tcckevin>code .

(venv) C:\Users\Kevin\tcckevin>_
```

Fonte: Autoria própria (2021)

Após a inicialização do VS Code, deve-se apertar “F1” e digitar “Python: Select Interpreter” para escolher um interpretador, e seleciona-se o Python disponível no ambiente virtual. Com isso, a versão utilizada do Python deve ficar disponível no canto inferior esquerdo da tela. Para inicializar o servidor, com o Prompt de Comando já na pasta do projeto (ou no próprio terminal do VS Code) digita-se “python manage.py runserver”, e a URL retornada (<http://127.0.0.1:8000/>) já pode ser digitada no navegador de preferência para acessar a aplicação web. Os arquivos do VS Code até aqui devem estar em conformidade com o mostrado na Figura 18.

Figura 18 - Arquivos iniciais do projeto no VS Code



Fonte: Autoria própria (2021)

A partir desse ponto, o projeto já pode começar a ser desenvolvido com Django.

4.2 Elaboração da Interface Web com Django

Para elaborar um formulário web capaz de coletar os dados de entrada utilizando o Django, primeiro abre-se um novo terminal do VS Code e digita-se “django-admin startapp main”. Dentro da pasta “main” que foi criada com esse comando, cria-se um novo arquivo intitulado “urls.py”, que ficará responsável por gerenciar a URL das diferentes páginas da aplicação web (formulário e resultados). O arquivo “urls.py” da pasta “tcckevin” deve então ser modificado de forma a ficar com a estrutura observada na Figura 19.

Figura 19 - urls.py da pasta “tcckevin”

```

16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('', include('main.urls')),
22 ]


```

Fonte: Autoria própria (2021)

O arquivo “urls.py” da pasta “main” ficará responsável por chamar as páginas “index”, “results” e “error”, conforme Figura 20. A página “index”, que conterá o formulário a ser preenchido, poderá ser chamada tanto digitando apenas a URL padrão no navegador “127.0.0.1:8000”, quanto digitando a chamada “/index” após essa mesma URL. A página “results”, que é responsável por mostrar ao usuário os

resultados obtidos, será chamada posteriormente com o auxílio do botão “Enviar” que estará presente ao final do formulário. Essa página também irá conter o botão “Voltar”, que permite o usuário retornar ao formulário inicial após ver os resultados obtidos, e conseqüentemente ingressar novos valores e realizar novos cálculos. Por fim, a página “error” será chamada quando ocorrer alguma exceção no programa, seja pelo preenchimento incorreto do formulário ou algum possível erro que ocorra durante a execução do código.


Figura 20 - urls.py da pasta “main”

```
main >  urls.py > ...
1  from django.urls import path
2  from main import views
3
4  urlpatterns = [
5      path('', views.indexpage, name='index'),
6      path('index', views.indexpage, name='index'),
7      path('results', views.results, name='results'),
8      path('error', views.results, name='error'),
9  ]
```

Fonte: Autoria própria (2021)

No arquivo “views.py” da pasta “main”, definimos as funções “indexpage” e “results” com o argumento “request”, como mostrado na Figura 21.

Figura 21 - views.py da pasta “main”

```
main >  views.py > ...
1  from django.http import response
2
3  from django.shortcuts import render, HttpResponseRedirect
4  from .functions import *
5
6  def indexpage(request):
7      return render(request, template_name='main/index.html')
8
9  def results(request):
10     try:
11         Sit = request.POST['ressalto']
12         Acab = request.POST['acabamento']
```

Fonte: Autoria própria (2021)

Na função “results” é feito o tratamento de exceção, utilizando os parâmetros “try” e “except”. Assim, quando o código é executado corretamente é retornada a página “results” para o usuário, mas quando ocorre algum erro durante sua execução é retornada a página “error”, conforme Figura 22.

Figura 22 - Tratamento de exceção

```

return render(request, "results.html", {"info

# Tratamento de exceção

except:
return render(request, "main/error.html")

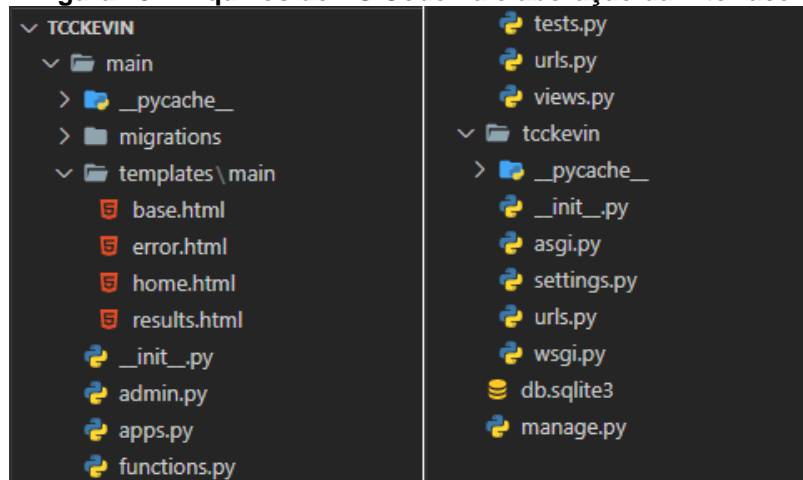
```

Fonte: Autoria própria (2021)

A aplicação “main” que foi criada também deve ser inclusa no arquivo “settings.py” da pasta “tcckevin”. Para isso, em “INSTALLED_APPS = [” adiciona-se o comando “‘main.apps.MainConfig’,”.

Dentro da pasta “main”, cria-se uma nova pasta chamada “templates”, que por sua vez também conterá uma outra pasta chamada “main”. Agora, pode-se criar os arquivos em HTML propriamente ditos das páginas que serão exibidas ao usuário, “base”, “index”, “results” e “error”. Para isso, dentro da pasta “templates/main” que acabou de ser criada, podemos criar os arquivos “base.html”, “index.html”, “results.html” e “error.html”. Neste ponto, a estrutura de arquivos do VS Code deve estar semelhante à Figura 23.

Figura 23 - Arquivos do VS Code na elaboração da interface



Fonte: Autoria própria (2021)


O arquivo “base.html” contém a formatação básica de todas as páginas e o corpo do HTML, sendo posteriormente estendido para as outras páginas utilizando o comando “{% extends ‘main/base.html’ %}”. Assim, também é possível desenvolver o código para as páginas “index”, “results” e “error”. O código de todas as páginas será disponibilizado no *GitHub*. Além do HTML, também foram usadas as linguagens CSS e JavaScript, com o auxílio de bibliotecas como o jQuery.

A linguagem CSS foi utilizada para personalizar o layout da página de maneira geral, centralizando os formulários, atribuindo um novo estilo de fonte, cor de fundo personalizada, posicionamento de imagens, entre outros. Isso faz com que a estética da interface web fique mais agradável ao usuário, tornando-a mais simples e intuitiva.

A linguagem JavaScript foi utilizada para tornar os formulários do site interativos, ou seja, mudarem de acordo com as opções que foram selecionadas previamente pelo usuário. Isso evita o fornecimento de informações irrelevantes e torna a interface mais amigável, já que serão disponibilizados apenas os campos que deverão ser obrigatoriamente preenchidos pelo usuário. Para definir quais campos devem aparecer de acordo com as seleções feitas, foram utilizadas as estruturas condicionais *if*, *else if* e *else*, baseadas nos valores disponíveis nos campos anteriores, em conjunto com os eventos “*hide*” e “*show*” do jQuery.

No final do formulário, foi inserido o botão de “Enviar”. Ao pressionar esse botão, é executada uma ação que redireciona para a página de resultados, contendo o código do arquivo “results.html”. Nessa página, será retornado o resultado dos cálculos de acordo com as informações que foram fornecidas pelo usuário. Também há um botão de “Voltar”, que faz o usuário ser redirecionado para o formulário novamente, podendo repetir esse processo. As interfaces do usuário são apresentadas nas Figuras 24, 25, 26 e 27.

Figura 24 - Início padrão do formulário



Análise de fadiga em eixos de transmissão

Escolha o tipo de geometria no ressalto

Arredondamento no ressalto

Digite o diâmetro da seção vizinha (mm)
 Digite o raio do entalhe (mm)

Escolha o acabamento superficial

Retificado

Escolha a temperatura

20 °C

Escolha a confiabilidade


50%

Escolha o tipo de carga

Flexão pura

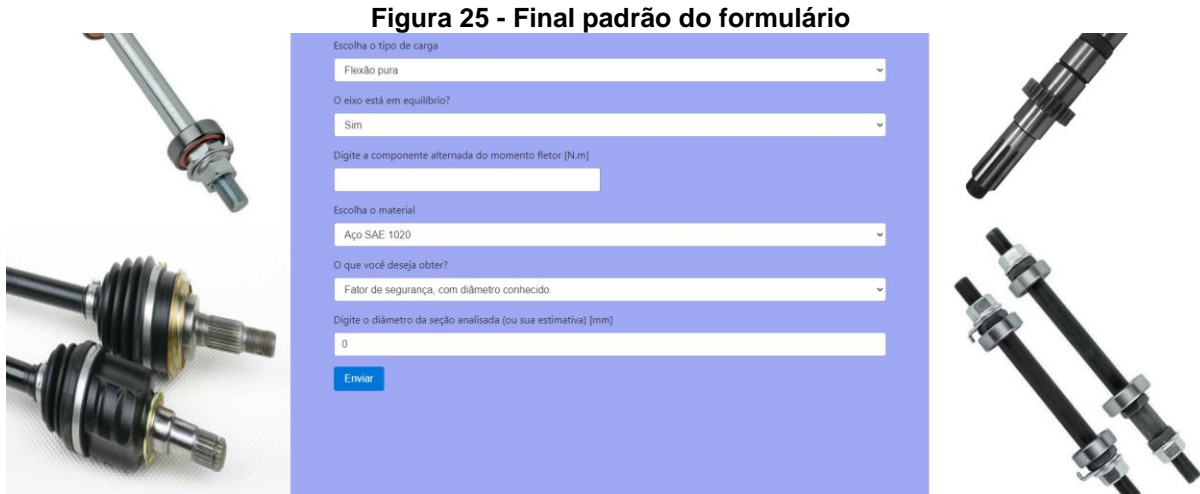
O eixo está em equilíbrio?

Sim



Fonte: Autoria própria (2021)

Figura 25 - Final padrão do formulário



Escolha o tipo de carga
Flexão pura

O eixo está em equilíbrio?
Sim

Digite a componente alternada do momento fletor [N.m]
0

Escolha o material
Aço SAE 1020

O que você deseja obter?
Fator de segurança, com diâmetro conhecido

Digite o diâmetro da seção analisada (ou sua estimativa) [mm]
0

Enviar

Fonte: Autoria própria (2021)

Figura 26 - Página de resultados

Dados de entrada

| | |
|--------------------|--|
| Ressalto | Arredondamento no ressalto |
| Acabamento | Usinado / Laminado a frio |
| Temperatura | 20 °C |
| Confiabilidade | 50% |
| Tipo de carga | Flexão pura |
| Eixo em equilíbrio | Sim |
| Função desejada | Fator de segurança, com diâmetro conhecido |
| Diâmetro | 40.00 mm |
| Fator de segurança | - |
| Material | Aço SAE 1040 |

Resultados

O fator de segurança de 0.50 foi escolhido pelo critério de Goodman. Contudo, este valor é inferior a 1, e portanto a sua utilização não é recomendada. Algumas sugestões para aumentar o fator de segurança são: Aumentar o diâmetro do eixo; Selecionar um material mais resistente; Alterar o tipo de acabamento superficial.

Voltar

Fonte: Autoria própria (2021)

Figura 27 - Página de erro

Erro

Ops! Ocorreu algum erro.

Verifique se todos os campos estão preenchidos e tente novamente.



Voltar

Fonte: Autoria própria (2021)

Os erros podem ocorrer por diversos motivos. Quando algum campo do formulário não é preenchido pelo usuário, quando os números decimais forem escritos com vírgula ao invés de ponto, quando forem preenchidas letras ao invés de números ou quando valores muito fora do padrão forem utilizados, a página de erro é exibida.

4.3 Tratamento dos dados e exibição dos resultados

Para receber e tratar os dados fornecidos pelo usuário, é utilizado o método “POST” na página “index.html” e atribuído um nome para cada campo do formulário. Posteriormente, é utilizado o comando “*request.POST*” dentro da função “results” no arquivo “*views.py*”, atribuindo esses dados à determinadas variáveis. Esse processo é exemplificado nas Figuras 28 e 29.

Figura 28 - Formulário da variável “ressalto”

```
<form class="p-3" action="results" method="POST">
  {% csrf_token %}

  <div class="form-group">
    <label for="pergunta_1"> Escolha o tipo de geometria no ressalto </label>
    <select class="form-control" id="pergunta_1" name="ressalto" value="0">
      <option value="1"> Arredondamento no ressalto </option>
      <option value="2"> Sulco no ressalto </option>
      <option value="3"> Canal de fundo plano </option>
      <option value="4"> Assento de chaveta de extremidade fresada r/d = 0,02 </option>
    </select>
  </div>
```

Fonte: Autoria própria (2021)

Figura 29 - Atribuição dos dados do formulário a variáveis

```
def results(request):
    try:
        Sit = request.POST['ressalto']
        Acab = request.POST['acabamento']
        d = request.POST['d_inicial']
        D = request.POST['d_vizinha']
        a = request.POST['l_canal']
        t = request.POST['p_canal']
        dinside = request.POST['p_sulco']
        r_entalhe1 = request.POST['r_entalhe1']
        r_entalhe2 = request.POST['r_entalhe2']
        r_entalhe3 = request.POST['r_entalhe3']
        Temp = request.POST['temperatura']
        conf = request.POST['confiabilidade']
        carga = request.POST['carga']
```

Fonte: Autoria própria (2021)

Com essas variáveis, é possível definir diversas funções dentro do arquivo “*functions.py*”, que serão responsáveis por realizar todos os cálculos e retornar os

resultados. Assim, toda a parte da coleta e tratamento de dados fica localizada na função “results” do arquivo “views.py” e no arquivo “functions.py”. A chamada das funções e o retorno do resultado final pode ser visualizada na Figura 30.

Figura 30 - Chamada das funções e cálculo do resultado

```

q1 = q(Sut, r)
qs1 = qs(Sut, r)
Kt1 = Kt(Sit, in1, in2)
Kts1 = Kts(Sit, in1, in2)
Kf1 = Kf(Sut, r, Sit, in1, in2)
Kfs1 = Kfs(Sut, r, Sit, in1, in2)
σa1 = σa(Kf1, Ma, d)
σm1 = σm(Kf1, Mm, d)
τa1 = τa(Kfs1, Ta, d)
τm1 = τm(Kfs1, Tm, d)
σa11 = σa1(σa1, τa1)
σm11 = σm1(σm1, τm1)
σmax1 = σmax(σa1, σm1, τa1, τm1)
Se1 = Se1(Sut)
ka1 = ka(Sut, Acab)
kb1 = kb(d)
kc1 = kc(carga)
kd1 = kd(Temp)
ke1 = ke(conf)
Se1 = Se(Sut, Acab, d, carga, conf, Temp)
ηvm1 = ηvm(σmax1, Sy)
AGerber1 = AGerber(Kf1, Ma, Kfs1, Ta)
BGerber1 = BGerber(Kf1, Mm, Kfs1, Tm)

# Resultado final

result = resultado(decide, Sit, Se1, Sy, Sut, r, D, d, a, t, Kf1, Kfs1, Ma,
return render(request, "main/results.html", {"info_ressalto": info_ressalto

```

Fonte: Autoria própria (2021)

Como o arquivo “views.py” importa todas as funções contidas no “functions.py”, essas funções também podem ser chamadas dentro da própria função “results”, atribuindo o resultado final à uma variável que posteriormente será exibida na página de resultados, junto com os dados de entrada utilizados e outras informações de interesse.

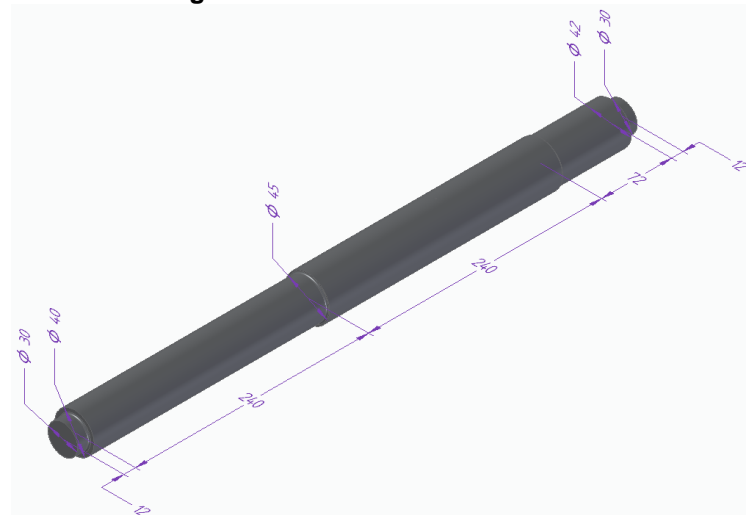
Um manual de uso para o site desenvolvido pode ser visto no Apêndice A.

4.4 Hospedagem da aplicação web em nuvem

Após a finalização de todo o projeto, foi realizado o *upload* do mesmo na plataforma *GitHub*. Assim, todos os códigos encontram-se disponíveis em github.com/Kvgr98/TCC-Kevin. Após a disponibilização do projeto, foi utilizada a plataforma em nuvem Heroku para realizar o *deploy* da aplicação com hospedagem

(ambiente). Como o eixo possui rotação constante, sabe-se que ele está em equilíbrio. Também não existe torção atuante, então a única carga presente no exercício é de flexão, havendo apenas a componente alternada do momento fletor (M_a). Para conseguir obter o valor de M_a nos diferentes pontos de concentração de tensão e verificar qual é o ponto mais crítico, o eixo foi modelado no Solid Edge, conforme representado na Figura 32.

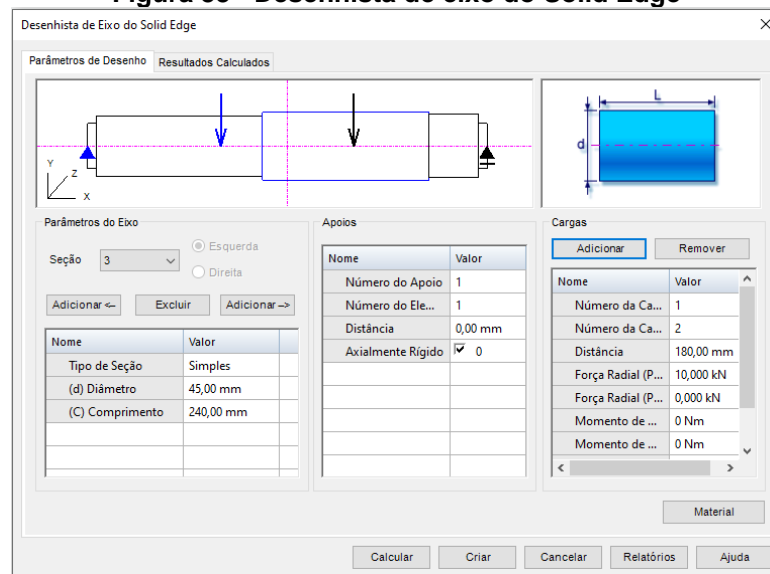
Figura 32 - Eixo modelado no CAD



Fonte: Autoria própria (2021)

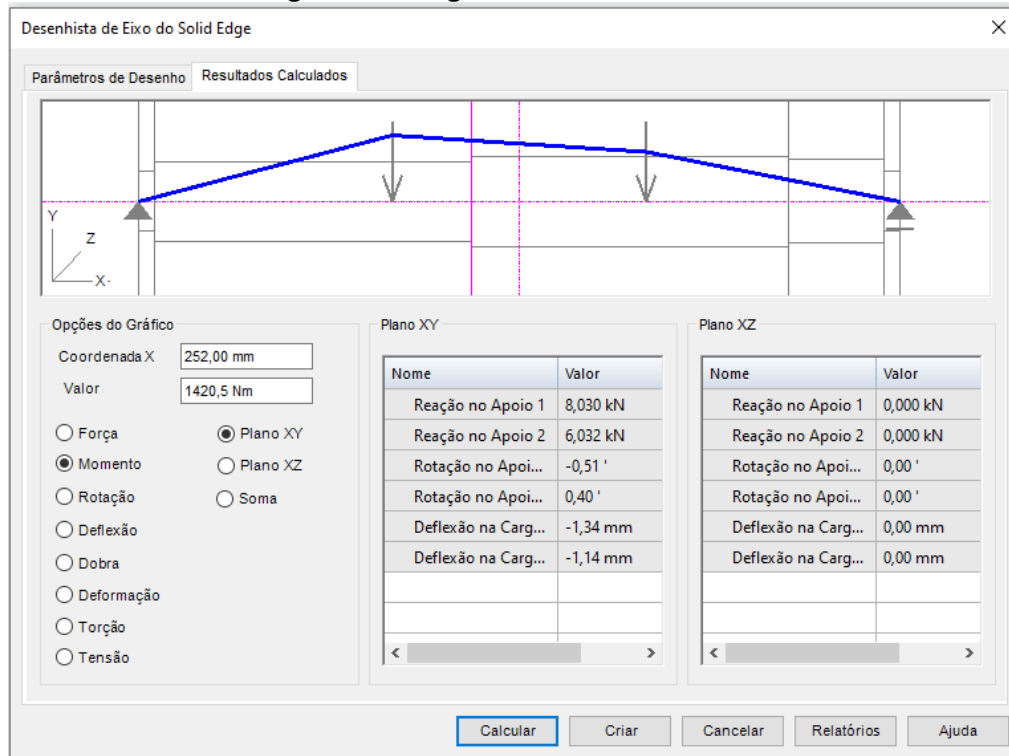
Assim, foi utilizado o desenhista de eixo do Solid Edge com as informações provenientes do enunciado para obter-se o diagrama de esforços internos para todos os pontos críticos, conforme mostrado nas Figuras 33 e 34.

Figura 33 - Desenhista de eixo do Solid Edge



Fonte: Autoria própria (2021)

Figura 34 - Diagrama de momento fletor



Fonte: Autoria própria (2021)

A partir do diagrama de momento fletor obtido, é possível encontrar os valores da componente alternada do momento fletor para os quatro pontos onde existe ressalto, mostrados na Tabela 6.

Tabela 6 - Valores de M_a nos pontos críticos

| Ponto | Distância do início do eixo [mm] | Diâmetro [mm] | Diâmetro da seção vizinha [mm] | Concentrador de tensão | M_a [Nm] |
|-------|----------------------------------|---------------|--------------------------------|--------------------------|------------|
| A | 12 | 30 | 40 | Ressalto ($r = 1,5mm$) | 96,4 |
| B | 252 | 40 | 45 | Ressalto ($r = 1,5mm$) | 1420,5 |
| C | 492 | 42 | 45 | Ressalto ($r = 1,5mm$) | 506,36 |
| D | 564 | 30 | 42 | Ressalto ($r = 1,5mm$) | 72,4 |

Fonte: Autoria própria (2021)

Assim, utilizando os dados disponíveis, é possível calcular o fator de segurança para cada ponto crítico utilizando a aplicação web criada. Os resultados obtidos para o fator de segurança em cada ponto estão na Tabela 7.

Tabela 7 - Fatores de segurança obtidos

| Ponto | Fator de segurança | Critério selecionado |
|-------|--------------------|----------------------|
| A | 3,25 | Goodman |
| B | 0,50 | Goodman |
| C | 1,64 | Goodman |
| D | 4,28 | Goodman |

Fonte: Autoria própria (2021)

Nota-se que o ponto B é o ponto mais crítico do eixo, com um fator de segurança de 0,5 selecionado pelo critério de *Goodman* e validado pelo critério de *von*

Mises. Assim, ocorrerá a fratura por fadiga neste ponto após um determinado número de ciclos, visto que o fator de segurança calculado é inferior a 1. Como esse fator de segurança não garante a vida infinita do eixo, será necessário alterar os parâmetros de projeto para obter um fator de segurança aceitável.

As sugestões provenientes da aplicação web para aumentar o valor do fator de segurança podem ser observadas na Figura 35.

Figura 35 - Sugestões do programa para aumentar o fator de segurança

Resultados

O fator de segurança de 0.50 foi escolhido pelo critério de Goodman. Contudo, este valor é inferior a 1, e portanto a sua utilização não é recomendada. Algumas sugestões para aumentar o fator de segurança são: Aumentar o diâmetro do eixo; Selecionar um material mais resistente; Alterar o tipo de acabamento superficial.

[Voltar](#)

Fonte: Autoria própria (2021)

Alterando o acabamento superficial do eixo para os outros valores, são obtidos os resultados de fator de segurança no ponto crítico conforme a Tabela 8.

Tabela 8 - Fator de segurança no ponto crítico para diferentes acabamentos

| Acabamento superficial | Fator de segurança | Critério selecionado |
|---------------------------|--------------------|----------------------|
| Retificado | 0,55 | <i>Goodman</i> |
| Usinado / Laminado a frio | 0,50 | <i>Goodman</i> |
| Laminado a quente | 0,35 | <i>Goodman</i> |
| Forjado | 0,29 | <i>Goodman</i> |

Fonte: Autoria própria (2021)

Ou seja, o maior fator de segurança com a troca do acabamento superficial foi de 0,55, obtido com o eixo retificado, que ainda é um valor muito baixo. Acatando outra sugestão, de realizar a troca do material por um mais resistente, podemos inserir valores superiores de tensão última, obtendo os resultados da Tabela 9.

Tabela 9 - Fator de segurança no ponto crítico para diferentes materiais e eixo laminado a frio

| Tensão última (S_{ut}) | Acabamento superficial | Fator de segurança | Critério selecionado |
|----------------------------|---------------------------|--------------------|----------------------|
| 1000 MPa | Usinado / Laminado a frio | 0,69 | <i>Goodman</i> |
| 1100 MPa | Usinado / Laminado a frio | 0,73 | <i>Goodman</i> |
| 1200 MPa | Usinado / Laminado a frio | 0,78 | <i>Goodman</i> |
| 1300 MPa | Usinado / Laminado a frio | 0,82 | <i>Goodman</i> |
| 1400 MPa | Usinado / Laminado a frio | 0,86 | <i>Goodman</i> |

Fonte: Autoria própria (2021)

Vale ressaltar que *Goodman* é o critério selecionado em todos os casos pois o fator de segurança obtido pelo critério de *von Mises* é superior, devido à tensão de escoamento utilizada. Caso a tensão de escoamento do material escolhido seja muito baixa, ocorrerá a falha por escoamento no primeiro ciclo, e portanto o critério de *Soderberg* seria selecionado. Ainda assim, mesmo com uma tensão última muito

elevada, de 1400 MPa, o resultado para fator de segurança obtido foi de 0,86, que ainda não é o suficiente. Porém, ao utilizar os mesmos valores elevados de tensão última combinados com um eixo de acabamento superficial retificado, obtemos os valores mostrados na Tabela 10.

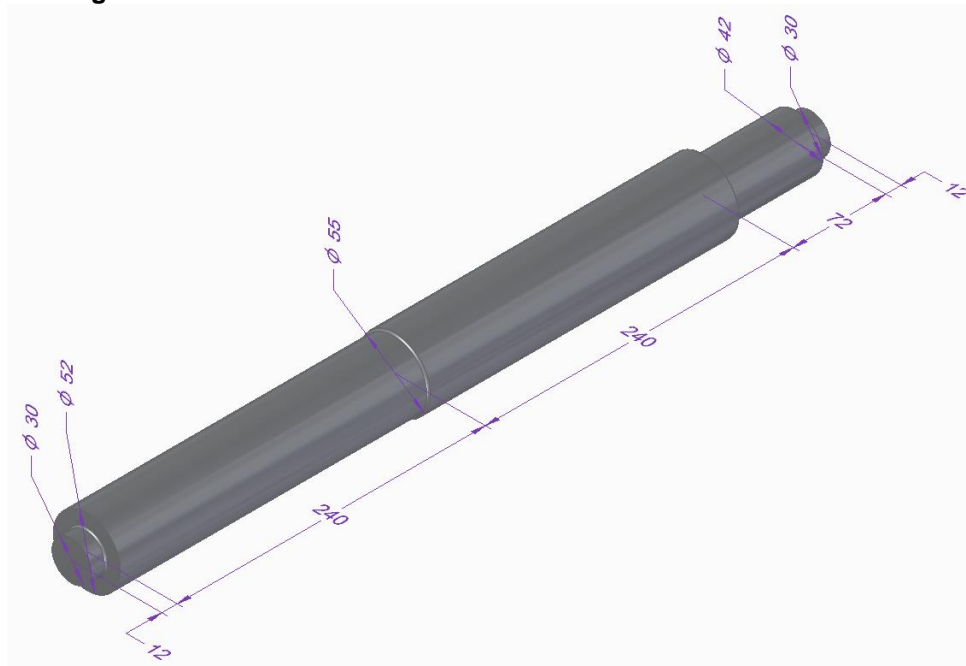
Tabela 10 - Fator de segurança no ponto crítico para diferentes materiais e eixo retificado

| Tensão última (S_{ut}) | Acabamento superficial | Fator de segurança | Critério selecionado |
|----------------------------|------------------------|--------------------|----------------------|
| 1000 MPa | Retificado | 0,84 | Goodman |
| 1100 MPa | Retificado | 0,91 | Goodman |
| 1200 MPa | Retificado | 0,97 | Goodman |
| 1300 MPa | Retificado | 1,04 | Goodman |
| 1400 MPa | Retificado | 1,11 | Goodman |

Fonte: Autoria própria (2021)

Então, uma das opções para que não ocorra a falha por fadiga nesse eixo seria trocar seu acabamento superficial e seu material por um mais resistente, mantendo suas dimensões originais. Como essa alternativa poderia influenciar no preço, uma outra opção para evitar a falha seria alterar os valores de diâmetro do eixo, caso isso seja possível na aplicação em questão, mantendo o mesmo acabamento superficial e material originais. Alterando o diâmetro da seção de 40 mm para 52 mm e o diâmetro de 45 mm para 55 mm, por exemplo, conforme a Figura 36, temos os resultados da Tabela 11 para todos os pontos críticos do eixo.

Figura 36 – Eixo modelado no CAD com novos valores de diâmetro



Fonte: Autoria própria (2021)

Tabela 11 - Fatores de segurança nos pontos críticos para novos valores de diâmetro

| Ponto | Diâmetro [mm] | Diâmetro da seção vizinha [mm] | Concentrador de tensão | M_a [Nm] | Fator de segurança |
|-------|---------------|--------------------------------|--------------------------|------------|--------------------|
| A | 30 | 52 | Ressalto ($r = 1,5mm$) | 96,4 | 3,12 |
| B | 52 | 55 | Ressalto ($r = 1,5mm$) | 1420,5 | 1,28 |
| C | 42 | 55 | Ressalto ($r = 1,5mm$) | 506,36 | 1,52 |
| D | 30 | 42 | Ressalto ($r = 1,5mm$) | 72,4 | 4,28 |

Fonte: Autoria própria (2021)

Essa alteração nos valores de diâmetro também resulta em fatores de segurança superiores a 1 para todos os pontos críticos, ou seja, não ocorreria a fratura por fadiga. Contudo, o ponto mais crítico possui um fator de 1,28, tendo apenas uma margem de erro de 28% para a prevenção da falha. Neste caso, seria recomendado aumentar ainda mais os valores de diâmetro, quando possível, para obter fatores de segurança superiores a 2, conforme sugerido na literatura clássica.

4.6 Segundo estudo de caso

Para o segundo estudo de caso, tem-se um eixo usinado e carregado com flexão e torção, que não se encontra em equilíbrio. Os valores dados das componentes alternadas e média de momento fletor e de torque são $M_a = 70 N.m$, $T_a = 45 N.m$, $M_m = 55 N.m$ e $T_m = 35 N.m$. Para o eixo, é considerada uma tensão última de $S_{ut} = 700 MPa$, uma tensão de escoamento de $S_y = 560 MPa$ e geometria no ressalto com assento de chaveta. Utilizando um fator de segurança de 2, pede-se o diâmetro mínimo aceitável para esse eixo, utilizando os critérios de *Goodman*, *Soderberg*, *Gerber* e ASME-elíptico. Também pede-se para discutir e comparar os resultados obtidos.

Ao ingressar os dados de entrada do enunciado no *site*, e considerando uma temperatura ambiente de 20°C com uma confiabilidade de 50%, obtém-se os resultados disponíveis na Tabela 12.

Tabela 12 - Diâmetros calculados para todos os critérios

| Critério | Estimativa inicial de diâmetro do eixo [mm] | Concentrador de tensão | Diâmetro [mm] |
|------------------|---|---|---------------|
| <i>Goodman</i> | 25 | Assento de chaveta de extremidade fresada | 25,65 |
| <i>Soderberg</i> | 25 | Assento de chaveta de extremidade fresada | 26,10 |
| <i>Gerber</i> | 25 | Assento de chaveta de extremidade fresada | 24,21 |
| ASME | 25 | Assento de chaveta de extremidade fresada | 24,11 |
| <i>von Mises</i> | 25 | Assento de chaveta de extremidade fresada | 21,77 |

Fonte: Autoria própria (2021)

Comparando os resultados encontrados para os diferentes critérios, é possível perceber que os critérios de *Gerber* e *ASME* são os menos conservadores, conforme comentado anteriormente. Assim, estes critérios oferecem um menor diâmetro aceitável do eixo, com o benefício da possível utilização de menos material, em troca de um maior risco de ocorrência da falha por fadiga.

O critério de *von Mises* resultou em um diâmetro de $21,77\text{ mm}$, que é o diâmetro mínimo para que não ocorra a falha por escoamento no primeiro ciclo, considerando um fator de segurança de 2. Como o diâmetro de *Goodman* resulta em $25,65\text{ mm}$, que é um valor superior, ele é aprovado pelo critério de *von Mises* e pode ser utilizado com a garantia de que não ocorrerá a falha por escoamento.

Por fim, nota-se que o critério de *Soderberg* é o mais conservador de todos, resultando em um maior diâmetro e oferecendo maior segurança, em troca da utilização de mais material do eixo.

5 CONCLUSÃO

Neste trabalho, o código desenvolvido por Carvalho Neto (2020) para análise de fadiga em eixos de transmissão foi modificado, simplificado e adaptado para funcionar como uma aplicação web em nuvem. Para isso, o *framework web* Django foi utilizado com o intuito de facilitar essa implementação através de uma interface gráfica amigável para o usuário utilizando a linguagem HTML, através de um formulário interativo que oculta as informações redundantes e mostra apenas o necessário, graças ao JavaScript. Os dados inseridos no formulário são então requisitados pelo arquivo em Python e utilizados para o cálculo do resultado final, que é exibido para o usuário na página de resultados.

Os estudos de caso realizados comprovaram a funcionalidade e praticidade do *site* desenvolvido, que permite a modificação de parâmetros de forma simples e direta na fase de projeto de um eixo. Isso poupa ao usuário todo o tempo necessário para realizar esses cálculos manualmente. Todos os objetivos iniciais do trabalho foram atendidos, visto que foi desenvolvido um sistema especialista que atua como uma aplicação web em nuvem, podendo ser facilmente acessada por qualquer pessoa que tenha acesso à internet, no computador ou no celular, através da URL tcc-kevin.herokuapp.com. Todos os códigos e arquivos utilizados no desenvolvimento do trabalho podem ser obtidos em github.com/Kvgr98/TCC-Kevin. Além disso, uma versão atualizada do site, que poderá ser melhorada por futuros contribuintes, também pode ser acessada em tcc-kevin-atualizado.herokuapp.com, com o seu código atualizado disponível em github.com/TCCKevin/TCC-Kevin.

Optou-se por deixar a interface gráfica da aplicação web sem imagens, visto que essas inviabilizam o seu uso em celulares ou em qualquer outro dispositivo com uma resolução abaixo de Full HD, devido à problemas de adaptação dos *pixels*, de forma que o formulário não pode ser preenchido. Assim, as imagens constam apenas como comentários no código, no início do arquivo HTML referente ao formulário, e uma sugestão de trabalho futuro é resolver esse problema e conseguir tornar as imagens adaptativas, possibilitando seu uso independentemente da resolução do dispositivo que está acessando o *site*.

Os trabalhos futuros que utilizarem este como base também poderão facilitar a reinserção dos dados de entrada na própria página de resultados do *site*, utilizando um banco de dados para facilitar a mudança dos parâmetros de projeto sem a

necessidade de preencher todos os campos novamente. Além disso, poderão exibir um comparativo de resultados com diferentes dados de entrada, melhorar a interface gráfica da aplicação web, aumentar o banco de dados de materiais, especificar melhor os tipos de exceções que podem ter ocorrido durante uma falha no código e adicionar novas possibilidades de cálculos para outros elementos mecânicos, além de eixos de transmissão.

Diversos métodos adicionais também podem ser implementados no código, como a identificação do número de ciclos em que ocorrerá a falha por fadiga, fornecendo essa informando ao usuário nas ocasiões em que o eixo não alcance vida infinita. Por fim, também seria possível transformar o programa desenvolvido em um sistema embarcado capaz de monitorar eixos de transmissão durante o seu processo de fabricação, utilizando sensores e um dispositivo como o Raspberry Pi para realizar a alteração de determinados parâmetros de projeto em tempo real.

REFERÊNCIAS

- ADEKUNLE, A. A.; ADEJUYIGBE, S. B.; ARULOGUN, O. T. International Conference on Modeling, Optimization and Computing (ICMOC 2012) Development of CAD Software for Shaft under various Loading Conditions. **Procedia Engineering**, v. 38, p. 1962–1983, 2012.
- AMERICAN SOCIETY OF MECHANICAL ENGINEERS. **ASME B106.1M-1985: Design of Transmission Shafting**. New York, 1985.
- ANDERSON, T. L. **Fracture Mechanics: fundamentals and applications**. 3. ed. Boca Ratón: CRC Press, 2005.
- AZEVEDO, T. F. et al. Failure analysis of a sugarcane loader rear shaft. **Engineering Failure Analysis**, v. 109, p. 104326, 1 jan. 2020.
- AZURE. **O que é computação em nuvem?** Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/#uses>>. Acesso em: 19 dez. 2021.
- BARATA, D. et al. System of Acquisition, Transmission, Storage and Visualization of Pulse Oximeter and ECG Data Using Android and MQTT. **Procedia Technology**, v. 9, p. 1265–1272, 1 jan. 2013.
- BUDYNAS, R. G.; NISBETT, J. K. **Elementos de Máquinas de Shigley**. 10. ed. Porto Alegre: Grupo A, 2016.
- CALLISTER JR, W. D. **Ciência e Engenharia dos Materiais: Uma Introdução**. 1. ed. Rio de Janeiro: LTC, 2002.
- CARVALHO NETO, A. P. **Automatização de análise de fadiga em eixos de transmissão para tomada de decisões na fase de projeto**. 2020. 166 f. Trabalho de Conclusão de Curso para obtenção de título de Bacharel em Engenharia Mecânica – Universidade Tecnológica Federal do Paraná. Guarapuava, 2020.
- CHAUDHARY, S. K.; RAJAK, A. K.; ASHISH, K. Failure analysis of rear axle shaft of a heavy vehicle. **Materials Today: Proceedings**, v. 38, p. 2235–2240, 1 jan. 2020.
- COLLINS, J. A.; BUSBY, H. R.; STAAB, G. H. **Projeto Mecânico de Elementos de Máquinas: Uma Perspectiva de Prevenção da Falha**. São Paulo: Grupo GEN, 2006.
- DJANGO. Disponível em: <<https://www.djangoproject.com/>>. Acesso em: 3 abr. 2021.
- FANGOHR, H. Introduction to Python for Computational Science and Engineering. **Environment**, p. 1–148, 2019.
- FISHER, A. E.; GRODZINSKY, F. S. **The Anatomy of Programming Languages**. Englewood Cliffs: Prentice Hall, 1993.
- FLORES, C. D. **Fundamentos dos Sistemas Especialistas**. Porto Alegre: Bookman, 2003.

GITHUB. Disponível em: <<https://www.github.com/>>. Acesso em: 5 dez. 2021.

GOMES, P. C. T. **Principais Protocolos de Rede**. Disponível em: <<https://www.opservices.com.br/protocolos-de-rede/>>. Acesso em: 29 abr. 2021.

GUIMARÃES, A. L. **O que são frameworks para desenvolvimento WEB**. Disponível em: <<https://blog.configur.com/frameworks-para-desenvolvimento-web/>>. Acesso em: 3 abr. 2021.

HANSEN, E. B.; IFTIKHAR, N.; BOGH, S. Concept of easy-to-use versatile artificial intelligence in industrial small & medium-sized enterprises. **Procedia Manufacturing**, v. 51, p. 1146–1152, 1 jan. 2020.

HEROKU. Disponível em: <https://www.heroku.com/>. Acesso em: 11 dez. 2021.

HIBBELER, R. C. **Resistência dos Materiais**. 7. ed. São Paulo: Pearson, 2010.

JUVINALL, R. C.; MARSHEK, K. M. **Fundamentos do Projeto de Componentes de Máquinas**. 5. ed. São Paulo: Grupo GEN, 2016.

KÖRBER, M.; FROMMEL, C. Automated planning and optimization of a draping processes within the CATIA environment using a python software tool. **Procedia Manufacturing**, v. 38, p. 808–815, 2019.

MATWEB. Disponível em: <<http://www.matweb.com/>>. Acesso em: 27 nov. 2021.

MOTT, R. L. **Elementos de Máquina em Projetos Mecânicos**. 5. ed. São Paulo: Pearson, 2015.

NORTON, R. L. **Projeto de Máquinas: Uma Abordagem Integrada**. 4. ed. Porto Alegre: Grupo A, 2013.

REIFF, C. et al. Web-based Platform for Data Analysis and Monitoring. **Procedia CIRP**, v. 86, p. 31–36, 1 jan. 2020.

ROBINSON, D. **The Incredible Growth of Python**. Disponível em: <<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>>. Acesso em: 26 mar. 2021.

SIEMENS. **Solid Edge**. Disponível em: <<https://solidedge.siemens.com/pt-br/>>. Acesso em: 25 mar. 2021.

VASCONCELOS, V.V.; MARTINS JUNIOR, P.P. **Protótipo de Sistema Especialista em Direito Ambiental para Auxílio à decisão em Situações de Desmatamento Rural**. NT-27. CETEC-MG. 2004. 80p.

WANG, M. S. et al. Failure analysis of screw shaft in screw compressor. **Engineering Failure Analysis**, v. 125, p. 105424, 1 jul. 2021.

WATERMAN, D.A. **A Guide to Expert Systems**. Addison-Wesley, 1986.

ZETTLER, Kev. **O que é computação em nuvem? Uma visão geral da nuvem.** Disponível em: <<https://www.atlassian.com/br/continuous-delivery/principles/cloud-computing>>. Acesso em: 19 dez. 2021.

ZHOU, H. B. et al. Research on Circular Area Search algorithm of multi-robot service based on SOA cloud platform. **Applied Soft Computing Journal**, v. 88, p. 105816, 1 mar. 2020.

APÊNDICE A - MANUAL DE USO

Ao acessar o site, o usuário deve preencher o formulário com todos os dados de entrada necessários, como segue:

1 – Escolha o tipo de geometria no ressalto. Dependendo da opção de geometria selecionada aqui, aparecerão novos campos para o preenchimento das dimensões.

2 – Escolha o acabamento superficial. A escolha do acabamento influencia no valor do fator de superfície k_a , conforme mostrado na Tabela 1.

3 – Escolha a temperatura. Essa escolha influencia no fator de temperatura k_d , conforme Tabela 3. Para eixos em temperatura ambiente, o valor padrão utilizado é de 20 °C.

4 – Escolha a confiabilidade. Essa escolha influencia no fator de confiabilidade k_e , conforme Tabela 4. O valor padrão utilizado aqui é de 50%.

5 – Escolha o tipo de carga. Aqui, o usuário deve escolher o tipo de carga a qual o eixo está submetido. Essa escolha influencia no fator de carga k_c , conforme Tabela 2, e afetará as opções de componentes médias e alternadas de momento fletor e torque que serão exibidas posteriormente.

6 – O eixo está em equilíbrio? Se o eixo estiver atuando em rotação constante, ele está em equilíbrio. Essa escolha também afeta as opções de componentes médias e alternadas de momento fletor e torque que serão preenchidas, juntamente com o tipo de carga selecionado.

Os valores de momento fletor e torque devem ser preenchidos em termos de tensões flutuantes, ou seja, através de componentes médias e alternadas, conforme representado graficamente na Figura 9. Para isso, é necessário obter os diagramas de esforços internos do eixo, o que pode ser feito utilizando o método das seções da mecânica dos sólidos, ou através de um *software* para realizar o procedimento automaticamente e retornar o diagrama, obtendo os valores nos pontos críticos.

7 – Escolha o material. O usuário tem a opção de selecionar materiais com valores de tensão última e de escoamento pré-estabelecidos, ou preencher esses valores por conta própria selecionando a opção “Digitar os valores de tensão manualmente”. Nos casos em que os valores exatos de tensão forem conhecidos, a última opção é mais recomendada.

8 – O que você deseja obter? Aqui, o usuário seleciona a opção de cálculo desejada, conforme Figura 16. As opções 1 e 2 irão calcular o fator de segurança ou diâmetro automaticamente e informar ao usuário. As opções 3 e 4 retornarão o resultado de fator de segurança ou diâmetro de acordo com todos os critérios apresentados. As opções 5 e 6 irão solicitar os valores desejados de fator de segurança ou diâmetro e compará-los com os valores calculados, informando ao usuário se os valores desejados podem ser utilizados.

Todos os campos exibidos no formulário devem ser preenchidos. Ao preencher valores decimais, deve-se utilizar ponto ao invés de vírgula. Se algum campo estiver em branco, for preenchido incorretamente ou ocorrer algum erro inesperado na execução do código, a página de erro será exibida.