

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DIOVANNI MIOTTO

**AUTOMATIZAÇÃO DA ANÁLISE DE CRATERAMENTO EM  
ENGRENAGENS CILÍNDRICAS DE DENTES RETOS**

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA  
2023

DIOVANNI MIOTTO

**AUTOMATIZAÇÃO DA ANÁLISE DE CRATERAMENTO EM  
ENGRENAGENS CILÍNDRICAS DE DENTES RETOS**

**AUTOMATION OF CRATERATION ANALYSIS IN CYLINDRICAL  
GEARS OF STRAIGHT TEETH**

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

Orientador: David Lira Nunez

GUARAPUAVA

2023



[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.

DIOVANNI MIOTTO

## **AUTOMATIZAÇÃO DA ANÁLISE DE CRATERAMENTO EM ENGRENAGENS CILÍNDRICAS DE DENTES RETOS**

Este Trabalho de Conclusão de Curso foi apresentado como requisito parcial para a obtenção do título de Bacharel em Engenharia Mecânica. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Data de aprovação: 11/Maio/2023

---

David Lira Nunez  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Andres Jesse Porfirio  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Renata Luiza Stange Carneiro Gomes  
Doutorado  
Universidade Tecnológica Federal do Paraná

GUARAPUAVA  
2023

## **AGRADECIMENTOS**

Gostaria de expressar minha profunda gratidão e reconhecimento a todas as pessoas que me ajudaram a concluir meu trabalho de conclusão de curso. Primeiramente, gostaria de agradecer ao meu pai, Dacir Miotto, e à minha mãe, Simone Miotto, pelo amor incondicional e apoio constante durante toda a minha trajetória acadêmica. Seu apoio emocional e incentivo foram fundamentais para eu seguir em frente e concluir esta etapa da minha vida.

Também gostaria de agradecer ao meu orientador, David Lira Nunez, por sua orientação valiosa, paciência e comprometimento em ajudar-me a desenvolver meu projeto. Sua expertise e insights foram essenciais para o sucesso do meu trabalho.

Agradeço ainda à Universidade Tecnológica Federal do Paraná (UTFPR) pela oportunidade de estudar em uma instituição de ensino reconhecida e com um corpo docente de excelência. Agradeço aos professores e demais membros do corpo docente que me ensinaram ao longo dos anos, compartilhando seus conhecimentos e experiências.

Gostaria de expressar minha gratidão aos meus amigos, que me acompanharam nesta jornada desafiadora e me apoiaram incondicionalmente. Também gostaria de agradecer a todos aqueles que, embora não tenham sido mencionados nominalmente, contribuíram com seu apoio e incentivo ao longo do caminho.

A todos vocês, meu mais sincero obrigado!

## RESUMO

Engrenagens são essenciais em sistemas mecânicos e podem ser projetadas para diversos fins. Para evitar falhas, é importante seguir as recomendações da norma AGMA (American Gear Manufacturers Association). Programas computacionais são eficazes no desenvolvimento de elementos mecânicos e podem simplificar o estudo dos mesmos. Ferramentas computacionais podem ser úteis na tomada de decisão em áreas específicas da engenharia mecânica, como no cálculo de engrenagens cilíndricas de dentes retos. Este trabalho tem como objetivo a automatização da análise de crateramento em engrenagens cilíndricas de dentes retos, por meio do desenvolvimento de uma ferramenta computacional em Python. Foi realizada uma revisão bibliográfica sobre o assunto e a implementação dos cálculos com base na norma ANSI/AGMA, seguida do desenvolvimento de uma plataforma web utilizando o Framework Django. A ferramenta desenvolvida se mostrou competente em sua finalidade, sendo capaz de realizar o cálculo da tensão ao contato AGMA e ainda sugerir diferentes durezas para que o fator de segurança normalizado seja igual a 1. Dessa forma, o usuário tem a possibilidade de escolher a melhor opção para sua aplicação com base nas sugestões fornecidas pela ferramenta. **Palavras-chave:** Python; engrenagens; Django; análise de crateramento.

## ABSTRACT

Gears are essential in mechanical systems and can be designed for various purposes. To avoid failures, it is important to follow the recommendations of the AGMA standard (American Gear Manufacturers Association). Computational programs are effective in the development of mechanical elements and can simplify the study of mechanical structures. Computational tools can be useful in decision-making in specific areas of mechanical engineering, such as in the calculation of straight-toothed cylindrical gears. This study aims to automate the analysis of pitting in straight-toothed cylindrical gears through the development of a computational tool in Python. A literature review was conducted on the subject, followed by the implementation of calculations based on the ANSI/AGMA standard and the development of a web platform using the Django framework. The developed tool proved to be modestly effective, capable of calculating the AGMA contact stress and suggesting different hardness values for a normalized safety factor of 1. Therefore, the user has the ability to choose the best option for their application based on the suggestions provided by the tool

**Keywords:** Python; gears; Django; crater analysis.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Engrenagem cilíndrica de dentes retos . . . . .	16
Figura 2 – Engrenagem cilíndrica de dentes retos 3D . . . . .	16
Figura 3 – Nomenclatura para engrenagens cilíndricas de dentes retos . . . . .	17
Figura 4 – Fator dinâmico, $K_v$ . . . . .	20
Figura 5 – Fator de forma de Lewis, Y . . . . .	21
Figura 6 – Fator de tamanho segundo o módulo . . . . .	21
Figura 7 – Constantes do fator de alinhamento . . . . .	22
Figura 8 – Fator de confiabilidade . . . . .	25
Figura 9 – Numero de tensão permitida ao contato para o aço . . . . .	25
Figura 10 – Ranking das linguagens de programação mais populares . . . . .	27
Figura 11 – Interesse em Python nos últimos cinco anos . . . . .	27
Figura 12 – Interesse em Python na área de ciências, tecnologia e engenharia. . . . .	28
Figura 13 – Mais "amadas" e "temidas" linguagens de programação. . . . .	28
Figura 14 – Interface do Visual Studio Code . . . . .	30
Figura 15 – Interface do Pycharm . . . . .	31
Figura 16 – Interface do Jupyter Notebook . . . . .	31
Figura 17 – Estrutura modular do sistemas especialistas . . . . .	33
Figura 18 – Arquitetura Django . . . . .	36
Figura 19 – Diagrama metodológico . . . . .	39
Figura 20 – Base Conda . . . . .	40
Figura 21 – Iniciar Projeto Django . . . . .	41
Figura 22 – Selecionando Interpretador . . . . .	41
Figura 23 – Rodando o servidor . . . . .	42
Figura 24 – Instalação bem sucedida . . . . .	43
Figura 25 – Estrutura inicial da aplicação . . . . .	43
Figura 26 – Pasta main . . . . .	44
Figura 27 – URL main . . . . .	44
Figura 28 – Página Index (Inserção de dados) . . . . .	45
Figura 29 – Página Index (Escolha de dados listados) . . . . .	45
Figura 30 – Página Results . . . . .	46
Figura 31 – Página Erro . . . . .	47
Figura 32 – Parâmetro Try . . . . .	47
Figura 33 – Parâmetro Except . . . . .	48
Figura 34 – Página Sobre . . . . .	48
Figura 35 – Estrutura da Pasta "main" e "templates" . . . . .	49
Figura 36 – Campo da página "index.html" . . . . .	50

Figura 37 – request.POST . . . . .	50
Figura 38 – Chamando os dados e atribuindo resultados . . . . .	51
Figura 39 – Pagina inicial . . . . .	53
Figura 40 – Escolha do dado de entrada . . . . .	54
Figura 41 – Escolha do material . . . . .	54
Figura 42 – Aplicando manualmente . . . . .	54
Figura 43 – Fonte de alimentação e Tipo de máquina . . . . .	55
Figura 44 – Fonte de alimentação . . . . .	55
Figura 45 – Tipo de máquina . . . . .	56
Figura 46 – Kv - Fator de qualidade Av . . . . .	57
Figura 47 – Demais cálculos . . . . .	58
Figura 48 – Calcular . . . . .	58
Figura 49 – Resultado . . . . .	58



## LISTA DE TABELAS

Tabela 1 – Fator de Sobrecarga . . . . .	19
Tabela 2 – Fator de confiabilidade pelo número de falha . . . . .	24

## LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AGMA	American Gear Manufacturers Association
ANSI	American National Standards Institute
DECOM	Departamento de Computação
IDE	Integrated Development Environment
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos

## LISTA DE SÍMBOLOS

$\Gamma$	Letra grega Gama
$\lambda$	Comprimento de onda
$\in$	Pertence

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	JUSTIFICATIVA	14
1.2	OBJETIVOS	15
1.2.1	Objetivos gerais	15
1.2.2	Objetivos específicos	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
2.1	ENGRENAGENS CILÍNDRICAS DE DENTES RETOS	16
2.1.1	Nomenclatura	17
2.1.1.1	Módulo	17
2.1.1.2	Passo circular	17
2.2	METODOLOGIA AGMA PARA ENGRENAGENS CILÍNDRICAS DE DENTES RETOS	18
2.2.1	Crateramento	18
2.2.2	Tensão de contato AGMA, $\sigma_h$	18
2.2.2.1	Coeficiente de elasticidade, $Z_E$	18
2.2.2.2	Carga tangencial, $W^t$	18
2.2.2.3	Fator de sobrecarga, $K_o$	19
2.2.2.4	Fator dinâmico, $K_v$	19
2.2.2.5	Fator de tamanho, $K_s$	20
2.2.2.6	Fator de distribuição de carga, $K_H$	21
2.2.2.7	Fator de condição superficial, $Z_R$	23
2.2.2.8	Fator geométrico ao crateramento, $Z_I$	23
2.2.3	Fator de segurança AGMA ao contato, $S_H$	23
2.2.3.1	Fator de ciclagem de tensão para resistência ao crateramento, $Z_N$	24
2.2.3.2	Fator de razão de dureza, $Z_W$	24
2.2.3.3	Fator de confiabilidade, $Y_Z$	24
2.2.3.4	Fator de temperatura, $Y_\theta$	25
2.2.3.5	Número de tensão ao contato permitida, $\sigma_{HP}$	25
2.3	LINGUAGEM DE PROGRAMAÇÃO PYTHON	26
2.3.1	Visual Studio Code	29
2.3.2	Pycharm Community	30
2.4	SISTEMAS ESPECIALISTAS	32
2.4.1	Aplicação e necessidade de uso de sistemas especialistas	34
2.4.2	Vantagens da utilização de sistemas especialistas	34
2.4.3	Classificação dos sistemas especialistas	34

2.5	FRAMEWORKS PYTHON . . . . .	34
2.6	DJANGO . . . . .	35
<b>3</b>	<b>ASPECTOS METODOLÓGICOS . . . . .</b>	<b>38</b>
<b>4</b>	<b>CONCEPÇÃO E DESENVOLVIMENTO DO TRABALHO . . . . .</b>	<b>40</b>
4.1	Inicialização da aplicação Django . . . . .	40
4.2	Criação da Interface web com o Django . . . . .	44
4.3	Coleta de dados e exibição de resultados . . . . .	49
4.4	<i>Deploy</i> da aplicação no Heroku . . . . .	51
4.5	Estudo de caso . . . . .	52
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>59</b>
	<b>Referências . . . . .</b>	<b>61</b>

## 1 INTRODUÇÃO

Como um dos elementos mecânicos mais relevantes associados ao projeto de máquinas, a engrenagem é uma roda dentada que engrena com outro mecanismo dentado para alterar a velocidade, torque ou a direção do movimento transmitido. Geralmente suas aplicações se definem basicamente em aumentar ou diminuir velocidade de rotação, alterar a quantidade de força ou torque, transmitir o movimento de rotação para outro eixo e inverter o sentido de rotação (BHATIA, 2014).

Engrenagens são elementos de transmissão de energia com capacidade de mudar a quantidade torque, são projetadas e otimizadas para melhorar a capacidade de carga dos dentes, alta eficiência, menor ruído e vibração. As engrenagens são geralmente fabricadas de acordo com os padrões estabelecidos pela *American Gear Manufacturers Association* (AGMA) e *American National Standards* (ANSI) (DAI et al., 2021).

As falhas de engrenagens são comuns e podem ser causadas por diversos fatores, como projeto inadequado, aplicação inapropriada, defeitos de fabricação e falta de manutenção. Elas podem apresentar sinais como ruído e vibração excessivos antes de se tornarem graves (ASI, 2006). De certo modo, diferentes tipos de falhas possuem características específicas e, as informações necessárias para sua solução podem ser obtidas através de uma inspeção detalhada para determinar a causa da falha. Os tipos gerais de modos de falha incluem desgaste, fratura por impacto, ruptura por tensão e fadiga. Existem muitas razões pelas quais há uma falha de engrenagem, como um projeto mal realizado, aplicação inadequada, defeitos de fabricação, falta de manutenção, lubrificação e assim por diante (FENG; FENG; MAO, 2020).

Com base neste contexto, um projeto de engrenagem bem planejado e estruturado é fundamental para prevenir falhas futuras. A norma AGMA é uma autoridade reconhecida na área de projeto e análise de engrenagens, especialmente no que diz respeito à resistência e desgaste. Seus métodos são amplamente utilizados e considerados confiáveis. Com base nisso, é importante seguir as recomendações da norma AGMA e utilizar seus métodos para garantir a segurança e a eficiência dos sistemas de transmissão (BUDYNAS; NISBETT, 2016).

A programação tem revolucionado o desenvolvimento de elementos mecânicos, desde o planejamento até a operação e manutenção. Com o aumento da capacidade de processamento dos computadores, as simulações computacionais se tornam cada vez mais populares e eficazes na simplificação de problemas complexos. Um estudo recente, liderado por Geus et al. (2017), sugere que a Transformação Rápida de Fourier é uma técnica especialmente útil para simplificar o estudo de microestruturas mecânicas, pois é mais rápida e eficiente em termos de consumo de memória do que outros métodos. Além disso, o estudo apresenta um código em Python, com apenas 59 linhas, para tornar o método acessível e fácil de implementar para pesquisadores que não possuem familiaridade com ele.

Com o crescimento do uso da tecnologia da informação e da eletrônica em dispositivos

mecânicos e máquinas-ferramentas, a criação de softwares que visam facilitar o trabalho de um engenheiro, realizando tarefas de rotina que, tomariam tempo de uma análise mais específica onde o computador ainda não substitui a ação do homem. Programas que são desenvolvidos tentando simular as tomadas de decisão e o raciocínio de um especialista são chamados de Sistemas Especialistas (ES - do inglês *Expert Systems*). Estes sistemas se destinam para uma área de conhecimento bastante específica, como no caso do presente trabalho, o cálculo de engrenagens cilíndricas de dentes retos (SCHLICHTING, 2014).

Os Sistemas Especialistas são aqueles capazes de solucionar um problema de maneira similar a um especialista humano em determinada área do conhecimento. Para a tomada de decisão em determinado assunto, um especialista o faz a partir dos fatos que encontra e hipóteses que formula, buscando na memória um conhecimento prévio (SCHLICHTING, 2014).

Na engenharia mecânica, se utiliza de várias técnicas para dimensionar ou analisar um elemento. Muitos destes métodos envolvem cálculos e desenhos que requerem além do conhecimento do engenheiro, requerem tempo, com isso o uso de sistemas especialistas pode colaborar para um desenvolvimento hábil dos objetivos necessários (SILVA, 2012).

Dentro deste contexto, o desenvolvimento de tais sistemas que reduzam o tempo de dimensionamentos analíticos com iterações, garantindo a confiabilidade dos resultados obtidos é indicado, para que possam ser aplicados nas mais diversas áreas, podendo auxiliar na fase inicial de um projeto, bem como servir de parâmetro ideal para comparação com um sistema já em funcionamento para a identificação de disparidades (SILVA, 2012).

## 1.1 JUSTIFICATIVA

Devido a evolução de diversas áreas da engenharia, com o passar do tempo o desenvolvimento que costumava se concentrar em melhorar ou aprimorar soluções existentes, passou a se focar mais em encontrar novas soluções. Outro foco de grande evolução é no sentido de melhorar a eficiência na realização de diversas atividades já conhecidas pelo engenheiro há algum tempo (SCHLICHTING, 2014). Como o estudo de R.Vundavilli et al. (2012) que busca criar um ES para previsão da profundidade de corte em processos de usinagem com jato de água abrasiva, onde a base de dados e a base de regras do sistema de lógica fuzzy são desenvolvidas com a ajuda de especialistas humanos.

A criação de um ES que automatiza os cálculos de engrenagens cilíndricas de dentes retos, visa auxiliar o engenheiro nas tomadas de decisões e tornar prático tais cálculos, visto que a demanda de tempo para a realização dos mesmos é alta e seus resultados são necessários para a continuação de um projeto. Essa praticidade facilita a entrega de um diagnóstico para a tomada de decisões por parte do projetista, que constantemente precisa fazer alterações no projeto inicial em razão das limitações do processo de fabricação, custos, fornecedores ou de outras exigências. Sua utilização não se limita apenas a área de projetos, pois seus resultados podem ser utilizados nas mais variadas situações, como manutenções, melhorias e adaptações. Além disso, sua implementação em uma plataforma web como no estudo de Garcia (2010) que

desenvolve uma plataforma web para aplicações de cálculo estrutural, onde o foco da aplicação é a análise e verificação de segurança de seções transversais metálicas. O autor ainda cita como motivação do estudo tornar a aplicação um *"meio catalisador de uma maior produtividade, com bastante utilidade educativa, e abrindo portas para o interesse dos engenheiros civis na criação de novas plataformas de desenvolvimento"*. A utilização de um framework web torna a aparência mais amigável para aqueles que não possuem muita intimidade com linguagem de programação e afins como Ambientes de Desenvolvimento Integrado (IDE do inglês - *Integrated Development Environment*) e compiladores para a execução de tal código.

## 1.2 OBJETIVOS

### 1.2.1 Objetivos gerais

Desenvolver um sistema especialista em Python, que vise o armazenamento nominal dos cálculos de projeto de uma engrenagem cilíndrica de dentes retos, para seu uso por meio de uma plataforma Web, implementando o código em um framework web afim de tornar sua aparência e utilização mais simples e amigáveis para o usuário.

### 1.2.2 Objetivos específicos

- Revisar as principais literaturas científicas sobre os conceitos relacionados a norma AGMA;
- Realizar um estudo mais detalhado sobre aplicações de Sistemas Especialistas na Engenharia;
- Demonstrar a aplicabilidade da linguagem Python na engenharia, bem como sua importância no desenvolvimento de soluções na área;
- Fazer o levantamento de plataformas e IDE's úteis para programação em Python;
- Explicar o que é um framework web;
- Propor um sistema especialista que seja capaz de mostrar resultados através dos dados obtidos;
- Contribuir para o meio acadêmico e industrial com a automatização dos conceitos de carateramento contidos na norma AGMA.



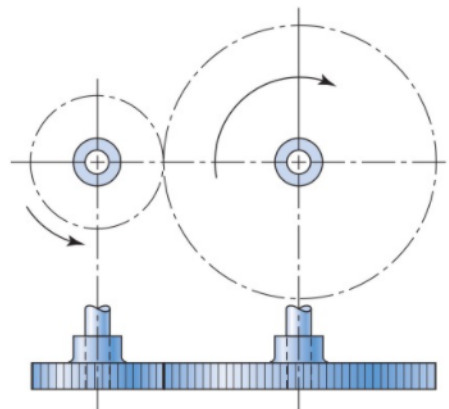
## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 ENGRENAGENS CILÍNDRICAS DE DENTES RETOS

A eficiência de engrenagens na transmissão de potência podem chegar a 98%. Porém, geralmente, engrenagens tendem a ser mais caras do que correntes e correias. Seus custos de fabricação variam de sua necessidade que, quanto maior a precisão, maior seu custo de fabricação, conforme se exige pela AGMA dados seus padrões de tolerância para a fabricação de engrenagens (JUVINALL; MARSHEK, 2013).

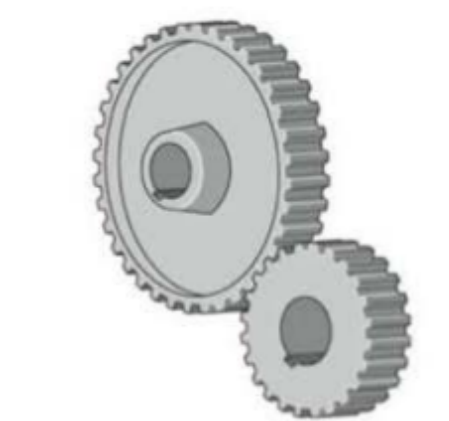
As engrenagens cilíndricas de dentes retos (conforme Figura 1 e Figura 2) são o tipo mais comum e simples de engrenagens. Elas possuem dentes paralelos ao eixo de rotação e são utilizadas para transmitir movimento de um eixo a outro (BUDYNAS; NISBETT, 2016).

Figura 1 – Engrenagem cilíndrica de dentes retos



Fonte: (BUDYNAS; NISBETT, 2016)

Figura 2 – Engrenagem cilíndrica de dentes retos 3D



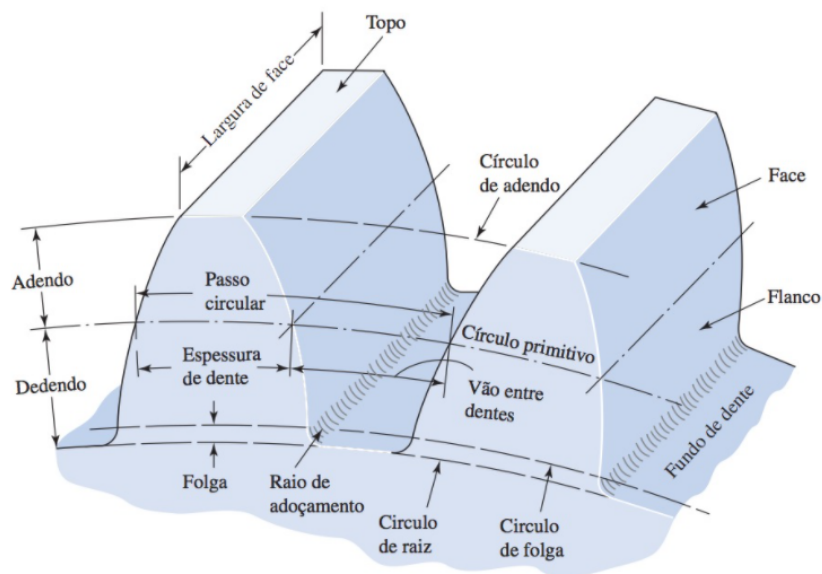
Fonte: (JUVINALL; MARSHEK, 2013)

### 2.1.1 Nomenclatura

Podemos ver pela [Figura 3](#) a nomenclatura, descrita a seguir.

O círculo primitivo é um círculo teórico sobre o qual todos os cálculos geralmente se baseiam. O adendo  $a$  é a distância radial entre o topo do dente e o círculo primitivo. O dedendo  $b$  é a distância radial do fundo do dente ao círculo primitivo. O círculo de folga é um círculo tangente ao círculo de adendo da engrenagem par. A folga  $c$  é a quantidade pela qual o dedendo em dada engrenagem excede o adendo da sua engrenagem par ([BUDYNAS; NISBETT, 2016](#)).

Figura 3 – Nomenclatura para engrenagens cilíndricas de dentes retos



Fonte: ([BUDYNAS; NISBETT, 2016](#))

Algumas relações são mostradas a seguir:

#### 2.1.1.1 Módulo

O módulo  $m$  é a razão entre  $d$  (diâmetro primitivo) e  $N$  (número de dentes).

$$m = \frac{d}{N} \quad (1)$$

#### 2.1.1.2 Passo circular

O passo circular  $p$  é a distância de um ponto do dente á outro correspondente no dente adjacente.

$$p = \frac{\pi * d}{N} = \pi * m \quad (2)$$

## 2.2 METODOLOGIA AGMA PARA ENGRENAGENS CILÍNDRICAS DE DENTES RETOS

Nesta seção serão abordados os cálculos da norma AGMA em relação ao crateramento que serão utilizados para a construção do Sistema especialista de forma resumida. O detalhamento destes cálculos com maior precisão e definições estão presentes no artigo de Nunez (2022), onde é realizado um detalhamento da norma AGMA e complementação com informações da literatura consolidada de Projeto de Máquinas, além de um estudo de caso que compara a norma AGMA com a norma ISO, para demonstrar que ambas as normas diferem muito.

### 2.2.1 Crateramento

A falha em dentes de engrenagens causada por tensões de contato é caracterizada pelo desgaste da face dos dentes em contato. Quando duas superfícies rolam e deslizam uma contra a outra com um determinado carregamento, poderá ocorrer falha por fadiga superficial (crateramento) após um certo número de ciclos de operação. Os fatores que influenciam a resistência do material a esse tipo de falhas são vários, sendo os principais, as tensões de Hertz, o número de ciclos ao qual as superfícies serão submetidas, o acabamento superficial, a dureza, o grau de lubrificação e a temperatura de operação (BUDYNAS; NISBETT, 2016).

### 2.2.2 Tensão de contato AGMA, $\sigma_h$

Para o cálculo de tensão de contato deve ser usada a Equação (3).

$$\sigma_h = Z_E \sqrt{\frac{W^t \cdot K_O \cdot K_V \cdot K_S \cdot K_H \cdot Z_R}{D_P \cdot b \cdot Z_I}} \quad [N/mm^2] \quad (3)$$

#### 2.2.2.1 Coeficiente de elasticidade, $Z_E$

$$Z_E = \sqrt{\frac{1}{\pi \left[ \left( \frac{1-v_1^2}{E_1} \right) + \left( \frac{1-v_2^2}{E_2} \right) \right]}} \quad [N/mm^2]^{0,5} \quad (4)$$

Onde

$v_1$  e  $v_2$  é a razão de Poisson do pinhão e a coroa, respectivamente;

$E_1$  e  $E_2$  e o módulo de elasticidade do pinhão e a coroa, respectivamente.

#### 2.2.2.2 Carga tangencial, $W^t$

$$W^t = \frac{P}{V} = \frac{T}{r_p} = \frac{P}{\pi \cdot D_p \cdot \omega} \quad [N] \quad (5)$$

Onde

$P$  é a potência transmitida, W;

$V$  é a velocidade linear, m/s;

$T$  é o torque transmitido, N-m.;

$r_p$  é o raio primitivo da engrenagem, m;

$D_p$  é o diâmetro primitivo da engrenagem, m;

$\omega$  é a rotação da engrenagem, rps.

### 2.2.2.3 Fator de sobrecarga, $K_o$

Os valores do fator de sobrecarga estão contidos na [Tabela 1](#).

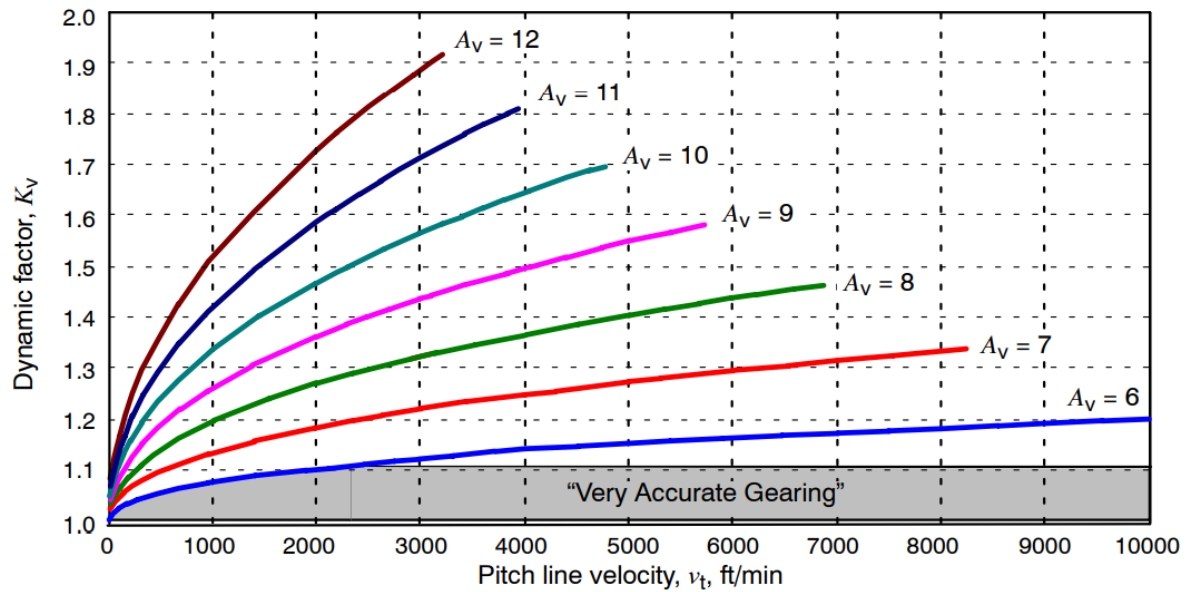
Tabela 1 – Fator de Sobrecarga

Fonte de alimentação	Tipo de máquina acionada			
	Uniforme	Choque leve	Choque moderado	Choque pesado
Uniforme	1,00	1,25	1,50	1,75
Choque leve	1,20	1,40	1,75	2,25
Choque moderado	1,30	1,70	2,00	2,75

Fonte: Adaptado de [Nunez \(2022\)](#)

### 2.2.2.4 Fator dinâmico, $K_v$

O fator dinâmico pode ser encontrado através de consulta gráfica (ver [Figura 4](#)) ou através das equações [6](#), [7](#) e [8](#).

Figura 4 – Fator dinâmico,  $K_v$ 

Fonte: Adaptado de (ANSI/AGMA-2101-D04, 2016)

$$K_V = \left[ \frac{C}{C + \sqrt{V}} \right]^{-B} \quad (6)$$

$$B = 0,25 (A_V - 5,0)^{2/3} \quad (7)$$

$$C = 3,5637 + 3,9914 (1,0 - B) \quad (8)$$

#### 2.2.2.5 Fator de tamanho, $K_s$

Há duas equações para o fator dinâmico  $K_s$ , dependendo da largura da face, se pode usar a [Equação \(9\)](#) (largura da face: 2,79 a 51 mm) ou [Equação \(10\)](#) (largura da face: 51 a 254 mm):

$$K_S = \frac{1}{1,1833 (b \cdot m \cdot \sqrt{Y})^{-0,0535}} \quad (9)$$

$$K_S = \frac{1}{1,4098 (b \cdot m \cdot \sqrt{Y})^{-0,0785}} \quad (10)$$

O fator de forma de Lewis  $Y$ , para engrenagens de aço e ângulo de pressão  $20^\circ$ , é encontrado na [Figura 5](#) e para o fator de tamanho em função do módulo usado, [Figura 6](#):

Figura 5 – Fator de forma de Lewis, Y

Número de dentes	Y	Número de dentes	Y
12	0,245	28	0,353
13	0,261	30	0,359
14	0,277	34	0,371
15	0,290	38	0,384
16	0,296	43	0,397
17	0,303	50	0,409
18	0,309	60	0,422
19	0,314	75	0,435
20	0,322	100	0,447
21	0,328	150	0,460
22	0,331	300	0,472
24	0,337	400	0,480
26	0,346	Cremalheira	0,485

Fonte: Adaptado de (BUDYNAS; NISBETT, 2016)

Figura 6 – Fator de tamanho segundo o módulo

Módulo	$K_s$
$\leq 5$	1,00
6	1,05
8	1,15
12	1,25
20	1,40
$> 20$	1,50

Fonte: Adaptado de (MOTT, 2013)

#### 2.2.2.6 Fator de distribuição de carga, $K_H$

$$K_H = 1,0 + K_{Hmc} (K_{Hpf} \cdot K_{Hpm} + K_{Hma} \cdot K_{He}) \quad (11)$$

Onde

$K_{Hmc}$  é o fator de formato da face do dente:

$$K_{Hmc} = \begin{cases} 1.0, & \text{para dente sem coroamento;} \\ 0.8, & \text{para dente coroado ou com correção de desvio.} \end{cases}$$

$K_{Hpf}$  é o fator de proporção do pinhão:

$$K_{Hpf} = \begin{cases} \text{Se } b \leq 25\text{mm}, & \text{então } \frac{b}{10 \cdot D_p} - 0,025 \\ \text{Se } 25 < b \leq 432\text{mm} & \text{então } \frac{b}{10 \cdot D_p} - 0,0375 + 0,000492 \cdot b \\ \text{Se } 432 < b \leq 1020\text{mm} & \text{então } \frac{b}{10 \cdot D_p} - 0,1109 + 0,000815 \cdot b - 0,000000353 \cdot b^2 \end{cases}$$

Para valores de  $\frac{b}{10 \cdot D_p}$  inferiores a 0,05, assumir  $K_{Hpf}$  igual a 0,05.

$K_{Hpm}$  é o fator de carga de deflexão:

$$K_{Hpm} = \begin{cases} 1.0, & \text{para pinhão montado com } \frac{S_1}{S} < 0.175 \\ 1.1, & \text{para pinhão montado com } \frac{S_1}{S} \geq 0.175. \end{cases}$$

Em que  $S_1$  é o deslocamento do pinhão; isto é, a distancia em relação ao centro do vão dos mancais. E  $S$  é o vão entre mancais; ou seja, a extensão entre as linhas de centro dos mancais.

$K_{Hma}$  é o fator de alinhamento de engrenamento:

$$K_{Hma} = A + B \cdot b + C \cdot b^2 \quad (12)$$

Valores a serem considerados nas constantes A, B e C apresentados na [Figura 7](#).

Figura 7 – Constantes do fator de alinhamento

Curve		A	B	C
Curve 1	Engrenagem aberta	$2.47 \times 10^{-1}$	$0.657 \times 10^{-3}$	$-1.186 \times 10^{-7}$
Curve 2	Engrenagem fechada industrial/comercial	$1.27 \times 10^{-1}$	$0.622 \times 10^{-3}$	$-1.69 \times 10^{-7}$
Curve 3	Engrenagem fechada de precisão	$0.675 \times 10^{-1}$	$0.504 \times 10^{-3}$	$-1.44 \times 10^{-7}$
Curve 4	Engrenagem fechada de alta precisão	$0.380 \times 10^{-1}$	$0.402 \times 10^{-3}$	$-1.27 \times 10^{-7}$

Fonte: Adaptado de [ANSI/AGMA-2101-D04 \(2016\)](#)

$K_{He}$  é o fator de ajuste:

$$K_{He} = \begin{cases} 0.8, & \text{quando a engrenagem e ajustada na montagem;} \\ 0.8, & \text{quando a compatibilidade da engrenagem e melhorada lapidando;} \\ 1.0, & \text{para todos os outros casos.} \end{cases}$$

2.2.2.7 Fator de condição superficial,  $Z_R$ 

Desde que a condição de superfície apropriada seja alcançada, o  $Z_R$  pode ser tomado como 1,0.

2.2.2.8 Fator geométrico ao crateramento,  $Z_I$ 

O fator geométrico para resistência ao crateramento pode ser obtido da

$$Z_I = C_c \cdot C_x \quad (13)$$

Onde

$C_C$  é o fator de curvatura na linha primitiva;

$C_x$  é o fator para ajuste da altura específica do LPSTC (ponto extremo inferior de contato de um dente, abaixo da linha primitiva).

As equações para se obter  $Z_1$  estão presentes a seguir:

$$C_c = \frac{\cos \theta \cdot \sin \theta}{2} \cdot \frac{i}{i + 1} \quad (14)$$

$$C_x = \frac{(C_1 - C_3 + C_4) \cdot (C_2 + C_3 - C_4)}{C_1 \cdot C_2} \quad (15)$$

$$C_1 = \frac{(N_p \cdot \sin \theta)}{2} \quad (16)$$

$$C_2 = C_1 \cdot i \quad (17)$$

$$C_3 = \pi \cdot \cos \theta \quad (18)$$

$$C_4 = 0,5 \left[ \sqrt{(N_p + 2)^2 - (N_p \cdot \cos \theta)^2} - \sqrt{N_p^2 - (N_p \cdot \cos \theta)^2} \right] \quad (19)$$

O fator geométrico também pode ser encontrado pelo gráfico presente no artigo de [Nunez \(2022\)](#).

2.2.3 Fator de segurança AGMA ao contato,  $S_H$ 

$$S_H = \frac{Z_N \cdot Z_W}{Y_\theta \cdot Y_Z} \cdot \frac{\sigma_{HP}}{\sigma_H} \quad (20)$$



2.2.3.1 Fator de ciclagem de tensão para resistência ao crateramento,  $Z_N$ 

$$Z_N = 1,4488 \cdot n_L^{-0,023} \quad (21)$$

Onde  $n_L$  é:

$$n_L = 60 \cdot L \cdot \omega \cdot q \quad (22)$$

$n_L$  é o número de ciclos de tensão;

$L$  é a vida nominal projetada(em horas);

$\omega$  é a rotação, rpm;

$q$  é o número de contatos por revolução.

2.2.3.2 Fator de razão de dureza,  $Z_W$ 

O valor de  $Z_W$  para o pinhão é fixado em 1,0. Já o valor de  $Z_W$  para a coroa é 1.0 ou conforme descrito no artigo de [Nunez \(2022\)](#) nas páginas 22 e 23.

2.2.3.3 Fator de confiabilidade,  $Y_Z$ 

Tabela 2 – Fator de confiabilidade pelo número de falha.

Requisitos de aplicação	$Y_z^{1)}$
Menos de uma falha em 10 000	1,5
Menos de uma falha em 1000	1,25
Menos de uma falha em 100	1,0
Menos de uma falha em 10	0,85 <sup>2)</sup>
Menos de uma falha em 2	0,7 <sup>2)3)</sup>

Fonte: Adaptado de ([NUNEZ, 2022](#))

Figura 8 – Fator de confiabilidade

Confiabilidade	$K_R (Y_z)$
0,9999	1,50
0,999	1,25
0,99	1,00
0,90	0,85
0,50	0,70

Fonte: (ANSI/AGMA-2101-D04, 2016)

#### 2.2.3.4 Fator de temperatura, $Y_\theta$

O fator de temperatura é geralmente 1,0 quando as próprias engrenagens ou a temperatura de óleo lubrificante opera com temperaturas não superiores a 120 °C. Quando as temperaturas de operação são abaixo de 0 °C, deve-se ter cuidado especial. Quando a temperatura do óleo ou da engrenagem opera acima de 120 °C, é atribuído ao  $Y_\theta$  um valor maior que 1,0 mas, isso depende da experiência do engenheiro projetista (NUNEZ, 2022).

#### 2.2.3.5 Número de tensão ao contato permitida, $\sigma_{HP}$

Os números de tensão permitidos para o aço estão presentes na Figura 9. Para os demais materiais, há uma faixa de valores e os mais baixos devem ser usados para fins gerais de projeto (NUNEZ, 2022).

Figura 9 – Numero de tensão permitida ao contato para o aço

Material designation	Heat treatment	Minimum surface hardness <sup>1)</sup>	Allowable contact stress number <sup>2)</sup> , $\sigma_{HP}$ N/mm <sup>2</sup>		
			Grade 1	Grade 2	Grade 3
Steel <sup>3)</sup>	Through hardened <sup>4)</sup>	see figure <sup>6)</sup>	see figure <sup>6)</sup>	see figure <sup>6)</sup>	--
	Flame <sup>5)</sup> or induction hardened <sup>5)</sup>	50 HRC	1170	1310	--
		54 HRC	1205	1345	--
	Carburized & hardened <sup>5)</sup>	see table 9	1240	1550	1895
Nitrided <sup>5)</sup> (through hardened steels)	83.5 HR15N	1035	1125	1205	
	84.5 HR15N	1070	1160	1240	
2.5% Chrome (no aluminum)	Nitrided <sup>5)</sup>	87.5 HR15N	1070	1185	1305
Nitalloy 135M	Nitrided <sup>5)</sup>	90.0 HR15N	1170	1260	1345
Nitalloy N	Nitrided <sup>5)</sup>	90.0 HR15N	1185	1300	1415
2.5% Chrome (no aluminum)	Nitrided <sup>5)</sup>	90.0 HR15N	1215	1350	1490

Fonte: adaptado de ANSI/AGMA-2101-D04 (2016)

### 2.3 LINGUAGEM DE PROGRAMAÇÃO PYTHON

Python é uma linguagem de programação criada no final da década de 80 por Guido van Russom. Com sua linguagem de alto nível, Python possui um código legível e acessível para iniciantes, além do mais o suporte a linguagem possui uma ampla quantidade de pesquisadores e comunidades de desenvolvimento de softwares (HOSMER, 2014). O Pypi.org (site oficial Python (2022) para bibliotecas) possui inúmeras bibliotecas para as mais diferentes aplicações.

Com a linguagem de alto nível, o uso do Python poupa o tempo dos programadores para criar o código, visto que sua linguagem permite criar linhas de códigos mais reduzidas devido suas funções internas. Contudo, o Python é uma linguagem de programação com uma compilação mais lenta que outras, tais como o C++ por exemplo (HOROWITZ, 2008). O Python tem como característica o uso de indentação como forma de criar blocos de código. Isso faz com que os códigos Python pareçam semelhantes, independentemente de quem os escreveu, o que aumenta a legibilidade do código (MILLMAN; AIVAZIS, 2011).

Por ser uma linguagem de propósito geral (aplicável na WEB, desktop, jogos...) ela vem se popularizando ao longo dos anos. Este fato se comprova com algumas pesquisas realizadas nos últimos anos, como a que foi realizada e postada na IEEE Spectrum por CASS (2020) que mostra a popularidade das linguagens por meio da análise de dados do Github, Google, Github, StackOverflow entre outras fontes. Python ocupa a primeira colocação, seguido de Java e linguagem C. Segundo CASS (2020) uma interpretação da alta classificação do Python é que suas métricas são infladas por seu uso crescente como linguagem de ensino: os alunos estão simplesmente perguntando e procurando as respostas para as mesmas perguntas elementares continuamente.

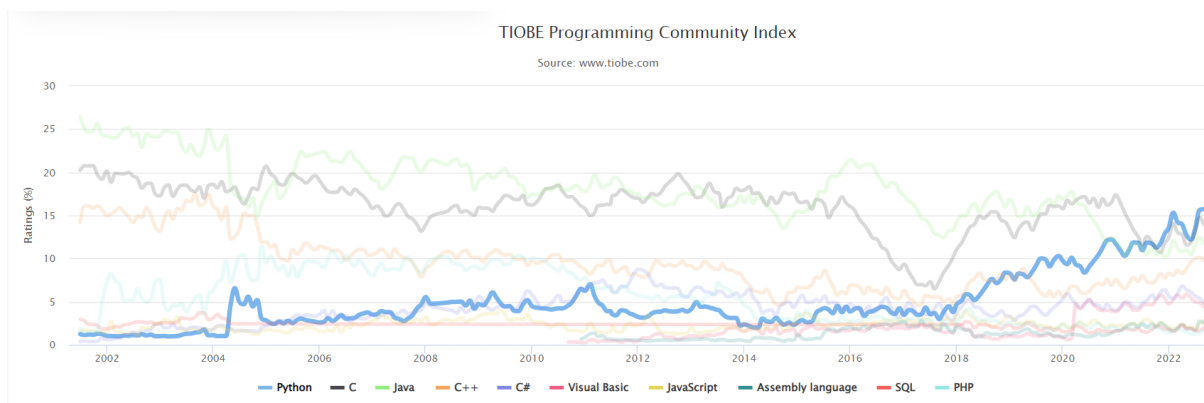
O site TIOBE mediu a popularidade das linguagens de programação através de buscas pela web. O TIOBE processa mensalmente um *ranking* de popularidade das linguagens de programação através de resultados dos motores de busca da Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube e outros 19 buscadores. O gráfico da Figura 10 deixa visível a evolução da popularidade de Python, em relação às demais linguagens, de 2002 até setembro de 2022.

1

---

<sup>1</sup><https://www.tiobe.com/tiobe-index/>

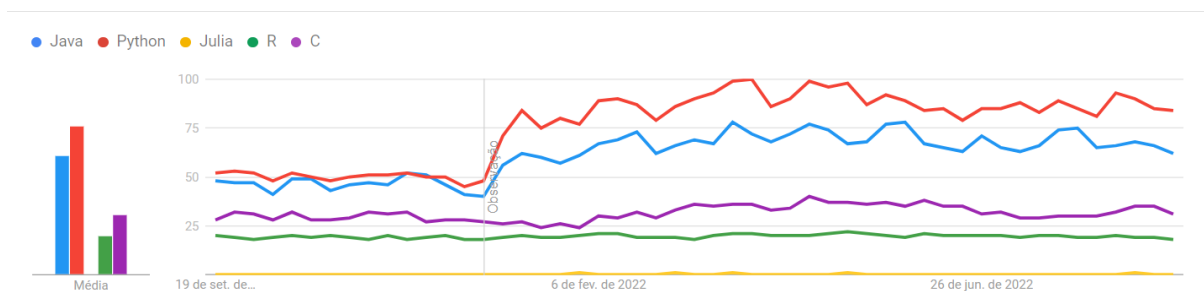
Figura 10 – Ranking das linguagens de programação mais populares



Fonte: Adaptado de TIOBE (2022)

O interesse nos últimos anos em Python tem crescido, como mostra a Figura 11 que ilustra seu destaque nos últimos cinco anos em todo o mundo, com base em pesquisas no Google, superando linguagens populares como Java. No Brasil, o interesse em Python também vem crescendo e segundo o Google Trends, em todo o mundo nos últimos 5 anos, com apenas Python e Java, filtrando na área de ciências, tecnologia e engenharia, o Brasil ocupa a posição 32 no ranking mundial de interesse em Python conforme a Figura 13.

Figura 11 – Interesse em Python nos últimos cinco anos



Fonte: adaptado de Google (2022)

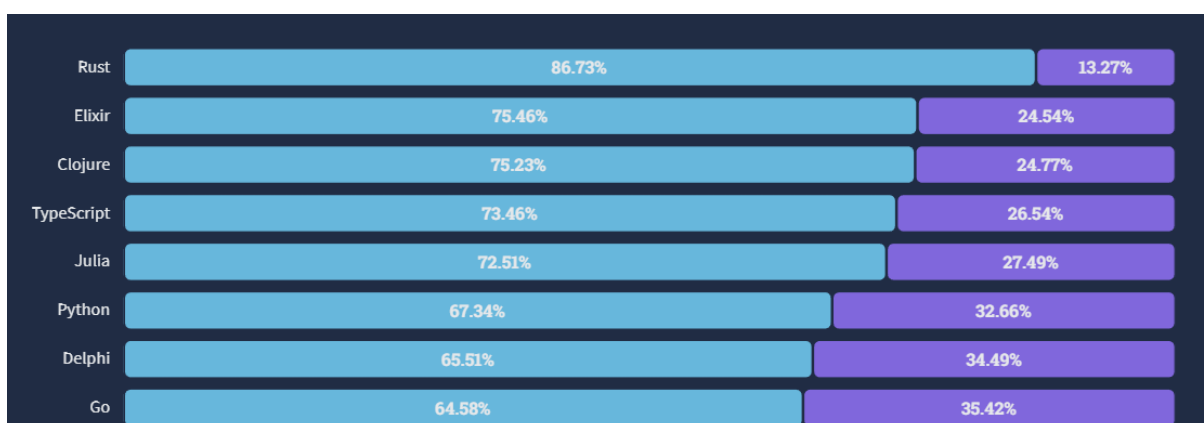
Figura 12 – Interesse em Python na área de ciências, tecnologia e engenharia.



Fonte: Adaptado de [Google \(2022\)](#)

Segundo o [StackOverflow \(2023\)](#) o Python está entre as linguagens de programação mais populares no interesse de desenvolver.

Figura 13 – Mais "amadas" e "temidas" linguagens de programação.



Fonte: Adaptado de [StackOverflow \(2023\)](#)

A utilização do Python é constante em pesquisas científicas, como por exemplo [Nedelcu e Gillich \(2021\)](#) que usam a linguagem Python para produzir um código de monitoramento de integridade estrutural para detectar pequenas mudanças nas frequências com o objetivo de detectar alterações estruturais, conseguindo uma observação precoce do estado da integridade estrutural. Da mesma maneira, [D.Thomas et al. \(2019\)](#) criaram um ambiente Python integrado para simulações de estrutura fluida acoplada com o objetivo de acoplar solventes sólidos e fluidos independentes. Por fim, [Guan, Burton e Sabol \(2020\)](#) desenvolveram uma plataforma computacional baseada em Python, com o objetivo de automatizar o projeto sísmico, na construção de um modelo estrutural não linear e análise de pórticos resistentes ao momento de aço.

Uma IDE (do inglês *Integrated Development Environment*) segundo Jeffery (2020) é um software que auxilia no desenvolvimento de aplicações, muito utilizado por desenvolvedores, com o objetivo de facilitar diversos processos ligados ao desenvolvimento, que combinam ferramentas comuns em uma única interface gráfica do usuário (GUI - do inglês *Graphical User Interface*). A maioria dos IDEs populares (e até mesmo os editores de código mais básicos) fornecem ferramentas para:

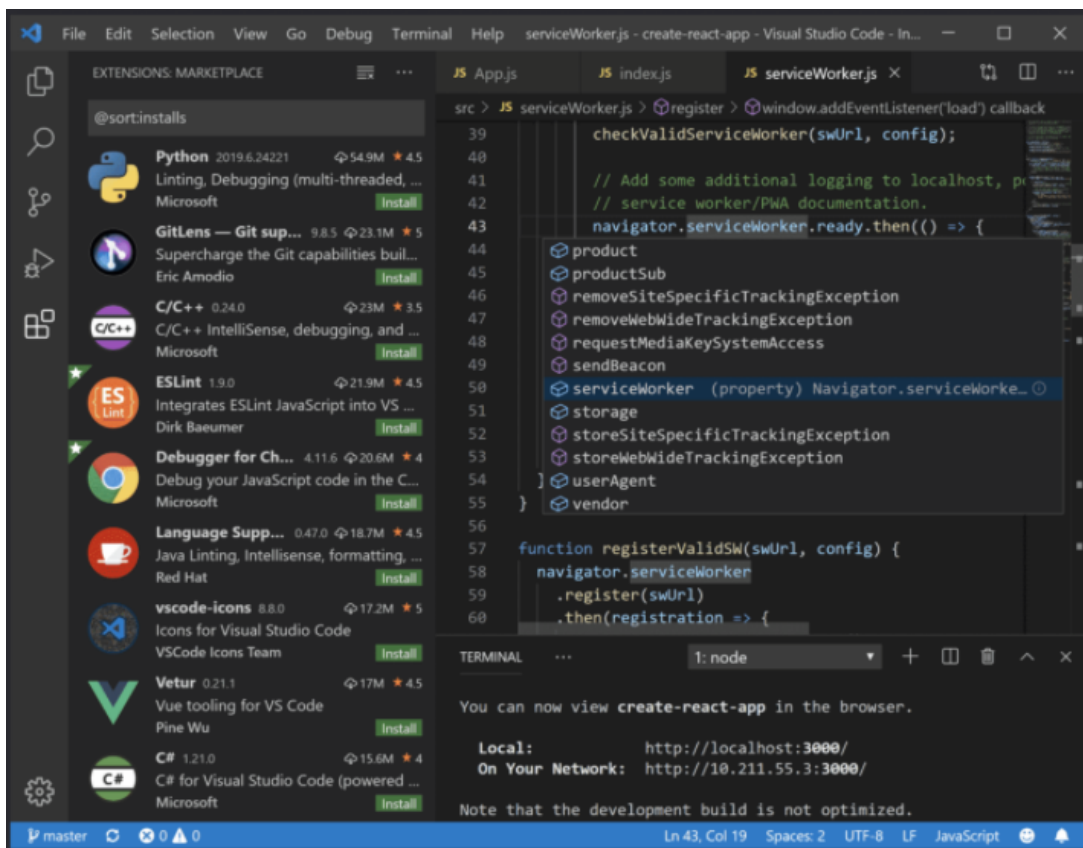
- **Realce de sintaxe com código de cores** - para ajudar na legibilidade e para separar diferentes tipos de comandos, variáveis, palavras e símbolos;
- **Deteção automática de erros** - com base na linguagem de programação específica que está sendo usada;
- **Recursos preditivos de autocompletar** - fornecendo sugestões à medida que o texto é digitado; e
- **Integração com repositórios de versionamento de código** - permitindo que o código seja compartilhado e garantindo o uso de versões atualizadas.

Programas bem conhecidos pela comunidade são o PyCharm e o Visual Studio Code, ambos os ambientes são gratuitos (PyCharm Community - Versão gratuita).

### 2.3.1 Visual Studio Code

O VS Code foi desenvolvido pela Microsoft e lançada sob uma licença do MIT para fornecer suporte aos profissionais que programavam em Python. O VS Code é totalmente personalizável e possui um rico conjunto de extensões que podem ser usados para melhorar as funcionalidades disponíveis (MICROSOFT, 2021).

Figura 14 – Interface do Visual Studio Code

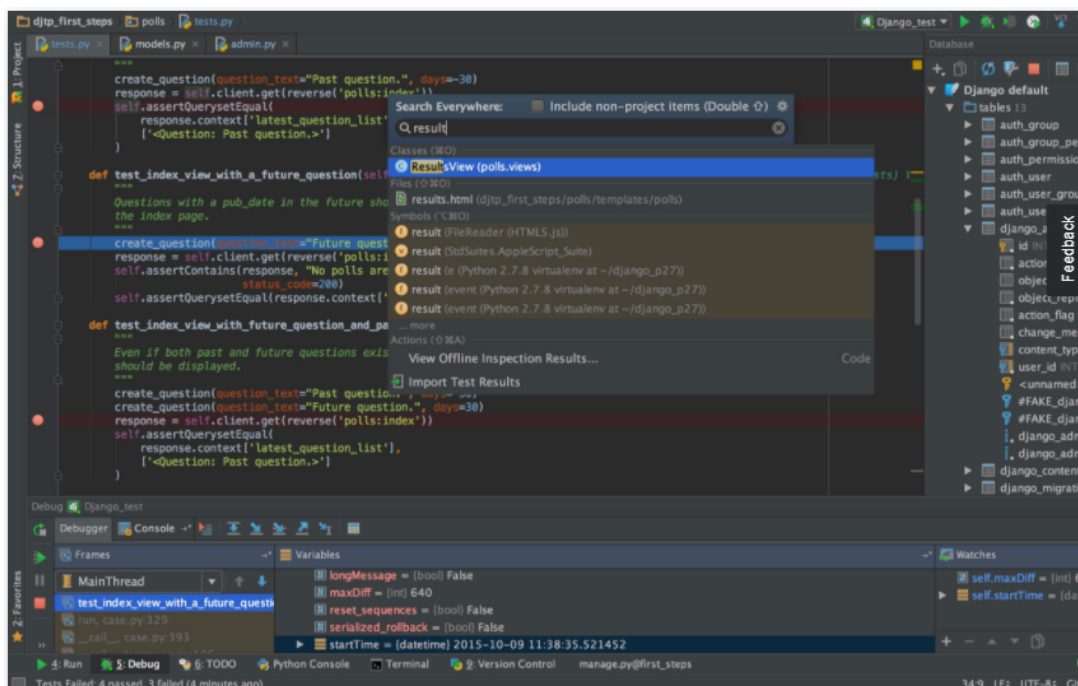


Fonte: Adaptado de [Microsoft \(2021\)](#)

### 2.3.2 Pycharm Community

Usado especificamente para programar em Python. Consiste em ferramentas de análise de código, debug, ferramentas de teste e opções de controle de versão. Os desenvolvedores podem criar seus próprios plugins Python com a ajuda de API's disponíveis no Pycharm. Você pode trabalhar diretamente com vários bancos de dados do próprio ambiente sem precisar integrá-los em uma ferramenta auxiliar ([JETBRAINS, 2021](#)).

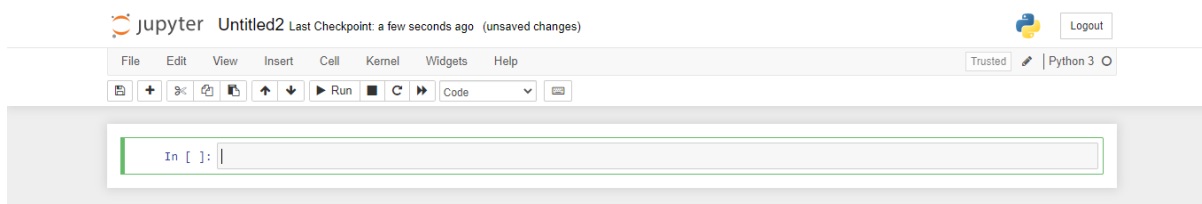
Figura 15 – Interface do Pycharm



Fonte: Adaptado de [JetBrains \(2021\)](#)

No presente trabalho, será utilizada a distribuição [Anaconda \(2021\)](#) com a versão 3.8.5 do Python. A Anaconda é uma distribuição de software livre (*individual edition*) que se auto intitula *"The World's Most Popular Data Science Platform"* utilizada por empresas como Cisco, Citi e Exxon mobil, ela é ideal para a programação científica, pelo fato de sua instalação já incluir vários pacotes importantes instalados, e com ambientes de trabalho, como o Jupyter Notebook (ambiente computacional web para a internet para criação de documentos em linguagem Julia, Python e R, daí seu nome Jupyter), Spyder (interface gráfica, que permite a utilização de Python num ambiente interativo), PyCharm Community e outras aplicações. Nota-se que há uma vasta quantidade de ambientes com os quais se pode trabalhar, a escolha neste trabalho é pelo Jupyter Notebook (ver [Figura 16](#)) para o equacionamento, devido ao seu ambiente amigável e por unir texto e código em um só programa, porém, o VS Code será utilizado para o desenvolvimento em Django.

Figura 16 – Interface do Jupyter Notebook



Fonte: Jupyter



Na presente proposta de trabalho, o armazenamento nominal dos cálculos de projeto de uma engrenagem cilíndrica de dentes retos, serão alimentadas na linguagem de programação Python, para seu uso em projeto por meio de uma plataforma web e que auxilie na tomada de decisão da escolha de uma geometria mais segura. Assim, o programa a ser desenvolvido se enquadra como um Sistema Especialista.

## 2.4 SISTEMAS ESPECIALISTAS

Sistemas Especialistas (ES do inglês - *Expert Systems*), também conhecidos como sistemas de conhecimento, são programas de computador que buscam transferir o conhecimento de especialistas humanos para uma base computacional. Tais sistemas são construídos com conhecimento específico em certos domínios de aplicação. Dada uma entrada, o sistema aciona uma regra correspondente para produzir uma resposta (C.Y.CHEN; POO, 2003).

Ainda segundo C.Y.Chen e Poo (2003) o uso de sistemas baseados em conhecimento é principalmente voltado para o auxílio de especialistas humanos para a realização de determinadas tarefas. Sua utilização em tais tarefas se dá pela função consultiva para análise, monitoramento e diagnóstico, enquanto a decisão final ainda é feita pelo especialista humano. Todo esse conhecimento obtido pelo sistema especialista para ser útil, deve ser posto em prática, e, essa função é do próprio especialista humano que irá utilizá-lo (E.ARONSON, 2003).

Um dos aspectos favoráveis da utilização de SE é a normatização de conhecimentos restritos e a sua difusão, tornando um conhecimento específico mais acessível. Para Gonçalves (2009) os SE fornecem respostas a perguntas de uma área muito específica, por meio de inferências humanas sobre conhecimento contido em uma base de conhecimento especializado. As respostas que os SEs fornecem são restritas às áreas específicas de aplicação, não tendo, portanto, a geração de novos conhecimentos, pois são criados com base em conhecimento especializado. As respostas aos usuários são baseadas em respostas previamente programadas.

Rodrigues (2010) destaca que a base de conhecimento é formada por regras e procedimentos usados pelo especialista humano na solução de problemas. O conhecimento do especialista pode ser adquirido através de meios de observação, testes ou entrevistas e ser sintetizado. A base de conhecimento fica separada da máquina de inferência, desse modo, o conhecimento permanece na base, facilitando sua modificação, se necessária. Quando realizadas mudanças nas bases de conhecimentos, podem haver mudanças de regras, remoção de bases antigas e adição de novas. Quando a representação do conhecimento é realizada por regras, uma opção muito útil para atividades repetitivas, o conhecimento pode ser representado por meio dos pares de condição IF e THEN, ou no caso do Python, IF, ELIF e ELSE.

Rodrigues (2010) afirma que um SE comum pode ser estruturado por módulos, nos quais é possível observar os seguintes elementos:

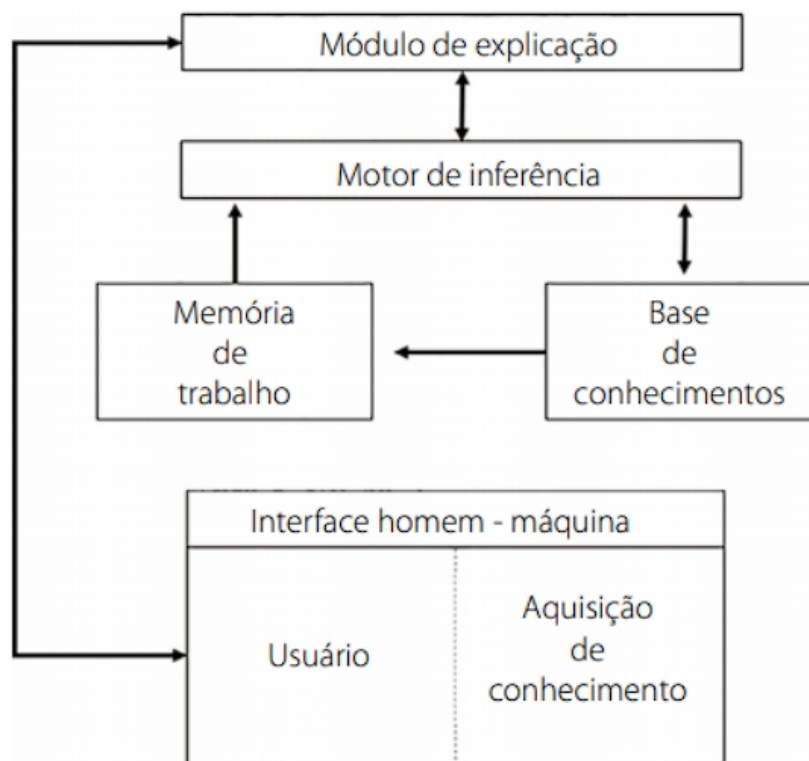
- Base de dados: É a base de dados onde se encontram os dados, arquivos, registros e relacionamentos que fazem correspondência com o conhecimento do especialista. Esta base interage com o usuário e com o motor de inferência, dessa forma se permite

identificar o problema resolvido, as possibilidades de solução e a solução do problema. Dessa interação baseada nos fatos e regras com o usuário, se obtêm as informações necessárias para a resolução do problema. O usuário é requerido pelo sistema para prestar informações adicionais e, a cada pergunta respondida pelo usuário, se reduz o espaço de busca pelo sistema, se aproximando da solução;

- Motor de inferência: Pelo intermédio do motor de inferência, os fatos, as regras e a heurística são aplicadas no processo de resolução do problema. Sua capacidade é baseada na combinação de procedimentos de raciocínios que se processam regressiva ou progressivamente. Na forma progressiva, ao passo que o usuário vai alimentado o sistema de dados, tais dados estimulam o desencadeamento do processo de busca por uma resposta. No modelo regressivo é justamente o contrário, o sistema parte de uma opinião conclusiva do assunto e inicia uma pesquisa, procurando provar se aquela conclusão é a mais adequada para o problema analisado;
- Interface com o usuário: Um SE que apresenta uma interface bem projetada deve considerar o grau de familiarização com o usuário, buscando facilitar o domínio de trabalho do ambiente. Janelas, caixa de diálogo, listas e botões são alguns dos elementos que facilitam o uso.

A [Figura 17](#) demonstra um fluxograma dos módulos:

Figura 17 – Estrutura modular do sistemas especialistas



### 2.4.1 Aplicação e necessidade de uso de sistemas especialistas

Segundo [Gonçalves \(2009\)](#), os SEs podem ser aplicados em várias áreas das empresas, promovendo maior agilidade na tomada de decisões por parte dos gestores, oferecendo, assim, soluções produtivas e mantendo a empresa com vantagem competitiva diante dos concorrentes. A tecnologia dos SEs é mais necessária em conhecimentos específicos, pois não possuem domínio múltiplo de conhecimento. Algumas categorias onde podem ser aplicados os SEs:

- Gerenciamento de decisões;
- Diagnóstico de problemas de operação;
- Projeto/configuração;
- Seleção/classificação;
- Monitoramento/controle de processo.

### 2.4.2 Vantagens da utilização de sistemas especialistas

Segundo [Mendes \(1997\)](#), uma das vantagens dos SEs é justamente eles possuírem uma bagagem de conhecimento especialista específico que o difere dos demais sistemas. Outras vantagens segundo o autor são:

- Facilitar a tomada de decisão;
- Melhorar a produtividade e desempenho dos usuários;
- Redução da dependência de pessoal para a realização de atividades que podem ser automatizadas.

### 2.4.3 Classificação dos sistemas especialistas

[Rodrigues \(2010\)](#) destaca uma classificação para SE:

- Interpretação: sistemas que descrevem a partir de dados e análises, soluções e interpretações para determinado objetivo;
- Diagnósticos: sistemas que realizam busca para verificação de falhas podendo ou não conduzir a uma solução;
- Monitoramento: sistemas que interpretam sinais, realizam verificações continuamente determinando comportamentos com limites pré-estabelecidos ou pré-existente, sinalizando necessidade de intervenção;
- Planejamento: sistemas que preparam iniciativas a serem tomadas através de etapas, cada etapa se aprofundando cada vez mais na análise.

## 2.5 FRAMEWORKS PYTHON

Os frameworks são softwares que facilitam a configuração de ambientes voltados para WEB, porque oferecem uma base estável para o desenvolvimento, com essas configurações de infraestrutura pré-programadas. Dessa forma, projetos que exigiriam uma carga maior de

conhecimento e trabalho exaustivo do programador se fossem realizados em linguagem de programação "pura" são simplificados por este software (CARDOSO; BISPO, 2019).

Os frameworks ditam a estrutura geral do projeto, sua divisão em classes e objetos e em como as responsabilidades-chave das classes de objetos colaboram. Um framework predefine tais parâmetros de projeto, de maneira que o projetista/implementador da aplicação, possa se concentrar nos aspectos específicos da sua aplicação (GAMA et al., 2007).

Em Python, existem diversos frameworks disponíveis, porém os dois mais populares na atualidade são Django e Flask. O Flask é um microframework que visa manter um núcleo simples. O Flask também não toma muitas decisões pelo usuário, como qual banco de dados usar e, as que toma são bem fáceis de mudar (RONACHER, 2021). O Django é o framework baseado em Python mais popular e ideal para o uso em ciência de dados. No presente trabalho o framework utilizado será o Django, devido a sua gama de recursos e pela comunidade ativa.

## 2.6 DJANGO

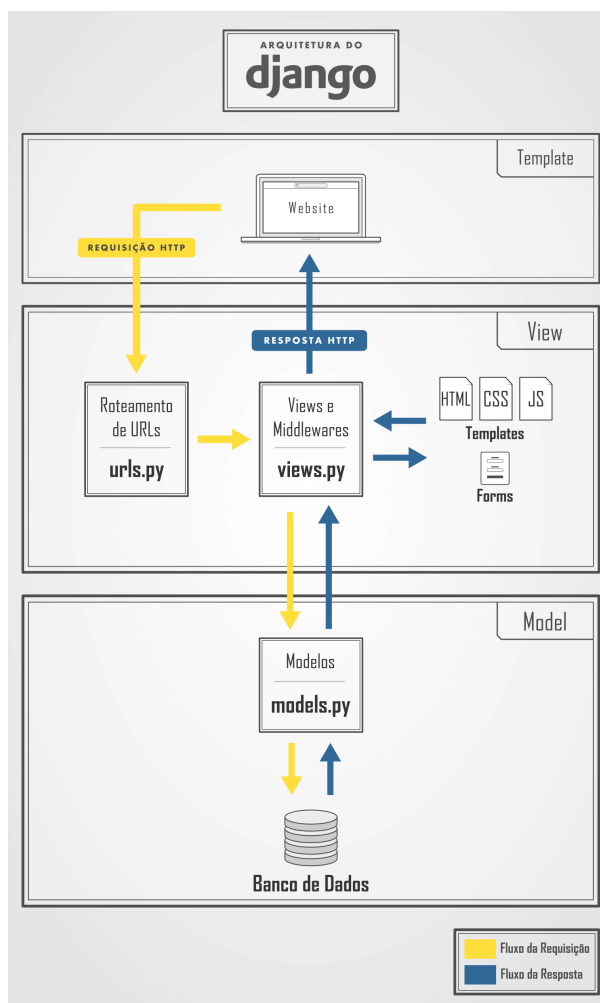
O Django foi criado em 2005 por Adrian Holovaty e Simon Willison, que dois anos antes resolveram abandonar o PHP para utilizar Python no desenvolvimento de seus sites. Seu crescimento se deu pela agilidade no desenvolvimento das aplicações web. Atualmente empresas como o Mozilla em seu site, o aplicativo do Instagram, o Pinterest utilizam o Django em seus projetos (SILVA; SILVA, 2019).

Django é um framework de alto nível que tem como proposta tornar tarefas comuns do desenvolvimento web rápidas e fáceis. Além disso, para testes, o Django possui um servidor que pode ser utilizado durante o processo de desenvolvimento, dessa forma não é necessário configurar um servidor externo como o Apache antes de efetivamente colocá-lo em produção (FOUNDATION, 2021).

O Django é definido como um framework MTV (*Model Template View*), onde o *Model* serve para o mapeamento do banco de dados para o projeto, o *Template* páginas para visualização de dados, onde normalmente fica o HTML que será renderizado nos navegadores e por fim, o *View*, que encapsula a lógica entre as requisições de usuários e as resposta. Toda esta arquitetura é interligada e conversam entre si. Uma depende da outra para realizar um determinado serviço e, no final, executar a tarefa que o usuário solicitou (FOUNDATION, 2021).

A arquitetura do Django esta ilustrada na [Figura 18](#):

Figura 18 – Arquitetura Django



Fonte: Extraído de (RAMOS, 2018)

A adoção de programação na engenharia não é nenhuma novidade e como qualquer tecnologia que vem para facilitar um projeto, sistema ou coleta de dados. O Django também tem suas aplicações nas mais diversas áreas como no estudo de Guha et al. (2021), onde notou que com o aumento de artigos publicados no campo da ciência dos materiais havia uma certa desestruturação das fontes de dados, dificultando a obtenção de várias informações úteis e e que tais informações necessárias para realizar cálculos posteriores com os dados nela contidos deveriam ser extraídos manualmente. Deste modo, eles criaram o MatSciE (*Material Science Information Extractor*), que pode extrair informações relevantes da literatura da ciência de materiais e fazer um banco de dados estruturado e muito mais fácil de usar para simulações de materiais. Tal banco de dados é alimentado por uma aplicação web onde os usuários podem carregar artigos publicados e visualizar / baixar as informações obtidas nesta ferramenta e podem criar seus próprios bancos de dados para seu uso pessoal. Todo o lado servidor da aplicação foi desenvolvido usando Django.

O estudo de Low et al. (2018) aprimora o modelo de Rede de Troca de Subproduto

(BEN) com um mecanismo de banco de dados para correspondência entre resíduos e recursos. Onde, o subsistema de correspondência de resíduos para recursos funciona como um sistema baseado em conhecimento que fornece aos usuários com recomendações sobre possíveis caminhos para conversões de resíduos para fontes. Este subsistema é alimentado por um banco de dados motor, que é implementado como um aplicativo da web usando o Framework Django. Por fim, [F.Eger et al. \(2020\)](#) abordam a redução de defeitos de fabricação de componentes rotativos com base nas medições do contorno interno e o cálculo de um novo eixo de rotação para os processos de fabricação subsequentes a fim de evitar desequilíbrios no controle de fim de linha. Para tais dimensionamentos e cálculos, diferentes algoritmos são desenvolvidos e integrados em um aplicativo da web baseado em navegador usando a web framework Django. No presente trabalho o Django servirá como interface para os cálculos de engrenagens.

### 3 ASPECTOS METODOLÓGICOS

Para a realização dos cálculos de engrenagens, foi realizado o levantamento bibliográfico de modo a obter dados para o desenvolvimento do trabalho com base em livros como Elementos de Máquinas de Shigley (BUDYNAS; NISBETT, 2016) e Fundamentos do Projetos de Componentes de Máquinas (JUVINALL; MARSHEK, 2013), artigos como de Nunez (2022), teses, nacionais e internacionais, e a norma ANSI/AGMA-2101-D04 (2016). Assim como para os cálculos, para a escolha de linguagem de programação, ferramentas de programação e framework, foi realizado um levantamento bibliográfico com base na documentação original de suas respectivas plataformas como as documentações oficiais Django (FOUNDATION, 2021) e Python (PYTHON, 2021).

Para desenvolver a aplicação web, será utilizado o framework Django, que oferece ferramentas para a construção de sites com facilidade, utilizando as linguagens de programação Python e HTML. A aplicação terá uma página inicial onde serão solicitados dados ao usuário através de campos preenchíveis ou listas de opções. Os resultados dos cálculos serão exibidos em uma página específica, caso haja algum erro na inserção dos dados, será direcionado para uma página de erro e uma página "sobre" será disponibilizada para fornecer mais informações sobre o conteúdo do site.

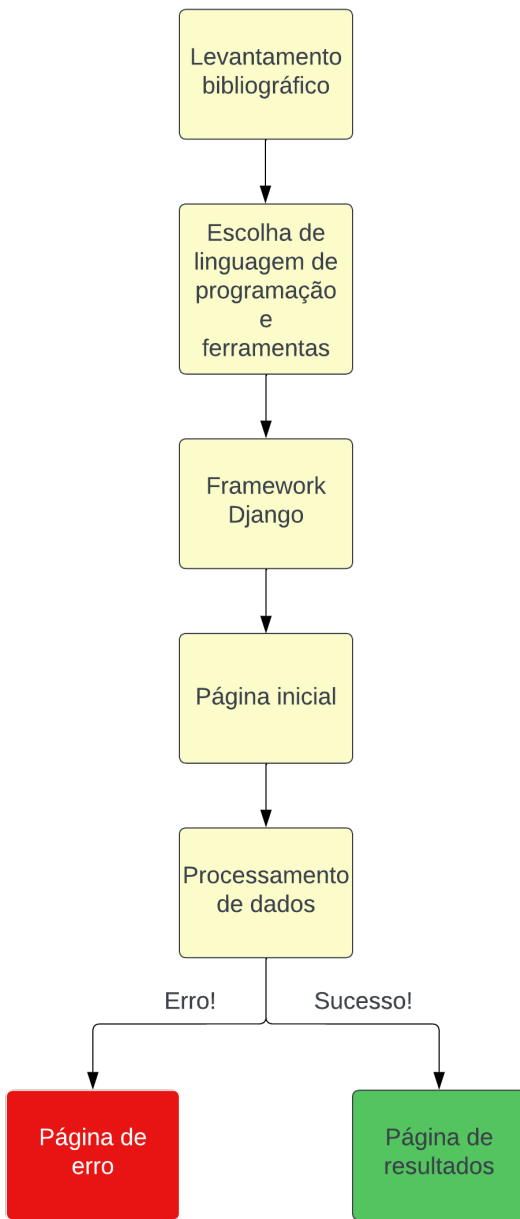
Como citado anteriormente, um dos casos onde a página levará à outra página chamada erro será quando nenhum ou apenas alguns dos campos do formulário visíveis para o usuário forem preenchidos, pois o sistema de construção do código que se baseia na norma ANSI/AGMA-2101-D04 (2016) é montado para trabalhar de forma interativa, ou seja, cada dado inserido depende de outro dado para calcular o resultado final.

A página HTML receberá os dados fornecidos pelo usuário e serão enviados para a camada View através do método POST, estes dados serão alocados em variáveis no arquivo "views.py". Com o recebimento de tais dados, as funções do arquivo "functions.py" que esta na camada Model realizará o procedimento de cálculo processando os dados recebidos.

Por último, o resultado final é retornado através de uma ou mais variáveis que conterão o processo de cálculo já realizado, esses dados passam pelo "views.py", processados pelo "functions.py", retornam ao "views.py" chamando a função através da variável e esta função será chamada para a página de resultados em HTML para apresentação dos dados. 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 18.

Também existe uma opção para o usuário escolher entre diferentes materiais já pré definidos com os valores de coeficiente de elasticidade já definidos ou podendo optar por digitar os dados de razão de Poison e módulo de elasticidade tanto do pinhão quanto da coroa manualmente.

Figura 19 – Diagrama metodológico



Fonte: Autoria Própria

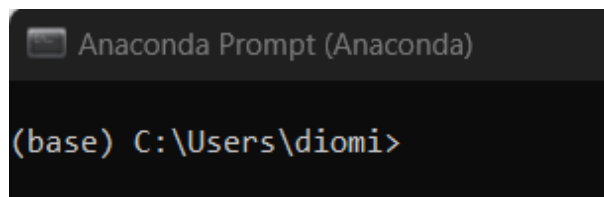


## 4 CONCEPÇÃO E DESENVOLVIMENTO DO TRABALHO

### 4.1 Inicialização da aplicação Django

Inicialmente, foi criado um ambiente virtual com o objetivo de separar um projeto, suas dependências e bibliotecas em um único lugar. Dessa forma, esse ambiente específico para o projeto não interfere em outros projetos. Para criar um ambiente virtual, é necessário seguir uma sequência de comandos. No caso, basta digitar **"python -m venv venv"** no Prompt de Comando, utilizado no exemplo foi o do Anaconda (conda prompt), pois ao finalizar a instalação do Anaconda, automaticamente é criado um ambiente virtual com o interpretador Python e os pacotes da instalação, chamado de "base". Por padrão, essas configurações fazem com que o ambiente seja automaticamente ativado, e ao abrir o terminal, o mesmo deve indicar o ambiente no prefixo do Terminal, como mostrado na [Figura 20](#):

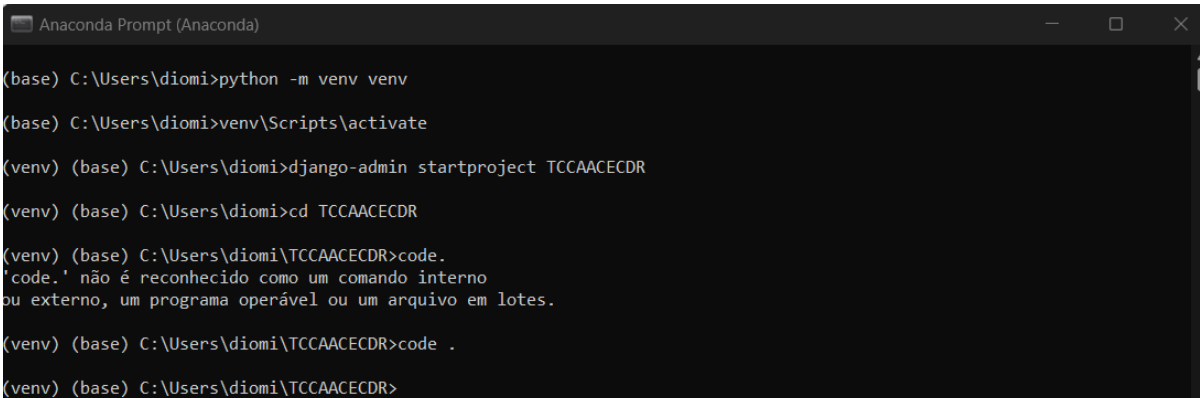
Figura 20 – Base Conda



Fonte: Autoria Própria

Para criar um ambiente virtual específico para o projeto, é necessário executar o comando **"python -m venv venv"** e, em seguida, **"venv/Scripts/activate"** para entrar no ambiente virtual. Uma vez no ambiente virtual, é possível iniciar o projeto em Django com o comando **"django-admin startproject TCCAACECDR"**, criando uma pasta específica chamada **"TCCAACECDR"** para o projeto. Então, digita-se **"cd TCCAACECDR"** para entrar na pasta e **"code ."** para inicializar o Visual Studio Code, que será utilizado para desenvolver o projeto, conforme mostrado na [Figura 21](#):

Figura 21 – Iniciar Projeto Django

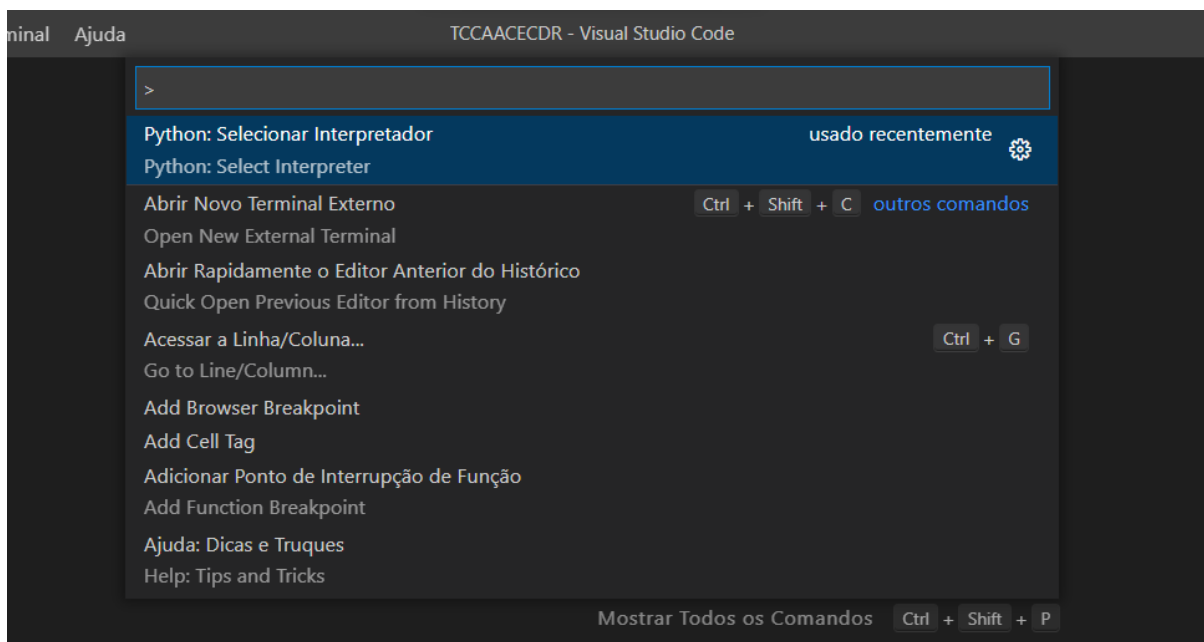


```
(base) C:\Users\diomi>python -m venv venv
(base) C:\Users\diomi>venv\Scripts\activate
(venv) (base) C:\Users\diomi>django-admin startproject TCCAACECDR
(venv) (base) C:\Users\diomi>cd TCCAACECDR
(venv) (base) C:\Users\diomi\TCCAACECDR>code.
'code.' não é reconhecido como um comando interno
ou externo, um programa operável ou um arquivo em lotes.
(venv) (base) C:\Users\diomi\TCCAACECDR>code .
(venv) (base) C:\Users\diomi\TCCAACECDR>
```

Fonte: Autoria Própria

Com o Visual Studio Code (VS Code) iniciado, é necessário selecionar o interpretador de Python correto através do comando "Python: Selecionar interpretador" e escolher o Python disponível no ambiente virtual criado anteriormente. Dessa forma, o projeto será executado com as configurações corretas e não haverá conflitos com outras versões de Python instaladas no sistema.

Figura 22 – Selecionando Interpretador



Fonte: Autoria Própria

Com o interpretador selecionado, é necessário abrir o terminal dentro do próprio VS Code, dentro da pasta do projeto e executar o comando "python manage.py runserver". Esse comando inicia um servidor local onde toda a aplicação em Django será construída. A URL

padrão retornada após o comando é "http://127.0.0.1:8000/". Acessando essa URL, é possível ver uma página que informa que a instalação e início da aplicação foram bem-sucedidos, mas essa página só aparecerá se a variável DEBUG estiver definida como "True".

Para acessar a variável DEBUG = True no Django:

Passo 1: Abra o diretório do seu projeto Django no seu editor de código ou terminal.

Passo 2: Navegue para a pasta do projeto, que geralmente possui o mesmo nome do seu projeto Django. Você pode usar o comando cd seguido do nome da pasta para entrar nela.  
cd nomedoprojeto/

Passo 3: Dentro da pasta do projeto, localize o arquivo settings.py. Esse arquivo contém as configurações do seu projeto Django.

Passo 4: Abra o arquivo settings.py no seu editor de código.

Passo 5: Procure pela linha que contém a configuração DEBUG = False. Você pode usar as teclas de navegação do seu editor de código para encontrar essa linha ou CTRL+F.

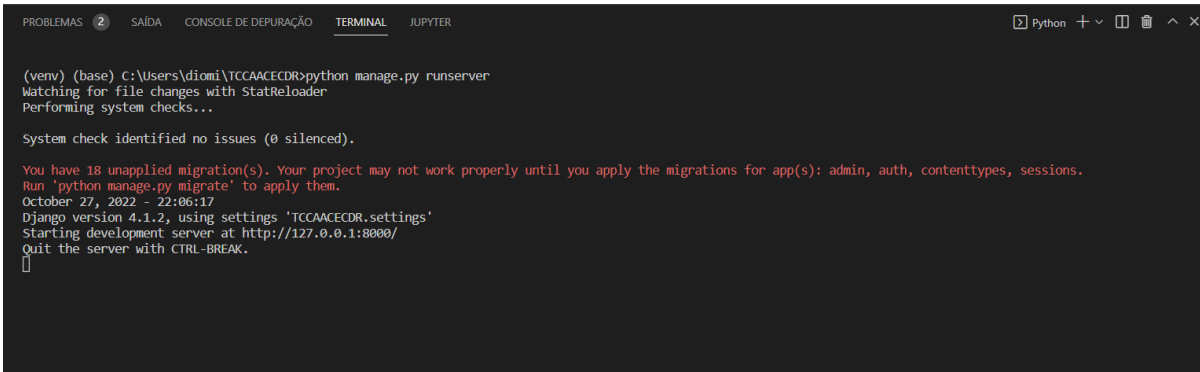
Passo 6: Altere o valor False para True, indicando que o modo de depuração está ativado.

Passo 7: Salve as alterações e feche o arquivo settings.py.

Passo 8: Agora você pode executar seu projeto Django com o modo de depuração ativado, permitindo que você veja detalhes de erros e mensagens de depuração.

Lembre-se de ter cuidado ao usar o modo de depuração em um ambiente de produção. Certifique-se de desativar o modo de depuração (DEBUG = False) antes de implantar seu aplicativo em produção para garantir a segurança e proteção dos dados sensíveis.

Figura 23 – Rodando o servidor



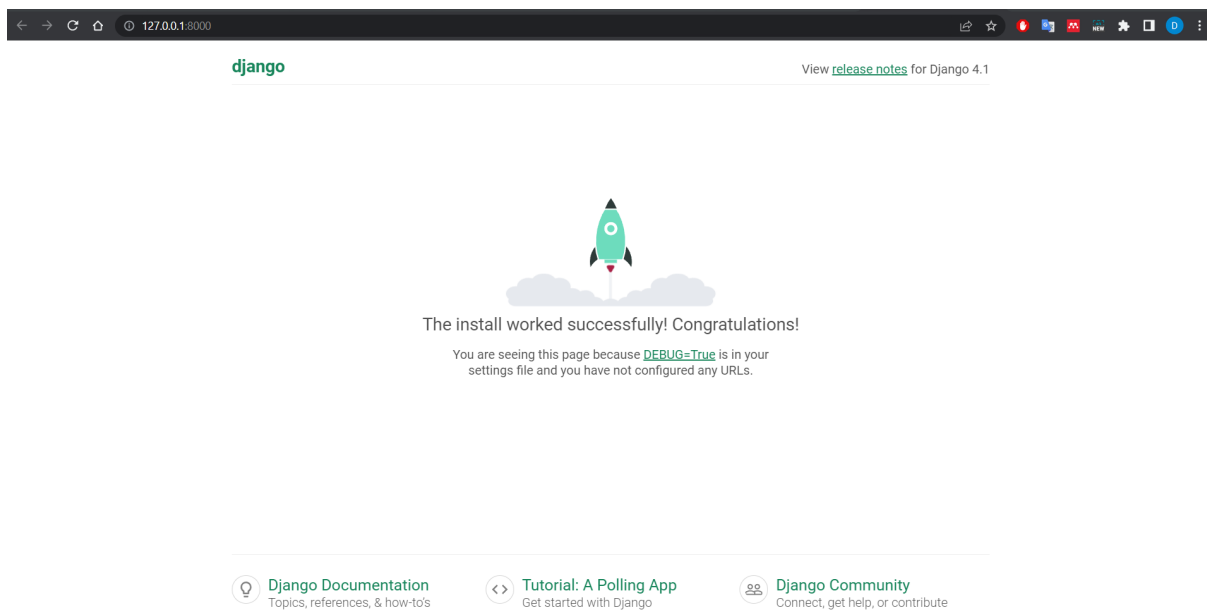
```
(venv) (base) C:\Users\diomi\TCCAACECDR>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 27, 2022 - 22:06:17
Django version 4.1.2, using settings 'TCCAACECDR.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
□
```

Fonte: Autoria Própria

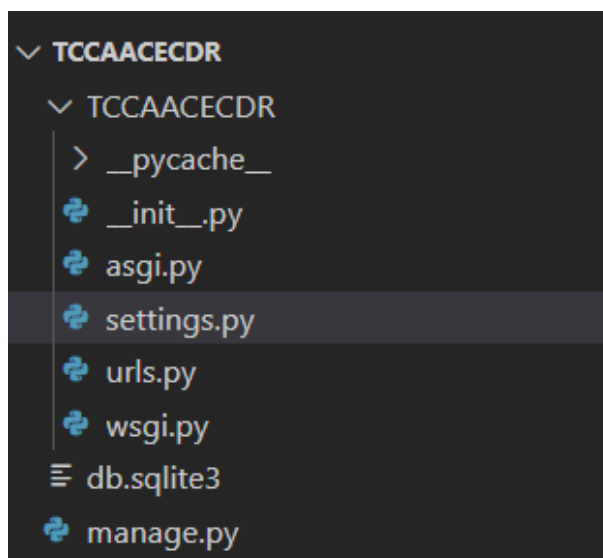
Figura 24 – Instalação bem sucedida



Fonte: Autoria Própria

A estrutura inicial da aplicação é mostrada na Figura 25:

Figura 25 – Estrutura inicial da aplicação



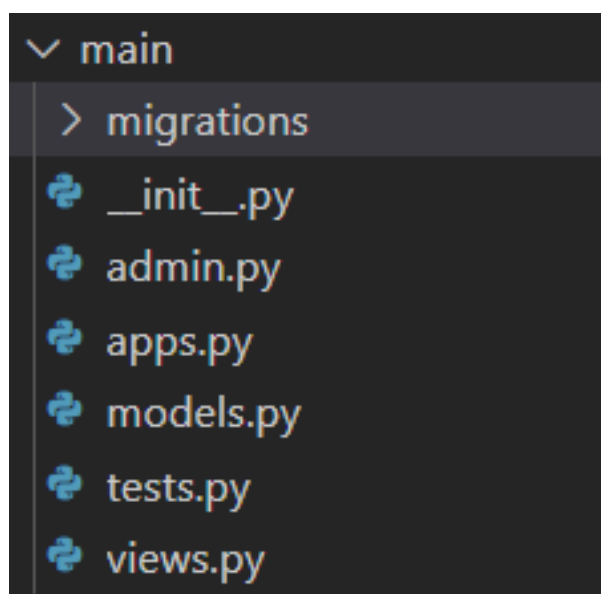
Fonte: Autoria Própria

A partir deste ponto, já é possível dar início ao desenvolvimento da aplicação.

## 4.2 Criação da Interface web com o Django

Na estrutura da aplicação, é criada uma pasta chamada "main", onde toda a aplicação Django capaz de calcular e mostrar os resultados serão armazenados. Para criar essa pasta, é necessário executar o comando "django-admin startapp main", o que irá gerar alguns arquivos Python, como mostrado na [Figura 26](#):

Figura 26 – Pasta main



Fonte: Autoria Própria

Dentre esses arquivos criados, "urls.py" é o arquivo python responsável por gerenciar as URL's de todas as páginas criadas pelo usuário. A [Figura 27](#) mostra como ele é configurado.

Figura 27 – URL main

```
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     path('sobre', views.sobrepagem, name='sobre'),
10 ]
```

Fonte: Autoria Própria

A página "index" contém um formulário que deve ser preenchido pelo usuário. É importante preencher todos os campos corretamente, para evitar erros na aplicação.

Figura 28 – Página Index (Inserção de dados)

The screenshot shows a web interface for data entry. At the top right, there is a blue button labeled 'Sobre'. The main title is 'Análise de Crateramento em Engrenagens Cilíndricas de Dentes Retos'. Below the title, there are several input fields and dropdown menus. The first field is labeled '{% csrf\_token %}' and 'Insira o valor da relação de transmissão' with the value '0'. The second field is 'Insira o valor do módulo'. The third field is 'Insira o valor da rotação em [RPM]' with the value '0'. The fourth field is 'Insira a quantidade de dentes do pinhão'. The fifth field is 'Insira o tempo de vida [H]' with the value '0'. The sixth field is a dropdown menu labeled 'Escolha o dado de entrada' with the selected option 'Torque'. The seventh field is 'Digite o valor do torque [N.m]' with the value '0'. The eighth field is a dropdown menu labeled 'Escolha o material' with the selected option 'Aço x Aço'.

Fonte: Autoria Própria

Figura 29 – Página Index (Escolha de dados listados)

The screenshot shows a web interface for data selection. It features several dropdown menus and input fields. The first dropdown is labeled 'Fator de formato da face do dente (Khmc)' with the selected option 'Dente sem coroamento'. The second dropdown is labeled 'Fator de carga de deflexão (Khpm)' with the selected option 'S1/S < 0.175'. The third dropdown is labeled 'Fator de alinhamento' with the selected option 'Engrenagem aberta'. The fourth dropdown is labeled 'Fator de ajuste (Khe)' with the selected option 'Engrenagem e ajustada na montagem'. The fifth dropdown is labeled 'Fator de confiabilidade' with the selected option '0,50'. The sixth field is labeled 'Digite o valor da dureza' with the value '0'. At the bottom, there is a green button labeled 'Calcular' with a calculator icon.

Fonte: Autoria Própria

A página "results" é responsável por exibir os resultados, calculados com base nas informações fornecidas pelo usuário na página "index". Após preencher todos os campos, o usuário deve enviar os dados para cálculo pressionando o botão "calcular". Assim, os resultados serão exibidos na tela.

Figura 30 – Página Results

Dados de Entrada	
Variável	Valor
Relação de Transmissão	{{info_reltrans1}}
Módulo	{{info_modulo1}}
Material	{{info_material1}}
Rotação	{{info_rpm1}}
Ze	{{info_Ze}}
Kh	{{info_Kh}}
Khma	{{info_Khma}}
Vp	{{info_Vp}}
Khpf	{{info_Khpf}}
Kv	{{info_Kv}}
Ks	{{info_Ks}}
Zi	{{info_Zi}}
ohpf	{{info_ohpf}}
HB	{{info_HB}}
Zn	{{info_Zn1}}
ohp	{{info_ohp}}

**Resultado**

Fonte: Autoria Própria

A página "error" é chamada sempre que uma exceção ocorre no programa, como por exemplo, devido a campos não preenchidos ou dados inseridos que não podem ser calculados pelo código. Com o Debug ativo no Django, é possível identificar e avaliar essas exceções durante o desenvolvimento do código. A página "error" foi criada para substituir a página de "debug", pois essas informações são úteis apenas para o desenvolvedor do código e não são relevantes para o usuário final.

Figura 31 – Página Erro



Fonte: Autoria Própria

Para esta página de erro ser mostrada ao usuário na função "results" que fica na página "views.py", é realizado um tratamento de exceção. Este tratamento se baseia nos parâmetros "try" e "except". Desta maneira, quando o código chega na função "results" ele tenta rodar o código que está dentro da função com o "try" como na [Figura 32](#).

Figura 32 – Parâmetro Try

```
def results(request):
    try:
        i = request.POST['p_transmissao']
        m = request.POST['p_modulo']
        n = request.POST['p_rpm']
        Np = request.POST['dentes_pinhao']
        L = request.POST['t_vida']
        T = request.POST['p_torque']
        P = request.POST['p_potencia']
        Wt = request.POST['p_forca_tangencial']
        Mat = request.POST['material']
```

Fonte: Autoria Própria



Se algum erro for detectado no código, a execução é interrompida e direcionada para o bloco de tratamento de exceção ("except"), onde o usuário é redirecionado para a página de erro correspondente, ver [Figura 33](#)

Figura 33 – Parâmetro Except

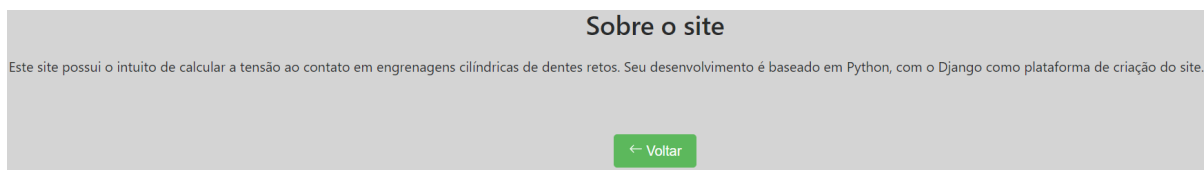
```
# Tratamento de exceção
except:
    return render(request, "error.html")
```

Fonte: Autoria Própria

Ao ser executado sem problemas, o código retorna a página "results" ao usuário. Contudo, se ocorrer algum erro durante a execução, a página "error" é exibida.

A página "sobre" contém informações sobre o site do Sistema Especialista (SE).

Figura 34 – Página Sobre

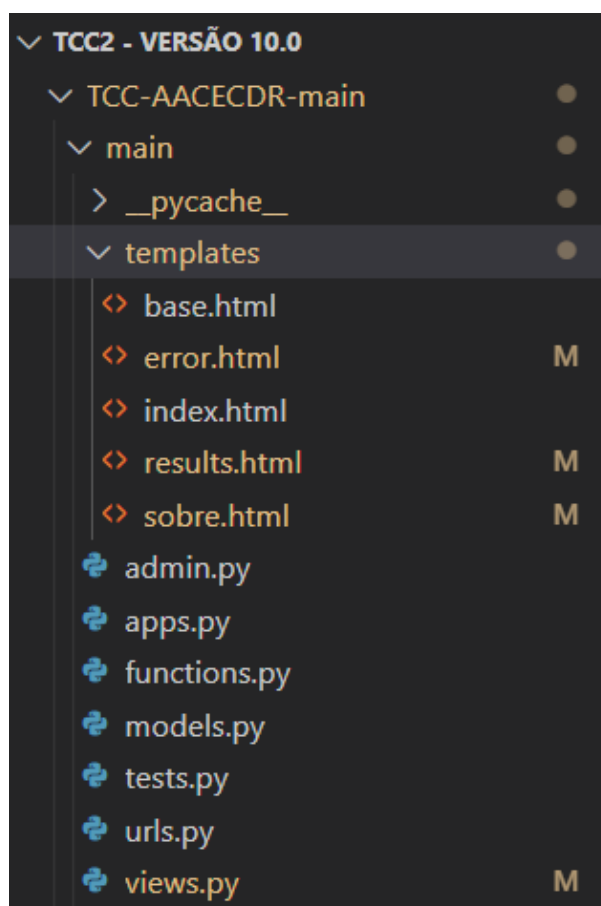


Fonte: Autoria Própria

Para que a aplicação "main" funcione, ela deve ser adicionada no arquivo "settings.py" da pasta "TCCAACECDR" através do comando "INSTALLED\_APPS = [" e incluindo "'main.apps.MainConfig',".

Na pasta "main" é criada uma pasta "templates" para armazenar os arquivos HTML, incluindo as páginas "index.html", "results.html", "error.html" e "sobre.html". Além disso, é criada a página "base.html" que contém a formatação básica para todas as páginas e o corpo do HTML, que é estendido para as outras páginas através do comando "% extends 'main/base.html' %". Com isso, é possível desenvolver o código para as páginas "index", "results", "error" e "sobre".

Figura 35 – Estrutura da Pasta "main" e "templates"



Fonte: Autoria Própria

### 4.3 Coleta de dados e exibição de resultados

Para que o código seja executado, é necessário que o usuário forneça os dados necessários nos campos correspondentes. Para isso, é utilizado o método "POST" na página "index.html" para receber esses dados. Cada campo tem um nome único que não deve ser repetido, pois é através dele que é identificado. A [Figura 36](#) mostra um exemplo de campo na página "index.html" que pode ser preenchido tanto por digitação quanto por escolha em lista. Os campos podem ter estruturas diferentes, mas possuem configurações semelhantes.

Figura 36 – Campo da página "index.html"

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

  <div class="form-group">
    <div class="row">
      <div class="col-6">
        <label for="L1_C1"> Insira o valor da relação de transmissão </label>
        <input type="float" class="form-control w-100" id="L1_C1" name="p_transmissao" value="0">
      </div>
      <div class="col-6">
        <label for="L1_C2"> Insira o valor do módulo </label>
        <input type="float" class="form-control w-100" id="L1_C2" name="p_modulo">
      </div>
    </div>
  </div>
</form>
```

Fonte: Autoria Própria

Em seguida, o comando "request.POST" é utilizado dentro da função "results" no arquivo "views.py" com o objetivo de atribuir os dados coletados pela página "index.html" às variáveis, para que possam ser utilizadas nas equações para o desenvolvimento do cálculo. Isso pode ser visualizado na [Figura 37](#).

Figura 37 – request.POST

```
def results(request):
    try:
        i = request.POST['p_transmissao']
        m = request.POST['p_modulo']
        n = request.POST['p_rpm']
        Np = request.POST['dentes_pinhao']
        L = request.POST['t_vida']
        T = request.POST['p_torque']
        P = request.POST['p_potencia']
        Wt = request.POST['p_forca_tangencial']
        Mat = request.POST['material']
        Ep = request.POST['E_p']
        Eg = request.POST['E_g']
        v1 = request.POST['poison_p']
        v2 = request.POST['poison_c']
        FA = request.POST['fontealimentacao']
        TM = request.POST['tipomaquina']
        Av = request.POST['carga']
        Khmc_z = request.POST['khmc']
        Khpm_z = request.POST['khpm']
        FA1 = request.POST['fatoralinhamento']
        Khe_z = request.POST['khe']
        Rr = request.POST['confiabilidade']
        HBt = request.POST['dureza']
        a = request.POST['dado_entrada1']
```

Fonte: Autoria Própria

A partir do momento que essas variáveis possuem dados atribuídos, é possível definir

diversas funções no arquivo "functions.py", este arquivo é responsável por realizar todos os cálculos e retornar os resultados. Toda a parte da coleta e tratamento de dados fica localizada na função "results" do arquivo "views.py" e no arquivo "functions.py". 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 a uma variável que posteriormente será exibida na página de resultados, junto com os dados de entrada utilizados e outras informações de interesse.

Figura 38 – Chamando os dados e atribuindo resultados

```
Zr1 = Zr
θ1 = θ
Zi1 = FatorGeometrico(Np,θ1,i)
b1 = LargFace(m)
Khp1 = fatorkhp1(b1, Dpp1)
Khma1 = FatorAlinhamento(FAL, b1)
Sh1 = Sh
Yz1 = FatorConfiabilidade(R)
Zn1 = FatorCiclagem(L,n,q)
Kh1 = FatorCarga(Khmc, Khpm, Khma1, Khe, Khpf1)
σH1 = sigmaH(Wt, Ko, Kv1, Ks1, Kh1, Zr1, Dpp1, b1, Zi1, Ze)
σhp1 = σhp12(σH1, Sh1, Yθ, Yz1, Zn1, Zw)
σhpf1 = Brinnel(HBt)
HB1 = Brinnel(σhp1)

# Resultado final
result = resultado(σhp1, Zn1, Zw, σH1, Yθ, Yz1, Wt, Ko, Kv1, Ks1, Kh1, Zr, Dpp1, b1, Zi1, Ze)
return render(request, "results.html", {"info_reltrans1": info_reltrans1, "info_modulo1": info_modulo1, "info_material1":
```

Fonte: Autoria Própria

#### 4.4 Deploy da aplicação no Heroku

O projeto da aplicação web em DJANGO encontra-se disponível na plataforma GitHub através do link <https://github.com/DMiotto/TCCAACECDR>. Para subir a aplicação para a nuvem, foi utilizado o Heroku que é uma plataforma de nuvem como serviço que permite subir aplicações para a web.

Para realizar o *Deploy*, foi aberta uma conta no site Heroku e criada uma aplicação escolhendo um servidor. Em seguida, foi instalado o Heroku CLI (Command Line Interface), que permite criar e gerenciar as aplicações Heroku diretamente pelo terminal.

Com o CLI instalado, foi iniciado o processo de *Deploy*, com o comando "Heroku login" para dar início. Na pasta "settings.py" do projeto foi adicionado as bibliotecas para configurar o projeto:

```
import django_heroku
django_heroku.settings(locals())
```

Na raiz da aplicação, foram adicionados os arquivos "Procfile" e "requirements.txt". O

arquivo "Procfile" define qual arquivo será utilizado para processar as requisições da aplicação, enquanto o "requirements.txt" instala os pacotes necessários para rodar a aplicação.

Com tudo pronto, para subir a aplicação, o comando:

```
git init
```

inicia um *branch* novo e

```
heroku git:remote -a tccaacecdr22
```

inicia a aplicação.

Para finalizar, o comando:

```
git add .
```

adiciona todas as modificações ao *branch* e o comando:

```
git commit -m ''Deploy da aplicação''
```

faz o deploy delas. Por fim,

```
git push -u heroku master
```

sobe a aplicação pronta para a nuvem.

O link da aplicação <https://tccaacecdr22.herokuapp.com/>.

#### 4.5 Estudo de caso

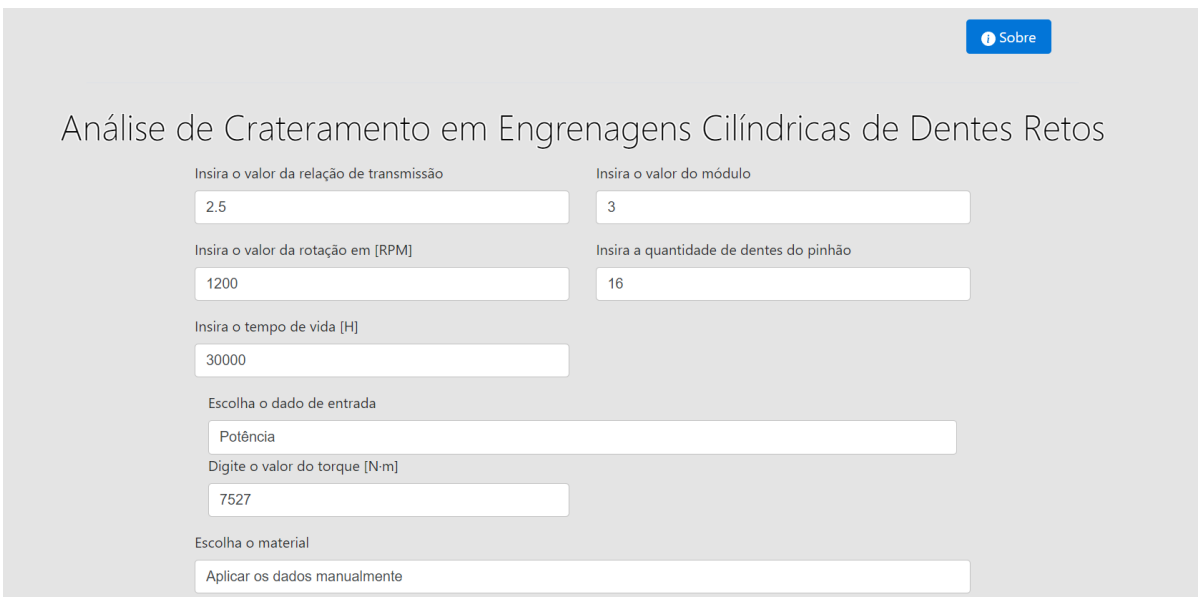
Este estudo de caso tem como objetivo validar os cálculos para comprovar sua precisão. Os dados foram obtidos a partir do artigo de Nunez (2022). Neste estudo de caso é analisado um sistema de transmissão de uma bomba centrífuga de baixa velocidade que é acionada por um motor elétrico. Para mais detalhes, ver o artigo citado anteriormente. Os dados necessários para o cálculo são dados a seguir:

- Pinhão cilíndrico com 16 dentes retos;
- Coroa com 40 dentes;
- Relação de transmissão de 2,5;
- Rotação do pinhão de 1.200 rpm;
- Diâmetro do eixo de 20 mm;
- Potência no pinhão de 7.527 W;
- Material da engrenagem em aço AISI 1045 trefilado a frio com diâmetro de 50-75mm, módulo de elasticidade 206 GPa e coeficiente de Poisson de 0,29;
- Fonte de alimentação motor elétrico uniforme;
- Máquina acionada bomba centrífuga de baixa velocidade com choque leve;

- Nível de precisão da transmissão AV igual a 10;
- Fator de carga de deflexão, considerando a engrenagem no meio dos mancais,  $S1/S < 0,175$ ;
- Fator de alinhamento de engrenagem, considerada engrenagem aberta, pois os mancais estão montados na mesma estrutura da máquina;
- Não há ajuste na montagem ou lapidação no dente, considera-se como fator de ajuste;
- Fator de temperatura, é considerado que a engrenagem não trabalhará acima de 120°C;
- Vida nominal é considerada como 30 000 h de trabalho para um equipamento industrial;

Tratando da dureza na superfície do dente, como a dureza não foi indicada, deve ser encontrada a dureza Brinell padronizada para a engrenagem de aço. O que diferencia o programa do método utilizado por Nunez (2022) é que se requisita uma dureza estimada, dessa maneira no primeiro cálculo retornara um fator de segurança estimado na dureza inserida, caso o fator de segurança fique abaixo de 1 o usuário recebera uma dureza sugerida para normalizar o fator de segurança, e para um fator de segurança maior que 5 o mesmo ocorre para evitar uso de material desnecessário, logo o usuário deve retornar aos cálculos e rever a dureza escolhida ate encontrar o melhor valor para seu fim.

Figura 39 – Pagina inicial



Insira o valor da relação de transmissão

2.5

Insira o valor do módulo

3

Insira o valor da rotação em [RPM]

1200

Insira a quantidade de dentes do pinhão

16

Insira o tempo de vida [H]

30000

Escolha o dado de entrada

Potência

Digite o valor do torque [N.m]

7527

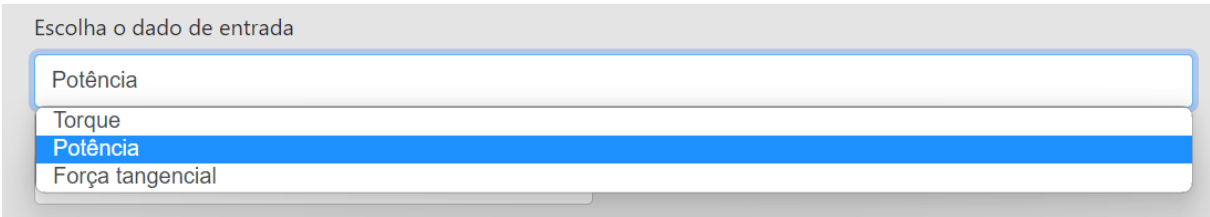
Escolha o material

Aplicar os dados manualmente

Fonte: Aatoria Própria

Na seção "Escolha o dado de entrada" o usuário pode escolher entre as 3 opções da figura abaixo, escolhendo uma, as outras duas serão calculadas de maneira automática independente da escolha.

Figura 40 – Escolha do dado de entrada



Escolha o dado de entrada

Potência

Torque

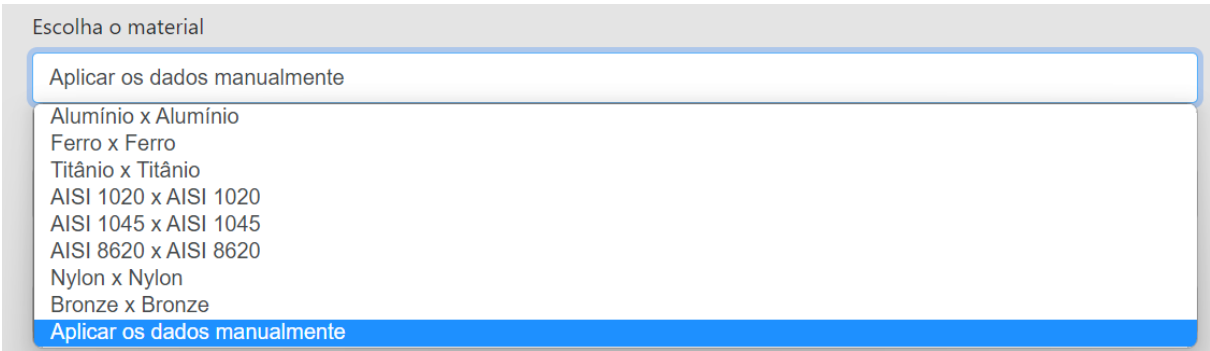
Potência

Força tangencial

Fonte: Autoria Própria

O mesmo vale para "Escolha o material", onde o usuário tem alguns materiais padrão no uso em engrenagens cilíndricas de dentes retos, mas também pode optar por aplicar os dados manualmente como foi o caso do estudo, ver [Figura 42](#).

Figura 41 – Escolha do material



Escolha o material

Aplicar os dados manualmente

Alumínio x Alumínio

Ferro x Ferro

Titânio x Titânio

AISI 1020 x AISI 1020

AISI 1045 x AISI 1045

AISI 8620 x AISI 8620

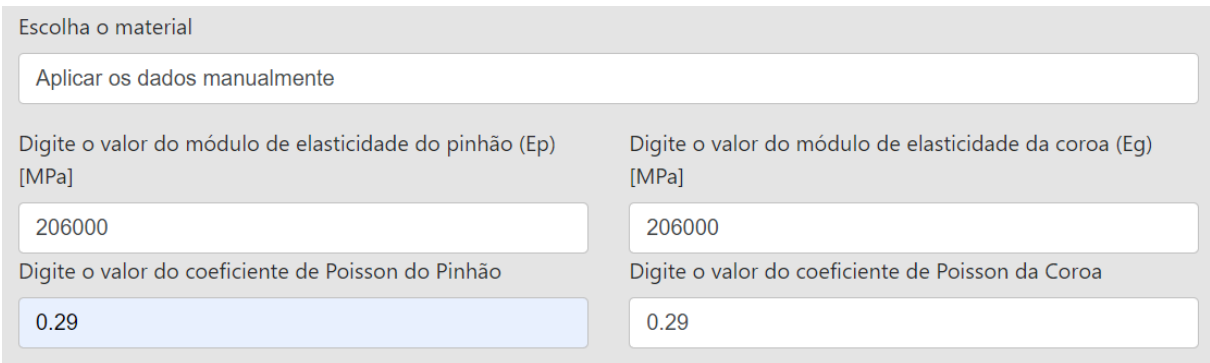
Nylon x Nylon

Bronze x Bronze

Aplicar os dados manualmente

Fonte: Autoria Própria

Figura 42 – Aplicando manualmente



Escolha o material

Aplicar os dados manualmente

Digite o valor do módulo de elasticidade do pinhão (Ep) [MPa]

206000

Digite o valor do módulo de elasticidade da coroa (Eg) [MPa]

206000

Digite o valor do coeficiente de Poisson do Pinhão

0.29

Digite o valor do coeficiente de Poisson da Coroa

0.29

Fonte: Autoria Própria

De maneira intuitiva, o programa pede a fonte de alimentação que será utilizada, sendo as opções mostradas na [Figura 44](#), onde:

Figura 43 – Fonte de alimentação e Tipo de máquina



Escolha a fonte de alimentação

Uniforme

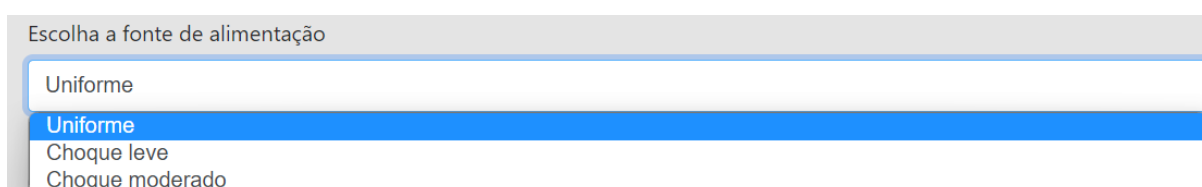
Escolha o tipo de máquina

Choque leve

Fonte: Autoria Própria

- Fonte de alimentação uniforme: Quando se trata de uma fonte de alimentação uniforme para engrenagens, pode-se pensar em uma fonte de alimentação elétrica que forneça energia constante e estável para um motor elétrico que aciona as engrenagens. Isso pode ser alcançado por meio de uma fonte de alimentação de corrente contínua (DC) ou corrente alternada (AC) que possua uma tensão e corrente nominal adequada para o motor;
- Fonte de alimentação de choque leve: Engrenagens que estão sujeitas a cargas leves de choque podem ser alimentadas por uma fonte de alimentação que fornece energia suave e uniforme, sem picos de energia repentinos que possam afetar a integridade das engrenagens. Um exemplo de fonte de alimentação de choque leve pode ser um sistema hidráulico que controla a velocidade do motor elétrico que aciona as engrenagens;
- Fonte de alimentação de choque moderado: Para engrenagens que estão sujeitas a cargas mais pesadas, uma fonte de alimentação de choque moderado pode ser necessária para garantir que elas funcionem de maneira confiável e segura. Uma fonte de alimentação de choque moderado pode ser um sistema pneumático que fornece um fluxo de ar comprimido para um motor pneumático que aciona as engrenagens. O ar comprimido fornece uma fonte de energia controlada que pode lidar com cargas de choque moderado sem danificar as engrenagens.

Figura 44 – Fonte de alimentação



Escolha a fonte de alimentação

Uniforme

Uniforme

Choque leve

Choque moderado

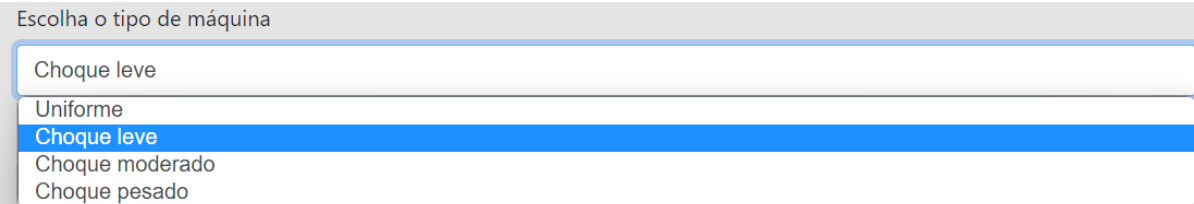
Fonte: Autoria Própria

Quanto ao tipo de máquina o mesmo ocorre:



- Máquina de velocidade uniforme: Exemplos de máquinas de velocidade uniforme incluem tornos, fresadoras, retificadoras, entre outras. As engrenagens nesses tipos de máquinas podem ser alimentadas por uma fonte de alimentação elétrica de corrente contínua ou corrente alternada que fornece energia constante e estável para o motor elétrico que aciona as engrenagens;
- Máquina de choque leve: Exemplos de máquinas de choque leve incluem impressoras, scanners, máquinas de costura, entre outras. As engrenagens nesses tipos de máquinas podem ser alimentadas por uma fonte de alimentação elétrica de corrente contínua ou corrente alternada que fornece energia suave e uniforme, sem picos de energia repentinos que possam afetar a integridade das engrenagens;
- Máquina de choque moderado: Exemplos de máquinas de choque moderado incluem cortadores de grama, trituradores de madeira, compactadores, entre outras. As engrenagens nesses tipos de máquinas podem ser alimentadas por uma fonte de alimentação hidráulica que fornece um fluxo controlado de óleo hidráulico para um motor hidráulico que aciona as engrenagens. O óleo hidráulico fornece uma fonte de energia que pode lidar com cargas de choque moderado sem danificar as engrenagens;
- Máquina de choque pesado: Para uma máquina que está sujeita a cargas pesadas de choque, como um caminhão ou uma escavadeira, pode ser necessário fornecer uma fonte de alimentação que possa lidar com cargas pesadas e extremas. Um motor diesel ou um motor elétrico de alta potência alimentado por uma fonte de alimentação de corrente contínua ou corrente alternada pode ser uma opção adequada para lidar com as cargas pesadas das engrenagens.

Figura 45 – Tipo de máquina



Escolha o tipo de máquina

Choque leve

Uniforme

Choque leve

Choque moderado

Choque pesado

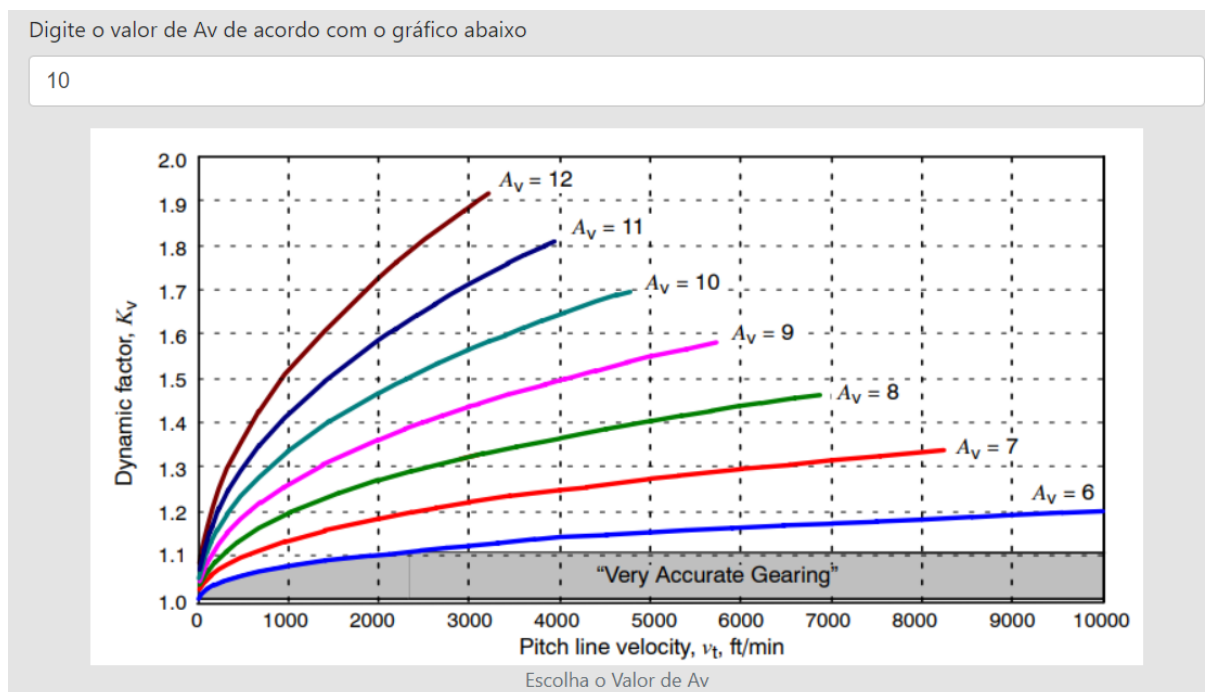
Fonte: Autoria Própria

O fator dinâmico  $K_v$  tenta compensar erros de manufatura e operação como:

- Erros geométricos e dimensionais nos dentes resultantes da fabricação;
- Vibração do dente durante o engrenamento devido a sua rigidez;
- Desbalanceamento do elementos em rotação;
- Desgaste e deformação plástica na face de contato;
- Desalinhamentos lineares e angulares, resultantes da flexão dos eixos;
- Atrito nos dentes

Para corrigir esses erros, a AGMA estabeleceu graus de qualidade (números de qualidade)  $A_v$ , que determinam as tolerâncias para engrenagens de diferentes tamanhos. Esses graus são classificados em 3 a 7 para engrenagens comerciais e de 8 a 12 para engrenagens de precisão.

Figura 46 –  $K_v$  - Fator de qualidade  $A_v$




Fonte: Autoria Própria

Figura 47 – Demais cálculos

Fator de formato da face do dente (Khmc)
Dente sem coroamento
Fator de carga de deflexão (Khpm)
$S1/S < 0.175$
Fator de alinhamento
Engrenagem aberta
Fator de ajuste (Khe)
Todos os outros casos
Fator de confiabilidade
0,99
Digite o valor da dureza
449.79

Fonte: Autoria Própria

Figura 48 – Calcular

 Calcular
--

Fonte: Autoria Própria

Figura 49 – Resultado

Resultado
Tensão ao contato AGMA 1079.35. Fator de segurança recalculado 1.03 (Padrão aplicado:1.0)

Fonte: Autoria Própria

O Valor da tensão ao contato agma difere dos 1106,36 MPa obtidos por [Nunez \(2022\)](#), pois lá é considerado o cálculo de Ks pelo fator de forma de Lewis obtendo Ks 1.05 e no programa foi considerado o fator de tamanho segundo o módulo com base na biografia de [Mott \(2013\)](#).

## 5 CONCLUSÃO

Nos dias de hoje, as engrenagens cilíndricas de dentes retos são amplamente utilizadas em uma gama de aplicações, desde equipamentos industriais até veículos automotores. O sucesso de sua operação depende de uma série de fatores, incluindo o cálculo da tensão ao contato AGMA. Foi nesse contexto que foi desenvolvido um Sistema Especialista (SE) em Django para realizar esses cálculos, integrando a linguagem de programação Python. Além de ser intuitivo e acessível para os usuários, este Sistema Especialista (SE) oferece resultados precisos e confiáveis. O *deploy* do Sistema Especialista (SE) no Heroku garante sua disponibilidade constante, tornando-o uma ferramenta valiosa para engenheiros, técnicos e estudiosos na área de engrenagens.

A integração dos cálculos da análise de tensão ao contato em engrenagens cilíndricas de dentes retos com as tecnologias da web é um exemplo de como a tecnologia pode ser utilizada para criar soluções eficientes e úteis. Além disso, essa integração permite a automação de processos, o que resulta em ganhos de tempo e aumento da eficiência.

Também é uma ferramenta valiosa para profissionais, este Sistema Especialista (SE) também possui um grande potencial educacional. Ele pode ser utilizado como uma ferramenta de ensino para estudantes de engenharia mecânica e profissionais em treinamento, ajudando-os a compreender os cálculos da tensão ao contato e sua importância para a operação eficiente das engrenagens. Além disso, a integração da tecnologia e da linguagem de programação Python no Sistema Especialista (SE), ajudará os estudantes a desenvolver habilidades técnicas valiosas e a entender como a tecnologia pode ser utilizada para solucionar problemas práticos.

Além de ser uma ferramenta útil e eficiente para profissionais na área de engrenagens, o Sistema Especialista (SE) desenvolvido em Django para o cálculo da tensão ao contato em engrenagens cilíndricas de dentes retos também tem um forte valor educacional. Ele é uma excelente ferramenta para ensinar os princípios da análise de tensão ao contato em engrenagens de forma interativa e acessível. Os estudantes e professores podem experimentar e ver como os diferentes parâmetros afetam os resultados da análise, tornando-os mais efetivos na compreensão dos conceitos subjacentes.

Ademais, a integração do cálculo com as tecnologias da web permite aos estudantes aprender sobre programação em Python e desenvolvimento web em um contexto relevante e aplicável. Isso ajuda a desenvolver habilidades em tecnologia e ciência, além de ser uma oportunidade para aprender sobre a criação e implementação de soluções tecnológicas na prática.

Em resumo, o SE desenvolvido em Django para o cálculo da tensão ao contato em engrenagens cilíndricas de dentes retos é uma ferramenta útil e eficiente para profissionais na área de engrenagens, além de ter um forte valor educacional. Ele é uma excelente ferramenta para ensinar princípios de análise de tensão ao contato em engrenagens de forma interativa e

acessível, e ajuda a desenvolver habilidades em tecnologia e ciência através da integração com Python e tecnologias da web.

Considerando os benefícios educacionais e profissionais do Sistema Especialista (SE) desenvolvido em Django para o cálculo da tensão ao contato em engrenagens cilíndricas de dentes retos, um possível trabalho futuro seria a expansão da plataforma para incluir outras análises e cálculos relevantes para a área de engrenagens. Isso poderia envolver a inclusão de recursos para o cálculo da carga de torção, desgaste de dentes, vida útil da engrenagem e outros fatores importantes que influenciam a operação eficiente das engrenagens.

Outra possível melhoria, seria a inclusão de uma folha com o memorial de cálculos feitos pelo programa. Isso permitiria que os usuários tivessem acesso a todas as informações utilizadas no cálculo, incluindo as equações e parâmetros, possibilitando uma análise mais detalhada dos resultados.

Além disso, essa funcionalidade seria especialmente útil para estudantes e profissionais em treinamento, pois permitiria que eles visualizassem todos os passos do cálculo e entendessem melhor como as equações foram aplicadas para chegar aos resultados. Isso ajudaria a fortalecer o aprendizado e a desenvolver habilidades mais profundas na área de engrenagens.

A folha de memorial de cálculos também seria útil para profissionais que precisam justificar suas decisões de projeto para clientes ou colegas de trabalho. Eles poderiam fornecer evidências concretas para apoiar seus resultados e tornar sua análise mais transparente e confiável.

Para expandir ainda mais essa funcionalidade, seria possível incluir opções para exportar o memorial de cálculos em diferentes formatos, como PDF, CSV ou Excel, para que os usuários possam facilmente compartilhar seus resultados com outras pessoas ou integrá-los em relatórios de projeto.

Por fim, a implementação de um mecanismo de validação de forms seria ideal para ao usuário um feedback (ex: qual campo está preenchido errado) sem direcioná-lo para uma página de erros.

## Referências

- ALVES Ítalo C. et al. **Sistemas especialistas**. 1. ed. Porto Alegre: SAGAH, 2019. ISBN 978-65-5690-093-3. Citado na página 33.
- ANACONDA. **Anaconda**. 2021. Disponível em: <<https://www.anaconda.com/>>. Acesso em: 17 de março de 2021. Citado na página 31.
- ANSI/AGMA-2101-D04. **Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth (Metric)**. [S.l.], 2016. Citado 4 vezes nas páginas 20, 22, 25 e 38.
- ASI, O. Failure analysis of a secondary driving helical gear in transmission of electric vehicle. **Engineering Failure Analysis**, Pergamon, v. 13, n. 7, p. 1116–1125, 2006. Citado na página 13.
- BHATIA, A. **Basic Fundamentals of Gears Drives**. [S.l.], 2014. 79 p. Disponível em: <<https://www.cedengineering.com/userfiles/Basic%20Fundamentals%20of%20Gear%20Drives%20R1.pdf>>. Citado na página 13.
- BUDYNAS, R. G.; NISBETT, J. K. **Elementos de máquinas de Shigley**. 10. ed. Porto Alegre: AMGH, 2016. Citado 6 vezes nas páginas 13, 16, 17, 18, 21 e 38.
- CARDOSO, N. S.; BISPO, T. M. da S. **Um estudo comparativo entre os principais frameworks de desenvolvimento web em linguagem Python**. 2019. Monografia (Bacharel em Sistemas de Informação), Universidade Federal Rural do Amazonas, Belém. Citado na página 35.
- CASS, S. **Top Programming Languages 2020**. 2020. Disponível em: <<https://spectrum.ieee.org/at-work/tech-careers/top-programming-language-2020>>. Acesso em: 05 de abril de 2021. Citado na página 26.
- C.Y.CHEN, P.; POO, A.-N. **Encyclopedia of Information Systems: Engineering, artificial intelligence** in. [S.l.]: Elsevier, 2003. 141-155 p. Citado na página 32.
- DAI, H. et al. An improved analytical model for gear mesh stiffness calculation. **Mechanism and Machine Theory**, v. 159, n. 104262, 2021. Citado na página 13.
- D.THOMAS et al. Cupydo - an integrated python environment for coupled fluid-structure simulations. **Advances in Engineering Software**, v. 128, p. 69–85, 2019. Citado na página 28.
- E.ARONSON, J. **Encyclopedia of Information Systems: Expert systems**. [S.l.]: Elsevier, 2003. 277-289 p. Citado na página 32.
- F.EGER et al. Reaching zero-defect manufacturing by compensation of dimensional deviations in the manufacturing of rotating hollow parts. **Procedia Manufacturing**, Elsevier, v. 51, p. 388–393, 2020. Citado na página 37.
- FENG, W.; FENG, Z.; MAO, L. Failure analysis of a secondary driving helical gear in transmission of electric vehicle. **Engineering Failure Analysis**, v. 117, n. 104934, 2020. Citado na página 13.

- FOUNDATION, D. S. **Documentação Django**. 2021. Disponível em: <<https://docs.djangoproject.com/en/3.2/>>. Acesso em: 07 de abril de 2021. Citado 2 vezes nas páginas 35 e 38.
- GAMA, E. et al. **Padrões de projeto**: Soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2007. Citado na página 35.
- GARCIA, J. G. M. **Desenvolvimento de uma plataforma web para aplicações de cálculo estrutural**. Junho 2012. 77 f. Dissertação (Mestrado em Engenharia Civil - Especialização em Estruturas) — Faculdade de Engenharia Universidade do Porto, Porto, 2010. Citado na página 14.
- GEUS, T. et al. Finite strain fft-based non-linear solvers made simple. **Computer Methods in Applied Mechanics and Engineering**, v. 318, p. 412–430, 2017. Citado na página 13.
- GONÇALVES, A. **Inteligência artificial e sistemas especialistas**.: Administradores. 2009. Disponível em: <<http://www.administradores.com.br/artigos/marketing/inteligencia-artificial-e-sistemas-especialistas/33860/>>. Acesso em: 28 de março de 2022. Citado 2 vezes nas páginas 32 e 34.
- GOOGLE. **Google Trends**. 2022. Disponível em: <<https://trends.google.com.br/trends>>. Acesso em: 14 de setembro de 2022. Citado 2 vezes nas páginas 27 e 28.
- GUAN, X.; BURTON, H.; SABOL, T. Python-based computational platform to automate seismic design, nonlinear structural model construction and analysis of steel moment resisting frames. **Engineering Structures**, v. 224, n. 111199, 2020. Citado na página 28.
- GUHA, S. et al. Matscie: An automated tool for the generation of databases of methods and parameters used in the computational materials science literature. **Computational Materials Science**, Elsevier, v. 192, n. 110325, 2021. Citado na página 36.
- HOROWITZ, E. **Handbook of Computer Science and Engineering**: Introduction to python for computational science and engineering. [S.l.: s.n.], 2008. 86-110 p. Citado na página 26.
- HOSMER, C. **Python Forensics**. [S.l.]: Elsevier, 2014. 1-11 p. Citado na página 26.
- JEFFERY, M. What would an integrated development environment for law look like? **MIT Computational Law Report**, 4 2020. Disponível em: <<https://law.mit.edu/pub/whatwouldanintegrateddevelopmentenvironmentforlawlooklike>>. Citado na página 29.
- JETBRAINS. **O IDE Python para desenvolvedores profissionais**. 2021. Disponível em: <<https://www.jetbrains.com/pt-br/pycharm/>>. Acesso em: 06 de abril de 2021. Citado 2 vezes nas páginas 30 e 31.
- JUVINALL, R. C.; MARSHEK, K. M. **Fundamentos do projeto de componentes de máquinas**. 4. ed. Rio de Janeiro: LTC, 2013. Citado 2 vezes nas páginas 16 e 38.
- LOW, J. S. C. et al. A collaboration platform for enabling industrial symbiosis: Application of the database engine for waste-to-resource matching. **Procedia CIRP**, Elsevier, v. 69, p. 849–854, 2018. Citado na página 36.
- MENDES, R. D. **Inteligência artificial: Sistemas especialistas no gerenciamento da informação**. 1997. Citado na página 34.

- MICROSOFT. **Visual Studio Code**. 2021. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 06 de abril de 2021. Citado 2 vezes nas páginas 29 e 30.
- MILLMAN, K. J.; AIVAZIS, M. **Python for Scientists and Engineers**. [S.l.: s.n.], 2011. 9-12 p. Citado na página 26.
- MOTT, R. L. **Machine Elements in Mechanical Design**. 5. ed. [S.l.]: Pearson, 2013. Citado 2 vezes nas páginas 21 e 58.
- NEDELCO, D.; GILLICH, G.-R. A structural health monitoring python code to detect small changes in frequencies. **Mechanical Systems and Signal Processing**, v. 147, n. 107087, 2021. Citado na página 28.
- NUNEZ, D. L. Análise da resistência à flexão e ao crateramento de uma engrenagem cilíndrica de dentes retos: avaliação da eficácia da norma ansi/agma 2101-d04. 2022. Citado 9 vezes nas páginas 18, 19, 23, 24, 25, 38, 52, 53 e 58.
- PYTHON. **Python Documentation**. 2021. Disponível em: <<https://docs.python.org/3.8/>>. Acesso em: 12 de abril de 2021. Citado na página 38.
- PYTHON. **Pypi.org**. 2022. Disponível em: <<http://www.pypi.org>>. Acesso em: 22 de maio de 2023. Citado na página 26.
- RAMOS, V. **Desenvolvimento WEB com Python e Django**. 2018. Disponível em: <<https://pythonacademy.com.br/assets/ebooks/desenvolvimento-web-com-Python-e-Django/desenvolvimento-web-com-Python-e-Django.pdf>>. Acesso em: 07 de abril de 2021. Citado na página 36.
- RODRIGUES, V. S. **Estudo sobre sistemas especialistas**. Dissertação (Trabalho de Conclusão de Curso - Graduação em Sistemas de Informação) — Faculdade do Litoral Sul Paulista, Praia Grande, 2010. Disponível em: <[http://www.fals.com.br/tcc/Sistema%20de%20Informa%C3%A7%C3%A3o/2010/ESTUDO\\_SOBRE\\_SISTEMAS\\_ESPECIALIS-TAS\\_SDI2010.pdf](http://www.fals.com.br/tcc/Sistema%20de%20Informa%C3%A7%C3%A3o/2010/ESTUDO_SOBRE_SISTEMAS_ESPECIALIS-TAS_SDI2010.pdf)>. Acesso em: 16 jan. 2018. Citado 2 vezes nas páginas 32 e 34.
- RONACHER, A. **Documentação Flask**. 2021. Disponível em: <<https://flask-ptbr.readthedocs.io/en/latest/foreword.html>>. Acesso em: 07 de abril de 2021. Citado na página 35.
- R.VUNDAVILLI, P. et al. Fuzzy logic-based expert system for prediction of depth of cut in abrasive water jet machining process. **Knowledge-Based Systems**, v. 27, p. 456–464, 2012. Citado na página 14.
- SCHLICHTING, R. F. **Sistemas Especialistas – Seleção de Par de Engrenagens cilíndricas helicoidais**. 2014. Monografia (Bacharel em Engenharia Mecânica), Universidade Federal do Rio de Janeiro, Rio de Janeiro. Citado na página 14.
- SILVA, I. R. S.; SILVA, R. O. da. Linguagem de programação python python programming language. **Tecnologias em Projeção**, v. 10, n. 1, 2019. Citado na página 35.
- SILVA, M. N. **Desenvolvimento de um programa para verificação de engrenagens cilíndricas de dentes retos e helicoidais**. 2012. Monografia (Bacharel em Engenharia Mecânica), Escola de engenharia de São Carlos da Universidade de São Paulo, São Carlos. Citado na página 14.



STACKOVERFLOW. **Most loved, dreaded, and wanted**. 2023. Disponível em: <<https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-language-love-dread>>. Acesso em: 23 de maio de 2023. Citado na página 28.

TIOBE. **TIOBE Index for September 2022**. 2022. Disponível em: <<https://www.tiobe.com/tiobe-index/>>. Acesso em: 14 de setembro de 2022. Citado na página 27.