UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE GUARATINGUETÁ

LEONARDO MORENO XAVIER

Programação em *Python* para dispositivos do laboratório de estruturas do departamento de mecânica FEG/UNESP.

Guaratinguetá
2022

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE GUARATINGUETÁ

Leonardo Moreno Xavier

Programação em *Python* para dispositivos do laboratório de estruturas para o departamento de mecânica da FEG/UNESP.

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Mecânica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Mecânica

Orientador: Prof. Dr. Marcelo Sampaio Martins

Guaratinguetá - SP

2022

X3p

Xavier, Leonardo Moreno
Programação em *Python* para dispositivos do laboratório de estruturas do departamento de mecânica FEG/UNESP / Leonardo Moreno Xavier – Guaratinguetá, 2023.
66 f : il.
Bibliografia: f. 45-47

Trabalho de Graduação em Engenharia Mecânica – Universidade Estadual Paulista, Faculdade de Engenharia e Ciências de Guaratinguetá, 2023.
Orientador: Prof. Dr. Marcelo Sampaio Martins

1. Software - Desenvolvimento. 2. Laboratórios experimentais.
3. Laboratórios - Testes de materiais. I. Título.

CDU 681.3.06

Luciana Máximo
Bibliotecária/CRB-8-3595

unesp UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE GUARATINGUETÁ

LEONARDO MORENO XAVIER

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE **"GRADUADO EM ENGENHARIA MECÂNICA"**

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

Prof. Dr. CELSO EDUARDO TUNA
Coordenador

BANCA EXAMINADORA:

Prof. Dr. MARCELO SAMPAIO MARTINS
Orientador/UNESP-FEG

Prof. Me. SOFIA GLYNIADAKIS
UNESP-FEG

Prof. Me. GERALDO CÉSAR ROSÁRIO DE OLIVEIRA
UNESP-FEG

Dezembro de 2022

DADOS CURRICULARES

LEONARDO MORENO XAVIER

NASCIMENTO 09.12.1993 – Ribeirão Preto / SP

FILIAÇÃO Roberto Xavier
Vilma de Castro Moreno Xavier

Dedico este trabalho aos meus pais por todo apoio, amor e compreensão durante meus anos de graduação.

AGRADECIMENTOS

Agradeço aos meus pais, por sempre terem me apoiado em todos os momentos. Foram anos muito difíceis e sem a ajuda deles, nada disso seria possível. Agradeço também aos meus companheiros de república onde morei todos esses anos e por último ao meu orientador Marcelo Sampaio Martins pela oportunidade de fazer meu trabalho de graduação e finalmente concluir meu curso.

RESUMO

O processo de construção do conhecimento passa por processos tanto práticos como teóricos. Esse é o motivo dos alunos de engenharia mecânica terem uma carga horária voltada tanto à aulas teóricas como de laboratório. Quando é possibilitado que os alunos acompanhem fisicamente os fenômenos estudados em sala, o conhecimento se complementa de forma satisfatória e definitiva. Neste trabalho foi desenvolvido um software utilizando a linguagem de programação *Python* para a simulação de seis dos dispositivos presentes no laboratório de estruturas do dep. mec. da Faculdade de Engenharia de Guaratinguetá. Esses dispositivos foram um guindaste de torre, uma treliça plana, um tubo de tensões principais, uma viga curva, uma viga hiperestática e um vaso de pressão. Cada um desses dispositivos apresenta características únicas e necessitam de uma modelagem teórica apropriada. O conhecimento adquirido ao longo na disciplina de Resistência dos Materiais durante o curso de Engenharia Mecânica serve como base para o entendimento destes conceitos. O laboratório de estruturas tem como objetivo complementar esse conhecimento visto em sala. Sendo assim, os alunos poderão fazer uso desse software para comparar seus experimentos no laboratório de estruturas com os valores teóricos de acordo com a literatura especializada.

PALAVRAS-CHAVE: Dispositivos Didáticos Deflexão. Deformação. Software Python.

ABSTRACT

The knowledge construction process goes through both practical and theoretical processes. This is why mechanical engineering students have a careful workload in both theoretical and laboratory classes. When it is possible for students to physically follow the phenomena studied in the classroom, knowledge complements itself in a satisfactory and definitive way. In this work, it was developed a software using *Python* programming language for the simulation of six of the devices present in the FEG structures laboratory. These devices were a tower crane, a flat truss, a main stress tube, a curved beam, a hyperstatic beam and a pressure vessel. Each of these devices has unique characteristics and need an appropriate theoretical model. The knowledge acquired throughout the subject of Strength of Materials during the mechanical engineering course serves as a basis for understanding these concepts. The structure laboratory aims to complement this knowledge seen in the classroom. Therefore, students will be able to use this software to compare their experiments in the laboratory of structures with the theoretical values according to the specialized literature.

KEYWORDS: Didactic Devices. Deflection. Deformation. Software Python.

LISTA DE FIGURAS

Figura 1 – Estrutura formada por treliças 18
Figura 2 – Forças internas e externas em uma treliça 19
Figura 3 – Treliça presente no laboratório de estruturas com elementos numerados 20
Figura 4 – Modelagem geométrica Guindaste de Torre em vista de perfil 21
Figura 5 – Guindaste de Torre do laboratório de estruturas do departamento de mecânica da FEG-UNESP 21
Figura 6 – Deformação de uma viga prismática submetida a conjugados 22
Figura 7 – Representação simplificada do dispositivo de medição de medição de deformação em uma viga 23
Figura 8 – Extensômetro colado na superfície da viga 24
Figura 9 – Aplicação da carga P na viga curva 25
Figura 10 – Aplicação de um carregamento em situação real 25
Figura 11 – Extensômetros A, B e C utilizados no dispositivo de medição de tensões e deformações principais. 26
Figura 12 – Representação esquemática do dispositivo medidor de tensões e deformações principais mostrando sua montagem e a posição da roseta retangular 27
Figura 13 – Dispositivo de medição de tensões e deformações principais após montagem. 27
Figura 14 – Posicionamento dos Vasos de Pressão 28
Figura 15 – Concepção do projeto de vaso de pressão de paredes finas 29
Figura 16 – Roseta instrumentada 29
Figura 17 – Dispositivo Didático montado 30
Figura 18 – Artigos acadêmicos relacionados com a linguagem Python ao longo dos anos 31
Figura 19 – Exemplo de código para criação de uma interface utilizando a biblioteca Tkinter 31
Figura 20 – Interface criada pelo código exemplo da Figura 19 32
Figura 21 – Cálculo da carga teórica do guindaste de torre no código fonte do software do laboratório de estruturas da FEG 34
Figura 22 – Nós da treliça plana 34
Figura 23 – Cálculo da deflexão do ponto C da treliça plana dentro do código fonte 35
Figura 24 –Eixos do tubo de tensões principais 36
Figura 25 – Cálculo das tensões e deformações do tubo de tensões principais 37

Figura 26 – Cálculo da deflexão do ponto C da viga curva dentro do código fonte 38
Figura 27 – Cálculo das tensões no costado, tangencial e longitudinal do vaso de pressão no código fonte 38
Figura 28 – Código de dados de entrada dentro do código fonte para a viga hiperestática 39
Figura 29 – Exemplo de dados de entrada inseridos pelo usuário e dados de saída do software para o guindaste de torre 40
Figura 30 - Exemplo de dado de entrada inserido pelo usuário e dado de saída do software para a treliça plana 40
Figura 31 - Exemplo de dado de entrada inserido pelo usuário e dado de saída do software para o tubo de tensões principais 41
Figura 32 - Exemplo de dados de entrada inserido pelo usuário e dados de saída do software para a viga curva 41
Figura 33 - Exemplo de dados de entrada inserido pelo usuário e dados de saída do software para o vaso de pressão 42
Figura 34 - Exemplo de dados de entrada inserido pelo usuário e dados de saída do software para a viga hiperestática 43
Figura 35 – Arranjo esquemático da viga hiperestática 43

LISTA DE TABELAS

Tabela 1 – Tabela para cálculo da deflexão do ponto C de acordo com o Teorema de Castigliano 35

LISTA DE ABREVIATURAS E SIGLAS

MEF Método De Elementos Finitos

LISTA DE SÍMBOLOS

F: Força [N]
M: Momento [N.m]
σ: Tensão [Pa]
P: Módulo da força [N]
ε: Deformação específica
δ: Variação absoluta do comprimento [m]
L: Comprimento [m]
E: Módulo de elasticidade [Pa]
I: Momento de inércia $[m^4]$
T: Tensão de cisalhamento

SUMÁRIO

1 INTRODUÇÃO 15
1.1 CONTEXTUALIZAÇÃO DO TRABALHO 15
1.2 OBJETIVOS 16
1.3 JUSTIFICATIVAS 16
2 REVISÃO BIBLIOGRÁFICA 17
2.1 ENSINO PRÁTICO DA ENGENHARIA 17
2.2 DISPOSITIVOS DIDÁTICOS NA ÁREA DE RESISTÊNCIA DOS MATERIAIS 18
2.3 PROGRAMAÇÃO EM PYTHON 30
2.3.1 INTERFACE EM PYTHON 31
3.0 MODELAGEM E DESCRIÇÃO TEÓRICA DOS DISPOSITIVOS 33
3.1 GUINDASTE DE TORRE 33
3.2 TRELIÇA PLANA 34
3.3 TUBO DE TENSÕES PRINCIPAIS 35
3.4 VIGA CURVA 37
3.5 VASO DE PRESSÃO 38
3.6 VIGA HIPERESTÁTICA 38
4.0 RESULTADOS E DISCUSSÃO 40
5.0 CONCLUSÕES E SUGESTÕES PARA PRÓXIMOS TRABALHOS 44
5.1 CONCLUSÕES 44
5.1 SUGESTÕES PARA PRÓXIMOS TRABALHOS 44
REFERÊNCIAS 45
APÊNDICE A – CÓDIGO FONTE DO SOFTWARE 48

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO TRABALHO

Segundo Dias (2017), a visualização e contemplação de fenômenos inerentes da Resistência dos Materiais em laboratório permite ao aluno um melhor aprendizado. Em razão do avanço tecnológico, busca-se para o ensino superior de Engenharia Mecânica novas alternativas para que as aulas de determinadas disciplinas, hoje, totalmente teóricas, possam apresentar-se, eventualmente, mais práticas através de dispositivos didáticos (BORTOLO, 2006). Assim sendo, fez-se necessária a confecção de dispositivos necessários para o estudo de resistência dos materiais para o Laboratório de Estruturas da Faculdade De Engenharia De Guaratinguetá (UNESP/FEG) que serão apresentados a seguir.

De acordo com Beer (2015), treliças são estruturas estacionárias totalmente vinculadas projetadas para suportar cargas. Elas são formadas por elementos retilíneos conectados em juntas localizadas nas extremidades de cada elemento. Sendo assim, em cada um dos membros de uma treliça são submetidas duas forças em suas extremidades de mesmo módulo, direção e sentidos opostos. No caso específico de treliças planas, pode-se dizer que são estruturas rígidas bidimensionais que, em sua forma mais elementar, consistem em três barras unidas por suas extremidades formando os lados de um triângulo. Essa foi uma das estruturas construídas para o laboratório de estruturas da referida instituição. O guindaste de torre é um dos equipamentos mais utilizados nos canteiros de obras (Chen et al., 2021). Assim sendo, fez-se necessária a construção de um protótipo de guindaste de torre no laboratório de estruturas dado sua importância no estudo de aplicações de engenharia. Segundo Beer (2015), a flexão pura ocorre em barras prismáticas submetidas à ação de dois conjugados iguais de sentidos contrários, que atuam em um mesmo plano longitudinal. Sendo assim, foi elaborado e construído um dispositivo para o estudo do caso de uma viga submetida a flexão pura que é capaz de demonstrar de forma clara, uma viga em flexão pura, que apresenta um trecho constante de seu diagrama de esforço cortante com força igual a zero e um trecho de seu respectivo diagrama de momento fletor com magnitude máxima constante, o que representa a flexão pura. Outro dispositivo didático presente no laboratório de estruturas trata-se de uma viga curva isostática para determinação experimental de tensões, deformações e deslocamentos. Essa viga curva simples é biapoiada com um apoio articulado em uma extremidade e um apoio deslizante na outra extremidade, permitindo rotação em ambos os apoios, mas restringindo a translação na primeira extremidade, com apoio articulado, e permitindo translação do apoio deslizante. O

engenheiro tem a responsabilidade de determinar se uma dada carga leva a tensões e deformações aceitáveis (BEER,1990). Assim sendo, para o estudo das tensões e deformações principais foi desenvolvido um dispositivo para o ensino de medições de tensões e deformações principais que consiste em um corpo tubular com uma extremidade soldada em uma base fixa e na extremidade livre sendo fixada uma alavanca onde é aplicada uma determinada carga. Essa carga provoca tensões normais, torcionais e de cisalhamento no tubo, e com os extensômetros devidamente posicionados, é possível realizar as medidas das tensões e deformações principais. Para o estudo da deflexão em vigas hiperestáticas prismáticas, foi elaborado um dispositivo que utiliza cargas cortantes e flexionais máximas em uma viga e simulação de distribuição e posição de cargas. Outro dispositivo importante no estudo de pressão e deformação presente no laboratório de estruturas é o vaso de pressão, dentro da teoria de membranas. Sendo assim, foi construído um vaso de pressão de aço comercial.

1.2 OBJETIVOS

O objetivo com a implementação do projeto de trabalho de graduação foi a possiblidade dos alunos que participarem dos procedimentos experimentais do laboratório, terem a possibilidade de utilizar dados de entrada que estejam além da capacidade máxima dos dispositivos presentes no mesmo para que seja possível simular os dados de saída desses dispositivos em situações customizadas em condições mais abrangentes por meio de programas em linguagem *Python*.

1.3 JUSTIFICATIVAS

Este trabalho possui como justificativa o interesse do pesquisador em conseguir utilizar um caso real para aplicar conhecimentos em programação com a linguagem *Python* assim como conhecimento prático em assuntos abordados ao longo do curso de engenharia mecânica.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados os conceitos fundamentais para o estudo e realização deste trabalho de graduação. Serão abordados assuntos referentes ao tipo de linguagem de programação utilizada na construção dos programas assim como aos dispositivos mecânicos presentes no laboratório de estruturas e que são objeto de estudo.

2.1 ENSINO PRÁTICO DA ENGENHARIA

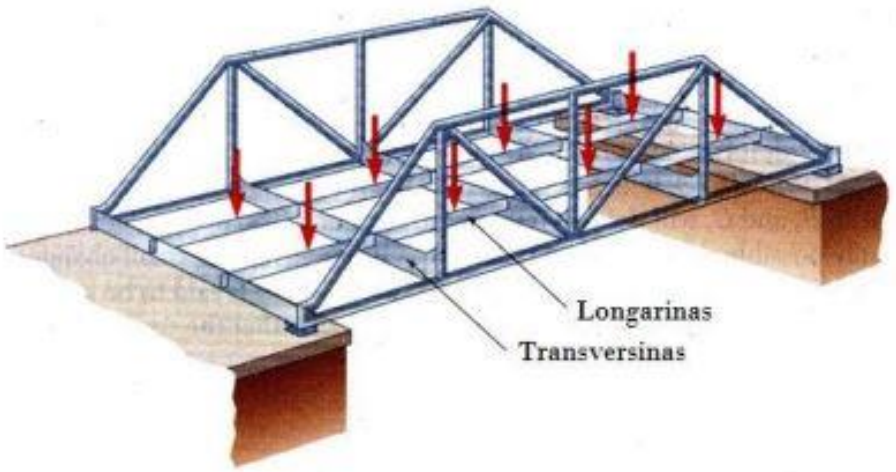
De acordo com a Unesp, ao longo da grade curricular dos cursos de engenharia mecânica de uma forma geral, é encontrada uma carga horária de aproximadamente 20% em matérias práticas ou laboratórios. Isso demonstra a importância do ensino prático dos conceitos encontrados na literatura teórica durante a formulação da grade curricular do referido curso. Além de matérias com grande parte de sua carga horária voltada ao conteúdo prático como por exemplo Propriedades Mecânicas Dos Materiais, também existem matérias que abordam seu conteúdo de forma prática em sua totalidade como é o caso de Laboratório de Mecânica. Em ambos os casos e também em outros exemplos encontrados na grade, é utilizado um grande aparato de dispositivos didáticos mecânicos e eletromecânicos para o ensino dos conceitos aos quais as disciplinas práticas se propõem a ensinar. Entre os tipos de dispositivos didáticos destacam-se os materiais virtuais desenvolvidos geralmente com o auxílio de ferramentas computacionais e o os materiais físicos compostos principalmente de protótipos (BORTOLO e LINHARES 2006). Segundo Diwakar (2013), o laboratório é o lugar onde várias teorias são testadas e onde é estabelecida a relação entre o conhecimento teórico e prático. Segundo Gaber (2013), o laboratório tem um grande papel no reforço de competências dos alunos, pois é um ambiente vital de uma variedade de atividades e experiências em que a ciência é estudada, sendo um ambiente onde a ciência é experimentada, onde as capacidades de resolver problemas são refinadas no contexto da investigação laboratorial. De acordo com Salaheddin (2015), a engenharia trata-se de uma ciência aplicada e os experimentos laboratoriais mantém a ciência viva. O ensino na engenharia deve ser autêntico, colaborativo, com aprendizagem ativa, baseado em problemas, e a direção adequada é aprender a pensar ativamente implementando os conhecimentos adquiridos em laboratórios práticos, incluindo laboratórios virtuais e remotos (SELL, 2014). Os laboratórios físicos beneficiam os alunos por incorporar objetos concretos naaprendizagem das ciências e segundo Feisel, Rosa (2005), alguns alunos podem se envolver melhor com ciência quando eles são capazes de tocar, mover e examinar objetos reais.

2.2 DISPOSITIVOS DIDÁTICOS NA ÁREA DE RESISTÊNCIA DOS MATERIAIS

A seguir, serão apresentados em maiores detalhes os dispositivos utilizados didaticamente dentro da área de mecânica dos materiais assim como sua importância acadêmica e didática dentro do contexto acadêmico.

Segundo Beer e Johnston (2009), treliças em elementos retos sujeitos a duas forças e unidos em nós localizados nas extremidades de cada elemento e além disso, nenhum elemento é contínuo através de um nó. Nas aplicações de engenharia encontradas atualmente, a maioria das estruturas reais é constituída de várias treliças unidas formando assim uma estrutura espacial. Cada treliça sustenta suas respectivas cargas em seu próprio plano e sendo assim, pode ser tratada como uma estrutura bidimensional. Além disso, de forma geral, os membros de uma treliça são esbeltos e podem suportar pouca carga lateral e sendo assim, todas as cargas devem ser aplicadas nos nós (BEER, 2009). Nesse contexto, a Figura 1 ilustra o arranjo esquemático de uma estrutura formada por treliças.

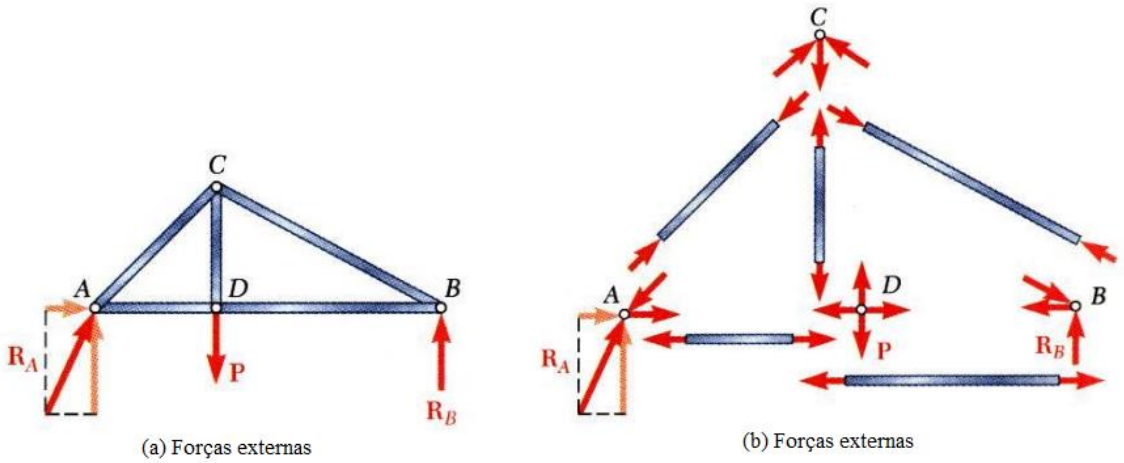
Figura 1 – Estrutura formada por treliças



Fonte: Beer e Jhonston (2009)

A Figura 2 por sua vez, ilustra o comportamento das forças presentes internamente e externamente e uma treliça simples por meio de um diagrama de corpo livre.

Figura 2 – Forças internas e externas em uma treliça



Fonte: Beer e Jhonston (2009)

As forças exercidas pelo elemento nos dois pinos ligados devem ser direcionadas ao longo desse elemento sendo iguais e opostas. Além disso, as condições de equilíbrio para a treliça inteira geram as equações 1, 2 e 3 que não são independentes das equações dos nós.

$$\sum F_x = 0 \tag{1}$$

$$\sum F_y = 0 \tag{2}$$

$$\sum M = 0 \tag{3}$$

Onde cada força *F* tem sua respectiva direção *y* ou *x* e *M* representa um momento atuante naquele elemento. As 3 equações trazem o elemento de somatório por tratarem-se de resultantes. As equações das forças internas são representadas a seguir:

$$\sum F_x = 0 \tag{4}$$

$$\sum F_y = 0 \tag{5}$$

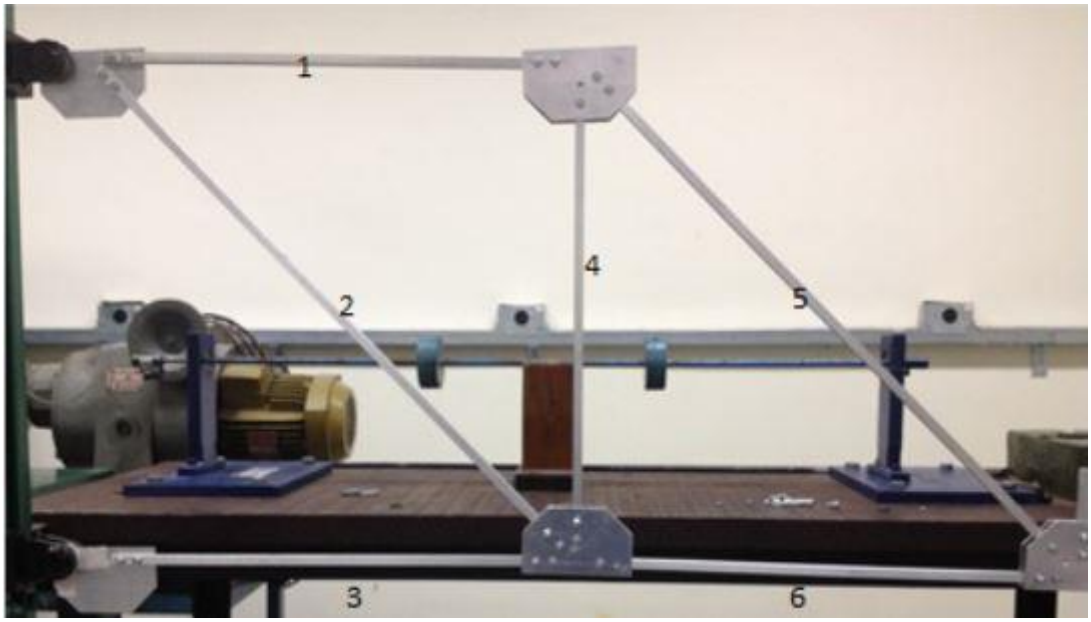
Por fim, as equações de tensão e deformação atuantes em elementos de treliças são representadas a seguir:

$$\boldsymbol{\varepsilon} = \frac{\boldsymbol{\delta}}{\boldsymbol{L}} \quad (6)$$

$$\sigma = \varepsilon.E \quad (7)$$

Onde **ε** representa a deformação específica por unidade de comprimento, ***δ*** uma deformação para um dado comprimento, e *L* o comprimento em questão. A Figura 3 por sua vez, mostra a treliça presente no laboratório de estruturas com a qual será feito o trabalho de graduação em questão:

Figura 3 – Treliça presente no laboratório de estruturas com elementos numerados

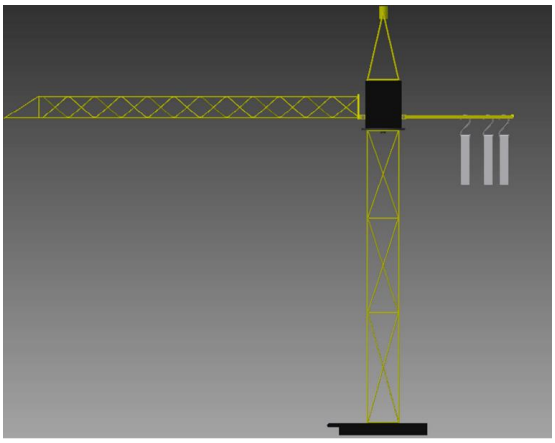


Fonte: Autoria própria (2022)

Segundo Scigliano (2008), no contexto da construção civil, máquinas, ferramentas e equipamentos que permitem erguer, descer ou ainda locomover de forma horizontal ações, tubulações, concreto e os demais tipos de cargas presentes em questão, são determinantes e, por isso, guindastes de torres amplamente utilizados. Ainda segundo o mesmo autor, as gruas são utilizadas não só na construção civil, mas também em indústrias, terminais portuários e aeroportuários.

Seu tamanho e formas variam, para que seja possível atender às necessidades de suas diferentes aplicações, e cabe ao engenheiro planejar e organizar padrões. Foi projetado e construído um Guindaste de Torre para o laboratório de estruturas da FEG – Faculdade de Engenharia de Guaratinguetá desde a fase de projeto até a execução. A Figura 4 ilustra a modelagem geométrica deste guindaste de torre.

Figura 4 – Modelagem geométrica Guindaste de Torre em vista de perfil



Fonte: Autoria própria (2022)

A figura 5 a seguir ilustra o referido dispositivo após fase de projeto, execução e montagem.

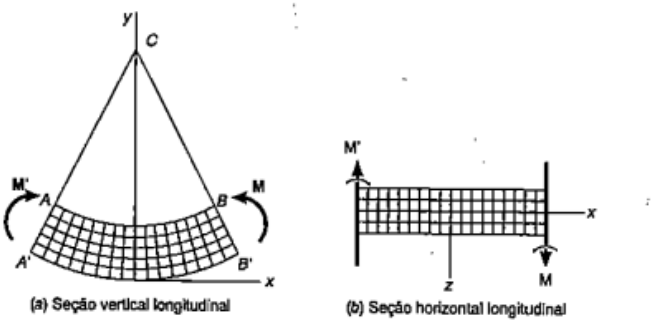
Figura 5 - Guindaste de Torre do laboratório de estruturas do departamento de mecânica da FEG-UNESP

Fonte: Autoria própria (2022)

Segundo Beer e Johnston (2009), os casos de flexão pura são muito comuns nas aplicações práticas. Ainda segundo o mesmo autor, a deformação atuante em uma barra

prismática em flexão pura sujeita a ação dos conjugados *M* e *M'* pode ser exemplificada pela figura 6.

Figura 6 – Deformação de uma viga prismática submetida a conjugados



Fonte: Beer e Jhonston (2008)

Foi projetado, executado e montado um dispositivo didático para o laboratório de estruturas da FEG conforme citado anteriormente com a função de medir e estudar a deformação presente em uma viga submetia a flexão pura. Mais especificamente, esse dispositivo apresenta um trecho constante de seu diagrama de esforço cortante com força igual a zero e um trecho do seu respectivo momento fletor com magnitude máxima constante, o que representa nesse caso flexão pura. As equações 8, 9 e 10 descrevem o funcionamento do referido dispositivo.

$$\sigma = \frac{Mc}{I} \tag{8}$$

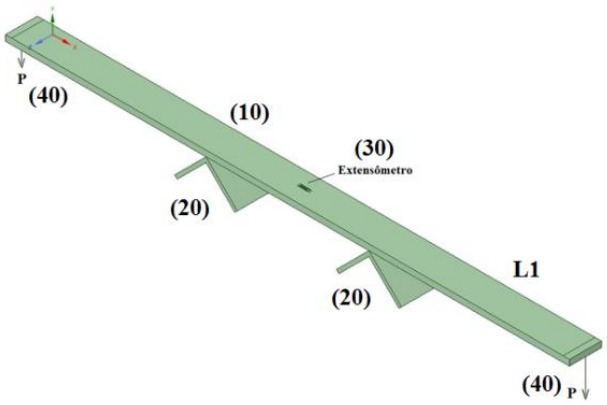
$$\sigma = \varepsilon.E \tag{9}$$

$$I = \frac{bh^3}{12} \tag{10}$$

Onde σ representa a tensão normal, *M* o momento atuante no elemento e *c* a distância da linha neutra. A segunda equação mostra matematicamente a Lei de Hooke para situações dentro do regime elástico onde ε representa a deformação enquanto *E* é o módulo de elasticidade que é uma característica específica do material. Por fim, para a terceira equação, *I* é o momento de inércia para a viga em questão sendo *b* sua base e *h* sua altura. De forma simplificada, a Figura

7 ilustra todas as partes presentes no dispositivo bem como a colagem do extensômetro que será utilizado nas medições requisitadas.

Figura 7 – Representação simplificada do dispositivo de medição de medição de deformação em uma viga



Fonte: Autoria própria (2022)

Outro dispositivo didático que será utilizado como objeto de estudo e que está presenta no laboratório de estruturas da FEG se trata de uma viga curva isostática para aferição de sua deflexão horizontal. Essa viga foi projetada de forma a ser biapoiada, com um apoio articulado em uma extremidade e um apoio deslizante na outra extremidade, permitindo rotação em ambos os apoios, mas restringindo a translação na primeira extremidade, com apoio articulado, e permitindo translação do apoio deslizante. Assim como os outros dispositivos citados, o projeto foi concebido para o melhor ensino da matéria de Resistência dos Materiais visto que permite a visualização e aplicação do conteúdo teórico sobre o Teorema de Castigliano para medição de deflexões em vigas visto nas aulas da referida matéria. A validade do projeto contará com três dados de deflexão. O primeiro será obtido diretamente no dispositivo, através da escala posicionada na extremidade que se deslocará. O segundo dado será obtido utilizando cálculos matemáticos, aplicando o Teorema de Castigliano. Os dados obtidos diretamente na viga, pela escala, serão comparados aos dados obtidos matematicamente. A partir da energia de deformação de uma estrutura, o teorema de Castigliano fornece um meio de se calcular a deflexão da estrutura quando submetida a um carregamento (GERE; GOODNO, 2010). Esse método aplica-se somente a corpos que permaneçam com temperatura constante e na região de cargas em que o material apresenta comportamento elástico (HIBBELER, 2010). Nesse contexto, a equação 11 representa do Teorema de Castigliano.

$$\delta = \frac{\partial U}{\partial Q} \tag{11}$$

Onde U representa a energia de deformação, δ o deslocamento generalizado e Q a força generalizada. Essa dedução foi realizada considerando um corpo arbitrário, porém o teorema pode ser usado para qualquer outro tipo de estrutura. A exigência para a aplicação deste teorema é de que a estrutura seja linear elástica, pois a dedução acima exige que somente forças conservativas sejam consideradas, e que o princípio de superposição seja aplicável (GERE; GOODNO, 2010). Segundo Hibbeler (2010), para tornar o procedimento matemático mais prático no caso de vigas, considerando E e I constantes, o Teorema de Castigliano pode ser enunciado como se segue:

$$\delta = \int_0^L M \left(\frac{\partial M}{\partial P}\right) dx \ \frac{1}{EI} \tag{12}$$

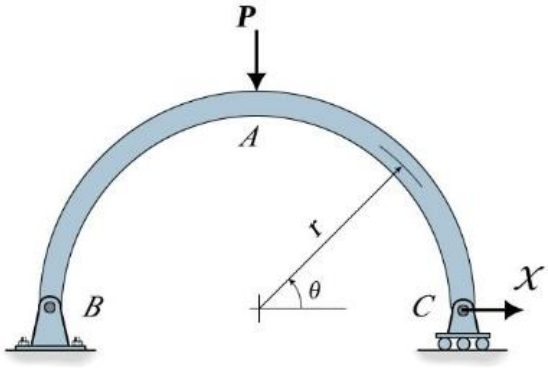
Onde δ representa o deslocamento do ponto provocado pelas cargas reais que agem sobre a viga, P a força externa de intensidade variável aplicada na viga na direção de δ, M o momento interno da viga, E o módulo de elasticidade do material da viga, I o momento de inércia da área da seção transversal, calculado em torno de um eixo neutro. Segundo Gere e Goodno (2010), essa abordagem é chamada de Teorema Modificado de Castigliano. Para a análise da deformação específica da viga e tensão de flexão atuante, foi realizada a colagem de dois extensômetros na viga, um na fibra externa e outro na fibra interna, a fim de avaliar tanto os esforços de compressão, quanto os esforços de tração atuantes como mostra a Figura 8.

Figura 8 – Extensômetro colado na superfície da viga

Fonte: Autoria própria (2022)

A aplicação da carga P na viga curva é representada pela figura 9.

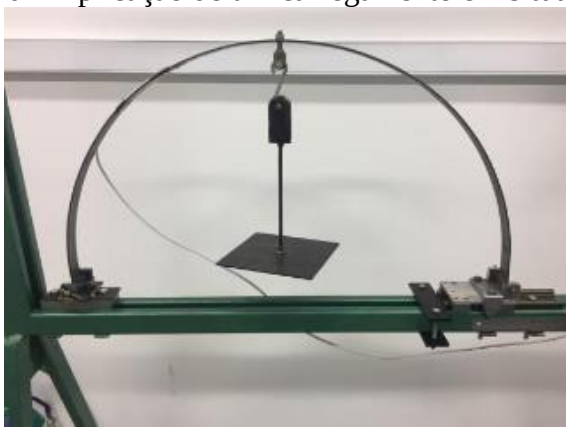
Figura 9 - Aplicação da carga P na viga curva



Fonte: Autoria própria (2022)

A aplicação do carregamento em uma situação real é vista na figura 10.

Figura 10 – Aplicação de um carregamento em situação real



Fonte: Autoria própria (2022)

Foi também elaborado um dispositivo didático para medição de tensões e deformações principais com combinações de carregamentos, flexional, torcional e as cargas cortantes na estrutura fazendo a utilização de extensômetros. Segundo Hibbeler (2010), o extensômetria é uma das ferramentas que auxiliam na determinação de esforços e deformações utilizando-se de sensores como rosetas extensométricas, e também sistemas de aquisição de dados para obtenção desses valores. O dispositivo para o ensino de medições de tensões e deformações principais compreende um corpo tubular uma extremidade soldada em uma base fixa e na extremidade livre sendo fixada uma alavanca onde é aplicada uma determinada carga. Uma força P é

aplicada na extremidade da alavanca que por sua vez está fixada na extremidade livre do tubo. A carga P provoca tensões normais, torcionais e de cisalhamento no tubo, e com os extensômetros devidamente posicionados, é possível realizar as medidas das tensões e deformações principais. As equações de tensão normal, conjugado, flexão em regime elástico, tensão de cisalhamento e torção respectivamente são mostradas a seguir:

$$\sigma = \frac{P}{A} \tag{13}$$

$$M = P.D \tag{14}$$

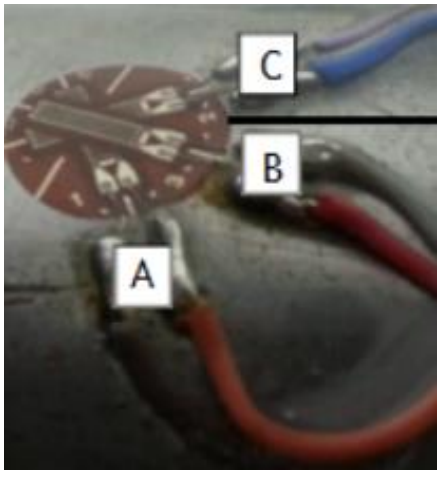
$$\sigma = \frac{M}{I} \tag{15}$$

$$\tau = \frac{V.Q}{I.t} \tag{16}$$

$$\tau = \frac{T.p}{J} \tag{17}$$

Onde σ representa a força normal, P a carga aplicada perpendicularmente à área, A a área onde P é aplicada, M o momento, d o comprimento do tubo, y a distância entre a superfície externa do tubo e a linha neutra, I o respectivo momento de inércia, V a força cortante, t a largura da seção horizontal, Q o momento estático de área, τ a torção, p a distância do eixo da barra e J o momento de inércia polar de um círculo. A disposição real dos extensômetro A, B e C utilizados na construção do dispositivo podem ser vistos na Figura 11.

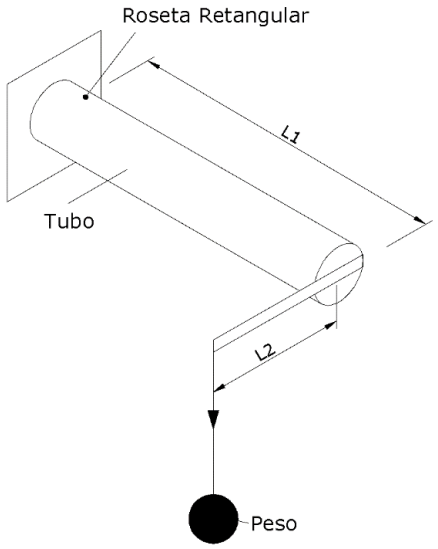
Figura 11 – Extensômetros A, B e C utilizados no tubo de tensões e deformações principais.



Fonte: Autoria própria (2022)

A Figura 12 mostra uma representação esquemática dos elementos citados envolvidos no projeto já montado enquanto a Figura 13 mostra o dispositivo real após a montagem.

Figura 12 – Representação esquemática do dispositivo medidor de tensões e deformações principais mostrando sua montagem e a posição da roseta retangular



Fonte: Autoria própria (2022)

Figura 13 – Dispositivo de medição de tensões e deformações principais em situação real após montagem

Fonte: Autoria própria (2022)

Por fim, foi também construído para o laboratório de estruturas da FEG um vaso de pressão. Segundo Borse; Sharma (2012), vasos de pressão são dispositivos para armazenamento e transporte de fluidos, comumente feitos de aço e tem sido amplamente utilizados em indústrias em operações de risco. Vasos de pressão são classificados majoritariamente de acordo com sua geometria, e cada uma destas é recomendada para sua aplicação específica de acordo com o código recomendado pela ASME (2015). Para Carraro et al. (2009), é possível distinguir dois principais componentes em um vaso de pressão: o tampo e o costado. O último é comumente concebido em forma cilíndrica, esférica, cônica ou até mesmo uma combinação das mesmas. Para Da; Cefet (2014), existem três grupos de vasos de pressão com relação ao posicionamento:

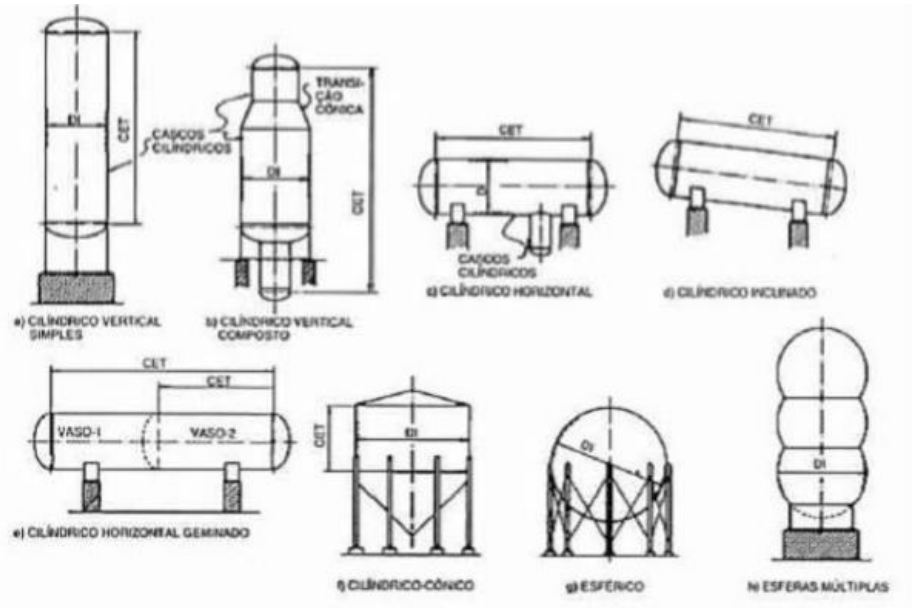
• Verticais em caso a interferência gravitacional seja essencial;

• Horizontais, quando se trata do caso de melhor disposição de sustentação

• Inclinados, quando ocorre o caso de um fluido muito viscoso de consequente difícil escoamento.

A Figura 14 ilustra os possíveis posicionamentos para vasos de pressão.

Figura 14 - Posicionamento dos Vasos de Pressão



Fonte: Telles (2001)

As as equações (18), (19) e (20) das tensões no costado, tangencial e longitudinal respectivamente em um vaso de pressão junto ao fator de correção ASME (2015):

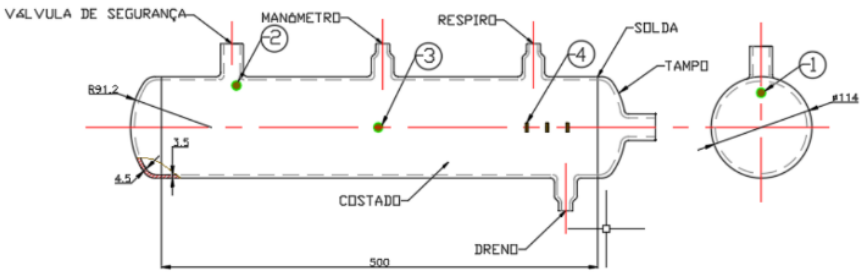
$$\sigma = (pr/2t) + 0{,}2.p \qquad (18)$$

$$\sigma t = (pr/t) + 0{,}6.p \quad (19)$$

$$\sigma l = (pr/2t) + 0{,}6.p \quad (20)$$

Onde *p* representa a pressão interna, *r* o raio interno e *t* a espessura do vaso considerando-se que esse valor t tem de ser 10% menor que o raio interno de acordo com a Teoria de Cálculo de Vasos de Pressão de Parede Fina. A Figura 15 ilustra a concepção do projeto bem como os elementos que integram esse dispositivo.

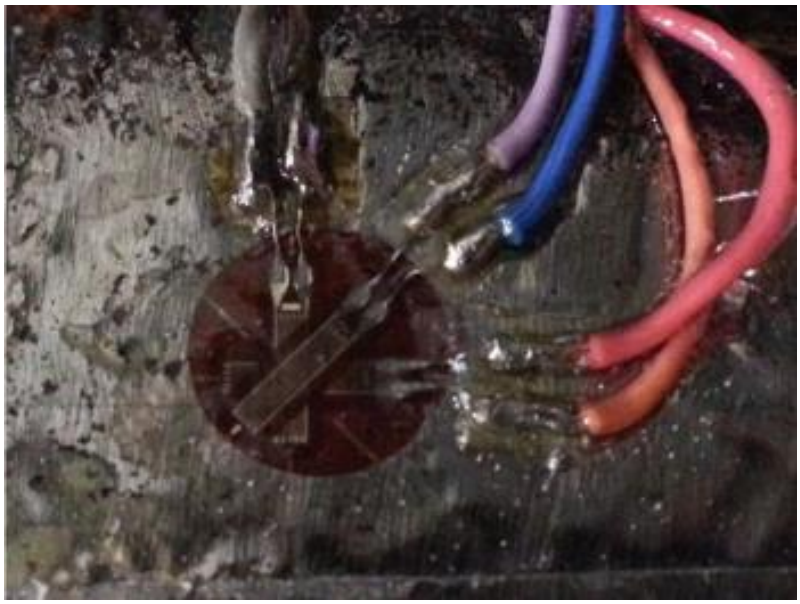
Figura 15 – Concepção do projeto de vaso de pressão de paredes finas



Fonte: Autoria própria (2022)

A Figura 16 a seguir mostra a roseta utilizada para instrumentação do dispositivo por meio de uma Roseta.

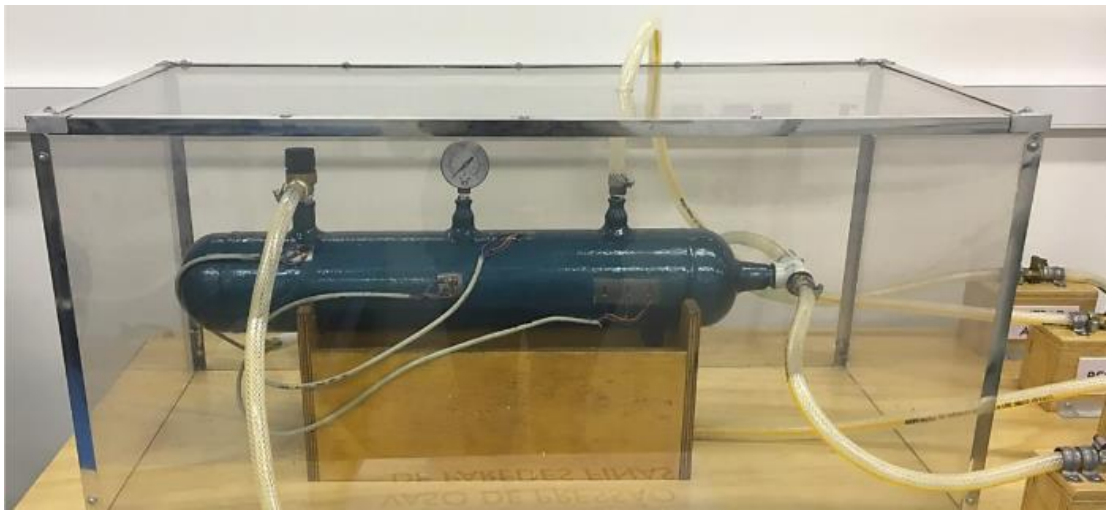
Figura 16 - Roseta instrumentada



Fonte: Autoria própria (2022)

A figura 17 mostra o dispositivo após o processo de montagem no laboratório de estruturas da FEG.

Figura 17 - Dispositivo Didático montado BR102019 0233320

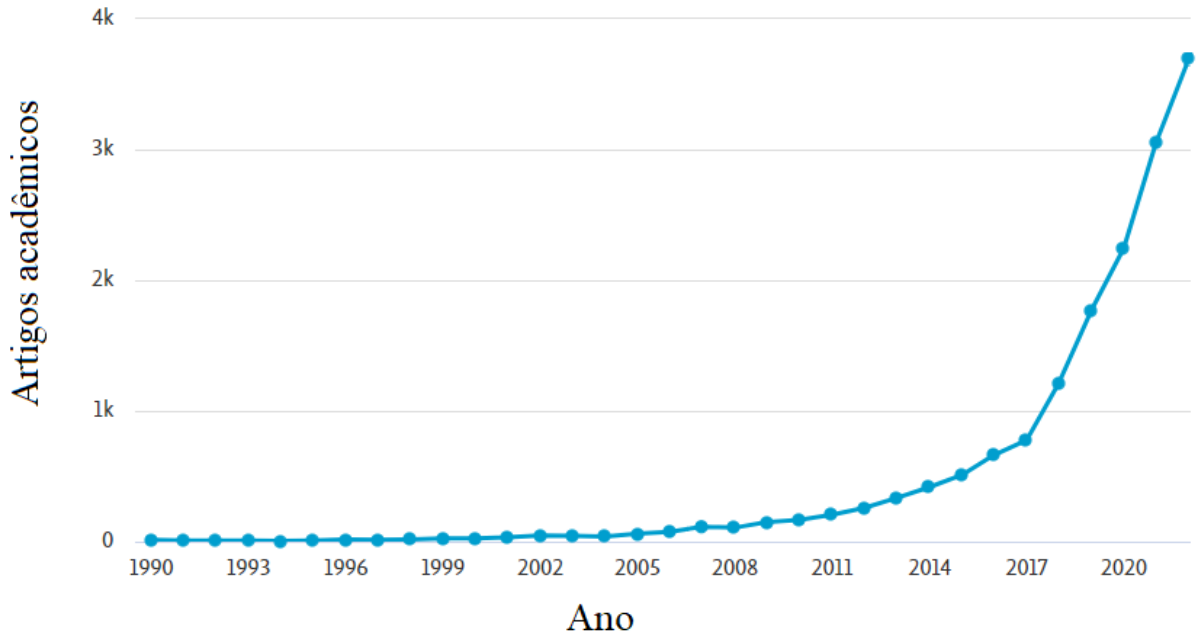


Fonte: Autoria própria (2022)

2.3 PROGRAMAÇÃO EM *PYTHON*

Segundo Lutz (2011), Python se trata de uma linguagem de programação de computadores de código aberto de uso geral sendo também otimizada para a qualidade de software, produtividade do desenvolvedor, portabilidade de programas e integração de componentes. Ainda de acordo com o mesmo autor, essa linguagem é hoje utilizada por pelo menos centenas de milhares de desenvolvedores em torno do mundo em áreas como programação de sistemas, interfaces de usuário, personalização de produtos, programação numérica, entre outros. É também considerado estar entre as cinco linguagens de programação mais amplamente utilizadas no mundo hoje. Python também conta hoje em sua base de usuários empresas como Google e Youtube. Seu conjunto de ferramentas o torna uma linguagem flexível e ágil, ideal tanto para tarefas táticas rápidas quanto para esforços de desenvolvimento de aplicações estratégicas de longo prazo. A Figura 18 representa a importância da linguagem Python em termos dos artigos científicos que de alguma forma utilizaram as ferramentas proporcionados por essa linguagem mostrando assim, a relevância atual dessa linguagem de programação no âmbito acadêmico.

Figura 18 – Artigos acadêmicos relacionados com a linguagem Python ao longo dos anos



Fonte: Adaptado de (Scopus, 2022)

2.3.1 INTERFACE EM PYTHON

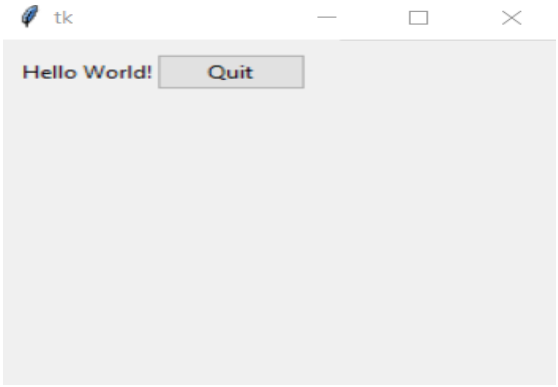
Dentre as possíveis interfaces para a construção de um programa em Python, para a aplicação desejada, será implementada a biblioteca *Tkinter* como é exemplificado pelas Figuras 19 e 20.

Figura 19 – Exemplo de código para criação de uma interface utilizando a biblioteca *Tkinter*

```
from tkinter import *
from tkinter import ttk
root = Tk()
frm = ttk.Frame(root, padding=10)
frm.grid()
ttk.Label(frm, text="Hello World!").grid(column=0, row=0)
ttk.Button(frm, text="Quit", command=root.destroy).grid(column=1, row=0)
root.mainloop()
```

Fonte: Documentação da biblioteca Tkinter (2022)

Figura 20 – Interface criada pelo código exemplo da Figura 19



Fonte: Autoria própria (2022)

Ainda segundo a documentação da biblioteca, o pacote *tkinter* é a interface padrão *Python* para o conjunto de ferramentas Tcl/Tk GUI. Tanto Tk como *tkinter* estão disponíveis na maioria das plataformas Unix, incluindo MacOS, bem como em sistemas Windows. O *PySimpleGU*I é capaz de transformar as estruturas GUI *tkinter*, *Qt*, *WxPython* e *Remi* em uma interface mais simples. A definição da janela é simplificada usando tipos de dados centrais *Python* compreendidos por iniciantes sendo tanto listas como dicionários. Uma simplificação adicional acontece mudando o tratamento de eventos de um modelo baseado em *callback* para um modelo que passa uma mensagem.

3 MODELAGEM E DESCRIÇÃO TEÓRICA DOS DISPOSITIVOS

Neste capítulo serão apresentadas as equações teóricas para os dispositivos anteriormente apresentados e suas respectivas versões dentro do contexto do código fonte do software desenvolvido para o laboratório de estruturas da FEG. Nas figuras apresentadas neste capítulo no contexto do código fonte, frequentemente haverá a ocorrência das funções *float()* para a transformação de valores de *input* para o tipo *float, pow*() para utilização de expoentes e *get()* para encontrar variáveis que foram inseridas pelo usuário.

3.1 GUINDASTE DE TORRE

Para o guindaste de torre, além das equações que já foram citadas e que serão citadas neste capítulo, a teoria foi embasada conforme Aluvino (2017). Para a obtenção da carga teórica, foi considerada a origem *x*, *y* e *z* no ponto de encontro da lança e a torre e foi utilizado o conceito de somatório de momentos conforme a equação (8):

$$m.g.l - F.sen(\theta).d + x.Cp = 0 \quad (21)$$

Onde *m* representa a massa alocada na extremidade da lança, *g* a aceleração da gravidade, *l* o comprimento da lança de guindaste, *F* a carga teórica, θ a angulação entre a torre e o cabo e *d* a sua respectiva distância, *x* a distância entre a torre e o contrapeso e *Cp* o contrapeso. Sendo assim, foi possível isolar a variável *F* considerando os dados retirados da documentação do dispositivo. No contexto do código fonte do software em questão, *m. x* e *Cp* foram utilizadas como variáveis de entrada enquanto *g, l* e θ tem valores fixos:

- g = 9,91 m/s²
- l = 1,95 m
- θ = 30°

A Figura 21 a seguir ilustra como foi calculada a variável *F* no código fonte do software utilizando o software de programação *Visual Studio Code* e auxílio da biblioteca *math.*

Figura 21 – Cálculo da carga teórica do guindaste de torre no código fonte do software do laboratório de estruturas da FEG

```
f = ((float(massa_carga.get())*9.81*1.95)-(float(x_cp.get())*float(cp.get())))/(math.sin(math.radians(angulo_torre_cabo))*dist_torre_cabo)
```

Fonte: Autoria própria (2022)

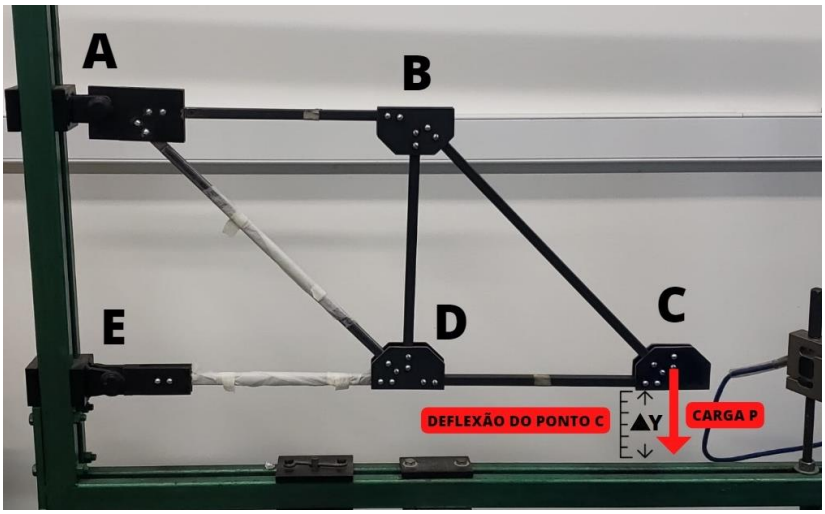
3.2 TRELIÇA PLANA

Para a treliça plana, além das equações que já foram citadas e que serão citadas neste capítulo, a teoria foi embasada conforme Cavaguti (2015). No contexto do software, foi utilizado o Teorema De Castigliano conforme a equação 9:

$$yc = \sum \left(\frac{FiLi}{AiE}\right)\frac{\partial \mathrm{Fi}}{\partial \mathrm{Q}} = \frac{1}{E}\sum \left(\frac{FiLi}{Ai}\right)\frac{\partial \mathrm{Fi}}{\partial \mathrm{Q}} \qquad (9)$$

Assim, foi montada uma tabela considerando os nós de acordo com a Figura 22.

Figura 22 – Nós da treliça plana



Fonte: Autoria própria (2022)

Dessa forma, utilizando as 3 equações da estática conforme as equações (1), (2) e (3), foi possível construir uma tabela com os elementos da treliça conforme a Figura 22 para o cálculo da deflexão conforme a equação (8) como é mostrado na Tabela 1:

Tabela 1 – Tabela para cálculo da deflexão do ponto C de acordo com o Teorema de Castigliano

Elemento	Fi	$\frac{\partial Fi}{\partial Q}$	Ai (m²)	Li (m)	$\left(\frac{FiLi}{Ai}\right)\frac{\partial Fi}{\partial Q}$
AB	P	1	6,4 x $(-10)^{-5}$	0,36	5625P
BC	P/0,712	1/0,712	6,4 x $(-10)^{-5}$	0,59	18184,95P
CD	-P	-1	6,4 x $(-10)^{-5}$	0,42	6562,5P
DB	-P	-1	6,4 x $(-10)^{-5}$	0,42	6562,5P
DE	-1,857P	-1,857	6,4 x $(-10)^{-5}$	0,36	19397,52P
DA	P/0,764	1/0,764	6,4 x $(-10)^{-5}$	0,55	14723P
-	-	-	-	∑	71055,47P

Fonte: Autoria própria (2022)

Com isso, foi possível obter a equação (9):

$$yc = 1/E.\sum((Fi/Li)/Ai).\ \partial Fi/\partial P \qquad (22)$$

Onde *yc* representa a deflexão no ponto C da treliça, *E* o módulo de elasticidade, *F* a força presente no elemento *i*, *A* a área do elemento *i* e *L* o comprimento da barra *i*. Dessa forma foi encontrada a equação (10):

$$yc = P.1{,}0151.\ (-10)^{-6} \qquad (23)$$

Onde *P* é a carga causada pela massa pendurada no ponto C da treliça plana. A Figura 23 a ilustra como foi calculada a variável *P* no código fonte do software:

Figura 23 – Cálculo da deflexão do ponto C da treliça plana dentro do código fonte

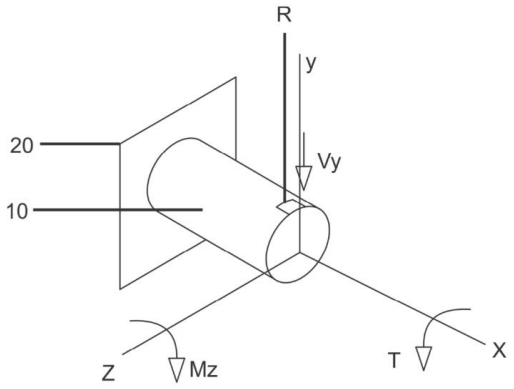
```
resultado = float(p.get())*1.0151*pow(10,-6)*1000
```

Fonte: Autoria própria (2022)

3.3 TUBO DE TENSÕES PRINCIPAIS

Para o tubo de tensões principais, além das equações que já foram citadas e que serão citadas neste capítulo, a teoria foi embasada conforme a Carta Patente do dispositivo. Para o cálculo das tensões teóricas foram utilizadas a equações (1), (2) e (3) com o auxílio do Círculo De Mhor. A Figura 24 ilustra como os eixos foram arbitrados.

Figura 24 – Eixos do tubo de tensões principais



Fonte: Autoria própria (2022)

Como a massa *m* pendurada na alavanca do tubo é uma variável, ela foi usada como variável de entrada do usuário no software e as tensões normais e de cisalhamento em *x* e *y* que servem como base para o restante dos cálculos foram feitos considerando-se esse *dado de* entrada conforme as equações (24), (25), (26), (27), (28) e (29).

$$Mx = m.g.Lt \quad (24)$$

$$My = m.g.(La+dext/2) \quad (25)$$

$$My = m.g.(La+dext/2) \quad (26)$$

$$\sigma x = m.g.(Lt+dext/2)/Iy \quad (27)$$

$$\sigma y = 0 \quad (28)$$

$$Txy = (Mx* dext/2)/Jx \quad (29)$$

Onde *g* é a aceleração da gravidade, *Lt* o comprimento do tubo, *La* o comprimento da alavanca e *dext* o diâmetro externo do tubo. Todas as características geométricas são conhecidas previamente. Para o cálculo das deformações representado pelas equações (30) e (31), foram

utilizados os conceitos de Círculo De Mohr, tensão normal máxima e mínima, Coeficiente Elástico *E* e Coeficiente de Poisson *v*.

$$\varepsilon máx = (\sigma máx - v\sigma mín)/E \quad (30)$$

$$\varepsilon mín = (\sigma mín - v\sigma máx)/E \quad (31)$$

Onde *εmáx* é a deformação máxima no ponto, *σmín* a tensão normal mínima, *εmín* a deformação mínima e *σmáx* a tensão normal máxima. A Figura 25 ilustra como foram calculados esses valores no código fonte do software.

Figura 25 – Cálculo das tensões e deformações do tubo de tensões principais

```
# Tensões teóricas
E = 200*pow(10,3)
v = 0.3
Iy = 1.1676*(pow(10,(-7)))
Jx = 2*Iy
Mx = (float(m.get()))*9.81*0.4
Gx = round((((float(m.get()))*9.81*(0.420+0.032))/Iy),2)
Gy = float(0)
Txy = (Mx*0.032)/Jx
G_med = (Gx+Gy)/2
dif = Gx - G_med
R = pow((pow(Txy,2)+pow(dif,2)),0.5)
T_max = R
T_min = -R
G_max = G_med + R
G_min = G_med - R
sen_dois_teta = Txy/R
e_max = (G_max - v*G_min)/E
e_min = (G_min - v*G_max)/E
sen_dois_teta = Txy/R
dois_teta = math.asin(sen_dois_teta)
teta = ((dois_teta)/2)*100

# Deformações Teóricas:

e_max = (G_max - v*G_min)/E
e_min = (G_min - v*G_max)/E
```

Fonte: Autoria própria (2022)

3.4 VIGA CURVA

Para a viga curva, além das equações que já foram citadas e que serão citadas neste capítulo, a teoria foi embasada conforme Dias (2017). Foi utilizado o Teorema De Castigliano para calcular a deflexão do ponto C conforme a Figura 9. A equação (32) o teorema nesse contexto:

$$\delta c = \frac{P.r^3}{2.E.I} \quad (32)$$

Onde δc representa a deflexão no ponto C, P a carga causada pelo peso pendurado e r o raio da viga (conforma Figura 9). A Figura 26 a seguir mostra como foi calculado esse valore no código fonte do software:

Figura 26 – Cálculo da deflexão do ponto C da viga curva dentro do código fonte

```
carga_total = (float(massa_anilhas.get())+m_sup)*10
deform = (float(carga_total)*pow(Rc,3)/(2*E*I))*1000
```

Fonte: Autoria própria (2022)

3.5 VASO DE PRESSÃO

No caso do vaso de pressão, a teoria foi embasada conforme Glyniadakis (2018). Os cálculos para a obtenção respectivamente das tensões teóricas no costado, tangencial e longitudinal foram calculadas considerando a correção ASME (2015) conforme e equação 8. Sendo assim, a Figura 27 ilustra como essas equações foram utilizadas no código fonte do software.

Figura 27 – Cálculo das tensões no costado, tangencial e longitudinal do vaso de pressão no código fonte

```
TVP = (float(p.get())*float(r.get()))/(2*(float(t.get()))) + 0.2*(float(p.get()))
TTVP = (float(p.get())*float(r.get())) + 0.6*(float(p.get()))
TLVP = (float(p.get())*float(r.get()))/(2*(float(t.get()))) + 0.6*(float(p.get()))
```

Fonte: Autoria própria (2022)

3.6 VIGA HIPERESTÁTICA

Para a viga hiperestática, além das equações que já foram citadas anteriormente, a teoria foi embasada conforme a Carta Patente do dispositivo referenciada neste documento. No caso da viga hiperestática, foi utilizado o simulador online *deflection* onde é possível simular uma grande quantidade de situações estáticas e hiperestáticas uma vez que existem vários elementos na viga que podem ser alterados como existência ou não de momento concentrado, força

concentrada, carregamento, seus respectivos módulos e sentidos. Isso foi feito sem a necessidade de sair do ambiente do software por meio da *Application Programming Interface* (API) do simulador. O software devolve além da deflexão em todos os pontos da viga, os diagramas de esforço cortante, momento fletor e ângulo como será mostrado no capítulo 4. A figura a seguir mostra como são feitos os *inputs* do usuário dentro do código fonte:

Figura 28 – Código de de dados de entrada dentro do código fonte para a viga hiperestática

```
# Entrada de dados
Label(janela, text = 'Insira a posição da força em metros que gera o momento concentrado. Caso ela nao exista, insira 0:', anchor = W).pla

p1_amount = Entry(janela, textvariable = p1)
p1_amount.place(x=910, y=350)

Label(janela, text = 'Insira o a força (em Newtons positiva ou negativa) que gera o momento concentrado. Caso ela não exista, insira 0:',
f1_amount = Entry(janela, textvariable = f1)
f1_amount.place(x=910, y=400)

Label(janela, text = 'Insira o momento concentrado (em N/m positivo ou negativo). Caso ele não exista, insira 0:', anchor = W).place(x=910
m1_amount = Entry(janela, textvariable= m1)
m1_amount.place(x=910, y=450)

Label(janela, text = 'Insira a posição da segunda força em metros. Caso ela não exista, insira 0:', anchor = W).place(x=910,y=480)
p2_amount = Entry(janela, textvariable= p2)
p2_amount.place(x=910, y=500)

Label(janela, text = 'Insira a força concentrada (em N/m positiva ou negativa). Caso ela não exista, insira 0:', anchor = W).place(x=910,y
f2_amount = Entry(janela, textvariable = f2)
f2_amount.place(x=910, y=550)

Label(janela, text = 'Insira em metros onde o carregamento começa:', anchor = W).place(x=910,y=580)
p3_amount = Entry(janela, textvariable= p3)
p3_amount.place(x=910, y=600)

Label(janela, text = 'Insira o valor do carregamento (em N/m negativo ou positivo):', anchor = W).place(x=910,y=630)
co_amount = Entry(janela, textvariable = co)
co_amount.place(x=910, y=650)

Label(janela, text = 'Insira em metros onde o carregamento termina:', anchor = W).place(x=910,y=680)
p4_amount = Entry(janela, textvariable = p4)
p4_amount.place(x=910, y=700)
```

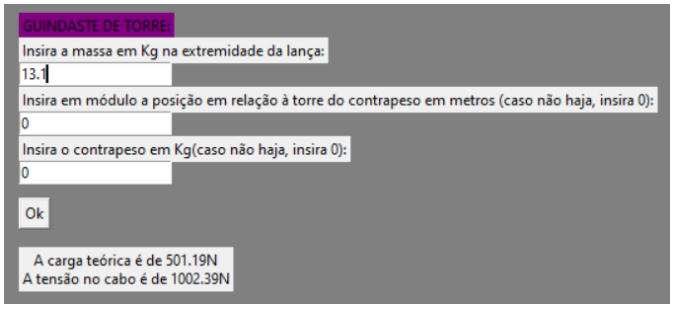
Fonte: Autoria própria (2022)

4 RESULTADOS E DISCUSSÃO

Neste capítulo, serão apresentados os resultados de todo projeto de construção do software referido anteriormente mostrando como um possível usuário inseriria valores por meio da interface e como o programa responderia de acordo com o código fonte já apresentado.

A Figura 29 ilustra um exemplo de funcionamento da simulação do guindaste de torre onde o usuário insere a massa pendurada na lança (13,1 kg), a posição do contrapeso (considerando-se a posição 0 dado que não existe contrapeso no exemplo) e seu módulo (considerando-se a módulo da força 0 dado que não existe contrapeso no exemplo) para obter a carga teórica aplicada à lança pela torre e a tensão no cabo.

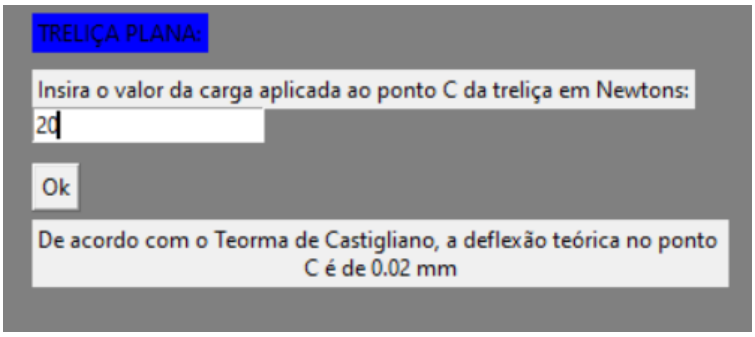
Figura 29 – Exemplo de dados de entrada inseridos pelo usuário e dados de saída do software para o guindaste de torre



Fonte: Autoria própria (2022)

A Figura 30 ilustra um exemplo de funcionamento da simulação da treliça plana onde o usuário insere a massa pendurada no ponto C (20 kg) como visto anteriormente para obter a deflexão do mesmo ponto:

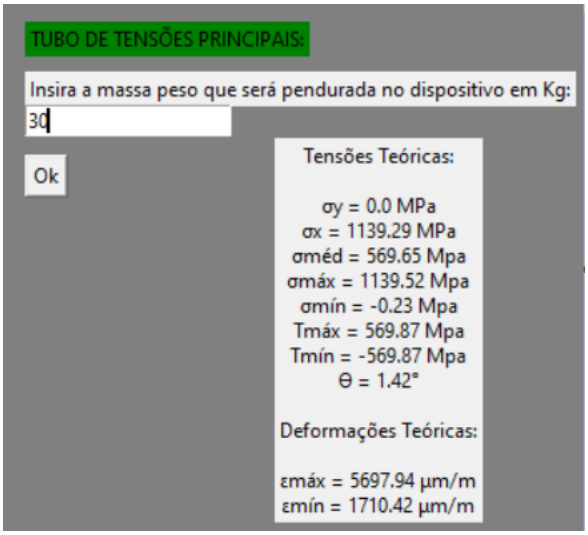
Figura 30 - Exemplo de dado de entrada inserido pelo usuário dados de saída do software para a treliça plana



Fonte: Autoria própria (2022)

A Figura 31 ilustra um exemplo de funcionamento da simulação do tubo de tensões principais no qual o usuário insere a massa pendurada na alavanca (30 kg) e obtém as tensões normais (x, y, média, mínima e máxima), tensões de cisalhamento (mínima e máxima), ângulo θ e as deformações teóricas (máxima e mínima) no ponto onde foram colocados os extensômetros.

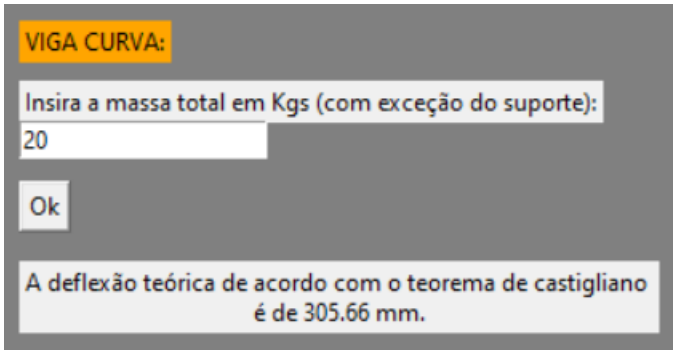
Figura 31 - Exemplo de dados de entrada inserido pelo usuário e dados de saída do software para o tubo de tensões principais



Fonte: Autoria própria (2022)

A Figura 32 a ilustra um exemplo de funcionamento da simulação da viga curva onde o usuário insere o valor pendurado no suporte da viga curva conforme foi visto nos capítulos anteriores (20 kg) e obtém a deflexão teórica de acordo com o Teorema De Castigliano:

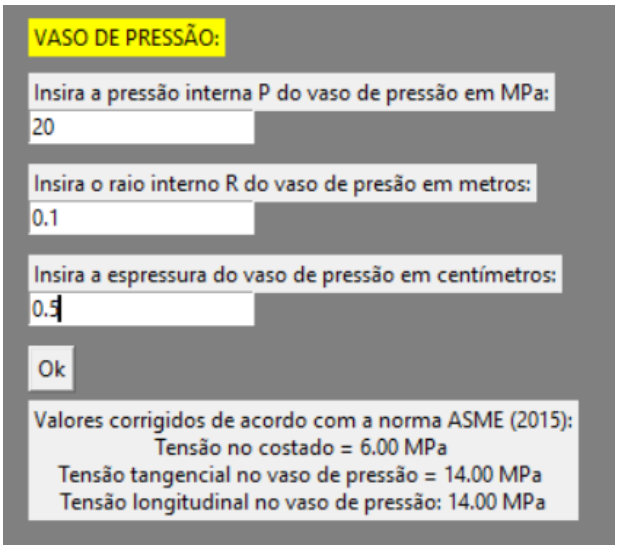
Figura 32 - Exemplo de dado de entrada inserido pelo usuário e dado de saída do software para a viga curva



Fonte: Autoria própria (2022)

A Figura 33 a ilustra um exemplo de funcionamento da simulação do vaso de pressão na qual o usuário insere valores para pressão interna (20 Mpa), o raio (0,1 m) e espessura (0.5 cm) obtendo assim os valores de acordo com a norma ASME (2015) para as tensões no costado, tangencial e longitudinal:

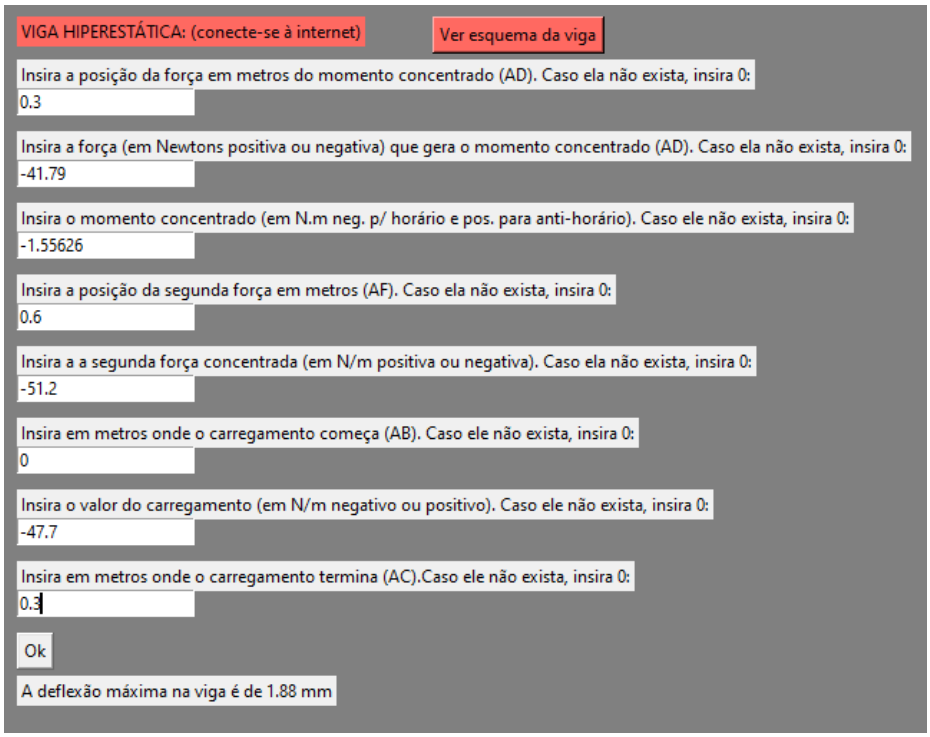
Figura 33 - Exemplo de dados de entrada inserido pelo usuário e dados de saída do software para o vaso de pressão



Fonte: Autoria própria (2022)

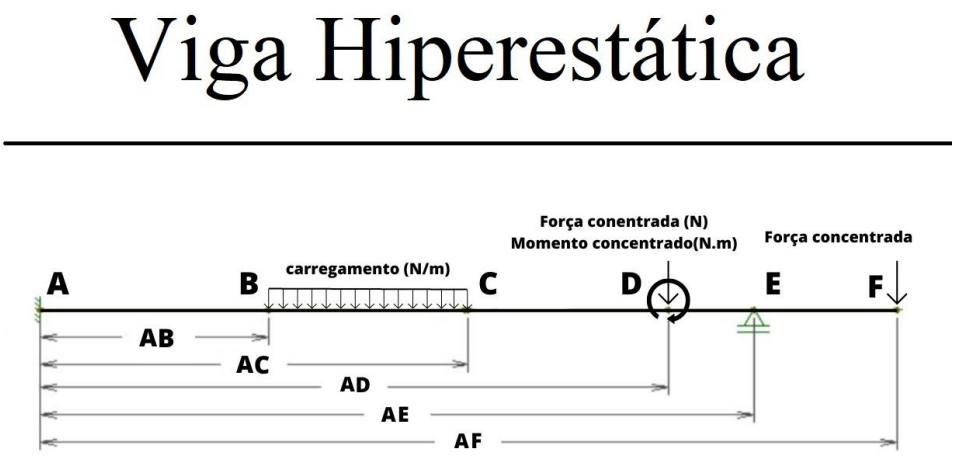
Por fim, a Figura 34 ilustra um exemplo de funcionamento da simulação da viga hiperestática onde o usuário insere a posição da força que gera o momento concentrado (0,3 m), a força que gera esse momento (-41,79 N), o momento concentrado gerado por essa força (-1,556260 N.m), a posição da segunda força (0.6 m), a força concentrada (-51,2 N), o ponto onde o carregamento começa (0), o valor do carregamento (-44,79 N/m) e o ponto onde o carregamento termina (0,3 m) para a obtenção da deflexão ao longo da viga. Também foi criado um botão para a visualização do esquema da viga para melhor orientação do usuário que quando clicado, abre a imagem ilustrada na Figura 35.

Figura 34 - Exemplo de dados de entrada inserido pelo usuário e dados de saída do software para a viga hiperestática



Fonte: Autoria própria (2022)

Figura 35 – Arranjo esquemático da viga hiperestática



Fonte: Autoria própria (2022)

5 CONCLUSÕES E SUGESTÕES PARA PRÓXIMOS TRABALHOS

Neste capítulo serão apresentadas as conclusões referentes ao software e também recomendações de possíveis melhorias adicionais para o mesmo ao longo do tempo.

5.1 CONCLUSÕES

O software mostrou-se prático e rápido dentro da proposta inicial que seria obter resultados com base teórica sólida para posterior comparação com os experimentos feitos no laboratório de estruturas da FEG. Além disso, é possível simular também valores de entrada que estão além das capacidades dos dispositivos por limitações de material, espaço, capacidade e etc.

5.2 SUGESTÕES PARA PRÓXIMOS TRABALHOS

Por hora, como já dito anteriormente o software atende parcialmente às necessidades dos alunos que utilizam o laboratório de estruturas visto que auxilia nos cálculos teóricos dos dispositivos presentes no mesmo.

No entanto, com o tempo o laboratório pode receber novos dispositivos didáticos e pode haver a necessidade de incorporá-los ao código fonte.

Outra melhoria possível seria uma interface mais comumente utilizada em softwares comerciais como *Ftool, Ansys* e *Solid Work.* Nesse contexto, é importante ressaltar que todo o código fonte foi comentado explicitando o que está sendo feito em cada função, método, objeto e etc.

Também seria importante acrescentar a verificação de casos absurdos de dados de entrada.

REFERÊNCIAS

ALUVINO, O. R. **Diretrizes para modelagem geométrica de um guindaste de torre destinado a aplicações em engenharia utilizando o software Inventor**. 2017. Trabalho de Iniciação Científica (PIBIC) (Técnico em Mecânica) - Colégio Técnico Industrial de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2017.

AMERICAN SOCIETY OF MECHANICAL ENGINEERS **ASME Seção VIII, Divisão 1**: código de projeto e de fabricação de vasos de pressão. Nova York, 2001.

BEER, F. P.; JOHNSTON J. R. **Mecânica dos materiais**. 3. ed. São Paulo: Pearson, 2008.

BEER, F. P. *et al*. **Mecânica dos materiais**. 7. ed. Porto Alegre: McGraw-Hill, 2015.

BEER, F. P. *et al*, **Mecânica dos materiais**. 9. ed. Porto Alegre: AMGH, 2012.

BORSE, Y.; SHARMA, A. K. Design analysis of pressure vessels at high stress zones using pro/e. v 4.0. **International Journal of Modern Engineering Research (IJMER)**, [*S. l.*], v. 2, n. 3, p. 991–995, 2012. Disponível em: https://www.ijmer.com/papers/vol2_issue3/BV23991995.pdf. Acesso em: 13 out. 2022.

BORTOLO, K. F.; LINHARES, J. C. Verificação da necessidade de dispositivos didáticos para o ensino na graduação em engenharia mecânica. *In*: CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA, 34., 2006, Passo Fundo. [**Anais eletrônicos...**]. Passo Fundo: ABENGE/UPF, 2006. p. 1128-1139. 1 CD-ROM. Disponível em: http://www.abenge.org.br/cobenge/legado/arquivos/8/sessoestec/art2117.pdf. Acesso em: 11 nov. 2022

CARRARO, D. F.; ABREU, J. P.; BARBOSA, M. M. **Projeto de um vaso de pressão para utilização em uma planta marítima de produção de petróleo.** 2013. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) – Fundação Oswaldo Aranha, Centro Universitário de Volta Redonda (UniFOA), 2013. Disponível em: https://www.academia.edu/19202081/UNIFOA_Eng_Mecanica. Acesso em: 12 jul. 2022

CAVAGUTI, A. S. **Análise comparativa de tensões e deformações específicas em uma estrutura treliçada, utilizando abordagem experimental e pelo método de elementos finitos**. 2015. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) - Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2015. Disponível em: http://hdl.handle.net/11449/124167. Acesso em: 28 jun. 2022.

CHEN, Y. *et al*. Safety supervision of tower crane operation on construction sites: an evolutionary game analysis. **Safety Science**, [*S.l.*], v. 152, 2022. Disponível em: https://doi.org/10.1016/j.ssci.2021.105578. Acesso em: 21 jul. 2022.

DIAS, R. A. A. **Dispositivo didático para aferição de deflexão horizontal em vigas curvas isostáticas.** 2017. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) - Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, Guaratinguetá, 2017.

DIWAKAR, A. *et al*. Control systems virtual labs: pedagogical and technological perspectives. **IEEE Xplore**, [*S.l.*], p. 483-488, 2013. Work presented at the IEEE International Conference On Control Applications (CCA), Hyderabad, 2013. Disponível em: https://ieeexplore.ieee.org/document/6662796. Acesso em: 17 set. 2022

ESTRUTURA CURRICULAR CURSO ENGENHARIA MECÂNICA INTEGRAL. **FEG-UNESP**, Guaratinguetá, 2007. Disponível em: https://www.feg.unesp.br/#!/graduacao/mecanica/integral/estrutura-curricular/. Acesso em: 20 jul. 2022.

FEISELl, L. D.; ROSA, A. J. The role of the laboratory in undergraduate engineering education. **Journal of Engineering Education**, [*S.l.*], v. 94, p. 121-130, 2005. Disponível em: https://onlinelibrary.wiley.com/doi/10.1002/j.2168-9830.2005.tb00833.x. Acesso em: 3 out. 2022.

GABER, A.; NASEEF, O. A.; ABDELBAKI, N. Creating an innovative generic virtual learning lab. In: WORLD CONGRESS ON COMPUTER AND INFORMATION TECHNOLOGY, 6 f., 2013, Sousse, Tunisia. **Proceedings**. New York: IEEE, 2013.

GERE, J. M.; GOODNO, B. J. **Mecânica dos materiais**.7. ed. São Paulo: Cengage Learning Brasil, 2010.

GLYNIADAKIS, S. *et al*. **Análise de tensões em um vaso de pressão para fins didáticos.** 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) - Faculdade de Engenharia De Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2018.

GRAPHICAL USER INTERFACES WITH TK. **Python**, [*S.l.*], [*s.d.*]. Disponível em: https://docs.python.org/3/library/tkinter.html#a-hello-world-program/. Acesso em: 14 nov. 2022

HIBBELER, R. C. **Resistência dos materiais**. 7. ed. São Paulo: Pearson Prentice-Hall, 2010.

IECKER, T. D. **Análise de tensões em vasos de pressão através do método de elementos finitos.** 2014. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ), Rio de Janeiro, 2014. Disponível em: https://www.docsity.com/pt/analise-de-tensoes-em-vasos-de-pressao-atraves-do-metodo-de-elementos-finitos/5033591/. Acesso em: 1 ago. 2022.

JOHNSTON, E. R. *et al*. **Mecânica dos materiais**. 7. ed. Porto Alegre: Bookman Companhia Ed, 2011.

LUTZ, M. **Programming python**. 4th ed. Sebastopol: O'Reilly Media, 2010.

MARTINS, M. S.; TOMAZINI, J. E. **Dispositivo para o ensino de medições de tensões e deformações principais**. Depositante: Universidade Estadual Paulista Júlio De Mesquita Filho. Depósito: 20 abr. 2016. Concessão: 24 out. 2017.

MARTINS, M. S.; TUPINAMBÁ, W. L. M; DE SOUZA, M. C. **Dispositivo didático para demonstração de deflexão em vigas**. Depositante: Universidade Estadual Paulista Júlio De Mesquita Filho. Depósito: 10 set. 2018. Concessão: 24 mar. 2020.

SALAHEDDIN, O.; SHANAB, S. A.; ANABTAWI, M. Augmented reality internet labs versus its traditional and virtual equivalence. **International Journal of Emerging Technologies in Learning**, [*S.l.*], v. 10, n. 3, p. 4-9, 2015. Disponível em: http://dx.doi.org/10.3991/ijet.v10i3.4354. Acesso em: 28 set. 2022.

SCIGLIANO, W. A. **Gruas:** o que é preciso para se usar grua. São Paulo: [*S.n.*], 2008.

SELL, R.; RÜÜTMANN, T.; SEILER, S. Inductive teaching and learning in engineering pedagogy on the example of remote labs. **International Journal of Engineering Pedagogy**, [*S.l.*], v. 4, n. 4, p. 12-15, 2014. Disponível em: http://dx.doi.org/10.3991/ijep.v4i4.3828. Acesso em: 4 July 2022.

TELLES, P. C. S. **Vaso de pressão**. 2. ed. Rio de Janeiro: LTC, 2001.

TKINTER — PYTHON INTERFACE TO TCL/TK. **Python**, [*S.l.*] [*s.d.*]. Disponível em: https://docs.python.org/3/library/tkinter.html/. Acesso em: 14 nov. 2022.

APÊNDICE A – Código fonte do software

APÊNDICE A – CÓDIGO FONTE DO SOFTWARE

```
# Bibliotecas:

from tkinter import *
from PIL import Image
import math
import urllib.request
import json
import time
import requests
import json
from svglib.svglib import svg2rlg
from reportlab.graphics import renderPDF, renderPM
import os
from PIL import ImageTk, Image



# Foto geral:

im = Image.open('geral.png')
im.show()
```

```
# Interface geral:

janela = Tk()
janela.title('Laboratório de estruturas - FEG')
janela.geometry("950x800")
janela['background'] = 'grey'


# Orientação:

texto_orientacao = Label(janela, text='Obs: Sempre utilize "."
em vez de "," como separador decimal!', bg='#add8e6',
font=("Arial", 12),fg='black')
texto_orientacao.place(x=15, y=750)

# Foto de saudação:

frame = Frame(janela, width=20, height=20, bg= 'grey')
frame.pack()
frame.place(x=110,y=20)
img = ImageTk.PhotoImage(Image.open("unesp_logo.png"))
label = Label(frame, image = img, bg='grey')
label.pack()

# 1) Treliça plana:
# resultado -> Deflexão do ponto C da treliça
# p -> Carga aplicada no ponto C


def deflexao_trelica_plana():

    # Título:

    lb = Label(janela, text ="TRELIÇA PLANA:", anchor = W, fg =
'black',bg='blue')
    lb.place(x=510,y=20)

    # Dados de entrada (interface):

    Label(janela, text = 'Insira o valor da carga aplicada ao
ponto C da treliça em Newtons:', anchor = W).place(x=510,y=50)
```

```
    p = Entry(janela)
    p.place(x=510, y=70)
    lb = Label(janela, text="Resultado:")
    lb.place(x=510, y=130)

    # Cálculo para saída:

    def bt_onclick():
        resultado = float(p.get())*1.0151*pow(10,-6)*1000
        lb["text"] = "De acordo com o Teorma de Castigliano, a
deflexão teórica no ponto \nC é de {:.2f} mm".format(resultado)

    # Botão "Ok (interface)"

    bt = Button(janela, text='Ok', command = bt_onclick)
    bt.place(x=510,y=100)

# 2) Vaso De Pressão:

# TVP-> Tensão no vaso de pressão
# TTVP-> Tensão Tangencial no vaso de pressão
# TLVP-> Tensão Longitudinal no vaso de pressão
# p -> Pressão interna
# r -> raio
# t -> expessura (10% menor que o raio interno)

def vaso_de_pressão():

    # Título:

    Label(janela, text = 'VASO DE PRESSÃO:', fg =
'black',bg='yellow').place(x=510, y=305)

    # Dados de entrada (interface):

    Label(janela, text = 'Insira a pressão interna P do vaso de
pressão em MPa:', anchor = W).place(x=510,y=335)
    p = Entry(janela)
    p.place(x=510, y=355)
```

```
    Label(janela, text = 'Insira o raio interno R do vaso de
presão em metros:', anchor = W).place(x=510,y=385)
    r = Entry(janela)
    r.place(x=510, y=405)
    Label(janela, text = 'Insira a espressura do vaso de
pressão em centímetros:', anchor = W).place(x=510,y=435)
    t = Entry(janela)
    t.place(x=510, y=455)
    lb = Label(janela, text="Resultado:")
    lb.place(x=510, y=515)

    # Resultado de saída:

    def bt_onclick():
        TVP =
(float(p.get())*float(r.get()))/(2*(float(t.get())))) +
0.2*(float(p.get()))
        TTVP = (float(p.get())*float(r.get())) +
0.6*(float(p.get()))
        TLVP =
(float(p.get())*float(r.get()))/(2*(float(t.get())))) +
0.6*(float(p.get()))
        lb["text"] = 'Valores corrigidos de acordo com a norma
ASME (2015):\nTensão no costado = {:.2f} MPa \nTensão
tangencial no vaso de pressão = {:.2f} MPa \nTensão
longitudinal no vaso de pressão: {:.2f} MPa'.format(TVP, TTVP,
TLVP)

    # Botão "Ok":

    bt = Button(janela, text='Ok', command = bt_onclick)
    bt.place(x=510,y=485)

# 3) Tubo de tensões principais:

# m -> Massa pendurada
# Mx-> Momento em x
# Iy-> Momento de inércia
# Jy-> Momento polar de inércia em y
# Txy-> Tensão de cisalhamento xy
```

```
def tensões_e_deformações_tubo():

    # Título:

    Label(janela, text = 'TUBO DE TENSÕES PRINCIPAIS:',fg =
'black',bg='green').place(x=80,y=305)

    # Dados de entrada (interface):

    Label(janela, text = 'Insira a massa peso que será
pendurada no dispositivo em Kg:', anchor = W).place(x=80,y=335)
    m = Entry(janela)
    m.place(x=80,y=355)

    lb = Label(janela, text="Resultado:")
    lb.place(x=80,y=415)

    def bt_onclick():

        # Tensões teóricas

        E = 200*pow(10,3)
        v = 0.3
        Iy = 1.1676*(pow(10,(-7)))
        J = (3.14/32)*(pow(0.064,4)-pow(0.0614,4))
        Mx = (float(m.get()))*10*0.40
        T = (float(m.get()))*10*0.42
        Iz = (3.14/64)*(pow(0.064,4)-pow(0.0614,4))
        Gx = (Mx*0.032)/Iz
        Gy = float(0)
        Txy = (T*0.032)/J
        G_med = (Gx+Gy)/2
        dif = Gx - G_med
        R = pow((pow(Txy,2)+pow(dif,2)),0.5)
        T_max = R
        T_min = -R
        G_max = G_med + R
        G_min = G_med - R
        sen_dois_teta = Txy/R
```

```
        CF = Gx - G_med
        tg_dois_teta = Txy/CF
        dois_teta = (math.atan(tg_dois_teta))*(180/3.14)
        teta = dois_teta/2

        # Deformações Teóricas:

        e_max = (G_max - v*G_min)/E
        e_min = -(-v*G_max + G_min )/E



        lb["text"] ='Tensões Teóricas:\n\n\u03C3y = {}
MPa\n\u03C3x = {:.2f} MPa\n\u03C3méd = {:.2f} Mpa\n\u03C3máx =
{:.2f} Mpa\n\u03C3mín = {:.2f} Mpa\n\u03A4máx = {:.2f}
Mpa\n\u03A4mín = {:.2f} Mpa\n\u03A4xy = {:.2f} Mpa \n\u03F4 =
{:.2f}°\n\nDeformações Teóricas:\n\nεmáx = {:.2f} µm/m\nεmín =
{:.2f} µm/m'.format(float(Gy),float(Gx)*pow(10,-
6),float(G_med)*pow(10,-6),float(G_max)*pow(10,-
6),float(G_min)*pow(10,-6),float(T_max)*pow(10,-
6),float(T_min)*pow(10,-6),float(Txy)*pow(10,-
6),float(teta),float(e_max),float(e_min)*(-1))



    # Botão "Ok":

    bt = Button(janela, text='Ok', command = bt_onclick)
    bt.place(x=80,y=385)


# 4-) Viga hiperestática:

def deflexao_viga_hiperestatica():

    global lb1
    lb1 = Label(janela, text="Resultado:")
    lb1.place(x=900, y=765)

    # Botão para visualizar a representação esquemática da viga
```

```
    def foto_viga_hip():
     im = Image.open('viga_hiperestatica.jpg')
     return im.show()

    bt_viga_hip = Button(janela, text='Ver esquema da viga',fg
= 'black',bg='#ff6961', command = foto_viga_hip)
    bt_viga_hip.place(x=1180,y=305)

    global p1
    global p2
    global p3
    global p4
    global f1
    global f2
    global co
    global m1

    p1 = StringVar()
    p2 = StringVar()
    p3 = StringVar()
    p4 = StringVar()
    f1 = StringVar()
    f2 = StringVar()
    co = StringVar()
    m1 = StringVar()


    # Título:

    Label(janela, text = 'VIGA HIPERESTÁTICA: (conecte-se à
internet)',fg = 'black',bg='#ff6961').place(x=900,y=305)


    # Entrada de dados
    Label(janela, text = 'Insira a posição da força em metros
do momento concentrado (AD). Caso ela não exista, insira 0:',
anchor = W).place(x=900,y=335)
    p1_amount = Entry(janela, textvariable = p1)
    p1_amount.place(x=900, y=355)
```

```
    Label(janela, text = 'Insira a força (em Newtons positiva
ou negativa) que gera o momento concentrado (AD). Caso ela não
exista, insira 0:', anchor = W).place(x=900,y=385)
    f1_amount = Entry(janela, textvariable = f1)
    f1_amount.place(x=900, y=405)

    Label(janela, text = 'Insira o momento concentrado (em N.m
neg. p/ horário e pos. para anti-horário). Caso ele não exista,
insira 0:', anchor = W).place(x=900,y=435)
    m1_amount = Entry(janela, textvariable= m1)
    m1_amount.place(x=900, y=455)

    Label(janela, text = 'Insira a posição da segunda força em
metros (AF). Caso ela não exista, insira 0:', anchor =
W).place(x=900,y=485)
    p2_amount = Entry(janela, textvariable= p2)
    p2_amount.place(x=900, y=505)

    Label(janela, text = 'Insira a a segunda força concentrada
(em N/m positiva ou negativa). Caso ela não exista, insira 0:',
anchor = W).place(x=900,y=535)
    f2_amount = Entry(janela, textvariable = f2)
    f2_amount.place(x=900, y=555)

    Label(janela, text = 'Insira em metros onde o carregamento
começa (AB). Caso ele não exista, insira 0:', anchor =
W).place(x=900,y=585)
    p3_amount = Entry(janela, textvariable= p3)
    p3_amount.place(x=900, y=605)

    Label(janela, text = 'Insira o valor do carregamento (em
N/m negativo ou positivo). Caso ele não exista, insira 0:',
anchor = W).place(x=900,y=635)
    co_amount = Entry(janela, textvariable = co)
    co_amount.place(x=900, y=655)

    Label(janela, text = 'Insira em metros onde o carregamento
termina (AC).Caso ele não exista, insira 0:', anchor =
W).place(x=900,y=685)
    p4_amount = Entry(janela, textvariable = p4)
```

```
    p4_amount.place(x=900, y=705)

    # Botão "Ok":

    bt = Button(janela, text='Ok', command = do_put)
    bt.place(x=900,y=735)

# Funções necessárias

def get_loads(force1_position, force1_mod,  moment_position,
moment_value, force2_position, force2_mod,
force_distributed_pos, force_distributed_mod,
force_distributed_pos2, force_distributed_mod2):
        return [
                {
                    'id': 'T-tlp7OuTL-Epd0dpSi7vQ',
                    'type': 'fixed',
                    'label': 'Engastamento',
                    'position': {
                        'value': 0,
                        'display': '0',
                        'units': 'm',
                        'unitsKey': 'meter',
                    },
                    'iconSvg': '<svg
xmlns:svg=\'http://www.w3.org/2000/svg\'
xmlns=\'http://www.w3.org/2000/svg\' viewBox=\'0 0 44 44\'
class=\'j_WRkA\'><style>.ky3ESg{stroke:rgba(72,72,72,1);stroke-
width:2}.j_WRkA{fill:none;stroke-linecap:round;stroke-
linejoin:round}</style><path d=\'M19.5,4.8L19.5,39.2
M24.5,4.8L24.5,39.2 M19.5,4.8L24.5,9.7 M19.5,9.7L24.5,14.6
M19.5,14.6L24.5,19.5 M19.5,19.5L24.5,24.5 M19.5,24.5L24.5,29.4
M19.5,29.4L24.5,34.3 M19.5,34.3L24.5,39.2\' class=\'ky3ESg load
_\'/></svg>',
                },
                {
                    'id': '4d5e8d63-6443-4af1-849d-
11e7648bb03c',
                    'type': 'force',
                    'label': 'Carga concentrada',
```

```
                    'position': {
                        'value': force1_position,
                        'display': '0.3',
                        'units': 'm',
                        'unitsKey': 'meter',
                    },
                    'magnitude': {
                        'value': force1_mod,
                        'display': '-41.8',
                        'units': 'N',
                        'unitsKey': 'newton',
                    },
                    'iconSvg': '<svg
xmlns:svg=\'http://www.w3.org/2000/svg\'
xmlns=\'http://www.w3.org/2000/svg\' viewBox=\'0 0 44 44\'
class=\'j_WRkA\'><style>.eRimDg{stroke:rgba(112,150,182,1);stro
ke-width:2}.j_WRkA{fill:none;stroke-linecap:round;stroke-
linejoin:round}</style><path d=\'M22,32.5L22,10.6
M18.6,26.7L22,32.5L25.4,26.7\' class=\'eRimDg load
_\'/></svg>',
                },
                {
                    'id': '2957a888-1685-45db-8fc6-
342b98056ea3',
                    'type': 'moment',
                    'label': 'Momento',
                    'position': {
                        'value': moment_position,
                        'display': '0.3',
                        'units': 'm',
                        'unitsKey': 'meter',
                    },
                    'magnitude': {
                        'value': moment_value,
                        'display': '-1.56',
                        'units': 'N-m',
                        'unitsKey': 'newton-meter',
                    },
                    'iconSvg': '<svg
xmlns:svg=\'http://www.w3.org/2000/svg\'
```

```
xmlns=\'http://www.w3.org/2000/svg\' viewBox=\'0 0 44 44\'
class=\'j_WRkA\'><style>.eRimDg{stroke:rgba(112,150,182,1);stro
ke-width:2}.j_WRkA{fill:none;stroke-linecap:round;stroke-
linejoin:round}</style><path d=\'M29.1,7.6A16 16,0,0,1,19.7
37.9 M25.4,42.2L19.7,37.9L24.9,32.8 M14.9,36.4A16 16,0,0,1,24.3
6.1 M18.6,1.8L24.3,6.1L19.1,11.2\' class=\'eRimDg load
_\'/></svg>',
                },
                {
                    'id': 'uQGZd_A5Q0uZ36EgxqX4yw',
                    'type': 'pin',
                    'label': 'Apoio simple',
                    'position': {
                        'value': 0.5,
                        'display': '0.5',
                        'units': 'm',
                        'unitsKey': 'meter',
                    },
                    'iconSvg': '<svg
xmlns:svg=\'http://www.w3.org/2000/svg\'
xmlns=\'http://www.w3.org/2000/svg\' viewBox=\'0 0 44 44\'
class=\'j_WRkA\'><style>.ky3ESg{stroke:rgba(72,72,72,1);stroke-
width:2}.j_WRkA{fill:none;stroke-linecap:round;stroke-
linejoin:round}</style><path d=\'M19.3,22a2.75,2.75 0 1,0
5.49,0a2.75,2.75 0 1,0 -5.5,0\' class=\'ky3ESg load _\'/><path
d=\'M36.9,30.6L7.1,30.6L22,4.8L36.9,30.6\' class=\'ky3ESg load
_\'/></svg>',
                },
                {
                    'id': 'fee94754-9755-4474-a316-
6362298bd5b7',
                    'type': 'force',
                    'label': 'Carga concentrada',
                    'position': {
                        'value': force2_position,
                        'display': '0.6',
                        'units': 'm',
                        'unitsKey': 'meter',
                    },
                    'magnitude': {
```

```
                        'value': force2_mod,
                        'display': '-51.2',
                        'units': 'N',
                        'unitsKey': 'newton',
                    },
                    'iconSvg': '<svg
xmlns:svg=\'http://www.w3.org/2000/svg\'
xmlns=\'http://www.w3.org/2000/svg\' viewBox=\'0 0 44 44\'
class=\'j_WRkA\'><style>.eRimDg{stroke:rgba(112,150,182,1);stro
ke-width:2}.j_WRkA{fill:none;stroke-linecap:round;stroke-
linejoin:round}</style><path d=\'M22,32.5L22,10.6
M18.6,26.7L22,32.5L25.4,26.7\' class=\'eRimDg load
_\'/></svg>',
                },
                {
                    'id': 'oisoCXKuSNC722b_RnI78w',
                    'type': 'distributed',
                    'label': 'Carga distribuída',
                    'position': {
                        'value': force_distributed_pos,
                        'display': '1',
                        'units': 'm',
                        'unitsKey': 'meter',
                    },
                    'magnitude': {
                        'value': force_distributed_mod,
                        'display': '-47.7',
                        'units': 'N/m',
                        'unitsKey': 'newton-per-meter',
                    },
                    'position2': {
                        'value': force_distributed_pos2,
                        'display': '0.3',
                        'units': 'm',
                        'unitsKey': 'meter',
                    },
                    'magnitude2': {
                        'value': force_distributed_mod2,
                        'display': '-47.7',
                        'units': 'N/m',
```

```
                        'unitsKey': 'newton-per-meter',
                    },
                    'iconSvg': '<svg
xmlns:svg=\'http://www.w3.org/2000/svg\'
xmlns=\'http://www.w3.org/2000/svg\' viewBox=\'0 0 44 44\'
class=\'j_WRkA\'><style>.eRimDg{stroke:rgba(112,150,182,1);stro
ke-width:2}.j_WRkA{fill:none;stroke-linecap:round;stroke-
linejoin:round}</style><path d=\'M10.6,29.6L10.6,13.4
M7.6,24.7L10.6,29.6L13.5,24.7 M18.2,29.6L18.2,13.4
M15.3,24.7L18.2,29.6L21.1,24.7 M25.8,29.6L25.8,13.4
M22.9,24.7L25.8,29.6L28.7,24.7 M33.4,29.6L33.4,13.4
M30.5,24.7L33.4,29.6L36.4,24.7
M10.6,13.4L18.2,13.4L25.8,13.4L33.4,13.4\' class=\'eRimDg load
_\'/></svg>',
                },
            ]

def do_put():

        force1_position = p1.get()
        force1_mod = f1.get()
        moment_position = p1.get()
        moment_value = m1.get()
        force2_position = p2.get()
        force2_mod = f2.get()
        force_distributed_pos = p3.get()
        force_distributed_mod = co.get()
        force_distributed_pos2 = p4.get()
        force_distributed_mod2 = co.get()



        var_list = [0.3, -41.79, 1.55626, 0.6, -51.2, 0.16, -
44.5, 0.440, -44.5]

        headers = {
            'Accept': 'application/json, text/plain, */*',
            'Accept-Language': 'en-DE,en;q=0.9,zh-
CN;q=0.8,zh;q=0.7,de-DE;q=0.6,de;q=0.5,en-US;q=0.4',
            'Connection': 'keep-alive',
```

```
            # Already added when you pass json=
            # 'Content-Type': 'application/json',
            'Origin': 'https://deflection.app',
            'Referer': 'https://deflection.app/',
            'Sec-Fetch-Dest': 'empty',
            'Sec-Fetch-Mode': 'cors',
            'Sec-Fetch-Site': 'cross-site',
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36',
            'sec-ch-ua': '"Google Chrome";v="107",
"Chromium";v="107", "Not=A?Brand";v="24"',
            'sec-ch-ua-mobile': '?0',
            'sec-ch-ua-platform': '"Windows"',
        }

        params = {
            'language': 'pt',
        }


        json_data = {
            'updatedOn': '2022-11-11T23:56:13Z',
            'beamLength': {
                'value': 0.6,
                'display': '0.6',
                'units': 'm',
                'unitsKey': 'meter',
            },
            'deflectionLimitDenominator': 360,
            'materialName': '',
            'elasticModulus': {
                'value': 73,
                'display': '73',
                'units': 'GPa',
                'unitsKey': 'gigapascal',
            },
            'weightDensity': {
                'value': 76982.2025,
                'display': '77×10³',
```

```
                'units': 'N/m³',
                'unitsKey': 'newton-per-cubic-meter',
            },
            'bendingStrength': {
                'value': 248211262.554061,
                'display': '248×10⁶',
                'units': 'Pa',
                'unitsKey': 'pascal',
            },
            'shearStrength': {
                'value': 143217898.493693,
                'display': '143×10⁶',
                'units': 'Pa',
                'unitsKey': 'pascal',
            },
            'yieldStrength': {
                'value': 248211262.554061,
                'display': '248×10⁶',
                'units': 'Pa',
                'unitsKey': 'pascal',
            },
            'crossSectionDesignation': '',
            'area': {
                'value': 114,
                'display': '114',
                'units': 'cm²',
                'unitsKey': 'centimeter-squared',
            },
            'momentOfInertia': {
                'value': 0.0257,
                'display': '25.7×10⁻³',
                'units': 'm⁴',
                'unitsKey': 'meter-fourth',
            },
            'applyBeamWeight': False,
            'loads': get_loads(force1_position,
force1_mod,  moment_position, moment_value, force2_position,
force2_mod, force_distributed_pos, force_distributed_mod,
force_distributed_pos2, force_distributed_mod2)
        }
```

```
        print(p1.get(), p2.get(), p3.get(), p4.get(), f1.get(),
f2.get(), co.get(), m1.get())

        response = requests.put('https://api-
2.deflection.ketchep.com/design/ejliCAfITveyzlGciAAsPg',
params=params, headers=headers, json=json_data)
        dado = json.loads(response.text)['results']['spans']
        print(dado)

        def get_max_value(dado):
            global lista_defl
            lista_defl = list()
            for elemento in dado:
                lista_defl.append(float(elemento['displacementM
ax']['value']))
            return print (max(lista_defl)),
        get_max_value(dado)

        lb1['text'] = 'A deflexão máxima na viga é de {:.2f}
mm'.format(max(lista_defl)*1000)


var_list = [0.3, -41.79, 1.55626, 0.6, -51.2, 0.16, -44.5,
0.440, -44.5]



# 5) Viga Curva:

# E -> Módulo de elasticidade do material
# Rc-> Raio de curvatura
# I -> Momento de inércia
# esp -> Espessura
# larg -> Largura
# m_sup -> Massa do suporte
# m_anilha_1 -> Massa da anilha 1
# m_anilha_2 -> Massa da anilha 2
# m_anilha_3 -> Massa da anilha 3
# deform -> Deformação
```

```
def deformação_viga_curva():

    #Túlulo:

    Label(janela, text = 'VIGA CURVA:',fg =
'black',bg='orange').place(x=510,y=645)

    # Dados:

    m_anilha_1 = 3.02
    m_anilha_2 = 4.64
    m_anilha_3 = 7.64
    larg = 31.75*pow(10,-3)
    esp = 3.175*pow(10,-3)
    Rc = 367.5*pow(10,-3)
    E = 200*pow(10,9)
    I = 84.68*pow(10,-12)
    m_sup = 0.86

    # Dados de entrada (interface):

    Label(janela, text = 'Insira a massa total em Kgs (com
exceção do suporte):', anchor = W).place(x=510,y=675)
    massa_anilhas = Entry(janela)
    massa_anilhas.place(x=510,y=695)

    lb = Label(janela, text="Resultado:")
    lb.place(x=510, y=755)

    def bt_onclick():

        carga_total = (float(massa_anilhas.get())+m_sup)*10
        deform = (float(carga_total)*pow(Rc,3)/(2*E*I))*1000


        lb["text"] ='A deflexão teórica de acordo com o teorema
de castigliano \né de {:.2f} mm.'.format(float(deform))
```

```
    # Botão "Ok":

    bt = Button(janela, text='Ok', command = bt_onclick)
    bt.place(x=510,y=725)


# 6) Guindaste de torre:

# massa_carga -> Massa de carga
# lanca_guindaste -> Comprimento da lança guindaste
# dist_torre_cabo -> Distância entre torre e cabo
# angulo_torre_cabo -> Ângulo entre torre e cabo
# f -> força F

# Equação de momento:

def guindaste_de_torre():

    #Túlulo:

    Label(janela, text = 'GUINDASTE DE TORRE:',fg =
'black',bg='purple').place(x=900,y=20)

   # Dados de entrada (interface):

    Label(janela, text = 'Insira a massa em Kg na extremidade
da lança:', anchor = W).place(x=900,y=50)
    massa_carga = Entry(janela)
    massa_carga.place(x=900,y=70)
    Label(janela, text = 'Insira em módulo a posição em relação
à torre do contrapeso em metros (caso não haja, insira 0):',
anchor = W).place(x=900,y=100)
    x_cp = Entry(janela)
    x_cp.place(x=900,y=120)
    Label(janela, text = 'Insira o contrapeso em Kg(caso não
haja, insira 0):', anchor = W).place(x=900,y=150)
    cp = Entry(janela)
    cp.place(x=900,y=170)
    lanca_guindaste = 1.95
    dist_torre_cabo = 1.00
```

```
    angulo_torre_cabo = 30
    lb = Label(janela, text="Resultado:")
    lb.place(x=900, y=230)

    # Resultado de saída:

    def bt_onclick():
        f = ((float(massa_carga.get())*9.81*1.95)-
(float(x_cp.get())*float(cp.get())))/(math.sin(math.radians(ang
ulo_torre_cabo))*dist_torre_cabo)
        lb["text"] = 'A carga teórica é de {:.2f} N\nA tensão
no cabo é de {:.2f}
N'.format(float(f),float(f)/math.sin(math.radians(30)))

    # Botão "Ok":

    bt = Button(janela, text='Ok', command = bt_onclick)
    bt.place(x=900,y=200)

guindaste_de_torre()
deformação_viga_curva()
deflexao_viga_hiperestatica()
tensões_e_deformações_tubo()
deflexao_trelica_plana()
vaso_de_pressão()

#Fechamento da janela:

janela.mainloop()
```