



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO MULTIDISCIPLINAR DE CARAÚBAS
CURSO DE BACHARELADO EM ENGENHARIA CIVIL

JOÃO MARCOS AZEVEDO DE OLIVEIRA

**DESENVOLVIMENTO DE UM PROGRAMA EM PYTHON PARA ANÁLISE
ESTRUTURAL DE PÓRTICOS PLANOS ISOSTÁTICOS PELO MÉTODO DAS
SEÇÕES**

CARAÚBAS

2019

JOÃO MARCOS AZEVEDO DE OLIVEIRA

**DESENVOLVIMENTO DE UM PROGRAMA EM PYTHON PARA ANÁLISE
ESTRUTURAL DE PÓRTICOS PLANOS ISOSTÁTICOS PELO MÉTODO DAS
SEÇÕES**

Monografia apresentada a Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia Civil.

Orientador: Prof. Dr. Manoel Dênis Costa Ferreira.

CARAÚBAS

2019

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

A48d Azevedo de Oliveira, João Marcos.
 Desenvolvimento de um programa em python para
 análise estrutural de pórticos planos isostáticos
 pelo método das seções / João Marcos Azevedo de
 Oliveira. - 2019.
 74 f. : il.

 Orientadora: Manoel Denis Costa Ferreira.
 Monografia (graduação) - Universidade Federal
 Rural do Semi-árido, Curso de Engenharia Civil,
 2019.

 1. Esforços internos. 2. Ftool. 3. Programação
 computacional. 4. Relatório. I. Costa Ferreira,
 Manoel Denis, orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

JOÃO MARCOS AZEVEDO DE OLIVEIRA

**DESENVOLVIMENTO DE UM PROGRAMA EM PYTHON PARA ANÁLISE
ESTRUTURAL DE PÓRTICOS PLANOS ISOSTÁTICOS PELO MÉTODO DAS
SEÇÕES**

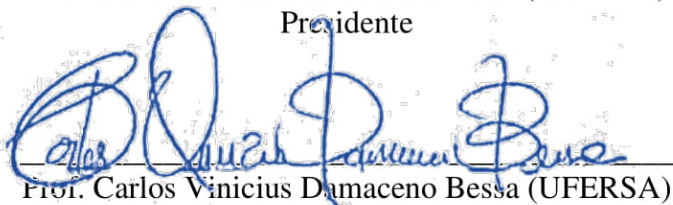
Monografia apresentada a Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Engenharia Civil.

Defendida em: 20 / 03 / 2019.

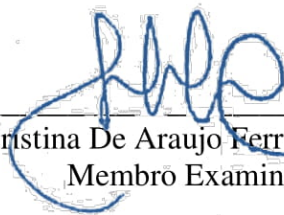
BANCA EXAMINADORA



Prof. Dr. Manoel Denis Costa Ferreira (UFERSA)
Presidente



Prof. Carlos Vinicius Damaceno Bessa (UFERSA)
Membro Examinador



Prof. MSc. Leonete Cristina De Araujo Ferreira Medeiros Silva (UFERSA)
Membro Examinador

A Deus, à minha família, a meus amigos e a todos que de alguma forma contribuíram para essa conquista.

AGRADECIMENTOS

A Deus, por tudo que tem feito em minha vida;

Aos meus pais, Miguel Arcanjo de Oliveira Filho e Soraia Maria Azevedo Cortes, por todo amor, apoio e incentivo para a conclusão deste trabalho;

À minha irmã, Maria Luiza Azevedo de Oliveira, pelo respeito e carinho, tornando minha vida muito mais alegre;

Aos meus grandes amigos, Layrton Saraiva, Thainny de Souza, Everton Diniz e Renata Marinho, pelo companheirismo, incentivo e ajuda no decorrer deste presente trabalho;

Às minhas Tias, Socorro e Maria, pela companhia e atenção em todos os momentos;

À minha segunda família, "os malucos da ufersa" (Anderson, Alexandre, Dennys, Túlio e Webert), por toda confiança, amizade, motivação e ajuda, que com certeza, foi fundamental para essa conquista;

A todos os moradores da "Batcaverna" (Arthur, Diogo, Everton, Heitor, Teslim e Wandick), por toda amizade e ajuda durante esse último período de faculdade;

Ao meu mais que um orientador, Manoel Denis Costa Ferreira, por compartilhar do seu conhecimento me ajudando sempre em todas as minhas dúvidas, sem ele seria praticamente impossível chegar até aqui;

Agradeço de forma especial, a minha namorada Valéria e seu filho Guilherme, por todo incentivo e amor. Sou feliz por tê-la ao meu lado. Me tornei uma nova pessoa desde o momento em que a conheci;

E a todos os docentes e discentes que de forma direta ou indireta contribuíram para minha formação;

Do fundo do coração, obrigado!

“Se construtores de edificios construíssem seus prédios como os programadores escrevem seus programas, o primeiro pica-pau que viesse poderia destruir uma civilização.”

(Gerald Weinberg)

RESUMO

Programas computacionais gráficos que apresentam resultados instantâneos e de melhor visualização, tornaram-se grandes aliados no ramo da construção civil, permitindo que estruturas complexas sejam melhor analisadas. A utilização de programas computacionais foi também inserida no ensino-aprendizagem, auxiliando os estudantes no entendimento do comportamento de estruturas. Mediante a necessidade de incentivar uma aproximação dos discentes com conceitos estruturais, o presente trabalho tem por objetivo desenvolver um software utilizando a linguagem Python, que possibilite analisar os esforços internos em pórticos planos isostáticos. Para validação do software, três exemplos foram testados. O primeiro teste foi de um pórtico isostático simples com carregamento distribuído de forma triangular, o segundo foi um pórtico simples com carregamento distribuído uniformemente composto por um elemento inclinado, ambos os problemas encontrados nas literaturas do ramo da análise estrutural, já o terceiro exemplo é baseado em parâmetros de uma estrutura com comportamento real. Após realizado o processo de entrada de dados pelo usuário e resolução dos problemas práticos, comprovou-se que o programa, comparado ao Ftool, em termos de resultados, revelou-se um ferramenta eficiente, ao fornecer soluções de forma rápida e fácil, o que torna possível ao usuário interpretar os resultados a partir de um relatório de cálculo.

Palavras-chave: Esforços internos. Ftool. Programação computacional. Relatório.

ABSTRACT

Graphical computer programs that present instantaneous results and better visualization, become great allies in the field of construction, allowing complex structures to be better analyzed. The use of computer programs was also inserted in teaching-learning, helping students in the behavior of structures. The idea of studying is one of the approaches aimed at the development of software using a Python language, which makes it possible for patients to analyze in isostatic flat plans. For software validation, three examples have been tested. The first test was a simple classic with triangular distributed distribution, the second was a simple simple with the distribution evenly distributed by an inclined element, both copies found in the literature of the branch of the structural analysis, is already available in parameters of one structure with real behavior. The process of entering data by the user and solving practical problems, the certification program, compliance with data in terms of results, the presentation of an efficient tool, the provision of solutions quickly and easily, which makes it possible to interpret the results from a calculation report.

Keywords: Internal efforts. Ftool. Computational programming. Report.

LISTA DE FIGURAS

Figura 1 - Níveis de abstração para uma estrutura na análise estrutural	19
Figura 2 - Estruturas reticulares formadas por elementos unidimensionais.....	20
Figura 3 - Estruturas bidimensionais	21
Figura 4 - Estruturas tridimensionais	21
Figura 5 - Representação dos apoios em modelos planos de estruturas	24
Figura 6 - Representação dos esforços internos de um elemento estrutural quando submetido a um sistema de forças.....	25
Figura 7 - Representação dos esforços internos em estruturas planas.....	26
Figura 8 - Diagramas ou Linhas de Estado dos esforços solicitantes internos.....	27
Figura 9 - Composição das estruturas reticulares	28
Figura 10 - Ação de um carregamento sobre uma estrutura.....	29
Figura 11 - Representação dos tipos de carregamentos distribuídos mais comuns em estruturas reticulares.....	30
Figura 12 - Equações do equilíbrio estático	30
Figura 13 - Sistema de referência global e local de um pórtico	33
Figura 14 - Pórticos simples	34
Figura 15 - Pórtico composto	35
Figura 16 - Blocos em python	37
Figura 17 - Exemplo de variáveis do tipo strings	38
Figura 18 - Representação de uma lista.....	40
Figura 19 - Representação de uma lista.....	40
Figura 20 - Representação de uma lista.....	40
Figura 21 - Caixa preta recebendo mensagens	43
Figura 22 - Esquema didático de hierarquia das classes	44
Figura 23 - Exemplo de código para geração de arquivo html.....	45
Figura 24 - Hierarquia das classes implementadas.....	46
Figura 25 - Classe problema estrutural dentro do IDE Eclipse	47
Figura 26 - Dados de entradas	48
Figura 27 - Código da função geradora do DLC	48
Figura 28 - Código da função calcular reações	49
Figura 29 - Código gerador do arquivo HTML	50
Figura 30 - Pórtico biapoiado com carregamento triangular	51

Figura 31 - Problema estrutural - exemplo 01	52
Figura 32 - DCL - exemplo 01	52
Figura 33 - Cálculo das reações de apoio - exemplo 01	53
Figura 34 - Seccionamento local da barra 1 - exemplo 01	53
Figura 35 - Função dos esforços internos da barra 1 - exemplo 01	54
Figura 36 - Seccionamento local da barra 2 - exemplo 01	54
Figura 37 - Função dos esforços internos da barra 1 - exemplo 01	55
Figura 38 - Diagrama de esforço normal – Exemplo 01	55
Figura 39 - Diagrama de esforço normal/ FTOOL - Exemplo 01	56
Figura 40 - Diagrama de esforço cortante – Exemplo 01	56
Figura 41 - Diagrama de esforço cortante/ FTOOL - Exemplo 01	57
Figura 42 - Diagrama de momento fletor – Exemplo 01	57
Figura 43 - Diagrama de momento fletor/ FTOOL - Exemplo 01	58
Figura 44 - Pórtico biapoiado com elemento inclinado e carregamento uniforme	58
Figura 45 - Problema estrutural - exemplo 02	59
Figura 46 - DCL - exemplo 02	59
Figura 47 - Cálculo das reações de apoio - exemplo 02	60
Figura 48 - Seccionamento local da barra 1 - exemplo 02	60
Figura 49 - Função dos esforços internos da barra 1 - exemplo 02	61
Figura 50 - Seccionamento local da barra 2 - exemplo 02	61
Figura 51 - Função dos esforços internos da barra 2 - exemplo 02	61
Figura 52 - Seccionamento local da barra 3 - exemplo 02	62
Figura 53 - Função dos esforços internos da barra 3 - exemplo 02	62
Figura 54 - Seccionamento local da barra 4 - exemplo 02	62
Figura 55 - Função dos esforços internos da barra 3 - exemplo 02	63
Figura 56 - Diagrama de esforço normal – Exemplo 02	63
Figura 57 - Diagrama de esforço normal/ FTOOL - Exemplo 02	63
Figura 58 - Diagrama de esforço cortante – Exemplo 02	64
Figura 59 - Diagrama de esforço cortante/ FTOOL - Exemplo 02	64
Figura 60 - Diagrama de momento fletor – Exemplo 02	65
Figura 61 - Diagrama de momento fletor/ FTOOL - Exemplo 02	65
Figura 62 - Galpão em arco	66
Figura 63 - Galpão metálico em arco sob influência do peso próprio	66
Figura 64 - Problema estrutural - exemplo 03	67

Figura 65 - DCL - exemplo 03	68
Figura 66 - Cálculo das reações de apoio - exemplo 03.....	68
Figura 67 - Diagrama de esforço normal – Exemplo 03	69
Figura 68 - Diagrama de esforço normal/ FTOOL - Exemplo 03.....	69
Figura 69 - Diagrama de esforço cortante – Exemplo 03.....	70
Figura 70 - Diagrama de esforço cortante/ FTOOL - Exemplo 03	70
Figura 71 - Diagrama de momento fletor – Exemplo 03.....	71
Figura 72 - Diagrama de momento fletor/ FTOOL - Exemplo 03	71

LISTA DE QUADROS

Quadro 1 - Classificação das ações	22
Quadro 2 - Cargas em modelos estruturais.....	29
Quadro 3 - Operações com números	38
Quadro 4 - Operadores relacionais	39
Quadro 5 - Operadores lógicos	39
Quadro 6 - Conceitos essenciais da Programação Orientada a Objetos	42

LISTA DE SIGLAS

HTML	HyperText Markup Language (linguagem de marcação de hipertexto)
IDE	Intergrated Development Environment (Ambiente de Desenvolvimento Integrado)
POO	Programação Orientada a Objetos
POP	Programação Orientada a Procedimentos

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 CONSIDERAÇÕES INICIAIS	15
1.2 OBJETIVOS.....	16
1.2.1 Objetivos Gerais.....	16
1.2.2 Objetivos Específicos.....	16
1.3 PROBLEMÁTICA E JUSTIFICATIVA.....	16
1.4 ESTRUTURA DO TRABALHO.....	17
2 REVISÃO DA LITERATURA.....	18
2.1 ABORDAGEM SOBRE A ENGENHARIA ESTRUTURA	18
2.2 ANÁLISE ESTRUTURAL	18
2.3 ESTRUTURAS E SEUS ELEMENTOS	19
2.3.1 Definição de estrutura	19
2.3.2 Tipos de sistemas estruturais.....	20
2.4 AÇÕES ATUANTES NAS ESTRUTURAS	21
2.4.1 Esforços solicitantes externos	22
2.4.2 Esforços solicitantes internos.....	24
2.5 ESTRUTURAS RETICULARES	27
2.5.1 Cargas em estruturas reticulares	28
2.5.2 Equilíbrio estático	30
2.5.3 Estabilidade e estaticidade de estruturas planas	31
2.6 PÓRTICOS ISOSTÁTICOS PLANOS	31
2.6.1 Sistema de referência.....	32
2.6.2 Pórticos simples.....	33
2.6.3 Pórticos compostos.....	34
2.7 LINGUAGEM PYTHON	35
2.7.1 Apresentando o Python	35
2.7.2 Conceitos básicos	36
2.7.3 Variáveis	37
2.7.4 Orientação a objetos	41
2.7.5 Arquivos – Geração HTML	44

3 METODOLOGIA.....	46
3.1 CLASSES IMPLEMENTADAS	46
3.2 DETERMINAÇÃO DO PROBLEMA ESTRUTURAL.....	47
3.3 RESOLUÇÃO DO PROBLEMA ESTRUTURAL	48
4 RESULTADOS	51
4.1 EXEMPLO 01 - PÓRTICO BIAPOIADO COM CARREGAMENTO TRIANGULAR.....	51
4.2 EXEMPLO 02 – PÓRTICO BIAPOIADO COM CARREGAMENTO UNIFORME	58
4.3 EXEMPLO 03 – GALPÃO COM COBERTURA EM ARCO	66
5 CONCLUSÃO.....	72
REFERÊNCIAS	73

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

A atual sociedade está sofrendo intensas transformações, em um ritmo cada vez mais acelerado, de maneira que necessita possuir informações rápidas e precisas, facilitando tarefas de modo a otimizar o tempo. Adaptando-se a esse novo cenário, a engenharia civil, associada às ferramentas tecnológicas está evoluindo e permitindo a facilidade de acesso a essas informações. Uma dessas ferramentas é a indústria de softwares, ambiente pelo qual a computação gráfica, sistemas de informação e métodos matemáticos e físicos aprimoram processos construtivos (BEZERRA; SILVA; LIMA, 2017).

Para Kimura (2007), a evolução na informática influencia diretamente a maneira como os projetos estruturais são processados, sendo sua aplicação relevante em todas as suas etapas. Programas computacionais gráficos que apresentam resultados instantâneos e de melhor visualização, tornaram-se grandes aliados no ramo da construção civil, permitindo que estruturas complexas sejam melhor analisadas. A utilização de programas computacionais foi também inserida no ensino-aprendizagem, auxiliando os estudantes no entendimento do comportamento de estruturas.

Nesse contexto, Branchier (2017), menciona a relevância da utilização da informática como ferramenta para o ensino e aprendizagem dentro da universidade, essencialmente nos cursos de ciências exatas, como nas engenharias para análises de sistemas estruturais. O desenvolvimento de programas são facilitadores na realização de cálculos e modelagem estrutural e têm como objetivo resumir e dinamizar situações para melhor compreensão dos usuários.

Dessa forma, a execução de análise estrutural, realizada de forma manual ou sem auxílio de softwares, pode tornar esse processo árduo e complexo. Porém, o crescimento tecnológico computacional, sobretudo no ramo da computação gráfica, proporcionou o surgimento de softwares com ambientes gráficos intuitivos, facilitando a automatização dessas etapas de análise (PRADO, 2012).

Assim, este trabalho procurou com o auxílio de um software desenvolvido em linguagem Python, resolver problemas estruturais isostáticos em pórticos planos sujeitos a diversas combinações de carregamentos e vinculações. Proporcionando ao acadêmico de engenharia civil a utilização de tal software como ferramenta de estudo para a comparação dos resultados calculados a mão.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

Desenvolver um programa capaz de apresentar ao seu usuário um relatório com todos os dados e cálculos utilizados para análise estrutural de pórticos isostáticos utilizando a linguagem de programação Python.

1.2.2 Objetivos Específicos

- Desenvolver um relatório de cálculo contendo todas as equações para as reações nos apoios, momentos fletores e esforços cortantes;
- Apresentar os gráficos de esforço normal, momento fletor e esforço cortante;
- Validar o software por meio de exemplos de aplicações.

1.3 PROBLEMÁTICA E JUSTIFICATIVA

O desenvolvimento acelerado da tecnologia computacional proporcionou a evolução de softwares voltados à área da engenharia estrutural, a fim de produzir mecanismos que possam resolver inúmeros problemas, reduzindo o tempo e custos com projetos. É esta percepção que motiva o emprego de métodos computacionais para a construção de um software que gere respostas dinâmicas e precisas para cada problema estrutural de pórticos planos isostáticos.

Ao longo da trajetória acadêmica, o aluno de engenharia civil se depara com diversos casos relacionados a análise estrutural, que em sua maioria, apresentam certa complexidade, tornando-os trabalhosos por demandar uma série de processos custando um longo período de tempo e muitas vezes inviáveis de serem solucionados sem o auxílio de algum artifício computacional.

Neste sentido, este trabalho busca de forma interativa apresentar ao usuário do software uma análise de todos os esforços presentes em um pórtico, por meio do traçado de diagramas e da disponibilização de um relatório (requisito diferencial entre outros softwares de análise estrutural) com todos os procedimentos necessários para a obtenção de tais gráficos, tornando a aprendizagem e atividade educacional mais intuitivas.

Para tanto, será possível o desenvolvimento de um programa que possa exibir para o seu usuário um relatório com o passo-a-passo oriundo da análise de pórticos?

1.4 ESTRUTURA DO TRABALHO

Ao longo desta seção foi apresentado a proposta do presente estudo, incluindo a problemática e justificativas para a sua realização, além da sua relevância e objetivos.

Na seção 2 são apresentados os princípios e elementos fundamentais das estruturas planas reticulares dando enfoque nas estruturas denominadas por pórtico. Como consequência delas, são apresentados as reações e esforços internos que nelas atuam quando submetidas a carregamentos. Ainda nessa seção é apresentado a linguagem de programação Python e suas funcionalidades para implementação computacional de cálculo estrutural.

Na sequência, é mostrado como funciona o programa criado para cálculo de pórticos: a estrutura do programa, a estrutura dos dados de entrada, e as etapas de cálculo, a qual permite realizar a análise estrutural.

Na seção 4 é feita a validação do programa por meio de exemplos de estruturas planas isostáticas, comparando os resultados obtidos com os apresentados pelo programa FTOOL e em algumas literaturas.

E por fim é realizado a síntese do trabalho e de seus resultados, e são feitas algumas sugestões para trabalhos futuros.

2 REVISÃO DA LITERATURA

2.1 ABORDAGEM SOBRE A ENGENHARIA ESTRUTURA

Segundo Martha (2010), a engenharia estrutural empiricamente origina-se, a partir das grandes obras do mundo antigo, tais como as pirâmides do antigo Egito, estradas, pontes e grandes muralhas da Grécia e da Roma Antiga. Por outro lado, a formalização teórica da engenharia estrutural surgiu a partir da publicação do livro *Duas Ciências*, de Galileu em 1638, dando origem a todo desenvolvimento da ciência desde o século XVII até os dias de hoje. Com o passar dos séculos, inúmeros matemáticos e cientistas contribuíram para formalizar a engenharia estrutural conhecida nos dias atuais.

As teorias científicas da engenharia estrutural permitem que os engenheiros estipulem as forças e solicitações que podem atuar nas estruturas ou nos elementos de forma segura, da mesma forma, os materiais e as dimensões devem ser apropriados a utilização sem que prejudiquem o funcionamento (MARTHA, 2010).

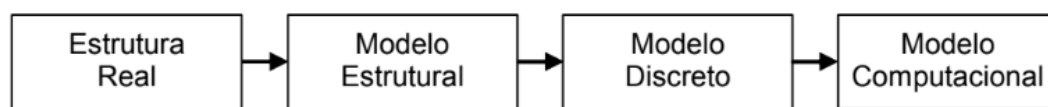
Desta maneira, nota-se que a engenharia estrutural tem enfoque no cálculo e determinação de padrões de estruturas adequados as construções reais de obras e edificações.

2.2 ANÁLISE ESTRUTURAL

Segundo Sussekind (1981), a análise estrutural é um dos ramos da mecânica dos materiais que trata das estruturas, onde é realizado o estudo para a determinação dos esforços e deformações a que essas estruturas ficam submetidas quando solicitadas devido a influência de agentes externos, como por exemplo os carregamentos.

Martha (2007) estabelece quatro níveis de abstração presentes na análise estrutural moderna (figura 1). O primeiro nível remete à estrutura física, isto é, a real, o segundo nível corresponde ao modelo analítico que serve para representar matematicamente a estrutura, tal modelo apresentará todas as teorias e hipóteses baseadas em leis físicas e leis constitutivas dos materiais.

Figura 1 - Níveis de abstração para uma estrutura na análise estrutural



Fonte: Martha (2010)

O terceiro nível refere-se ao modelo discreto inserido de diversos parâmetros ao modelo estrutural de forma a representar o comportamento de uma estrutura, e por fim, o quarto nível é a conversão do modelo discreto em modelo computacional através da implementação em softwares, sendo esse último, o objeto de estudo deste trabalho.

2.3 ESTRUTURAS E SEUS ELEMENTOS

2.3.1 Definição de estrutura

As estruturas podem ser definidas como a composição de uma ou mais peças que são ligadas entre si e ao meio exterior, de modo a formar um conjunto estável, isto é, aquele capaz de receber solicitações externas, absorve-las internamente e transmiti-las até seus apoios ou vínculos, onde estas encontrarão seu sistema de forças externas equilibrantes, chamadas de forças reativas (SUSSEKIND, 1981).

O conceito de estrutura referindo-se à engenharia civil, especificamente, denomina-se a parte resistente de uma construção, à qual se aplica o conceito geral apresentado anteriormente (ALMEIDA, 2009).

Para Soriano (2010), toda e qualquer estrutura ao ser construída deve apresentar as propriedades de resistência e de rigidez em seus elementos estruturais, isto é, ter capacidade resistente ao próprio peso e demais cargas, dentro de certos limites, sem sofrer deformações e vibrações excessivas que gerem grandes variações de suas dimensões originais.

De acordo com Almeida (2009), "resistência é a capacidade de transmitir as forças internamente, molécula por molécula, dos pontos de aplicação aos apoios, sem que ocorra a ruptura da peça."

Para a análise da resistência de uma estrutura é necessário a determinação dos esforços solicitantes internos (Realizado na Análise Estrutural) ou Estática das Construções, e das tensões internas (Realizado na Resistência dos Materiais).

Por outro lado, a rigidez é a resistência do material em não apresentar deformações excessivamente, para a carga estabelecida, caso contrário, comprometeria os aspectos visuais e

funcionais do elemento estrutural. A realização dos cálculos das deformações no componente estrutural é encontrada na resistência dos materiais (ALMEIDA, 2009).

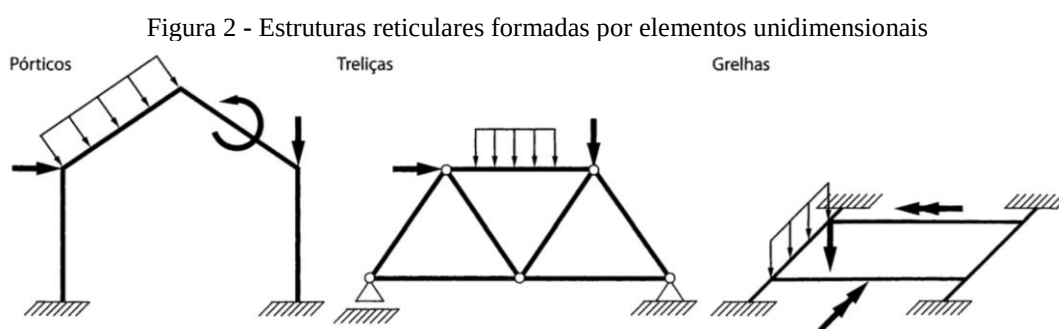
2.3.2 Tipos de sistemas estruturais

Todos os sistemas estruturais são compostos de vários elementos estruturais básicos. Em função das dimensões e direções das ações, tais elementos podem ser classificados em unidimensionais, bidimensionais ou tridimensionais (ALMEIDA, 2009).

- Unidimensionais

Também denominada estruturas reticulares, são aquelas compostas por elementos unidimensionais, ou seja, em que o comprimento prevalece sobre as outras duas dimensões (ALMEIDA, 2009).

Sussekind (1981) afirma que para as peças estudadas deste caso recebem a denominação de barras, e que sua análise pode ser realizada considerando-a representada pelo seu eixo. Na imagem abaixo são fornecidos alguns exemplos de estruturas reticulares.

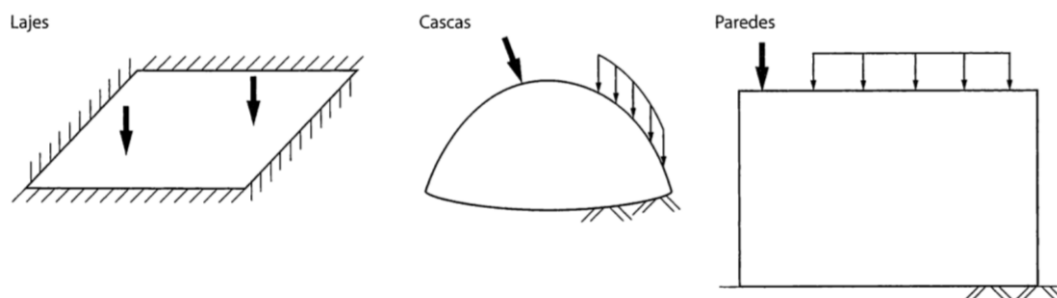


Fonte: Almeida (2009)

- Bidimensionais

Conforme Almeida (2009), estruturas bidimensionais são aquelas em que duas de suas dimensões prevalecem em relação a terceira. A figura 3 apresenta exemplos desse tipo de estrutura.

Figura 3 - Estruturas bidimensionais

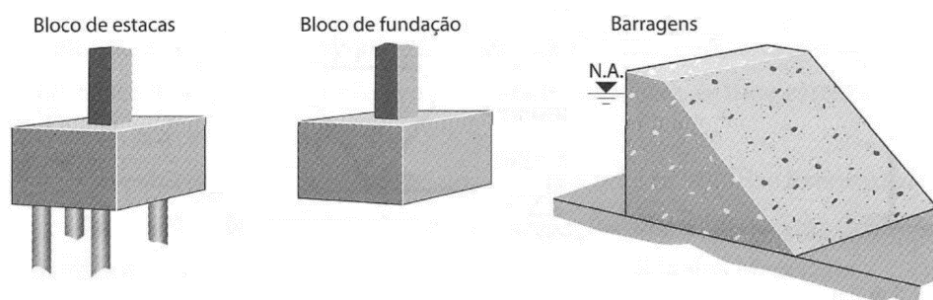


Fonte: Almeida (2009)

- Tridimensionais

Já as estruturas tridimensionais, Almeida (2009) consideram que são estruturas sólidas, onde suas três dimensões possuem mesma ordem de grandeza, logo todas elas são levadas em consideração. Estão ilustradas na figura 4 exemplos de estruturas tridimensionais.

Figura 4 - Estruturas tridimensionais



Fonte: Almeida (2009)

2.4 AÇÕES ATUANTES NAS ESTRUTURAS

Para Soriano (2010), as ações atuantes numa estrutura podem vir a surgir devido a forças, deformações impostas ou de comportamento do material ao longo do tempo, e por variações de temperatura. Em resumo, essas ações se classificam conforme indicado no Quadro 1.

Quadro 1 - Classificação das ações

Esforços Solicitantes	Externos	Ativos	Permanentes
			Acidentais
			Excepcionais
	Internos	Reativos	
		Forças: N , Q_x e Q_y	
		Momentos: T , M_y e M_z	

Fonte: Adaptado Soriano (2010)

Este trabalho prioriza o estudo dos esforços solicitantes que podem atuar em uma estrutura, e por esse motivo será apresentado conceitos acerca dessa temática.

2.4.1 Esforços solicitantes externos

2.4.1.1 Ações externas ativas

As ações externas são aquelas que agem sobre a estrutura e podem ser classificadas em ativas e reativas. As consideradas ativas, são provocadas por agentes do meio externo e podem ser classificadas quanto à posição, à duração, à forma de aplicação e à variação com o tempo (ALMEIDA, 2009).

- **Quanto à posição**

Se subdividem em fixas e moveis. As fixas são carregamentos que não variam sua posição, enquanto as móveis são o caso contrário (ALMEIDA, 2009).

- **Quanto à duração**

Dividem-se em permanentes, acidentais e excepcionais. A primeira são as que ocorrem durante toda a vida útil da estrutura e surgem devido as forças gravitacionais atuantes na estrutura, como o peso próprio da mesma e das partes que se apoiam permanentemente sobre ela, mas não possuem funções estruturais. As acidentais são oriundas de ações que podem ou não agir sobre as estruturas, como o peso das pessoas, veículos, assim como a ação do vento e da variação da temperatura. Já as excepcionais são as que possuem duração extremamente curta,

intensidade elevada e de ocorrências mínimas, pode ser citado nesse caso as explosões e incêndios (SORIANO, 2010).

- **Quanto à forma de aplicação**

Existem as cargas concentradas, que são aquelas que transmitem uma força por meio de um único ponto, e as distribuídas, quando à transmissão de uma força de maneira distribuída, seja ao longo de um comprimento ou de uma superfície (ALMEIDA, 2009). No tópico 2.5 é apresentado a forma como essas forças são representadas em uma estrutura.

- **Quanto à variação com o tempo**

Para Almeida (2009), as cargas podem ser subdivididas em estáticas e dinâmicas. As estáticas são as que não variam de intensidade ao longo do tempo. Enquanto nas dinâmicas ocorre variação da ação ao longo do tempo.

2.4.1.2 Ações externas reativas

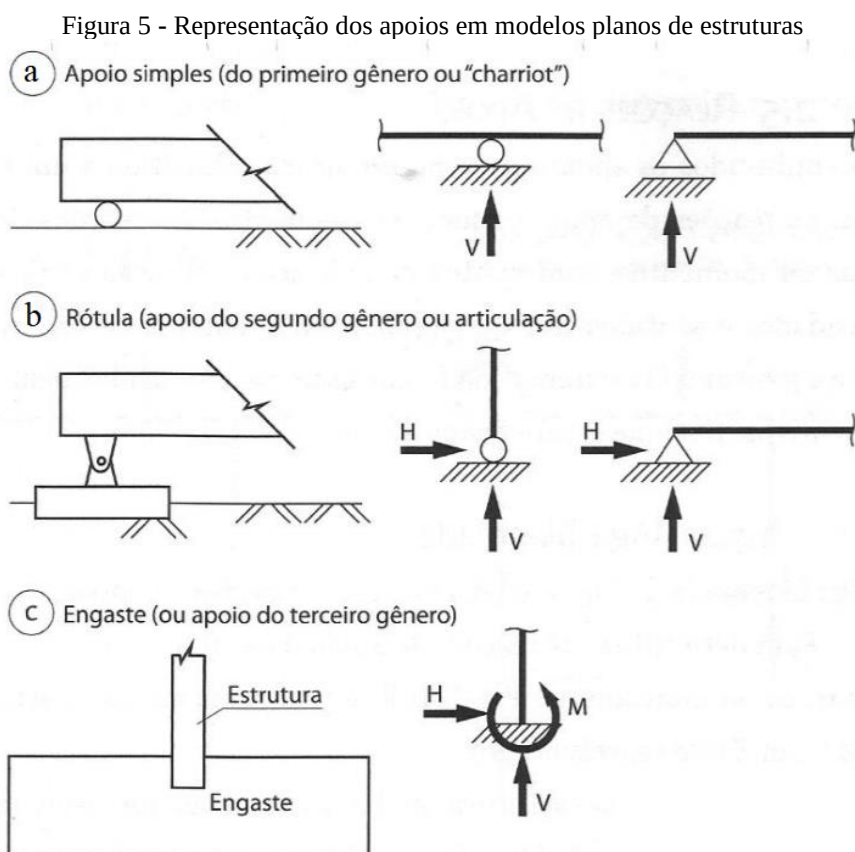
São os esforços que se desenvolvem nos vínculos externos ou apoios a partir da restrição aos movimentos de uma estrutura, e são classificados em função do número de restrições impostas (SORIANO, 2010).

Segue a classificação dos apoios em função do número de deslocamentos impedidos:

- **Apoio simples (do primeiro gênero):**
 - **Impede** a translação em uma das direções.
 - **Permite** a translação na direção perpendicular à impedida.
 - **Permite** a rotação em torno do eixo Z.
- **Rótula (apoio do segundo gênero ou articulação):**
 - **Impede** as translações nas duas direções (X e Y).
 - **Permite** a rotação em torno do eixo Z.
- **Engaste (ou apoio do terceiro gênero):**
 - **Impede** as translações nas duas direções (X e Y).

- **Impede** a rotação em torno do eixo Z.

A figura 5 apresenta, para os modelos estruturais planos de vigas, treliças e pórticos, esse último sendo o objeto de estudo deste trabalho, os apoios associados a estas direções de deslocamento e as formas simplificadas de serem representados.



Fonte: Almeida (2009)

2.4.2 Esforços solicitantes internos

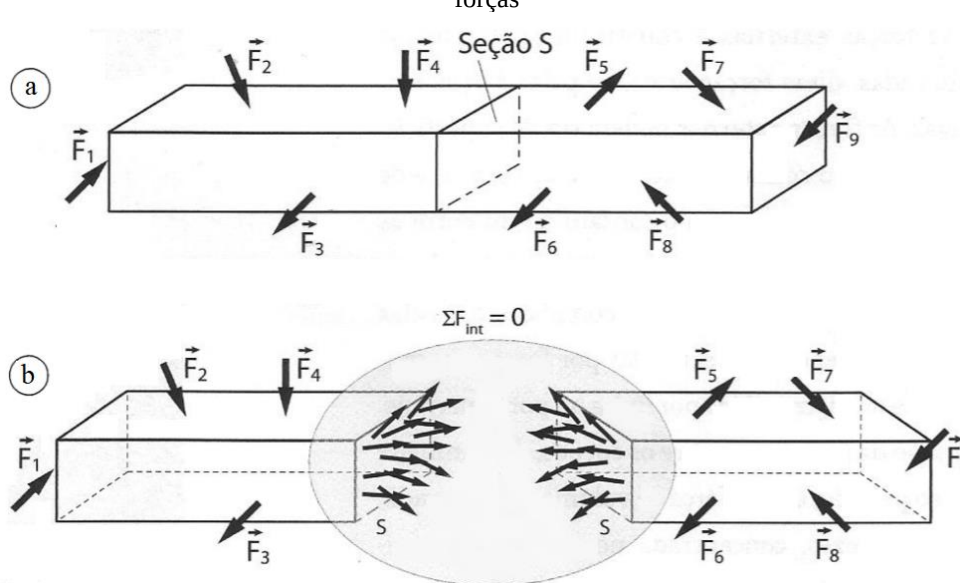
A Análise Estrutural tem como objetivo a determinação das reações de apoio e dos esforços solicitantes internos. Descobrendo-se as reações de apoio, no caso de estruturas isostáticas, é possível determinar o comportamento interno da mesma (SORIANO, 2010).

Conforme Almeida (2009), "a interação entre as partes do corpo que está sendo analisado se dá através das forças internas. Estas forças internas surgem entre todas as seções contíguas de um corpo submetido à ação de um sistema de forças externas".

Para efeito de demonstração, um corpo submetido a um sistema de forças externas em equilíbrio é ilustrado na figura 6. Ao seccionar imaginariamente o elemento em duas parcelas

pela seção S, vê-se a necessidade de introduzir um sistema de forças internas a fim de manter o equilíbrio das duas partes do corpo em análise, conforme apresentado na figura 6a. De acordo com o princípio da ação e reação essas forças possuem as mesmas direções, intensidades e ponto de aplicação, mas de sentidos opostos. A parte direita do elemento age sobre a parte esquerda e vice-versa, de tal maneira que as forças que aparecem em ambos os lados formam também um sistema de forças, internas, nesse caso, que estão em equilíbrio, ilustrado na figura 6b (ALMEIDA, 2009).

Figura 6 - Representação dos esforços internos de um elemento estrutural quando submetido a um sistema de forças



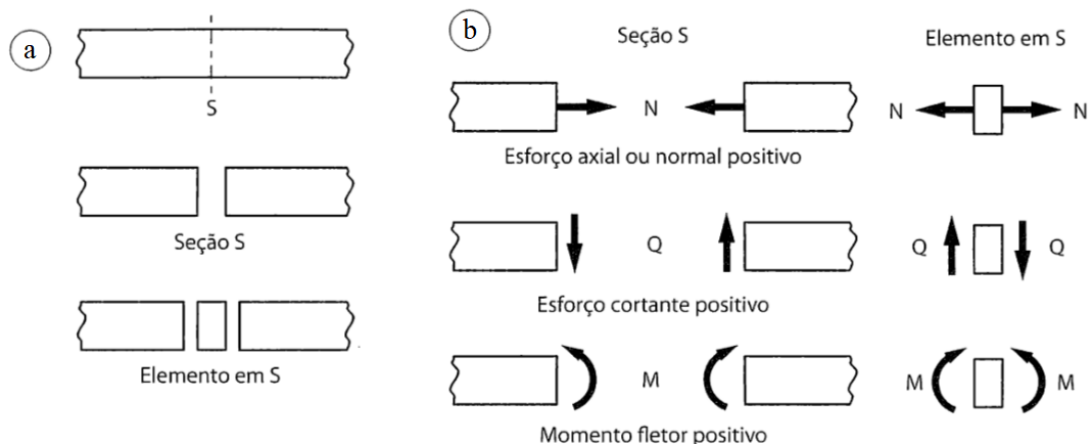
Fonte: Almeida (2009)

2.4.2.1 Esforços internos em estruturas planas

Uma estrutura é tida como plana quando ela e as forças que nela atuam pertencem a um mesmo plano. Desta forma a estrutura pode ser analisada segundo um modelo plano apresentando até três esforços solicitantes internos em qualquer seção ao longo de seu comprimento, são eles: esforço normal ou axial (N), esforço cortante (Q) e momento fletor (M) (ALMEIDA, 2009).

A figura 7a ilustra duas formas de representações dos esforços internos em uma seção S, uma através de um dos lados da seção, outra por meio do elemento infinitesimal contendo S. Já a 7b indica os sentidos positivos desses esforços.

Figura 7 - Representação dos esforços internos em estruturas planas



Fonte: Adaptado de Almeida (2009)

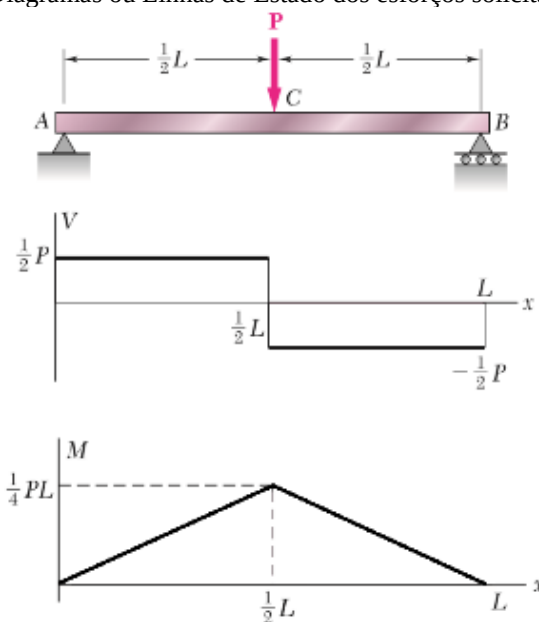
2.4.2.2 Diagramas dos esforços solicitantes internos

Também denominadas de linhas de estado, são representações gráficas dos esforços internos. Esses gráficos ou diagramas descrevem os valores dos esforços ao longo de toda estrutura. Assim, ao ser observado um diagrama, nota-se como os esforços variam de seção para seção na estrutura (GORFIN E OLIVEIRA, 1975).

Usualmente o diagrama dos esforços normais são identificados simplificadaamente por DEN, o diagrama dos esforços cortantes receber a identificação de DEC e o diagrama de momento fletor é denominado por DMF.

A figura 8 apresenta um exemplo dos diagramas dos esforços internos em uma viga plana biapoiada com carregamento pontual vertical.

Figura 8 - Diagramas ou Linhas de Estado dos esforços solicitantes internos



Fonte: Beer (2011)

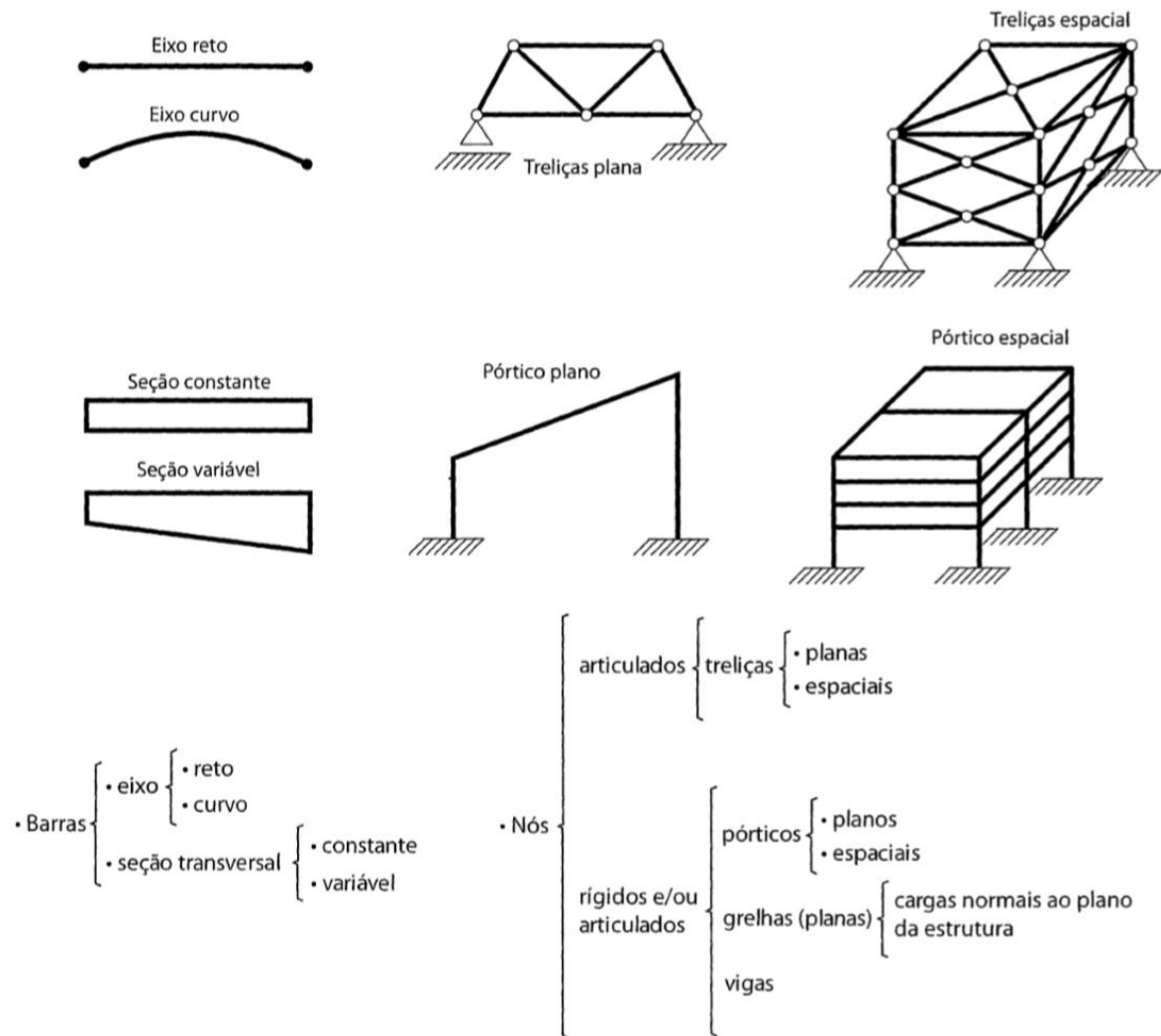
2.5 ESTRUTURAS RETICULARES

Como já mencionado, as estruturas unidimensionais ou reticulares, são formadas por elementos ou barras, caracterizando-se por apresentar uma de suas dimensões, o comprimento, muito maior que as outras duas (largura e altura).

Para efeito de cálculo, na elaboração dos modelos matemáticos para análises, tais estruturas são simplificadas geometricamente como constituídas por barras interconectadas por nós, isto é, a barra apresenta um nó inicial e um final, podendo ser de eixo reto ou curvo, de seção constante ou não (ALMEIDA, 2009).

Enquanto que os nós, Almeida (2009), divide-os em nós articulados, que são os que permitem a rotação relativa de elementos a eles conectados; e em nós rígidos, aqueles que não permitem rotação relativa. A figura 9 apresenta um resumo do que foi discutido acima e apresenta as estruturas que podem ser formadas a partir dessas configurações.

Figura 9 - Composição das estruturas reticulares



Fonte: Almeida (2009)

2.5.1 Cargas em estruturas reticulares

As cargas em uma estrutura, como já visto, podem ser representadas de maneiras reais ou aproximadas, classificadas, quanto ao tipo, em forças e momentos; e quanto à forma de aplicação em concentradas e distribuídas por unidade de comprimento e por unidade de área (ALMEIDA, 2010). O quadro 2 apresenta as possíveis representações dos carregamentos numa estrutura.

Quadro 2 - Cargas em modelos estruturais

Representação	Real		
	Aproximada		
A cargas podem ser	Forças	Concentradas	
		Distribuídas	Uniformes, triangulares, trapezoidais, outras
	Momentos	Concentradas	
		Distribuídas	

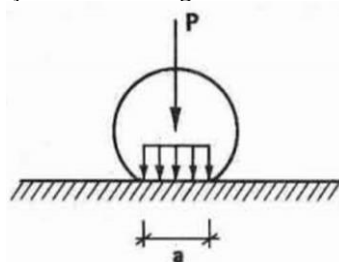
Fonte: Almeida (2009)

2.5.1.1 Cargas concentradas

Para Sussekind (1981), as cargas concentradas ou pontuais, são uma forma aproximada de tratar cargas distribuídas aplicadas em superfícies muito pequenas comparadas com as dimensões da estrutura, podendo ser consideradas nulas.

Para efeito de demonstração, a figura 10 esquematiza uma roda de um caminhão descarregando ao longo de uma ponte uma reação P sobre uma pequena área denotada por a . Nela, não haverá a aplicação de uma carga concentrada, mas sim de uma carga distribuída, entretanto segundo uma área tão pequena que se pode considerá-la nula em relação as dimensões da estrutura.

Figura 10 - Ação de um carregamento sobre uma estrutura



Fonte: Sussekind (1981)

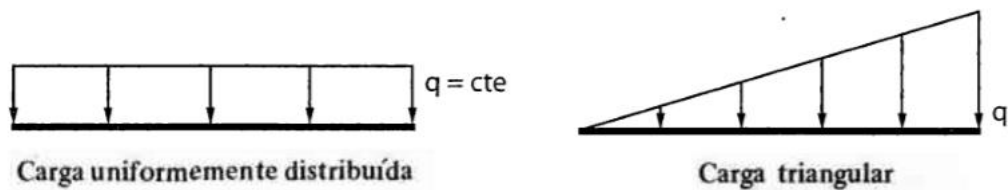
2.5.1.2 Cargas distribuídas

Assim como no caso anterior, as forças e momentos também podem ser de forma simplificada, considerados como distribuídos ao longo de um comprimento. Sendo que agora, uma das dimensões da área sobre a qual a força se transfere é pequena quando comparada com a outra dimensão. Nas estruturas reticulares as forças ou momentos distribuídos linearmente são considerados ao longo dos eixos dos elementos que as constituem (ALMEIDA, 2009).

Os tipos de carregamentos distribuídos que mais ocorrem na prática são as cargas uniformemente distribuídas ou constantes e as cargas triangulares, enquanto os que ocorrem com menor frequência são os carregamentos parabólicos e, em casos mais excepcionais, carregamentos de forma inteiramente aleatória (SUSSEKIND, 1981).

A seguir são indicados, para estruturas planas, os carregamentos distribuídos mais utilizados na prática.

Figura 11 - Representação dos tipos de carregamentos distribuídos mais comuns em estruturas reticulares



Fonte: Sussekind (1981)

2.5.2 Equilíbrio estático

De acordo com Sussekind (1981), para que um corpo, sob influência de um sistema de forças, permaneça em equilíbrio, é necessário que elas não provoquem nenhuma tendência de translação nem de rotação a este corpo, isto é, a resultante das forças e dos momentos dever ser nulos.

Em se tratando de estruturas, o que impedem seus deslocamentos quando submetidas a forças ativas (aplicadas à estrutura) são os apoios, que geram forças e momentos reativos, onde juntamente com as ações ativas formam um sistema de forças em equilíbrio (ALMEIDA, 2009).

Matematicamente, a partir desses conceitos, são fornecidas as equações do equilíbrio estático para uma estrutura espacial nas direções x, y e z, conforme ilustra a figura 12.

Figura 12 - Equações do equilíbrio estático

Equilíbrio de Forças	Equilíbrio de Momentos
$\sum F_x = 0$	$\sum M_x = 0$
$\sum F_y = 0$	$\sum M_y = 0$
$\sum F_z = 0$	$\sum M_z = 0$

Fonte: Almeida (2009)

2.5.3 Estabilidade e estaticidade de estruturas planas

Quanto a estabilidade das estruturas, Almeida (2009) classifica-as em estáveis e instáveis. As estruturas estáveis ocorrem quando o sistema de forças reativas estiver em equilíbrio com as forças ativas, mantendo-a estática, já as instáveis ocorrem quando as forças reativas forem em número insuficiente, isto é, não são capazes de manter a estrutura em equilíbrio e por esse motivo estão propícias a se moverem.

Segundo Soriano (2010), quanto a estaticidade, as estruturas de barras são classificadas em hipostáticas, isostáticas e hiperestáticas.

As estruturas hipostáticas são aquelas onde os vínculos internos ou externos são em número inferior ao necessário para impedir todos os movimentos possíveis ou de suas partes, sob ações externas, caracterizando-se dessa forma uma estrutura instável. Já quando os apoios são em número estritamente necessário para impedir todos os movimentos possíveis da estrutura, ou seja, o número de reações de apoio a determinar é igual ao número de equações de equilíbrio, é dito que a estrutura é isostática.

Quando os apoios são em número superior ao necessário para impedir todos os movimentos possíveis da estrutura, haverá uma menor quantidade de equações em relação as incógnitas, quanto à estaticidade, é classificada como hiperestática. Dessa forma, além das equações de equilíbrio, será necessário adicionar novas equações, a fim de tornar o problema possível.

Para este trabalho somente interessam as estruturas estáveis classificadas em isostáticas, uma vez que para serem analisadas basta aplicar as equações do equilíbrio estático.

2.6 PÓRTICOS ISOSTÁTICOS PLANOS

Pórtico plano é uma estrutura constituída de barras retas ou curvadas, situadas em um plano vertical, sob efeito de ações que o solicita nesse plano, de maneira que tenha apenas esforço normal, cortante e momento fletor (SORIANO, 2010)

Almeida (2009) define os pórticos planos como modelo composto por elementos cujos eixos apresentam orientações arbitrárias, pertencentes a um único plano, assim como as cargas que nela atuam. São conectados em suas extremidades por nós que podem ser rígidos ou articulados.

Ainda para Almeida (2009), na ligação por nós rígidos, ocorre a transmissão de momentos entre as barras. Os nós rígidos das estruturas deformadas apresentam rotação relativa

nula nos elementos conectados. Na estrutura indeformada, os ângulos entre os elementos, permanecem os mesmos após a aplicação do carregamento e a consequente deformação da estrutura. Já nos pontos com nós articulados, não ocorre a transmissão de momentos fletores entre as barras. Os nós articulados admitem a rotação relativa entre os elementos conectados, e o momento fletor é sempre nulo na rótula.

2.6.1 Sistema de referência

A fim de identificar e ordenar as ações mecânicas e os deslocamentos existentes numa estrutura, é fixado, inicialmente, um sistema de referência para a mesma (MOREIRA, 1977). Esse sistema de coordenadas pode ser caracterizado de duas formas: global ou local.

Para Almeida (2009), para as estruturas formadas por elementos que apresentam diversas orientações é necessário fazer distinção entre os eixos globais da estrutura completa e os eixos locais dos elementos presentes nela.

2.6.1.1 Eixos globais

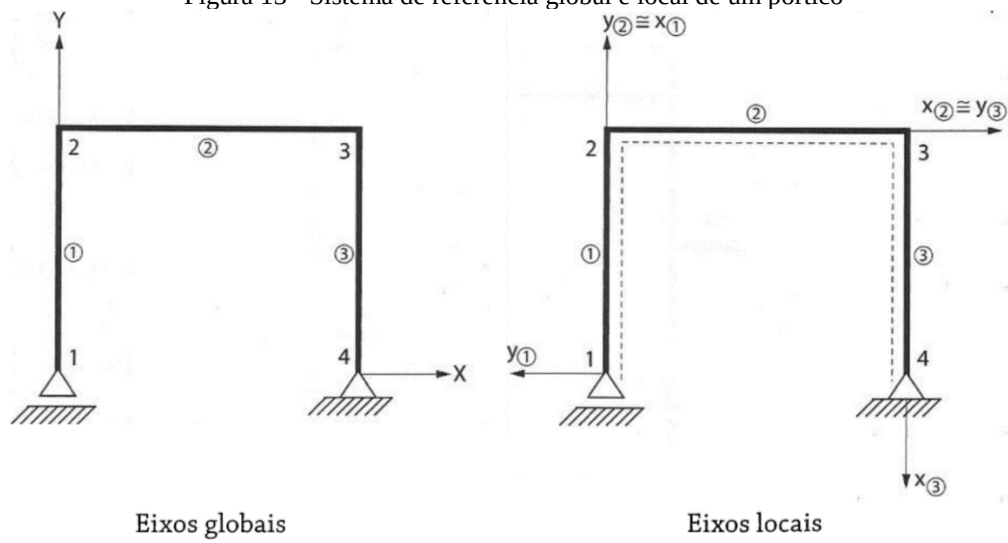
O sistema de referência global trata a estrutura como um todo. De acordo com Soriano (2005), a esse sistema de referência ficam vinculadas as grandezas relativas aos nós da estrutura, como coordenadas, deslocamentos, reações e cargas concentradas, isto é, para a determinação das reações de apoio em estruturas formadas por elementos com orientações diversas é necessário de um sistema de eixos globais.

2.6.1.2 Eixos locais

De acordo com Soriano (2005), se torna prático trabalhar com um sistema de referencia local para cada barra, de forma que o eixo x tenha origem no nó inicial da barra e seja dirigido para o nó final da mesma.

Almeida (2009) diz que para determinar os esforços solicitantes internos em uma estrutura, é necessário que seja definido para cada elemento que a compõe, um sistema de eixos locais. A figura 13 ilustra um exemplo de um pórtico plano no sistema de eixos global e local.

Figura 13 - Sistema de referência global e local de um pórtico

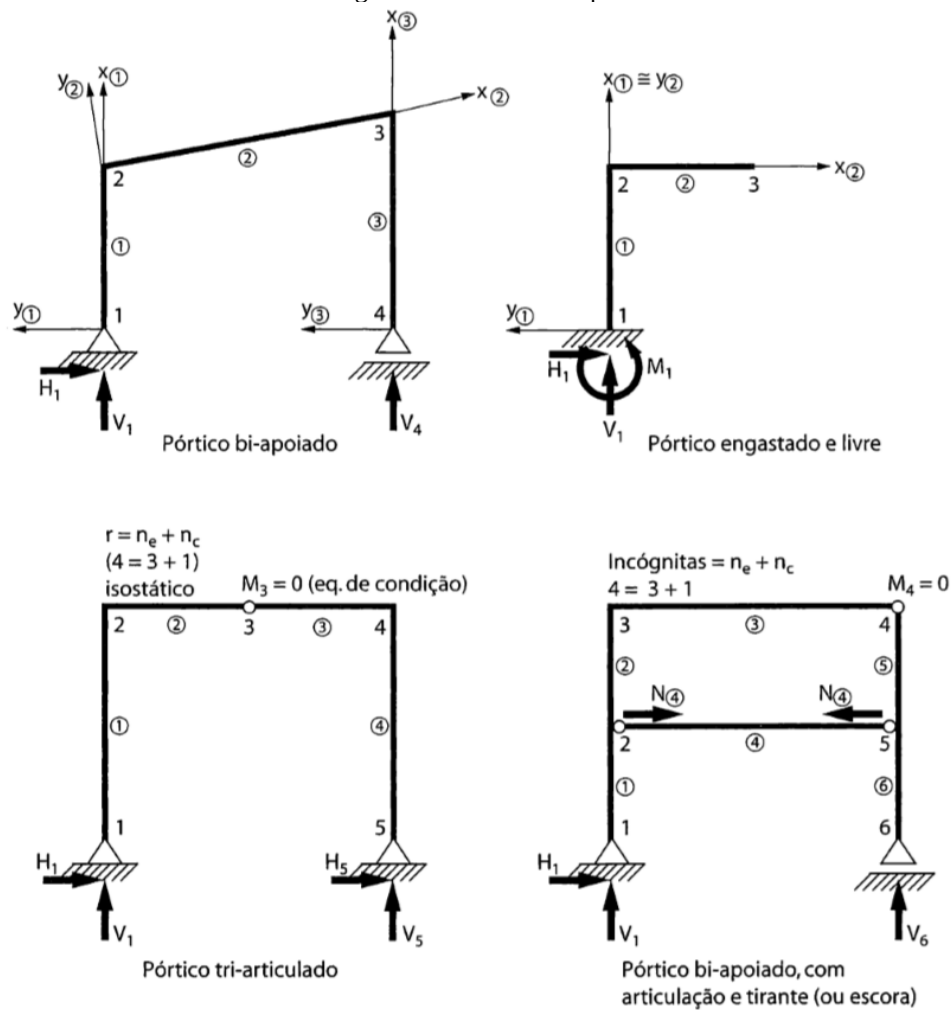


Fonte: Almeida (2009)

2.6.2 Pórticos simples

Sussekind (1981) classifica os pórticos em simples e compostos, aqueles considerados simples são divididos em quatro tipos fundamentais (figura 14), aos quais que quando associados entre si formam os pórticos compostos.

Figura 14 - Pórticos simples

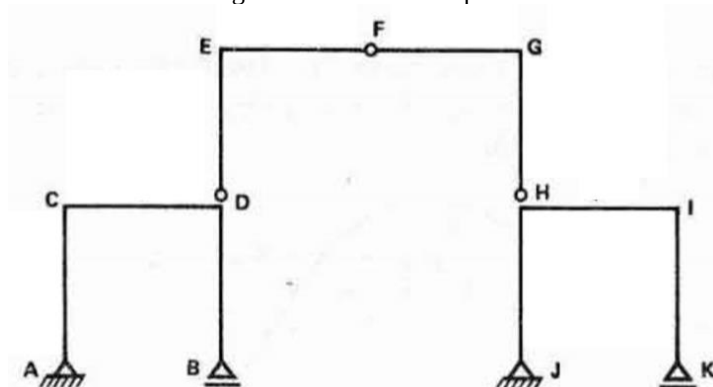


Fonte: Almeida (2009)

2.6.3 Pórticos compostos

Como mencionado, os pórticos compostos são formados pela associação de dois ou mais pórticos simples, as vigas simples também podem se associar a pórticos simples para formar pórticos compostos. A figura 15 ilustra alguns exemplos desses pórticos.

Figura 15 - Pórtico composto



Fonte: Sussekind (1981)

2.7 LINGUAGEM PYTHON

2.7.1 Apresentando o Python

Segundo Borges (2010), Python é uma linguagem de programação de alto nível, interpretada, imperativa, e orientada a objetos. Foi criada pelo holandês Guido Van Rossum em 1991, onde foi concebido a partir da linguagem existente na época, chamada ABC, e apesar de não ser tão recente, esta linguagem está em ascensão no mercado de tecnologia da informação, justamente por ser uma ferramenta robusta e capaz de realizar tarefas complexas, e ao mesmo tempo de fácil manipulação e aprendizado.

A PyScience-Brasil a define como uma linguagem de programação com o objetivo de buscar um código flexível e de fácil manutenção, sendo possível destacar como as principais características:

1. linguagem pseudo-código executável com o uso mínimo de caracteres especiais;
2. marcação de blocos por endentação;
3. uso de palavras-chave desnecessário nas compilações;
4. gerenciador de memória automático através do coletor de lixo;
5. suporta técnicas mais complexas como orientação a objetos;
6. é uma linguagem livre onde os desenvolvedores podem modificar seu código;
7. não possui um propósito específico como Java e PHP, podendo ser utilizada para atender diversas necessidades;
8. multiplataforma, funciona em vários sistemas e dispositivos.

Além das características acima, essa linguagem possui uma filosofia própria denominada Zen of Python, traduzida por Rocha (2018), os quais possuem frases para que o programador possa refletir acerca do desenvolvimento de suas aplicações que pretende realizar, a seguir são apresentadas alguma delas.

1. Bonito é melhor que feio;
2. Explícito é melhor que implícito;
3. Simples é melhor que complexo;
4. Complexo é melhor que complicado;
5. Linear é melhor do que aninhado;
6. Esparso é melhor que denso;
7. Legibilidade conta;
8. Casos especiais não são especiais o bastante para quebrar as regras;
9. Ainda que praticidade vença a pureza;
10. Erros nunca devem passar silenciosamente;
11. A menos que sejam explicitamente silenciados;
12. Diante da ambiguidade, recuse a tentação de adivinhar;
13. Agora é melhor que nunca;
14. Se a implementação é difícil de explicar, é uma má ideia.

Como dito anteriormente, sua filosofia enfoca na importância do esforço do programador sobre o esforço computacional e prioriza a legibilidade do código sobre a velocidade ou expressividade. Essa linguagem ainda apresenta a característica de ser executado em diversas plataformas, funcionando da mesma maneira em diferentes sistemas operacionais.

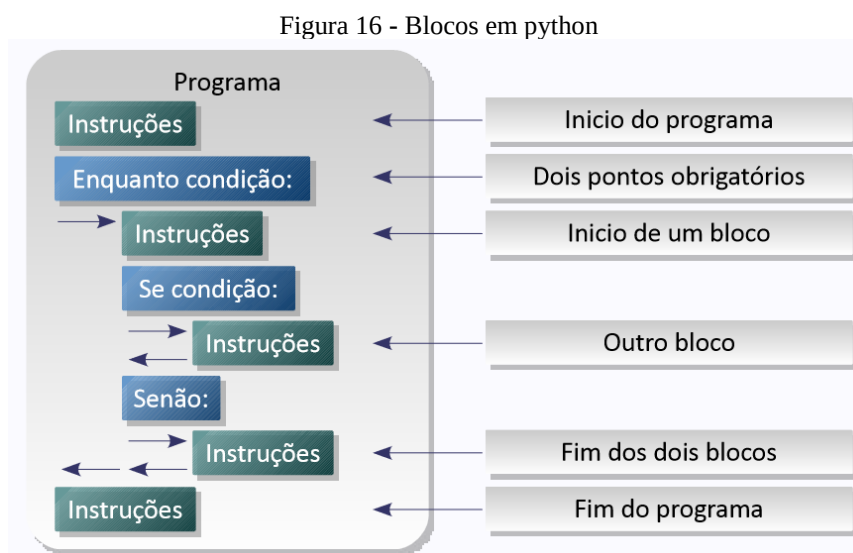
Outra vantagem é a disponibilidade de uma vasta biblioteca, que oferece recursos para a realização das principais tarefas comuns no desenvolvimento de software (DEV MEDIA, 2019).

2.7.2 Conceitos básicos

A sintaxe do Python é uma das suas mais fundamentais características. Um programa construído por essa linguagem é composto de linhas, que podem continuar nas linhas seguintes, pelo uso do caractere de barra invertida (\) ao final da linha ou parênteses (BORGES, 2010).

Os blocos de códigos em Python são um conjunto de algoritmos que devem ser executados um seguido do outro, de cima para baixo da esquerda para a direita; dessa forma, bloco de código é uma associação de comandos que se encontram numa mesma distância da margem esquerda, limitados por endentação. Segundo Ceder e Yergler (2003), endentar é o ato de inserir tabulações no código, de forma a hierarquizar seu fluxo de estruturação, laços e funções são exemplos de execuções comandadas pela endentação.

O uso desse artifício traz benefícios por organizar de forma rápida o código, tornando-o funcional e legível, sendo basicamente um padrão utilizado por programadores do mundo todo. A linha anterior ao bloco sempre é encerrada com dois pontos (:) o que irá representar uma nova estrutura de controle de linguagem ou uma declaração de uma nova estrutura, figura 16 (BORGES, 2010).



Fonte: Borges (2010)

Python possui uma biblioteca padrão imensa, que contém classes, métodos e funções para a realização de inúmeras tarefas. Essa característica da linguagem é comumente chamada baterias inclusas, significando que tudo de que você precisa para rodar um programa está na maior parte das vezes presente na instalação básica (BORGES, 2010).

2.7.3 Variáveis

Uma variável é uma denominação que se refere a um conteúdo. Os nomes de variáveis devem começar com letra sem acentuação ou sublinhado (_) e seguido por letras sem

acentuação, dígitos ou por sublinhados (_), sendo que letras maiúsculas e minúsculas são tidas como diferentes. Uma das características das variáveis, dentro da linguagem de programação é a possibilidade de elas serem manipuladas de acordo com a necessidade. Python possui vários tipos de dados, sendo os mais comuns os números inteiros, números de ponto flutuante e strings. Além dos tipos básicos, existem as que funcionam como coleções, isto é, podem armazenar números e letras, cujas principais são: listas, tuplas e o dicionário (BORGES, 2010).

2.7.3.1 Variáveis string

Variáveis do tipo string são aquelas que armazenam uma sequência de símbolos como nomes e textos em geral. Este tipo de variável é muito útil para exibir mensagens ou mesmo para gerar outros arquivos (MENEZES, 2014).

Para referenciar uma variável como do tipo string, basta coloca-la entre aspas, como demonstrado na figura 17.

Figura 17 - Exemplo de variáveis do tipo strings

```
>>> a = "Olá mundo!"
>>> print (a)
```

Fonte: Autoria Própria (2019)

2.7.3.2 Variáveis numéricas

É dito variável numérica aquela que armazena números inteiros ou de ponto flutuante. Python possibilita uma infinidade de operações matemáticas com esse tipo de variável, desde operações básicas, como somar e subtrair, até as mais complexas.

Quadro 3 - Operações com números

Operador	Operação
+	soma
-	subtração
*	multiplicação
/	divisão
%	resto da divisão
**	potenciação

Fonte: Menezes (2014)

2.7.3.3 Variáveis do tipo lógico

Também conhecidas como booleano, são variáveis que armazenam um conteúdo simples: verdadeiro ou falso.

- Operadores relacionais

Para realizar comparações lógicas, utiliza-se operadores relacionais. A lista de operadores relacionais suportados em Python é apresentada no quadro 4.

Quadro 4 - Operadores relacionais

Operador	Operação	Símbolo matemático
==	igualdade	=
>	maior que	>
<	menor que	<
!=	diferente	≠
>=	maior ou igual	≥
<=	menor ou igual	≤

Fonte: Menezes (2014)

- Operadores lógicos

Para agrupar operações com lógica booleana, utiliza-se operadores lógicos, essa linguagem suporta três operadores básicos.

Quadro 5 - Operadores lógicos

Operador	Operação
not	não
and	e
or	ou

Fonte: Menezes (2014)

2.7.3.4 Lista

São um tipo de variável que permite o armazenamento de vários valores entre colchetes, acessados por um índice. Uma lista pode conter nenhum ou mais de um elemento de um mesmo tipo ou de tipos diferentes, podendo conter outras listas (Menezes, 2014).

Segundo Borges (2010), as listas são mutáveis, podendo ser alteradas a qualquer instante. Além disso, podem ser fatiadas da mesma forma que as strings, mas como as listas são mutáveis, é possível fazer atribuições a itens da lista.

Figura 18 - Representação de uma lista

```
lista = [a, b, ..., z]
```

Fonte: Borges (2010)

2.7.3.5 Tupla

Tuplas são variáveis que podem ser vistas como listas, porém são imutáveis: não se pode acrescentar, apagar ou fazer atribuições aos itens inicialmente construídos (BORGES, 2010). Um tupla é representada segundo ilustrado na figura 19, sendo os parênteses opcionais.

Figura 19 - Representação de uma lista

```
tupla = (a, b, ..., z)
```

Fonte: Borges (2010)

2.7.3.6 Dicionário

Dicionário é uma variável com estrutura de dados semelhante às listas, mas com propriedades de acesso diferentes. O dicionário em si consiste em relacionar uma chave a um valor específico. A chave precisa ser de um tipo imutável, de forma geral são usadas strings. Já os itens (valores) dos dicionários podem ser tanto mutáveis quanto imutáveis (BORGES, 2010). Um dicionário é criado a partir de ({}), como pode ser percebido na figura 20.

Figura 20 - Representação de uma lista

```
dicionario = {'a': a, 'b': b, ..., 'z': z}
```

Fonte: Borges (2010)

2.7.4 Orientação a objetos

Este paradigma de programação baseia-se na uniformização e comunicação dos elementos da estrutura global do programa, isto é, todo elemento é um objeto com a capacidade de se comunicar com outros objetos. Assim a POO surge como uma nova maneira de programar, tentando simular o mundo real (LEITE; RAHAL, 2006).

Segundo Chagas e Oliveira (2009), na POO, os objetos passam a simular a realidade através de uma abstração do cotidiano, onde a ideia básica é representar as características relevantes dos objetos envolvidos no sistema computacional em desenvolvimento. Essa simulação da realidade, ocorre de forma diferente da Programação Orientada a Procedimentos (POP), pois foi incluída a concepção de visibilidade, que pode ser definida como o nível de acesso de propriedades e métodos de uma classe, esse conceito restringe o acesso aos métodos, impedindo o acesso direto aos atributos, tornando possível controlar as operações.

Tais aspectos tornaram a Programação Orientada a Objetos mais eficaz do que a Programação Orientada a Procedimentos. A POO ainda traz outros benefícios que fazem dela um dos paradigmas mais utilizados, como: reuso, modularização, simplificação e redução dos custos de manutenção (CHAGAS; OLIVEIRA, 2009).

No quadro 6, são apresentados diversos conceitos da POO, designados por seus idealizadores, afim de tornar esse paradigma um importante instrumento de elaboração de programas.

Além dos conceitos elementares já vistos, alguns autores consideram que a Orientação a Objetos é baseada em três fundamentos. São eles: encapsulamento, herança e o Polimorfismo.

Quadro 6 - Conceitos essenciais da Programação Orientada a Objetos

Classe	Representa um conjunto de objetos com características comuns. Uma classe define o comportamento dos objetos, através de métodos, e quais estados ele é capaz de manter, através de atributos. Exemplo de classe: Pórtico.
Objeto	É uma instância de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos. Exemplo de objetos da classe Pórtico: Material, Seção.
Atributos	São as características de um objeto. Exemplos: nome, altura etc. Por sua vez, os atributos possuem valores. Por exemplo, o atributo tipo da seção pode conter o valor triangular.
Estado	É o conjunto de valores dos atributos de um determinado objeto.
Métodos	Definem as habilidades dos objetos. Material é uma instância da classe Pórtico, portanto tem habilidade para adicionar nome do material, implementada através do método adicionar_nome(). Em uma classe, um método é apenas uma definição. A ação só ocorre quando o método é invocado através do objeto, no caso Ana. Dentro do programa, a utilização de um método deve afetar apenas um objeto em particular; Todos os seres humanos podem caminhar, mas você quer que apenas Ana caminhe. Normalmente, uma classe possui vários métodos, que no caso da classe pórtico poderiam ser adicionar_material(), atribuir_seção().
Mensagem	É uma chamada a um objeto para invocar um de seus métodos, ativando um comportamento descrito por sua classe. Também pode ser direcionada diretamente a uma classe (através de uma invocação a um método estático). Em síntese, é a forma que os objetos utilizam para se comunicar.
Abstração	É a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de software.
Interface	É um contrato entre a classe e o mundo externo. Quando uma classe implementa uma interface, ela está comprometida a fornecer o comportamento publicado pela interface.

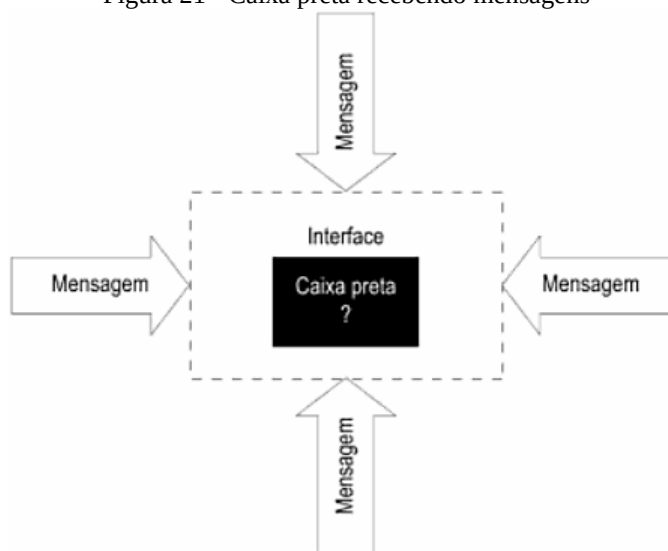
Fonte: Chagas e Oliveira (2009)

2.7.4.1 Encapsulamento

Encapsulamento é uma das principais características que define a programação orientada a objetos, que por sua vez se baseia em agrupar em um determinado lugar, códigos que aparecem em conjunto dentro de um programa, a esse grupo é atribuído um nome, e a partir desse atributo sempre que for necessário a utilização do grupo de códigos, basta recorrer ao nome do atributo (GOETTEN; WINCK, 2006).

Para Chagas e Oliveira (2009), através do encapsulamento, é possível utilizar entidades de software como se fossem caixas-pretas (figura 21). Em outras palavras, o solicitante, espera obter uma determinada resposta e tendo apenas uma visão superficial da operação, envia mensagens a uma caixa preta sem necessariamente se atentar aos detalhes da implementação.

Figura 21 - Caixa preta recebendo mensagens

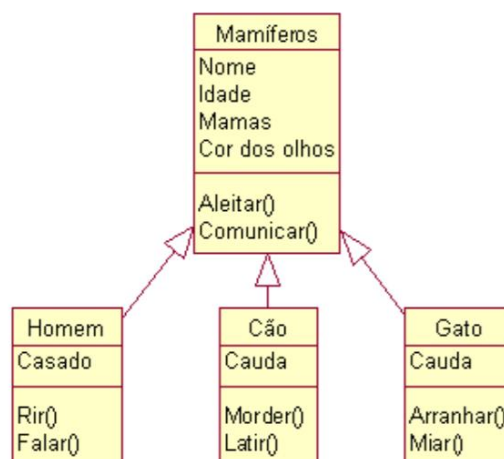


Fonte: Goetten e Winck (2006)

2.7.4.2 Herança

Herança é um mecanismo que permite altos graus de reutilização de código. Segundo Leite e Rahal (2006), tal método pode ser entendido como um conjunto de instâncias criadas a partir de um outro conjunto de instâncias que possuem características semelhantes, e os elementos desse subconjunto herdam todas as características do conjunto original. Assim esse mecanismo se torna uma técnica muito eficiente para construir, organizar e reutilizar códigos.

Figura 22 - Esquema didático de hierarquia das classes



Fonte: Leite e Rahal (2006)

A figura 22 exemplifica de maneira didática a utilização dessa técnica, podendo destacar a semelhança entre a forma de organização do mundo real com a POO, onde é possível perceber que a classe dos Mamíferos possui as características de nome, idade, mamas, cor dos olhos. Essas características são atributos desta classe, da mesma forma que cauda é um atributo da classe Cão. Observa-se também que todo mamífero pode se aleitar, ou comunicar-se, que são caracterizados como métodos desta classe. Assim, nota-se que as classes homem, cão e gato herdam características da classe Mamíferos.

2.7.4.3 Polimorfismo

No universo da POO, segundo Leite e Rahal (2006), o polimorfismo é definido como sendo um código que incorpora ou que produz vários comportamentos, isto é, um código que pode ser aplicado à várias classes de objetos. Um método polimórfico é aquele que pode ser aplicado à várias classes de objetos sem que haja qualquer inconveniente, em outras palavras, o polimorfismo é a qualidade ou estado de um objeto ser capaz de assumir diferentes formas.

2.7.5 Arquivos – Geração HTML

Um arquivo é uma área em disco onde se pode ler e gravar informações. Páginas web não passam de combinações entre textos e imagens, interligadas por links. Como o formato HTML é definido apenas com textos simples, ou seja, sem caracteres especiais de controle, ele utiliza marcações, que são sequências especiais de texto delimitadas pelos caracteres de menor (<) e maior (>). Essas sequências são chamadas de tags e podem iniciar ou finalizar um

elemento. O elemento de mais alto nível de um documento HTML é chamado de <html>. Escreve-se páginas web entre as tags <html> e </html>, onde a primeira marca o início do documento; e a segunda, delimita seu final (MENEZES, 2014).

Figura 23 - Exemplo de código para geração de arquivo html

```
>>> <|DOCTYPE html>
>>> <html lang = "pt-BR">
>>> <head>
>>> <meta charset = "utf-8">
>>> <title>Título da Página</title>
>>> </head>
>>> <body>
>>> Olá!
>>> </body>
```

Fonte: Menezes (2014)

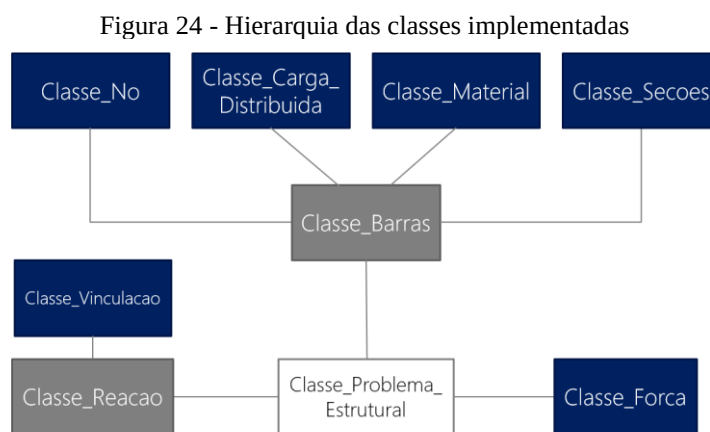
3 METODOLOGIA

Neste tópico, será abordada a metodologia utilizada para desenvolvimento do trabalho em questão. Para a sua confecção, foram realizados estudos na área da mecânica das estruturas, afim de analisar o comportamento de estruturas quando sujeitas a esforços externos; e no campo da programação para a implementação computacional dos componentes do programa.

Em seguida, utilizando a linguagem Python e algumas das suas bibliotecas (Numpy, Matplotlib, Scipy) realizou-se a implementação de elementos em duas dimensões para análise estrutural de pórticos isostáticos. Tais implementações foram realizadas no ambiente de desenvolvimento Eclipse, que é uma ferramenta com objetivo de facilitar a implementação e a depuração de códigos computacionais.

3.1 CLASSES IMPLEMENTADAS

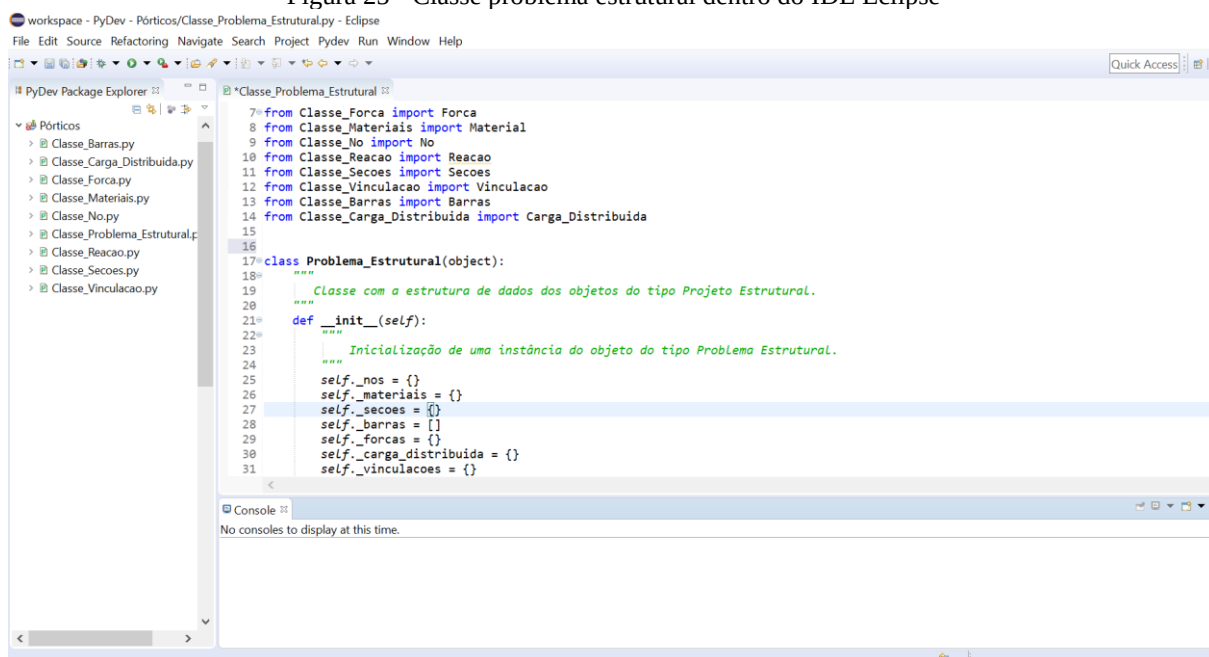
A implementação foi particionada em nove classes (figura 24). A classe problema estrutural recebe funcionalidades de todas as outras e a partir da criação de um módulo “exemplo” é possível gerar e apresentar um problema prático.



Fonte: Autoria Própria (2019)

A figura 25 ilustra o IDE Eclipse com a implementação da classe problema estrutural, onde é possível verificar todas as outras classes sendo importadas para dentro dela.

Figura 25 - Classe problema estrutural dentro do IDE Eclipse



Fonte: Autoria Própria (2019)

3.2 DETERMINAÇÃO DO PROBLEMA ESTRUTURAL

Com a obtenção de todas as considerações necessárias para análise de uma estrutura, o programa realizará uma leitura dos dados de entrada. Assim, o usuário deverá fornecer ao software todas as informações necessárias sobre o referente problema. Esta sequência será apresentada a seguir:

- Quantidade e posição de nós;
- Conectividades dos nós com as barras;
- Adição de vinculações, indicando o nó e a direção da restrição;
- Cargas atuantes, indicando o nó e intensidade;
- Área da seção transversal e módulo de elasticidade de cada barra.

A partir das informações fornecidas, o programa gerará um arquivo no formato html com os resultados obtidos no programa (equação dos esforços internos e seus respectivos gráficos). A figura 26, apresenta os dados de entradas descritos pelo usuário.

Figura 26 - Dados de entradas

```

1 from Classe_Problema_Estrutural import Problema_Estrutural
2
3 portico = Problema_Estrutural()
4
5 portico.adicionar_no(id='A', x=0, y=0.0)
6 portico.adicionar_no(id='B', x=0, y=3.0)
7 portico.adicionar_no(id='C', x=4, y=3.0)
8 portico.adicionar_no(id='D', x=8, y=3.0)
9
10 portico.adicionar_material(id = 'Mat01', young = 1.0, poison = 0.1, alfa = 1.0, peso_esp = 1.0, tensao_adm = 1.0)
11
12 portico.adicionar_secao(id = 'Sec01', area = 1.0, inercia = 1.0)
13
14 portico.adicionar_barra('A', 'C', 'Mat01', 'Sec01')
15 portico.adicionar_barra('B', 'C', 'Mat01', 'Sec01')
16 portico.adicionar_barra('C', 'D', 'Mat01', 'Sec01')
17
18 portico.adicionar_forca(id = 'P1', fx = -5.0, fy = 0.0, m = 0.0)
19 portico.tribuir_forca(id_no = 'D', id_forca = 'P1')
20
21 portico.adicionar_carga_distribuida(id = 'Q1', fxi = 0.0, fxf = 0.0, fyi = -10.0, fyf = -10.0)
22 portico.tribuir_carga_distribuida(id_barra = 1, id_carga_distribuida = 'Q1')
23
24 portico.adicionar_vinculacao(id = 'Vic01', rest_x = 1, rest_y = 1, rest_r = 0)
25 portico.adicionar_vinculacao(id = 'Vic02', rest_x = 0, rest_y = 1, rest_r = 0)
26 portico.tribuir_vinculacao(id_no = 'A', id_vinculacao = 'Vic01')
27 portico.tribuir_vinculacao(id_no = 'D', id_vinculacao = 'Vic02')
28

```

Fonte: Autoria Própria (2019)

3.3 RESOLUÇÃO DO PROBLEMA ESTRUTURAL

Com os dados de entrada fornecidos, o programa inicia sua análise a partir do cálculo das reações de apoio, para tanto, representa-se o problema estrutural em um diagrama de corpo livre deixando explícitas todas as cargas ativas e reativas que nela atuam. A Figura 27 apresenta grande parte da função pela qual é gerado o DCL do problema.

Figura 27 - Código da função geradora do DCL

```

*Classe_Problema_Estrutural
288 def desenhar_dcl(self):
289     x_max = self.barras[len(self.barras)-1].no_f.x + 1
290     fig, ax = plt.subplots()
291     fig.set_size_inches(13.5, 10)
292     vmax = 0.0
293
294     for barra in self.barras:
295
296         if barra.carga_distribuida != None:
297             vaux = barra.carga_distribuida.fyi
298
299             if abs(vmax) < abs(vaux):
300                 vmax = vaux
301
302             vaux = barra.carga_distribuida.fyf
303             if abs(vmax) < abs(vaux):
304                 vmax = vaux
305
306         barra.desenhar_barra(ax)
307     for id, no in self.nos.items():
308         no.desenhar(ax, id)
309
310     for id, no in self.nos.items():
311         no.desenhar_forca(ax)
312
313     for barra in self.barras:
314         barra.desenhar_carga_distribuida_vertical(ax, vmax)
315
316     for barra in self.barras:
317         barra.desenhar_carga_distribuida_horizontal(ax)
318
319     for id, no in self.nos.items():

```

Fonte: Autoria Própria (2019)

Após a representação do DCL, dentro da função “desenhar_calcular_reacoes” presente na classe problema estrutural, foram calculadas as reações de apoio a partir das equações de equilíbrio estático no plano ($\Sigma F_x = 0$, $\Sigma F_y = 0$ e $\Sigma M_o = 0$), tomando o ponto inicial “A” como o escolhido para a realização do cálculo do momento. A figura 28 expõe trechos do código escrito.

Figura 28 - Código da função calcular reações

```
*Classe_Problema_Estrutural
330
331 def desenhar_calcular_reacoes(self):
332     fs = 0.5
333     sfh = ''
334     sfv = ''
335     sm = ''
336     simbolos = []
337
338     for id, no in self.nos.items():
339         if no.vinculacao:
340             if no.vinculacao.rest_x:
341                 sfh = sfh + str('+H%s' % id)
342                 simbolos.append('H%s' % id)
343
344             if no.vinculacao.rest_y:
345                 sfv = sfv + str('+V%s' % id)
346                 dist = no.x - self.nos['A'].x
347                 sm = sm + str('+V%s * %.2f' % (id, dist))
348                 simbolos.append('V%s' % id)
349
350             if no.vinculacao.rest_r:
351                 sm = sm + str('-M%s' % id)
352                 simbolos.append('M%s' % id)
353
354     for id, no in self.nos.items():
355         if no.forca:
356             if no.forca.fx > 0:
357                 dist = no.y - self.nos['A'].y
358                 sfh = sfh + str('+%.2f' % no.forca.fx)
359                 if dist > 0:
360                     sm = sm + str('-(%.2f * %.2f)' % (no.forca.fx, dist))
361
```

Fonte: Autoria Própria (2019)

Tendo conhecimento dos valores das reações de apoio o passo seguinte foi o traçado de um novo DCL, dessa vez, seccionando cada elemento do pórtico e os representando por suas coordenadas locais conforme esses elementos mudam de direção. No método das seções o eixo em estudo é dividido a medida em que ocorrer a mudança no carregamento imposto. Dessa forma, por meio da função “desenhar seção”, o programa mostrará ao usuário uma representação de seu elemento com as devidas representações dos eixos locais e dos trechos que deverão ser seccionados para a resolução de suas equações de esforços internos.

Posterior aos seccionamentos realizados ao longo do pórtico, o próximo passo realizado pelo programa é expor as equações dos esforços internos presentes em cada trecho analisado. Como já apresentado na seção 2, as equações que podem estar presentes em um problema estrutural, são: equações para forças normais, que atuam paralelamente ao eixo horizontal da barra, equações de esforços cortantes, onde se tem todos os carregamentos e reações impostos de forma perpendicular ao elemento e as equações para o momento fletor.

De maneira semelhante ao cálculo das reações, foram utilizadas funções condicionantes, além de bibliotecas gráficas e de resolução matemática para implantação dessa etapa do relatório a ser gerado.

A última etapa para a resolução do problema estrutural se deu pela representação gráfica das funções dos esforços internos ao longo da barra, dessa forma, o programa apresentará ao usuário os desenhos dos diagramas de esforço normal, esforço cortante e momento fletor.

Com o fim do processamento do programa, toda saída dos resultados é convertida em um único arquivo html para visualização do relatório por completo, de modo, que ao ser aberto pelo usuário seja possível observar todos os passos do relatório desenvolvido. Na figura 29 pode-se visualizar o código responsável por gerar o arquivo do relatório.

Figura 29 - Código gerador do arquivo HTML

```

2558
2559 def gerar_relatorio(self):
2560     ntrx = len(self.barras)
2561     texto = '<p><strong>PROBLEMA ESTRUCTURAL:</strong></p>'
2562     texto = texto + '<p><strong></strong></p>'
2563     texto = texto + '<p><strong>DCL:</strong></p>'
2564     texto = texto + '<p><strong></strong></p>'
2565     texto = texto + '<p><strong>C&Aacute;LCULO DAS REA&Ccedil;&Otilde;ES:</strong></p>'
2566     texto = texto + '<p><strong></strong></p>'
2567     texto = texto + '<p><strong>FUN&Ccedil;&Otilde;ES DOS ESFOR&Ccedil;OS INTERNOS:</strong></p>'
2568     for i in range(len(self.barras)):
2569         texto = texto + '<p><strong>* TRECHO 0%d: %.2f m &nbsp;&nbsp;&Le;&nbsp;&nbsp;&nbsp; X &nbsp;&nbsp;&Le;&nbsp;&nbsp;&nbsp; %.2f m</strong></p>' \
2570             % (i+1, self.barras[i].no_i.x, self.barras[i].no_f.x)
2571         texto = texto + '<p><strong></strong></p>' % (i+4)
2572         texto = texto + '<p><strong></strong></p>' % (i+ntrx+4)
2573         texto = texto + '<p><strong>DIAGRAMAS DE ESFOR&Ccedil;OS SOLICITANTES:</strong></p>'
2574         texto = texto + '<p><strong></strong></p>' % (5+2*ntrx)
2575
2576         texto = texto + '<p><strong></strong></p>' % (6+2*ntrx)
2577
2578         texto = texto + '<p><strong></strong></p>' % (7+2*ntrx)
2579         arq = open('PÓRTICO.html', 'w')
2580         arq.write(texto)
2581         arq.close()

```

Fonte: Autoria Própria (2019)

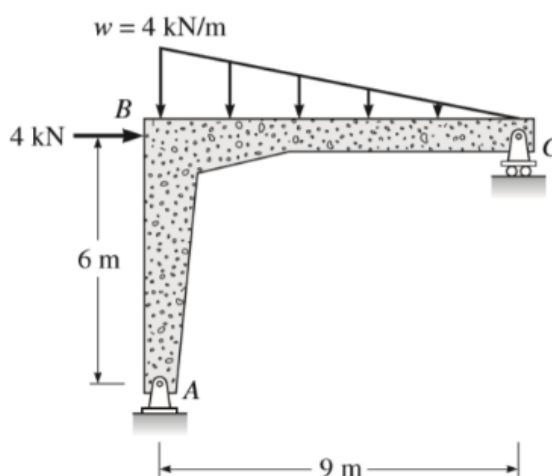
4 RESULTADOS

Para realizar a validação do programa desenvolvido neste trabalho, foram confeccionados três exemplos. Tanto o primeiro quanto o segundo são problemas existentes em conteúdo acadêmico, mais precisamente no livro Fundamentos da análise estrutural de Leet e Kenneth, enquanto o terceiro é uma simulação baseada em parâmetros de uma estrutura real.

4.1 EXEMPLO 01 - PÓRTICO BIAPOIADO COM CARREGAMENTO TRIANGULAR

O primeiro exemplo é o exercício proposto 10, presente no capítulo 5 do livro citado anteriormente. Esse exemplo apresenta um carregamento triangular distribuído de cima para baixo sobre o elemento horizontal da estrutura e uma carga horizontal da direita para esquerda com intensidade de 4kN, as dimensões se encontram na abaixo.

Figura 30 - Pórtico biapoiado com carregamento triangular

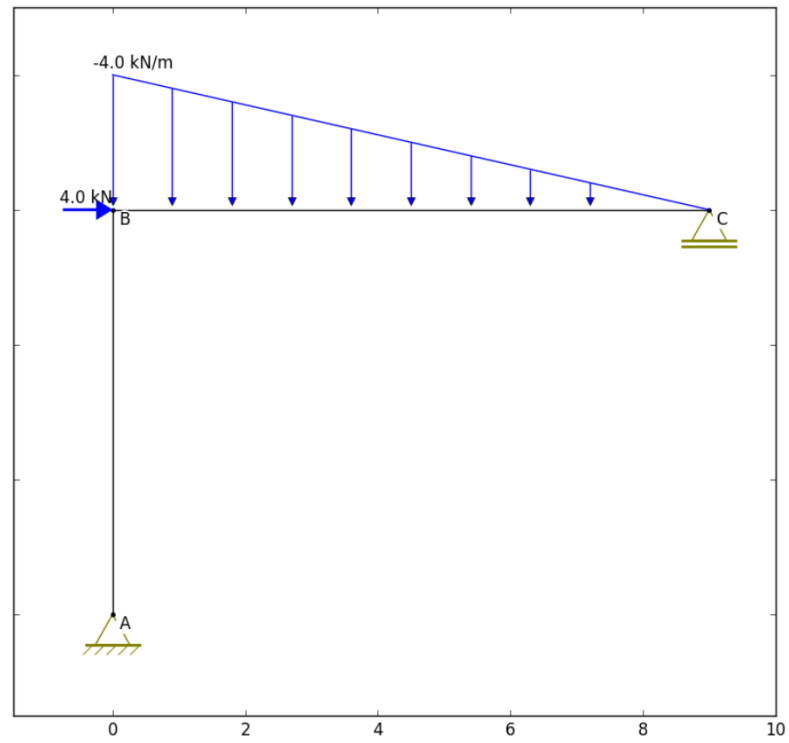


Fonte: Leet e Kenneth (2009)

A primeira etapa para a resolução do problema apresentado é a implantação dos dados de entrada que o problema oferece. Com isso, é feita a representação do problema estrutural para o relatório (ver figura 31).

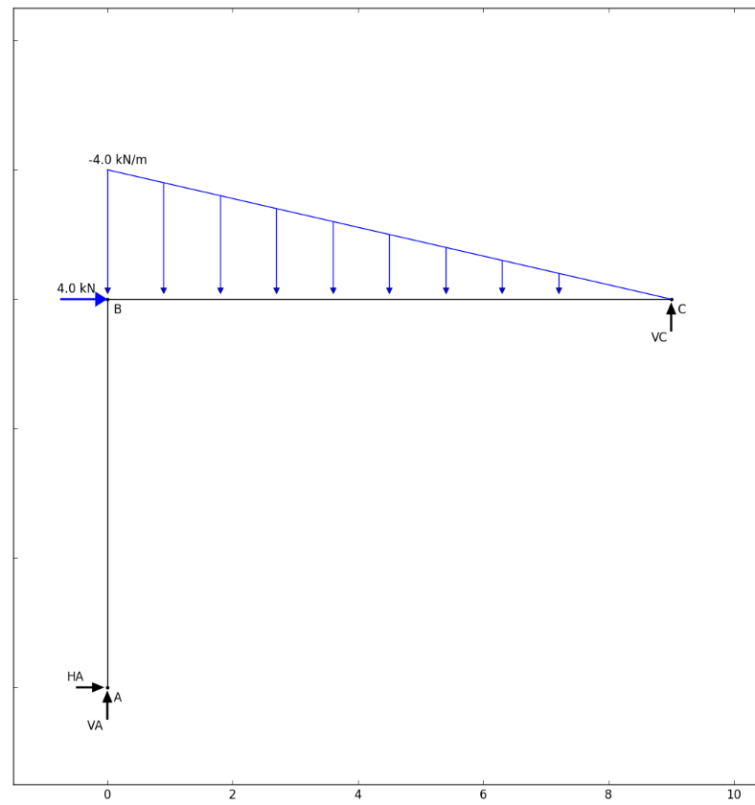
Em seguida é gerado o DCL do problema, exibindo todas as forças ativas e reativas atuantes no pórtico. Tendo representado o DCL (figura 32), a próxima tarefa para resolução do exercício foi a determinação dos valores de cada reação aplicando-se as equações de equilíbrio estático (ver figura 33).

Figura 31 - Problema estrutural - exemplo 01



Fonte: Autoria Própria (2019)

Figura 32 - DCL - exemplo 01



Fonte: Autoria Própria (2019)

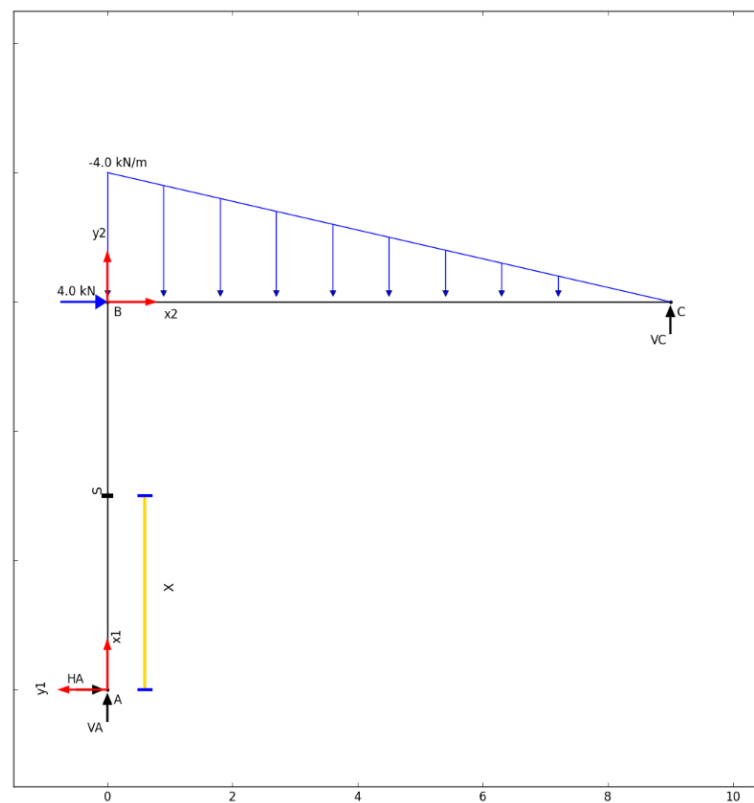
Figura 33 - Cálculo das reações de apoio - exemplo 01

$$\begin{aligned}
 \sum F_H &= 0 \\
 +H_A + 4.00 &= 0 \\
 \sum F_V &= 0 \\
 +V_A + V_C - 18.00 &= 0 \\
 \sum M_A &= 0 \\
 +V_A * 0.00 + V_C * 9.00 - (4.00 * 6.00) - (18.00 * 3.00) &= 0 \\
 V_A &= 9.33 \\
 V_C &= 8.67 \\
 H_A &= -4.00
 \end{aligned}$$

Fonte: Autoria Própria (2019)

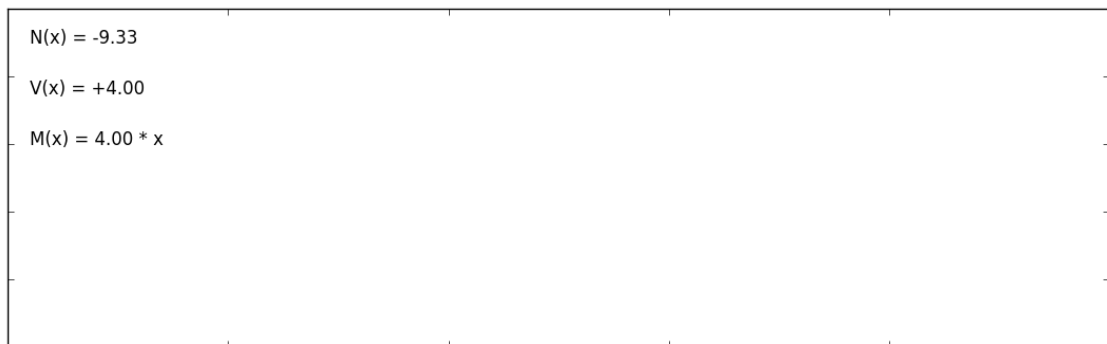
Posteriormente, ocorreu a análise local de cada elemento, aplicando-se o método das seções (figura 34) e por fim, foram obtidas as equações dos esforços normal, cortante e fletor, para as determinadas seções (ver figura 35).

Figura 34 - Seccionamento local da barra 1 - exemplo 01



Fonte: Autoria Própria (2019)

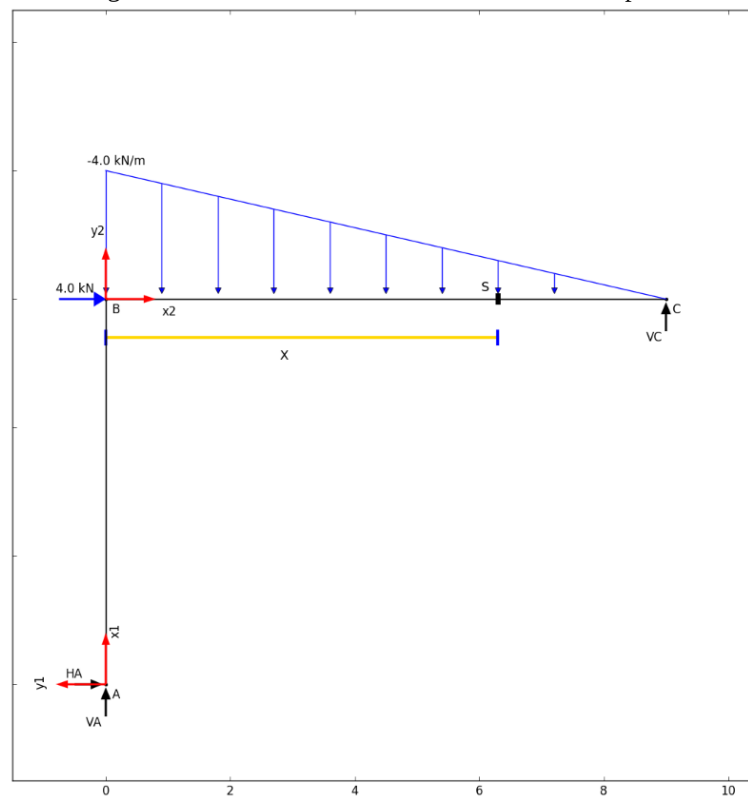
Figura 35 - Função dos esforços internos da barra 1 - exemplo 01



Fonte: Autoria Própria (2019)

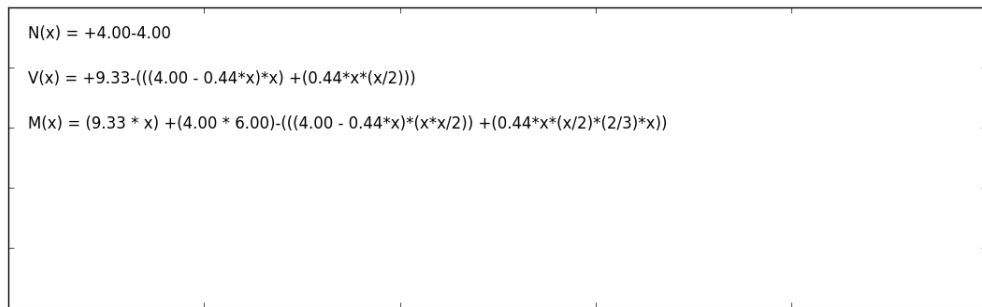
De forma semelhante, o segundo elemento foi analisado, observar na figura 36. Na sequência é apresentada suas respectivas funções dos esforços internos ao longo do trecho da barra (ver figura 37).

Figura 36 - Seccionamento local da barra 2 - exemplo 01



Fonte: Autoria Própria (2019)

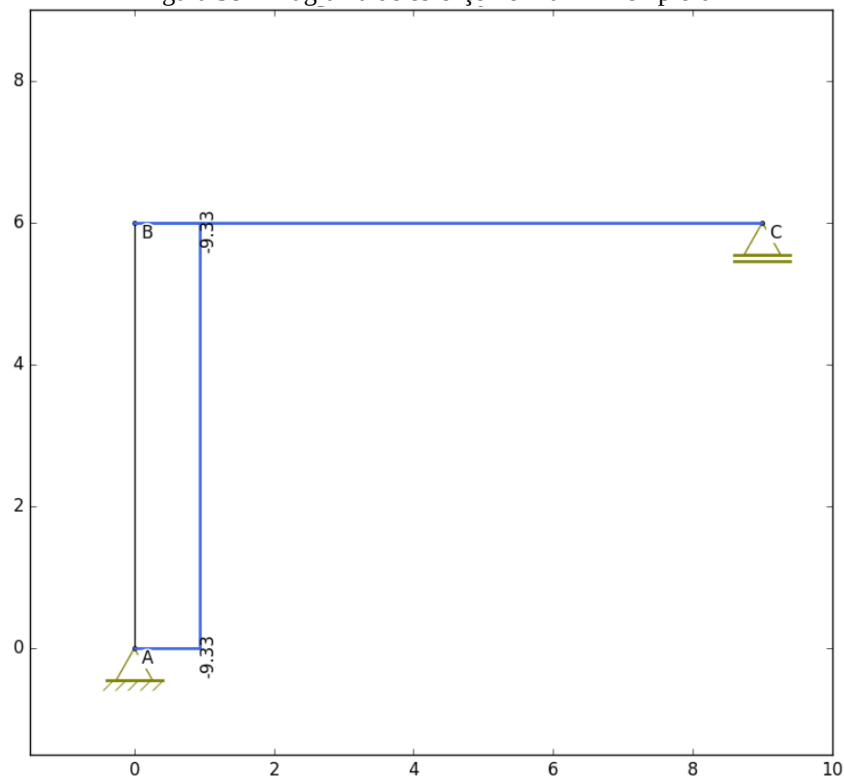
Figura 37 - Função dos esforços internos da barra 1 - exemplo 01



Fonte: Autoria Própria (2019)

Após apresentação dos resultados em forma de equações, os mesmos são plotados em gráficos, os quais serão comparados como os obtidos pelo Ftool. O primeiro resultado é o diagrama de esforço normal, representado na figura 38.

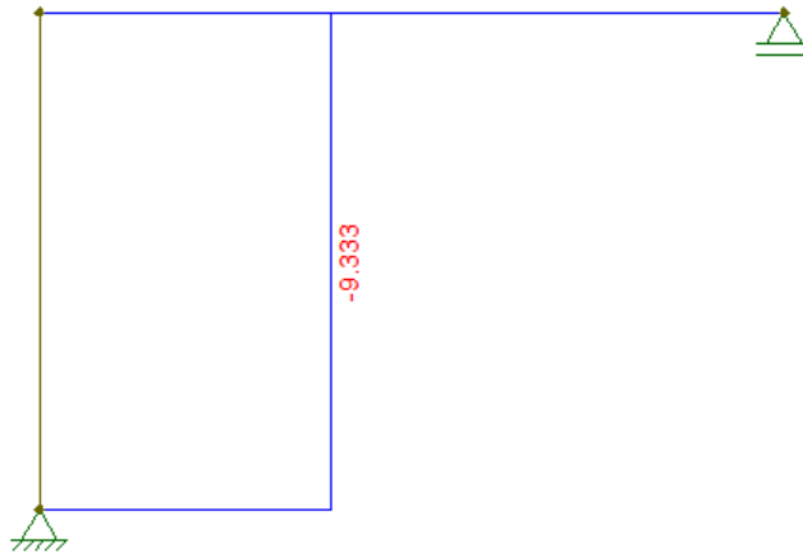
Figura 38 - Diagrama de esforço normal – Exemplo 01



Fonte: Autoria Própria (2019)

Observa-se que o traçado do diagrama feito pelo software é semelhante ao processado pelo Ftool (figura 39), havendo uma diferenciação apenas a escala de projeção.

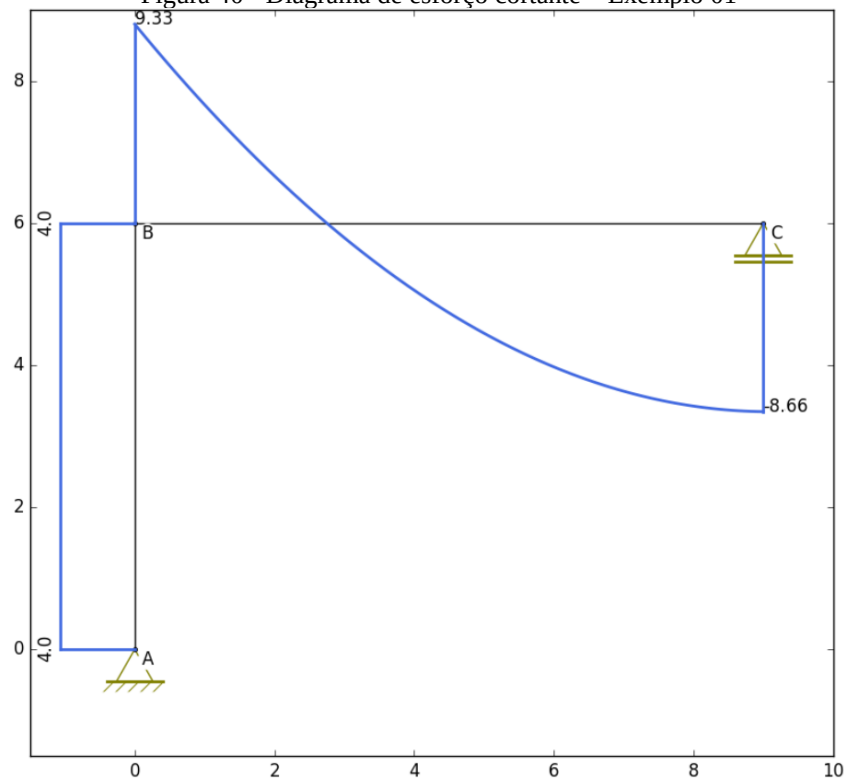
Figura 39 - Diagrama de esforço normal/ FTOOL - Exemplo 01



Fonte: Ftool (2019)

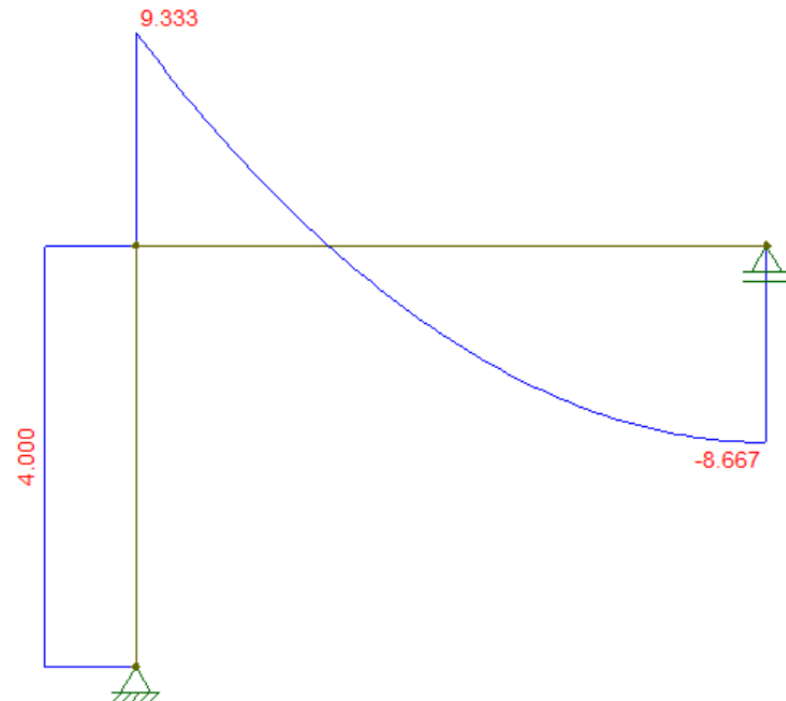
De forma idêntica, se comparou os resultados dos diagramas de esforço cortante e momento fletor. Ambos comparados com os do Ftool, e o seus resultados saíram como o esperado, como ilustra as figuras na sequência.

Figura 40 - Diagrama de esforço cortante – Exemplo 01



Fonte: Autoria Própria (2019)

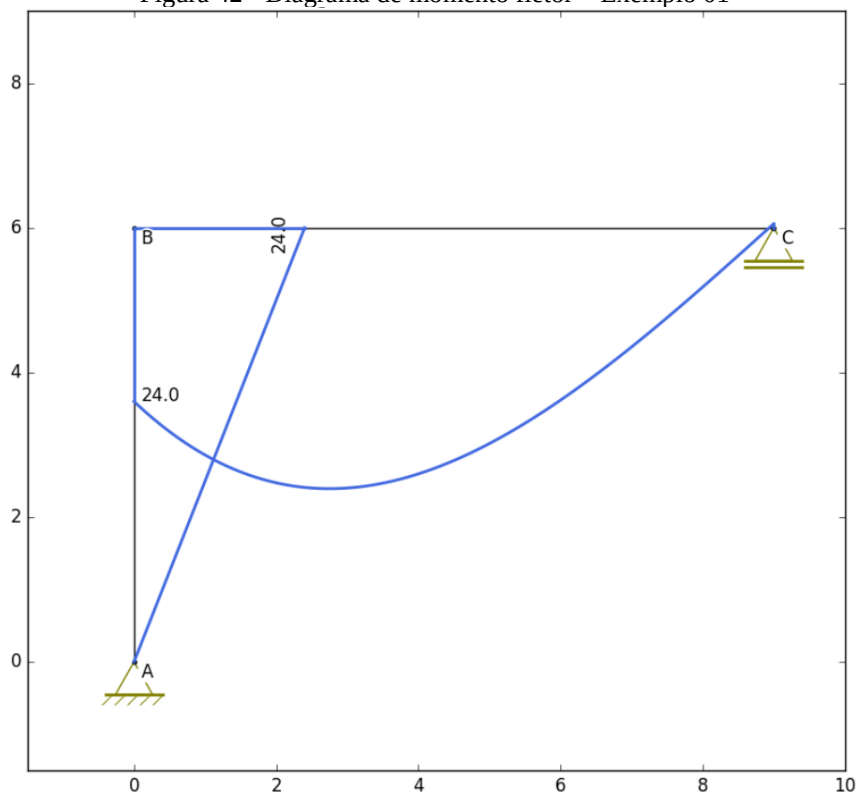
Figura 41 - Diagrama de esforço cortante/ FTOOL - Exemplo 01



Fonte: Ftool (2019)

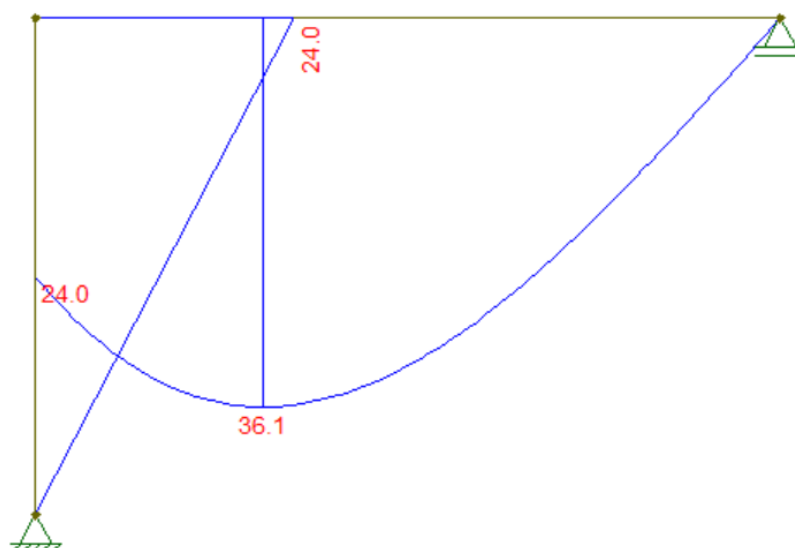
A seguir são demonstrados os diagramas de momento fletor.

Figura 42 - Diagrama de momento fletor – Exemplo 01



Fonte: Autoria Própria (2019)

Figura 43 - Diagrama de momento fletor/ FTOOL - Exemplo 01

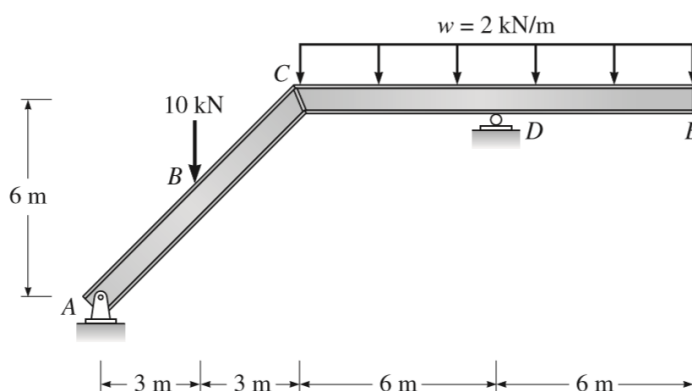


Fonte: Ftool (2019)

4.2 EXEMPLO 02 – PÓRTICO BIAPOIADO COM CARREGAMENTO UNIFORME

Esse exemplo, assim como o anterior é voltado ao âmbito estudantil e pertence ao exercício proposto 19, também presente no capítulo 5 do livro de fundamentos da análise estrutural. Nele, é possível visualizar, a partir da figura 44, que apresenta um carregamento uniformemente distribuído de cima para baixo com intensidade de 2 kN/m sobre o elemento horizontal da estrutura e uma carga vertical aplicada também de cima para baixo com intensidade de 10 kN sobre um elemento inclinado.

Figura 44 - Pórtico biapoado com elemento inclinado e carregamento uniforme

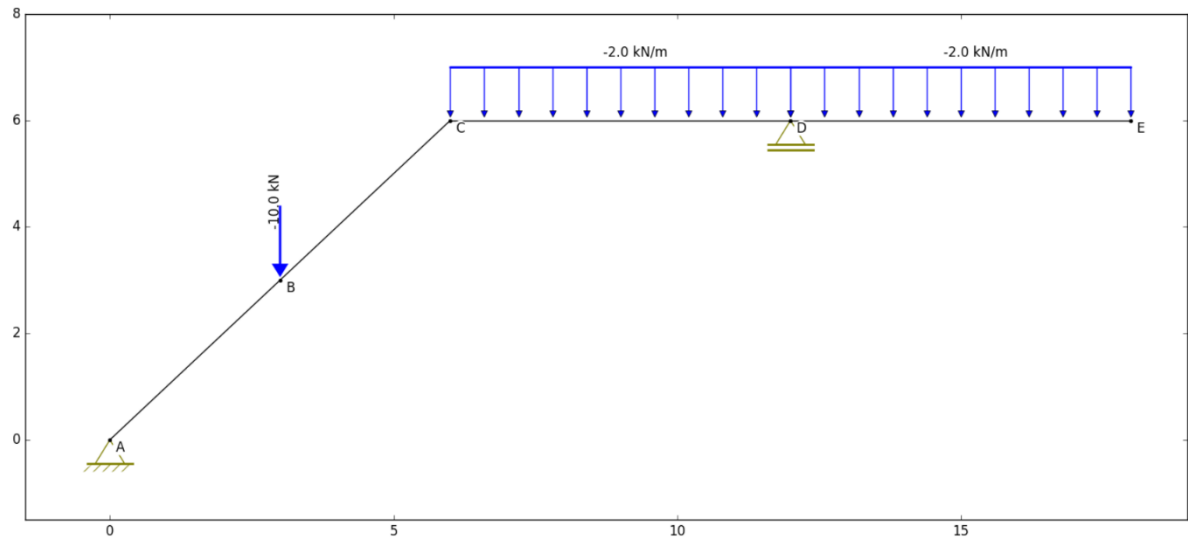


Fonte: Leet e Kenneth (2009)

Realizada a adição dos dados de entrada, a estrutura é processada e calculada e ao fim são gerados novamente os diagramas dos esforços internos de forma semelhante ao processo utilizado para o exemplo 01, onde os diagramas são comparados com o software Ftool.

Assim, por meio dos dados fornecidos, foi feita a plotagem do problema estrutural. (ver figura 45).

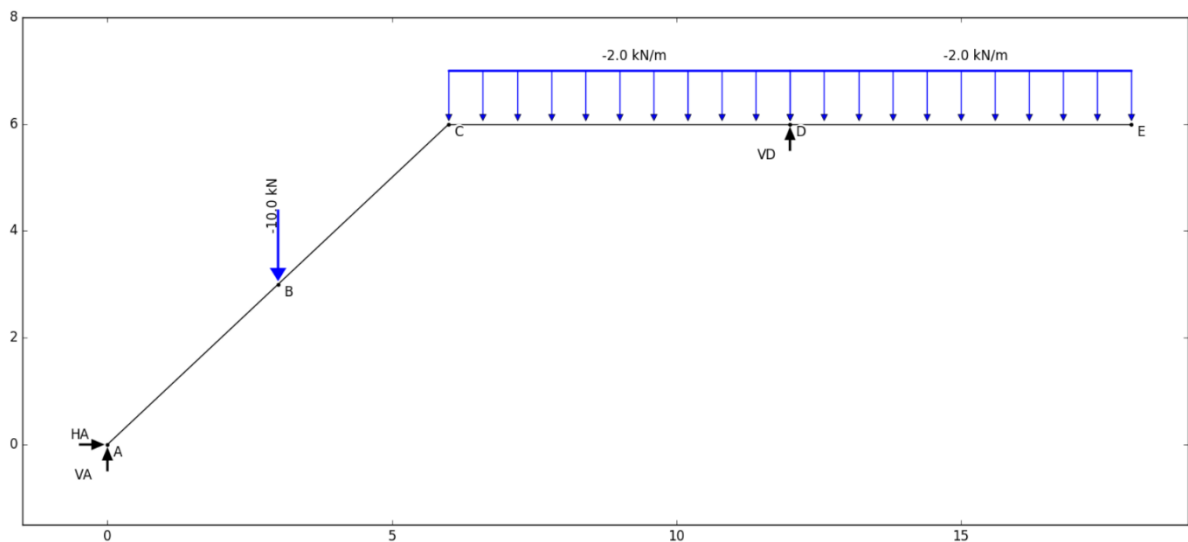
Figura 45 - Problema estrutural - exemplo 02



Fonte: Autoria Própria (2019)

Tomando-se conhecimento do problema, na sequência foi desenvolvido o diagrama de corpo livre, onde todas as forças foram discriminadas (ver figura 46). A partir do DCL, foram calculados os valores para as reações de apoio atuantes no elemento, assim como mostra a figura 47.

Figura 46 - DCL - exemplo 02



Fonte: Autoria Própria (2019)

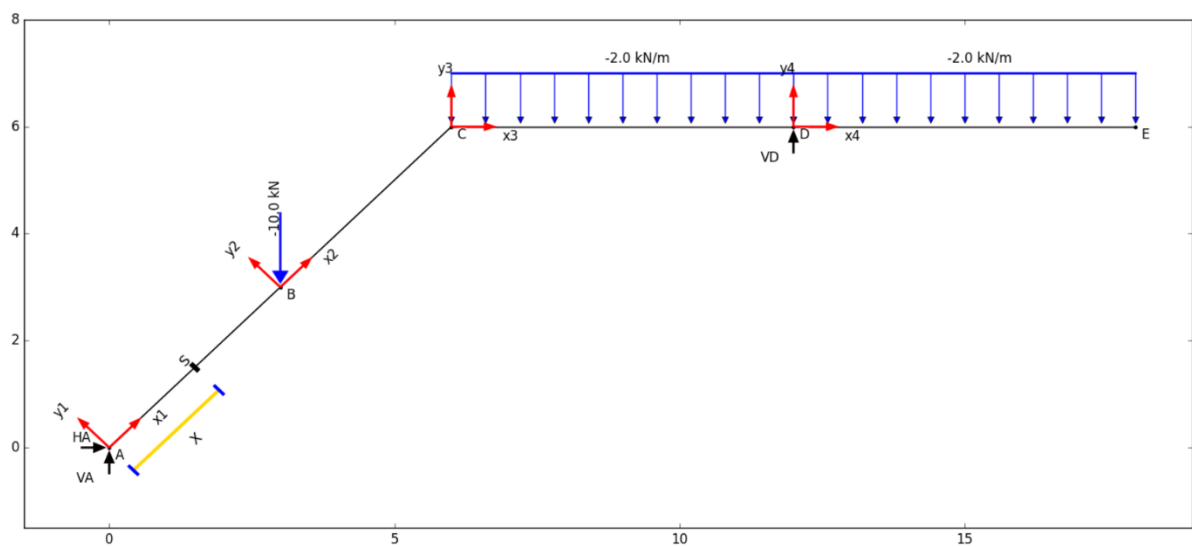
Figura 47 - Cálculo das reações de apoio - exemplo 02

$$\begin{aligned}
 \sum F_H &= 0 \\
 +H_A &= 0 \\
 \sum F_V &= 0 \\
 +V_A + V_D + (-10.00 - 12.00 - 12.00) &= 0 \\
 \sum M_A &= 0 \\
 +V_A * 0.00 + V_D * 12.00 + (-10.00 * 3.00) - (12.00 * 9.00) - (12.00 * 15.00) &= 0 \\
 V_A &= 7.50 \\
 V_D &= 26.50 \\
 H_A &= 0.00
 \end{aligned}$$

Fonte: Autoria Própria (2019)

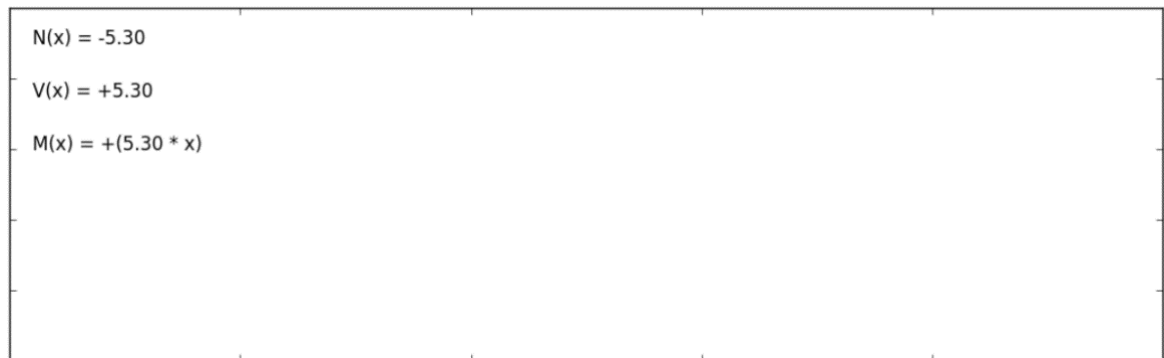
Encontradas as reações, o programa realiza o seccionamento dos elementos a partir de suas coordenadas locais e faz a análise de seus esforços internos, afim de se descobrir suas equações para cada porção da estrutura. Nas figuras 48 e 49, podem ser encontradas o primeiro trecho seccionado, assim como, suas equações.

Figura 48 - Seccionamento local da barra 1 - exemplo 02



Fonte: Autoria Própria (2019)

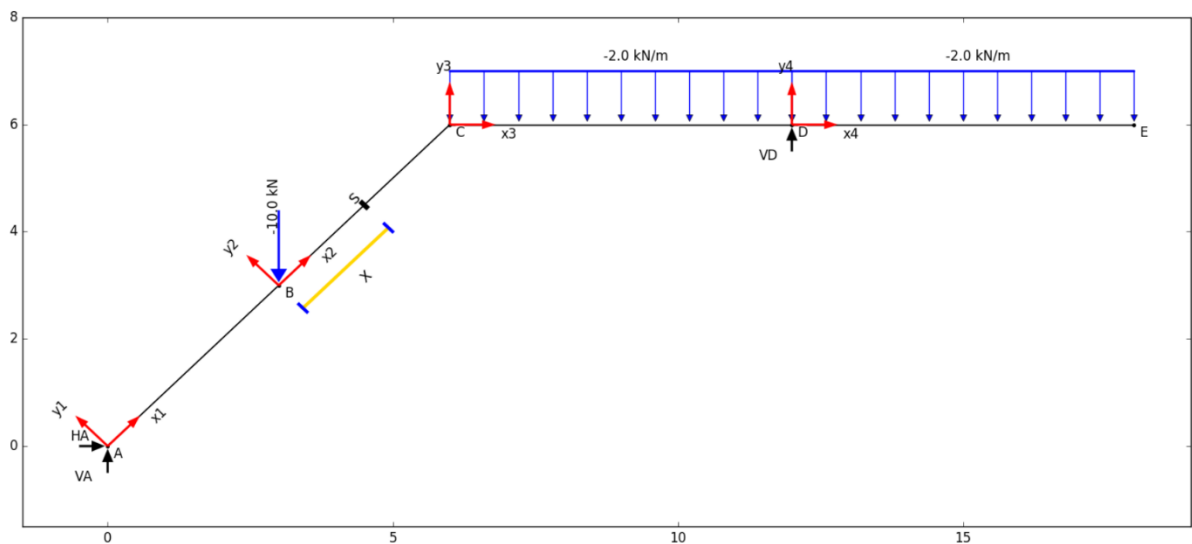
Figura 49 - Função dos esforços internos da barra 1 - exemplo 02



Fonte: Autoria Própria (2019)

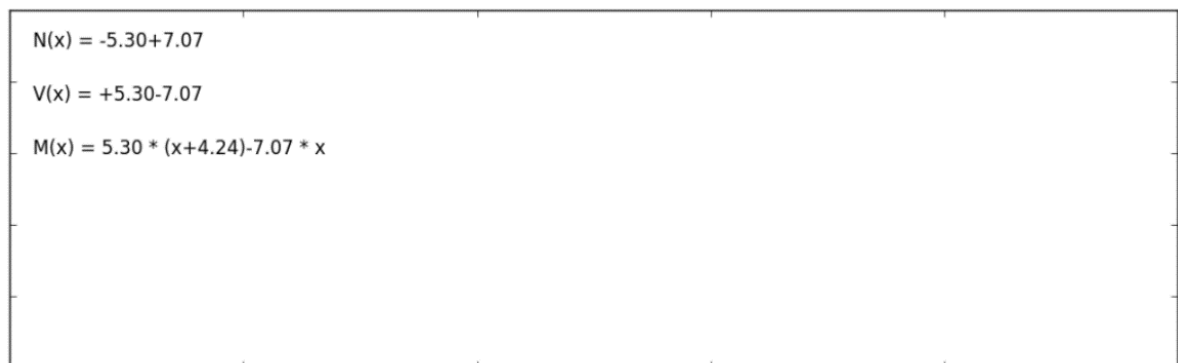
A seguir são apresentadas as próximas barras seccionadas e seus respectivos cálculo para determinação dos esforços.

Figura 50 - Seccionamento local da barra 2 - exemplo 02



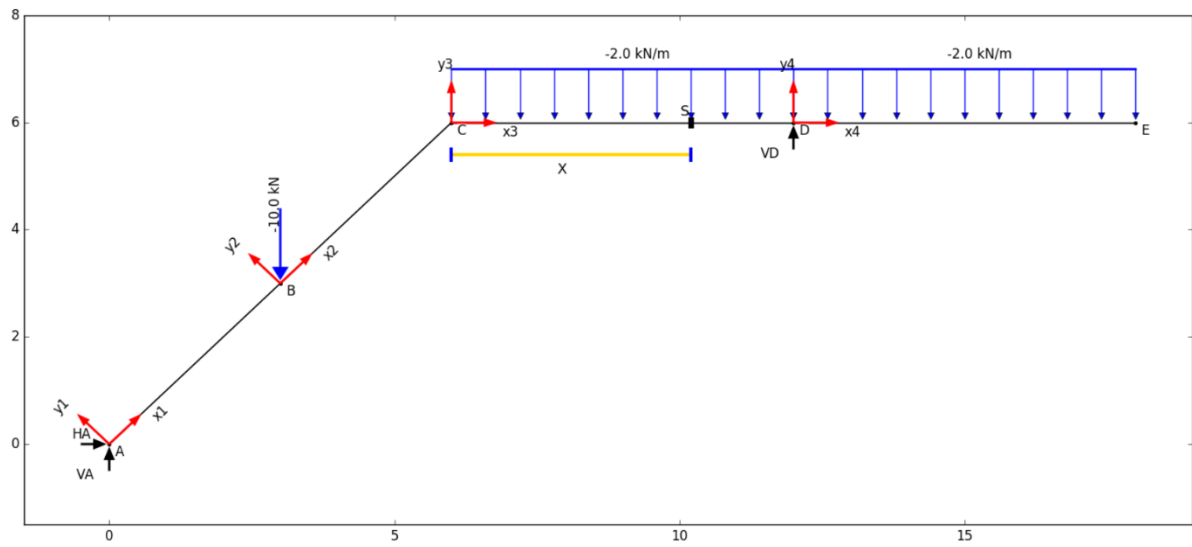
Fonte: Autoria Própria (2019)

Figura 51 - Função dos esforços internos da barra 2 - exemplo 02



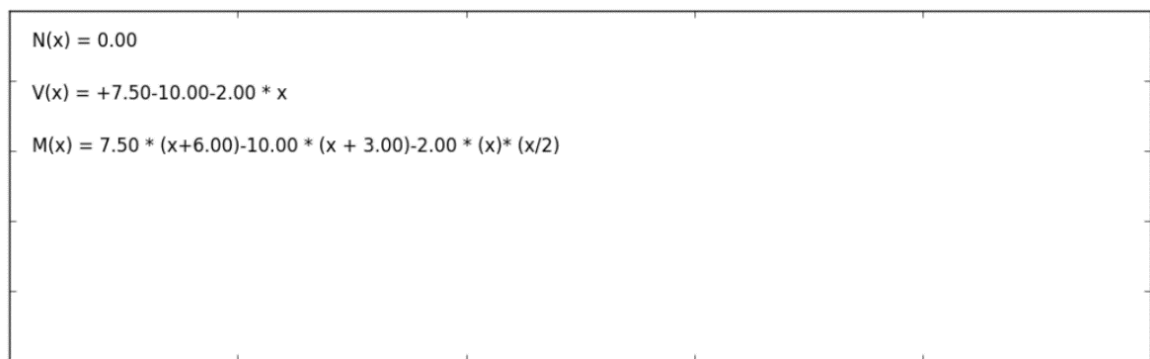
Fonte: Autoria Própria (2019)

Figura 52 - Seccionamento local da barra 3 - exemplo 02



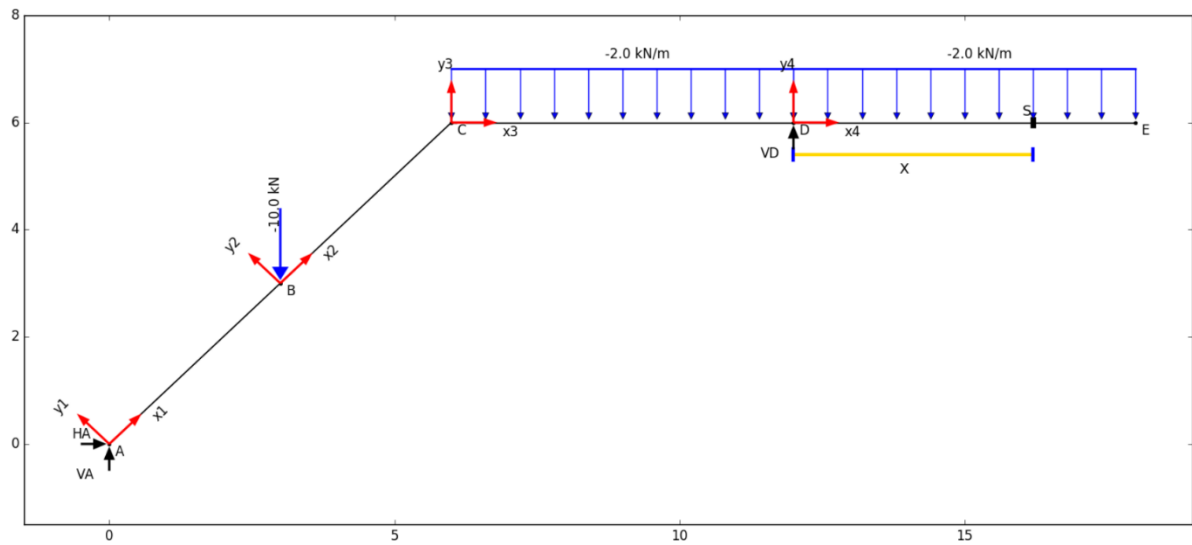
Fonte: Autoria Própria (2019)

Figura 53 - Função dos esforços internos da barra 3 - exemplo 02



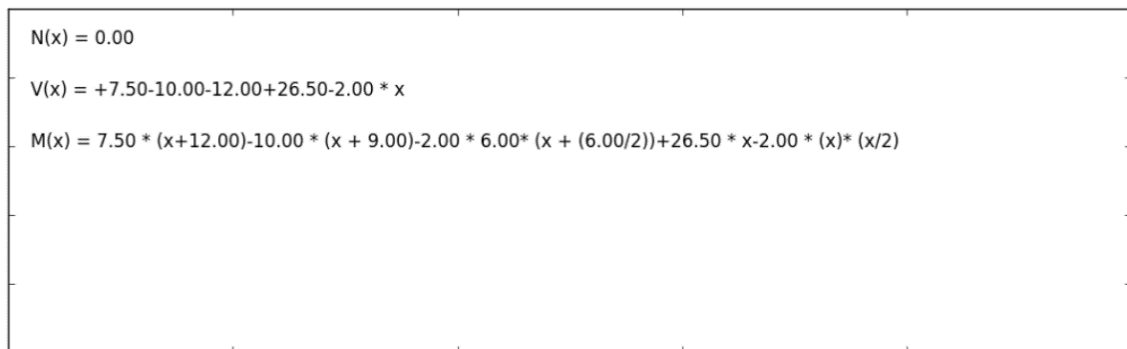
Fonte: Autoria Própria (2019)

Figura 54 - Seccionamento local da barra 4 - exemplo 02



Fonte: Autoria Própria (2019)

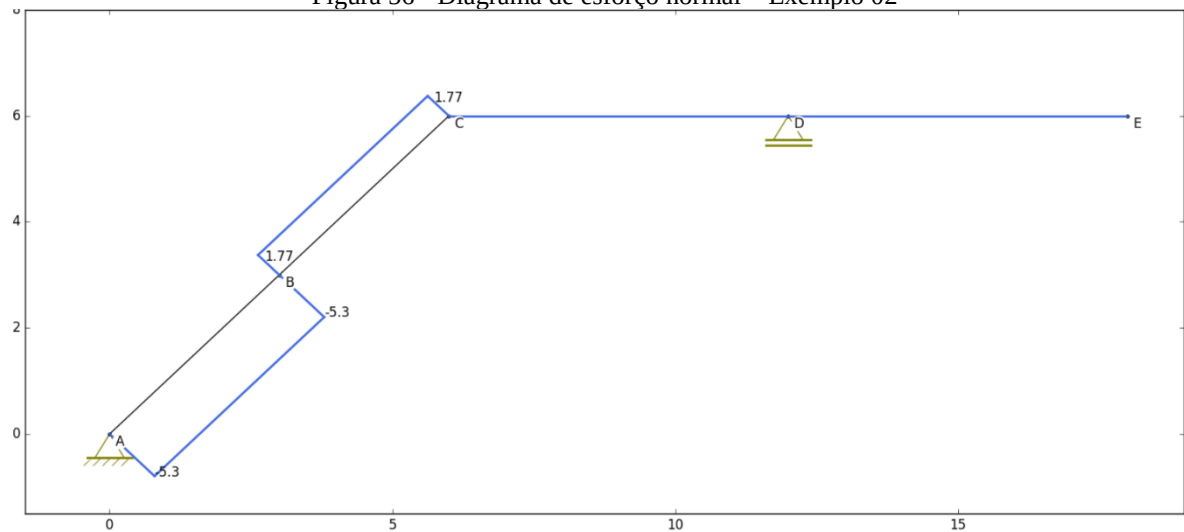
Figura 55 - Função dos esforços internos da barra 3 - exemplo 02



Fonte: Autoria Própria (2019)

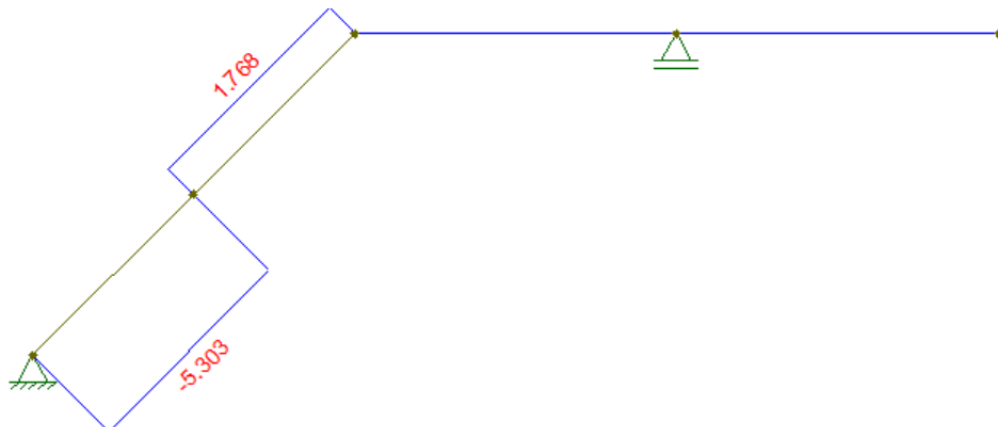
Tendo posse das equações, pode-se então realizar a plotagem dos diagramas de esforço internos, e por fim compará-los com o Ftool. Primeiramente é apresentado o diagrama dos esforços normais (figura 56).

Figura 56 - Diagrama de esforço normal – Exemplo 02



Fonte: Autoria Própria (2019)

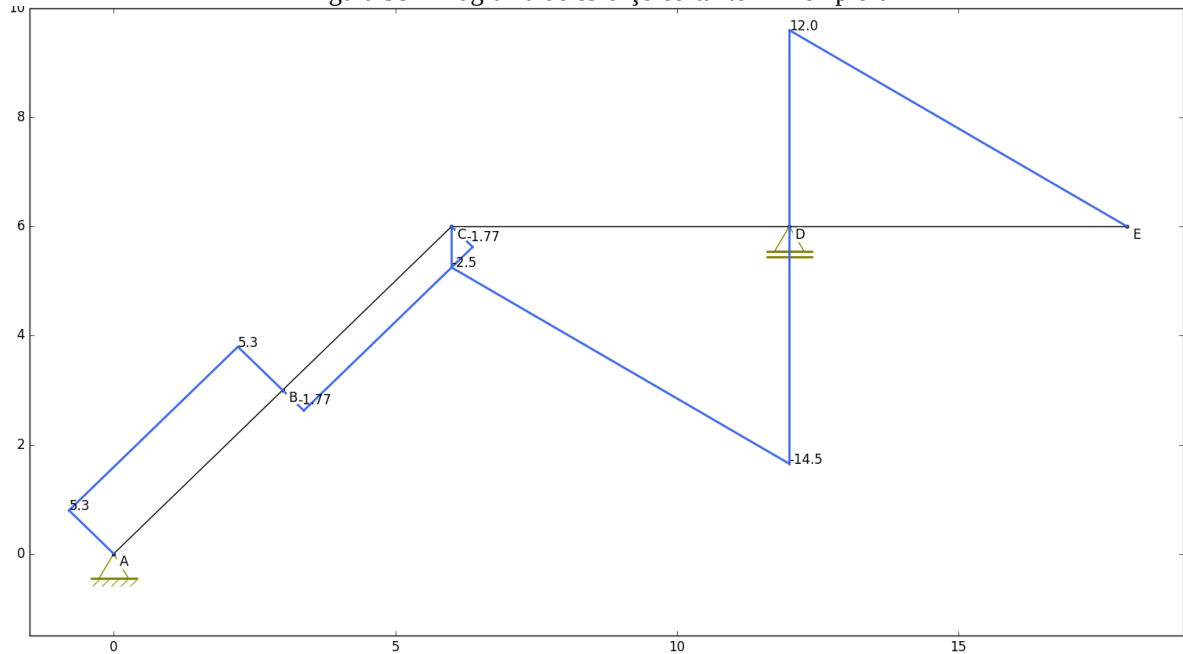
Figura 57 - Diagrama de esforço normal/ FTOOL - Exemplo 02



Fonte: Ftool (2019)

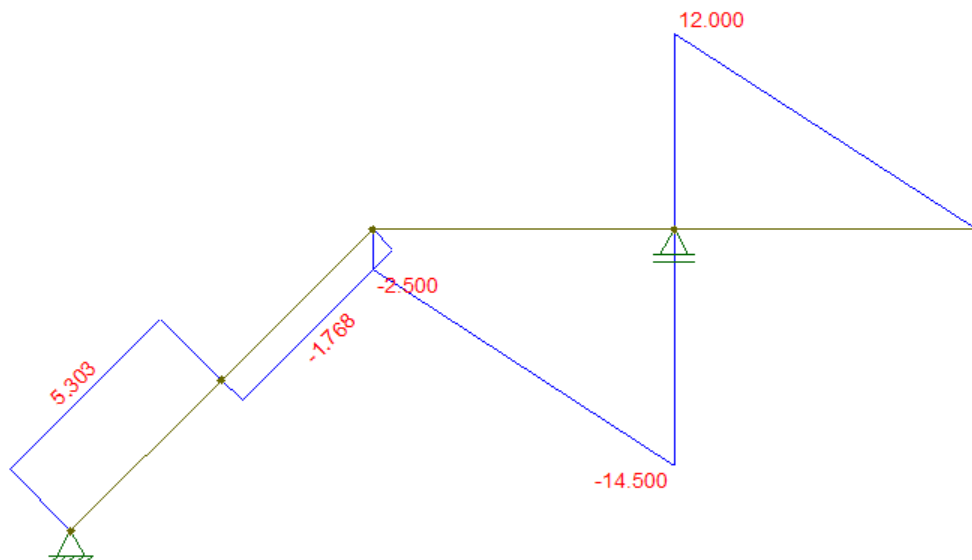
Novamente ocorreu semelhança nos traçados ao se comparar com os dados gerados pelo Ftool (Figura 57). Da mesma maneira, segue a lógica para os próximos diagramas.

Figura 58 - Diagrama de esforço cortante – Exemplo 02



Fonte: Autoria Própria (2019)

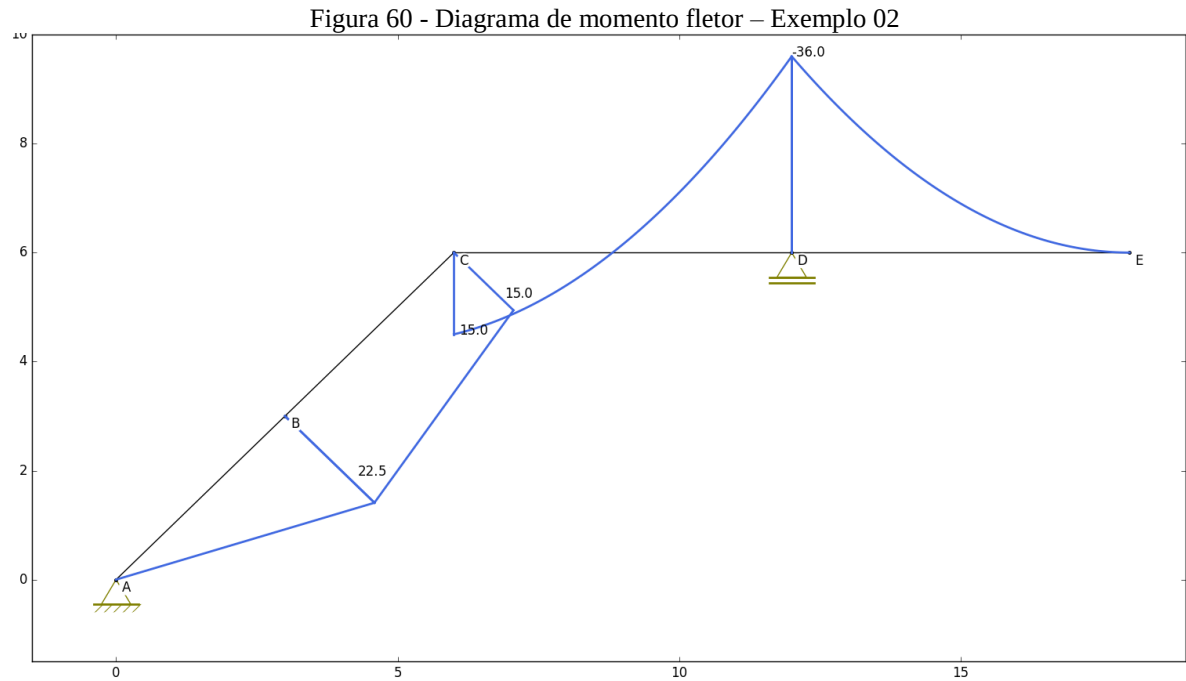
Figura 59 - Diagrama de esforço cortante/ FTOOL - Exemplo 02



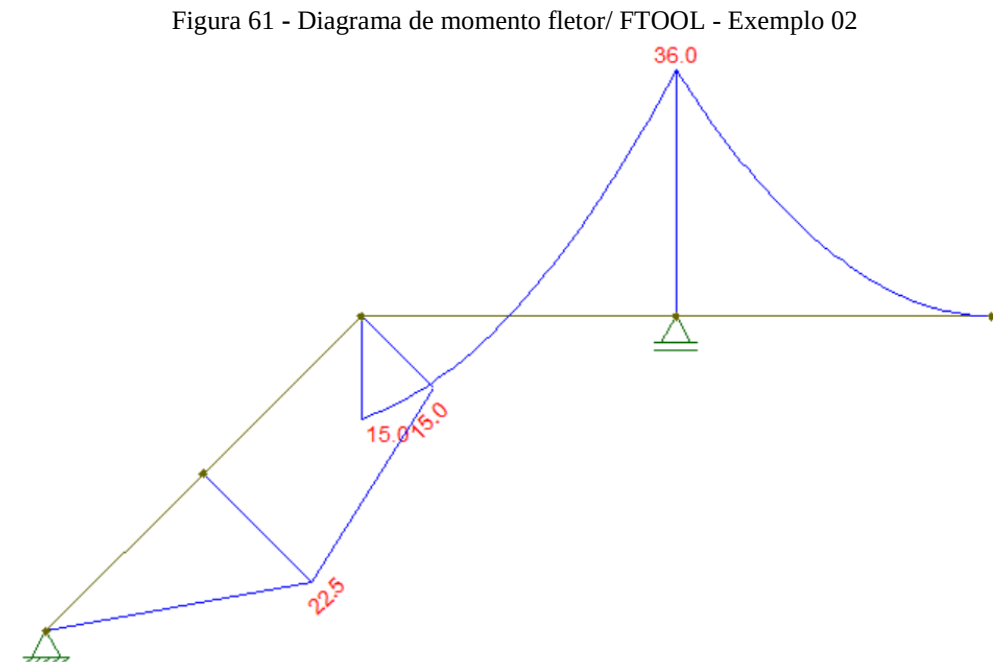
Fonte: Ftool (2019)

Mais uma vez, agora para o diagrama dos esforços cortantes os traçados se comportaram de forma idêntica, ver figura (58 e 59).

Por último, o programa gerou o traçado do diagrama de momento fletor (figura 60) e apresentou a mesma configuração do que foi processado pelo Ftool (figura 61).



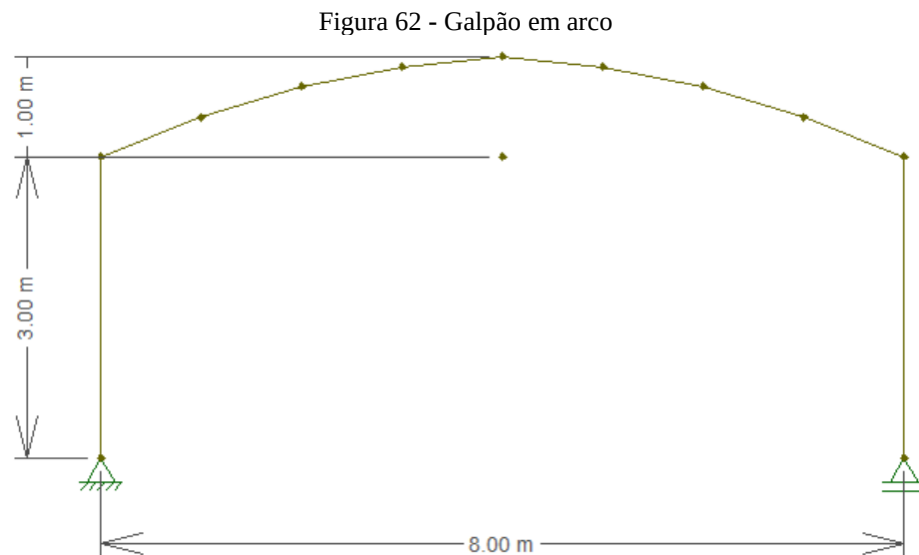
Fonte: Autoria Própria (2019)



Fonte: Ftool (2019)

4.3 EXEMPLO 03 – GALPÃO COM COBERTURA EM ARCO

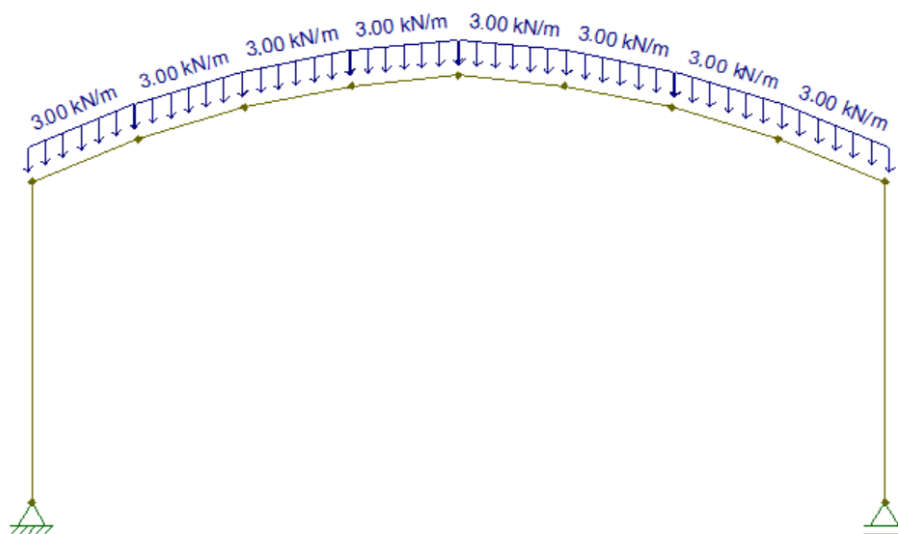
O terceiro exemplo, baseia-se em analisar um galpão com cobertura em arco. Para tanto, foi estipulado suas dimensões (largura e altura) conforme presentes em norma (Figura 62).



Fonte: Ftool (2019)

Para a modelagem dessa estrutura, algumas considerações foram feitas. Nesse caso, vai ser analisado a influência do seu peso próprio, para esse exemplo, optou-se por considerar o peso próprio agindo de forma distribuída ao longo da seção transversal com intensidade de 3 kN/m (Figura 63).

Figura 63 - Galpão metálico em arco sob influência do peso próprio

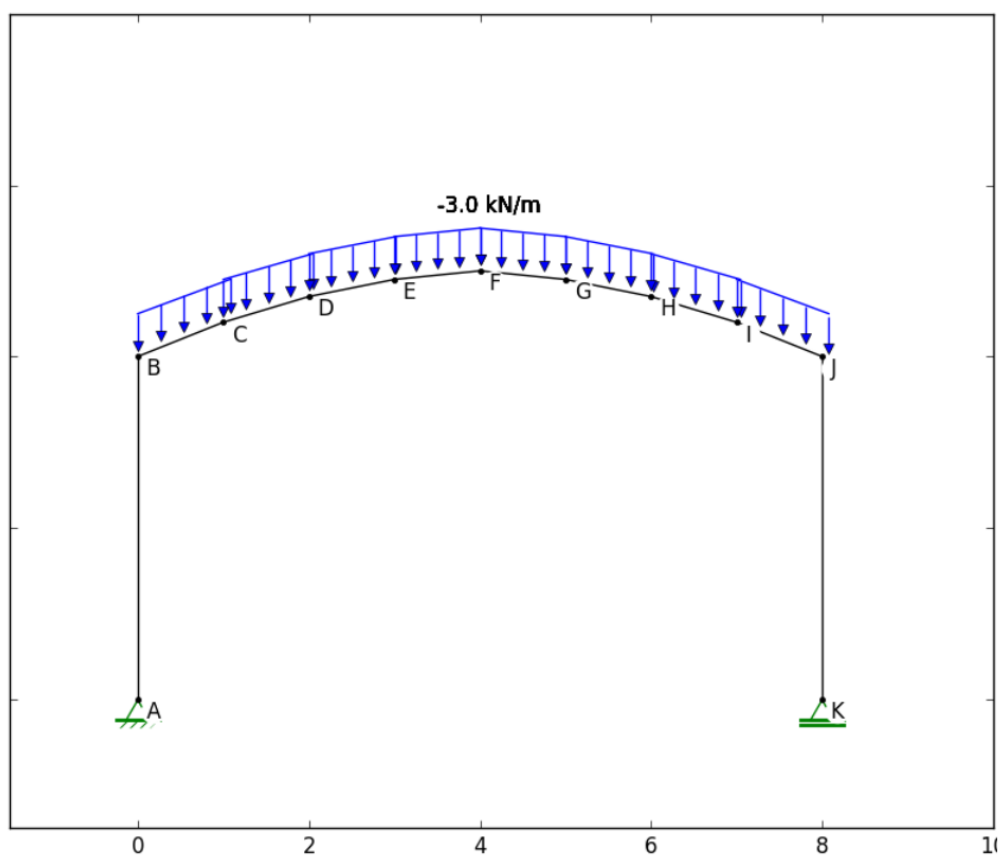


Fonte: Ftool (2019)

De forma análoga aos exemplos anteriores, de início, são introduzidos os dados de entrada, afim de modelar o problema em questão. Assim, a figura 64, ilustra o problema gerado.

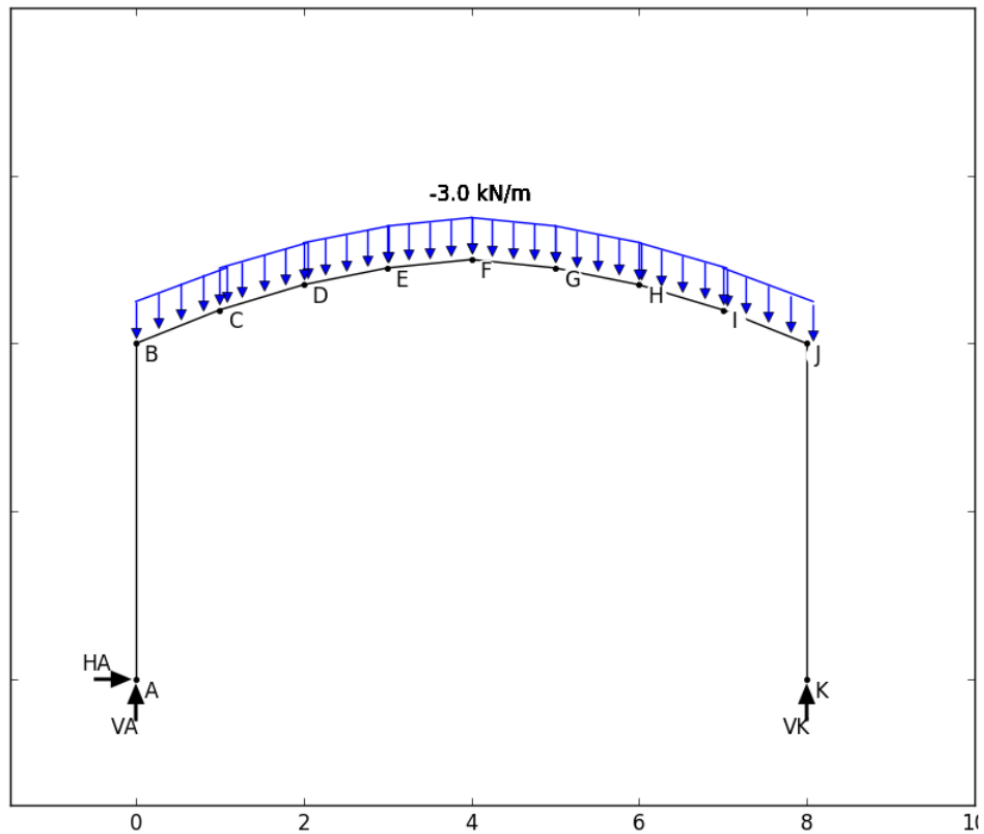
A figura 65 demonstra o seu DCL, e em seguida na figura 66 é apresentado o cálculo das reações de apoio.

Figura 64 - Problema estrutural - exemplo 03.



Fonte: Autoria Própria (2019)

Figura 65 - DCL - exemplo 03



Fonte: Autoria Própria (2019)

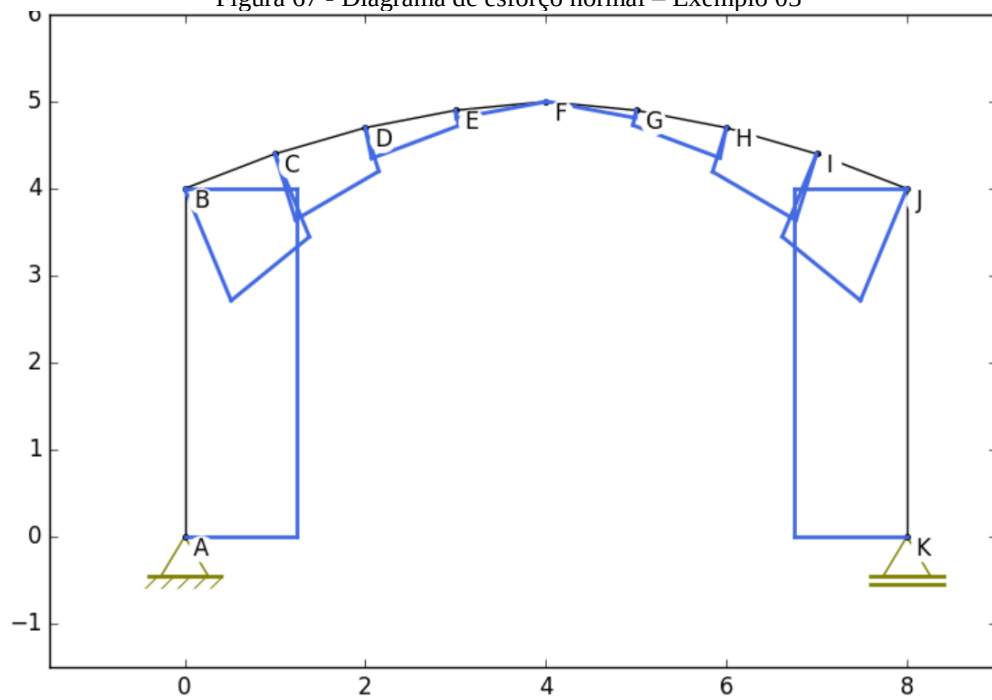
Figura 66 - Cálculo das reações de apoio - exemplo 03

$$\begin{aligned} \sum F_H &= 0 \\ +H_A &= 0 \\ \sum F_V &= 0 \\ +V_K + V_A - 3.23 - 3.13 - 3.06 - 3.01 - 3.01 - 3.06 - 3.13 - 3.23 &= 0 \\ \sum M_A &= 0 \\ +V_K * 8.00 + V_A * 0.00 - 3.23 * 0.50 - 3.13 * 1.50 - 3.06 * 2.50 - 3.01 * 3.50 - 3.01 * 4.50 - 3.06 * 5.50 - 3.13 * 6.50 - 3.23 * 7.50 &= 0 \\ H_A &= 0.00 \\ V_K &= 12.43 \\ V_A &= 12.43 \end{aligned}$$

Fonte: Autoria Própria (2019)

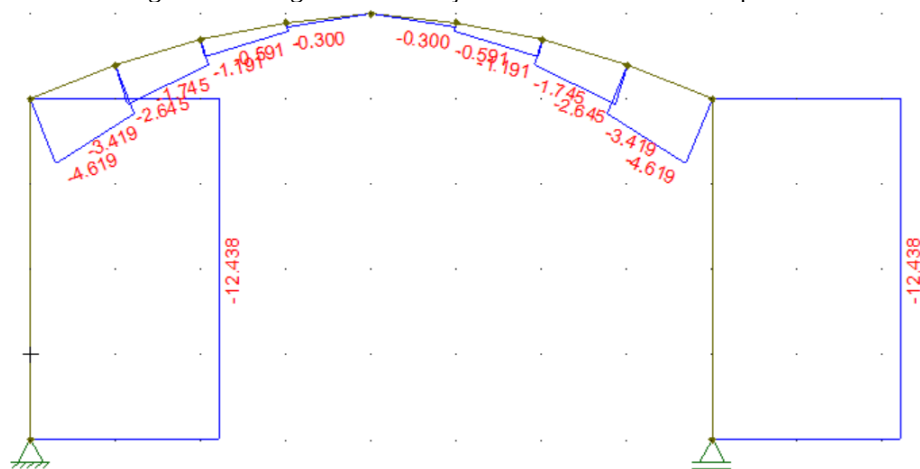
Afim de não tornam uma leitura repetitiva e cansativa, os exemplos anteriores já são capazes de demonstrar a eficiência do cálculo para apresentar as equações dos esforços internos. Dessa forma, são apresentados apenas os diagramas dos esforços internos e os comparar com os gerados no software Ftool.

Figura 67 - Diagrama de esforço normal – Exemplo 03



Fonte: Autoria Própria (2019)

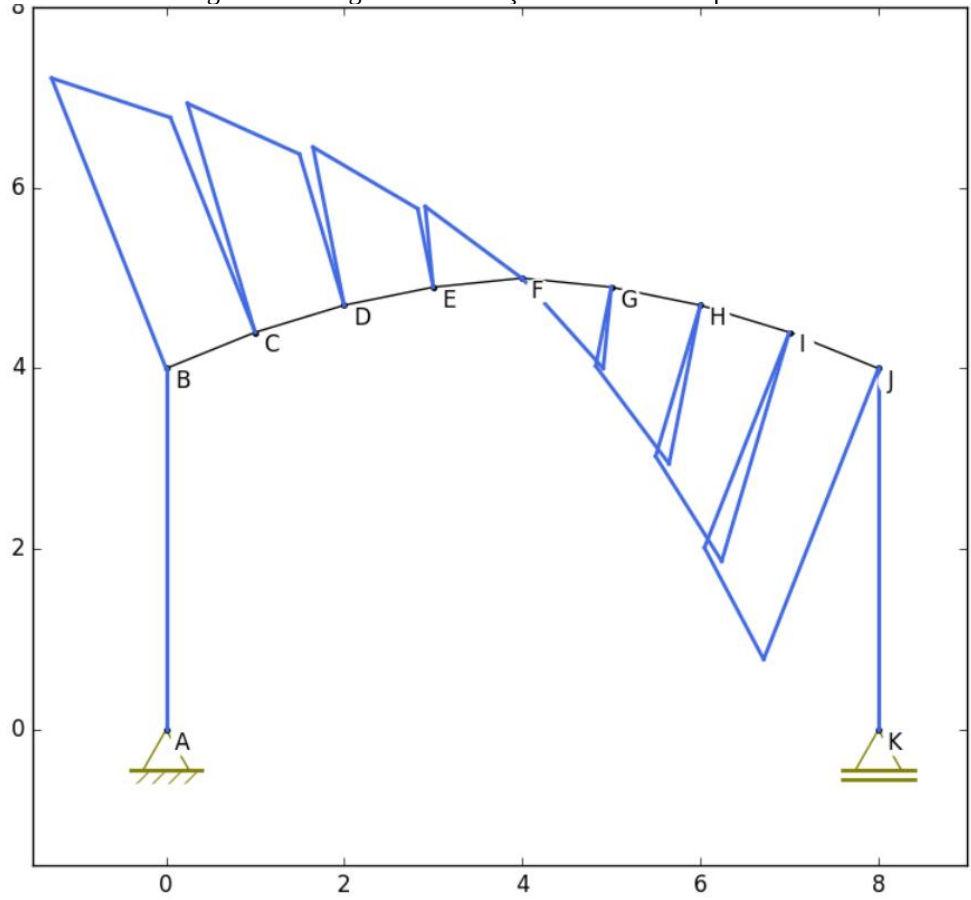
Figura 68 - Diagrama de esforço normal/ FTOOL - Exemplo 03



Fonte: Ftool (2019)

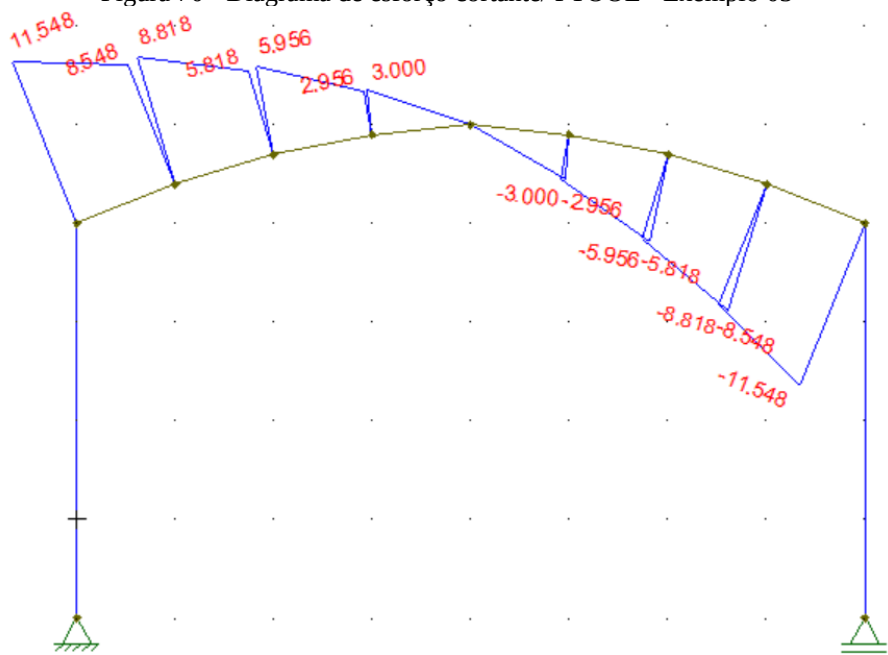
Para o caso acima, na figura 68, observa-se que o esforço normal da barra vertical da direita apresenta-se voltado para fora da estrutura, diferentemente da figura 67, isso ocorre pelo simples fato de se adotar sentidos contrários no momento da análise local da barra. A seguir são apresentados os diagramas de esforço cortante e momento fletor.

Figura 69 - Diagrama de esforço cortante – Exemplo 03



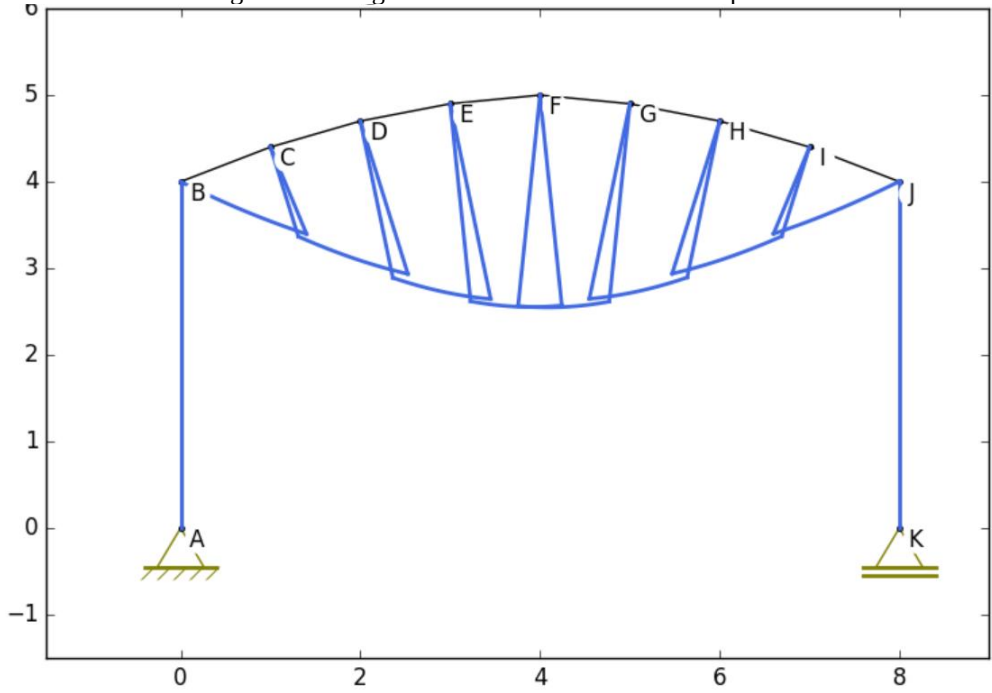
Fonte: Autoria Própria (2019)

Figura 70 - Diagrama de esforço cortante/ FTOOL - Exemplo 03



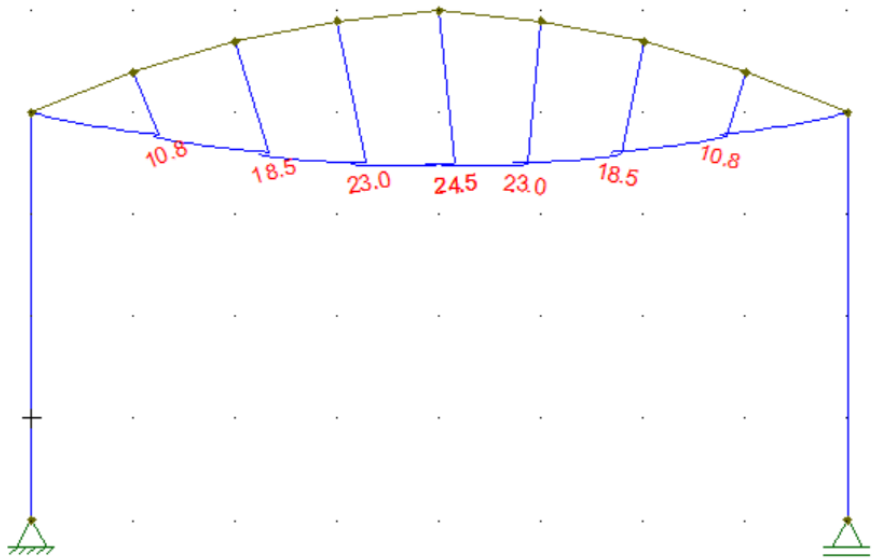
Fonte: Ftool (2019)

Figura 71 - Diagrama de momento fletor – Exemplo 03



Fonte: Autoria Própria (2019)

Figura 72 - Diagrama de momento fletor/ FTOOL - Exemplo 03



Fonte: Ftool (2019)

5 CONCLUSÃO

Pode-se concluir, que de forma geral o programa implementado neste trabalho se apresentou como uma ferramenta de enorme poder para a resolução de problemas envolvendo estruturas isostáticas de pórticos planos, pois além de revelar resultados idênticos ao ser comparado com FTOOL, fornece soluções de forma rápida e fácil, tornando o usuário capaz de interpretar os resultados com uma maior agilidade.

Além disso, o software se mostrou como uma ferramenta de grande proveito no ensino-aprendizagem, pois tanto alunos quanto professores podem utiliza-lo, gerando consequentemente uma redução de tempo ao se analisar determinado problema.

Um diferencial desse para outros programas de análises estruturais é o fato dele poder gerar um relatório completo com todos os passos para determinação das equações dos esforços internos e tudo isso de forma ilustrativa, tornando-se assim muito mais detalhado, visto que o usuário poderá acompanhar o desenvolvimento do problema estrutural.

A linguagem de programação Python, permitiu produzir uma cadeia de códigos de forma funcional e de fácil interpretação. Por se tratar de uma linguagem orientada ao objeto, essa linguagem tornou possível dividir a estrutura do software, em classes, facilitando a escrita dos algoritmos que compõe o programa.

Para realização de futuros trabalhos associados a esse tema, pode-se incluir a análise de pórticos hiperestáticos, além de inserir o elemento denominado por rótula, ou ainda a adição de novas funcionalidades ao software, seja na plotagem de novas características, seja tornando a ferramenta mais completa e com menos possibilidades de erros.

REFERÊNCIAS

- ALMEIDA, M. C. F. **Estruturas Isostáticas**. São Paulo: Oficina de textos, 2009.
- BEER, F. P. et al. **Mecânica dos materiais**. Tradução de José Benaque Rubert; Walter Libardi. 5 ed. Porto Alegre: AMGH, 2011. 803 p.
- BEZERRA, Alysson Aldrin Barreto; SILVA, Luanda Maria Sousa da; LIMA, Antônio Wagner de. **Desenvolvimento de um programa computacional para análise de vigas Euler-Bernoulli utilizando a linguagem Pytho**. Principia, João Pessoa, v. 38, p.54-60, 2 fev. 2017.
- BORGES, L. E. **Python para desenvolvedores**. 2. ed. Rio de Janeiro: Edição do Autor, 2010.
- BRANCHIER, H. S. **Contribuições dos softwares na aprendizagem de análise e cálculo de elementos estruturais**. Lajeado: Universidade do Vale do Taquari, 2017. 110 p.
- CHAGAS, J. D. E.; OLIVEIRA, M. V. D. G. **Programação Orientada a Objetos Versus Programação Orientada a Aspectos**: Um estudo de caso comparativo através do desenvolvimento de um banco de questões. São Cristóvão/SE: [s.n.], 2009. Disponível em: <<https://danielettinger.files.wordpress.com/2010/12/monografia.pdf>>. Acesso em: 20 fevereiro 2019.
- DEVMEDIA. **Site da DEVMEDIA**, 2018. Disponível em: <<https://www.devmedia.com.br>>. Acesso em: 05 março 2019.
- GORFIN, B.; OLIVEIRA, M.M. **Estruturas Isostáticas**. Livros Técnicos e Científicos Editora S.A. - Rio de Janeiro, 1975.
- GOETTEN, J.; WINCK, D. AspectJ. **Programação Orientada a Aspectos com Java**. São Paulo/SP: Novatec Editora, 2006.
- KIMURA, A. **Informática aplicada em estruturas de concreto armado**. Editora Pini, v. 2, 2007, p. 8-9.
- LEET, KENNETH M. **Fundamentos da análise estrutural** - 3ª edição. 2009. 810 f. Traduzido por J. E. N. Tortello, 2009. Revisão técnica por P. V. P. Mendonça, 2009.
- LEITE, M.; RAHAL, N. A. S. **Programação orientada ao objeto: uma abordagem didática**. Universidade Estadual de Campinas, Taquara - Rio Grande do Sul - Brasil, p. 8, 2006. 31
- MARTHA, L. F. **Análise de Estruturas**: Conceitos e Métodos básicos. Rio de Janeiro: Elsevie, 2010.

MENEZES, N.N.C. **Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes**. São Paulo: Novatec, 2014.

MOREIRA, D. F. **Análise matricial de estruturas**. Rio de Janeiro, Livros Técnicos e Científicos; São Paulo, Editora da Universidade de São Paulo, 1977.

PYSCIENCE-BRASIL. **Python: O que é? Por que usar?**. Portal Wikidot. Disponível em: <<http://pysciencebrasil.wikidot.com/python:python-oq-e-pq>>. Acessado em 05 de fevereiro de 2019.

PRADO, I. M. **CS-ASA Preprocessador: Sistema Gráfico Interativo de Pre-processamento para Análise Avançada de Estruturas**. Ouro Preto: Universidade Federal de Ouro Preto, 2012. 163 p.

ROCHA, Z. **Lições aprendidas com o mundo Python**. 2011. Disponível em: <<https://zenorocha.com/licoesaprendidas-com-o-mundo-python/>>. Acessado em 06 de fevereiro de 2019.

SORIANO, H. L. **Estática das Estruturas**. 2ª edição. Rio de Janeiro: Editora Ciência Moderna Ltda, 2010.

SORIANO, H. L. **Análise de Estruturas: formulação matricial e implementação computacional**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2005.

SUSSEKIND, J. C. **Curso de Análise Estrutural - Volume I: Estruturas Isostáticas**, 1981.