



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

FELIPE TAVARES RIBEIRO DE SOUZA

**DESENVOLVIMENTO DE UM PROGRAMA PARA ANÁLISE DE VIGAS
HIPERESTÁTICAS PELO MÉTODO DAS FORÇAS**

CARAÚBAS

2019

FELIPE TAVARES RIBEIRO DE SOUZA

**DESENVOLVIMENTO DE UM PROGRAMA PARA ANÁLISE DE VIGAS
HIPERESTÁTICAS PELO MÉTODO DAS FORÇAS**

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.

SS719 SOUZA, FELIPE TAVARES RIBEIRO.
d DESENVOLVIMENTO DE UM PROGRAMA PARA ANÁLISE DE
VIGAS HIPERESTÁTICAS PELO MÉTODO DAS FORÇAS /
FELIPE TAVARES RIBEIRO SOUZA. - 2019.
105 f. : il.

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

1. ESTRUTURAS ESTATICAMENTE INDETERMINADAS. 2.
HIPERESTÁTICA. 3. RECURSOS COMPUTACIONAIS . 4.
SOFTWARE APLICADO. 5. CONSTRUÇÃO CIVIL. I.
Ferreira, Manoel Dênis Costa, 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.

FELIPE TAVARES RIBEIRO DE SOUZA

**DESENVOLVIMENTO DE UM PROGRAMA PARA ANÁLISE DE VIGAS
HIPERESTÁTICAS PELO MÉTODO DAS FORÇAS**

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

Defendida em: 13 / 08 / 2019

BANCA EXAMINADORA


Manoel Dênis Costa Ferreira, Prof. Dr. (UFERSA)
Presidente


Carlos Vinicius Damaceno Bessa, Prof. (UFERSA)
Membro Examinador


Francisca Ires Vieira de melo, Prof. (UFERSA)
Membro Examinador

AGRADECIMENTOS

A Deus, por cada vez que pedi forças para continuar com essa jornada que não é tão fácil, que mesmo diante das fraquezas minha fé persistiu acima de tudo;

Aos meus pais, Rivaldo Tavares e Joana Lucia, que me educaram de forma a buscar sem o melhor e me mostraram que devo continuar com os meus sonhos e nunca desistir, superando todas as barreiras que estiverem pela frente;

À minha noiva, Aricia Miele e mãe da nossa filha Marina que está por vir nos próximos meses, pela paciência e forças para continuar essa jornada, com toda certeza me tornei uma pessoa melhor desde o momento que de conheci;

A todos que participaram diretamente e indiretamente que se tornaram minha família nesses longos anos de estudo;

A meu orientador, Manoel Denis Costa Ferreira, por disponibilizar um pouco do seu tempo para transmitir um pouco de seus conhecimentos, e todas as horas dedicadas a ajudar com minhas dúvidas, os seus ensinamentos serão de enorme valia em minha vida profissional;

E a todos os docentes que se tornaram além de professores, amigos e serão levados para toda minha vida;

“Caíram mil homens a tua esquerda e dez mil a tua direita, tu não serás atingido. ”

(Salmo 90)

RESUMO

Um dos grandes aliados em resolução de problemas complexos de hiperestaticidade, são recursos computacionais, pois propiciam uma melhor visualização e descrição dos dados, tornando cada vez mais precisos as análises e fortalecendo o laço com a construção civil. Durante o aprendizado sobre o tema, o uso de recursos computacionais pode otimizar o tempo de forma simples proporcionando uma melhor aprendizagem. Com as necessidades abordadas o trabalho tem objetivo de criar um programa em linguagem de Python para fornecer a resolução detalhada de problemas de vigas hiperestáticas, fazendo com que assim possa aproximar os conceitos estruturais por formas de questões bem resolvidas. Para efetivar a validação do programa foram analisados dois exemplos, o primeiro se tratando de um modelo simples presente nas literaturas, contendo três barras distribuídas em quatro apoios, onde, nas duas barras iniciais estará presente uma carga distribuída uniforme. Já o segundo exemplo será apresentado um problema em nossa realidade, uma viga com quatro barras distribuídas em cinco apoios, contendo carregamento distribuído uniforme e forças pontuais ao longo do trecho, essa viga está inserida em uma edificação unifamiliar sendo representada posteriormente. Para validação do programa serão analisados os gráficos gerados apoio do software Ftool, e com essas comparações foram verificadas que o programa possui eficiência para fornecer soluções de forma simples, rápidas e precisas, tornando possível a visualização através de um relatório detalhado.

Palavras-chave: Estruturas estaticamente indeterminadas, Hiperestática, Construção Civil. Recursos computacionais. Relatório.

ABSTRACT

One of the great allies in solving complex problems of hyperstaticity, are computational resources, because they provide a better visualization and description of the data, making the analysis increasingly accurate and strengthening the bond with the construction industry. While learning about the subject, the use of computational resources can simply optimize time providing better learning. With the needs addressed the work aims to create a program in Python language to provide detailed resolution of problems of hyperstatic beams, so that can approach the structural concepts by forms of well-resolved questions. To validate the program two examples were analyzed, the first being a simple model present in the literature, containing three bars distributed in four supports, where in the two initial bars a uniform distributed load will be present. Already the second example will be presented a problem in our reality, a beam with four bars distributed in five supports, containing uniform distributed loading and punctual forces along the stretch, this beam is inserted in a single-family building being represented later. For the validation of the program will be analyzed the graphs generated with the Ftool software graphs, and with these comparisons it was verified that the program has efficiency to provide solutions in a simple, fast and accurate way, making possible the visualization through a detailed report.

Keywords: Statically undetermined structures, Hyperstatic, Civil Construction. Computational resources. Report.

LISTA DE FIGURAS

Figura 1 - Estruturas reticulares formadas por elementos unidimensionais: Barras, vigas, pilares e Treliças;	21
Figura 2 - Estruturas bidimensionais: lâminas e lajes	22
Figura 3 - Estruturas tridimensionais	22
Figura 4 - Representação dos apoios em modelos planos de estruturas	25
Figura 5 - Representação dos esforços internos de um elemento estrutural quando submetido a um sistema de forças.....	26
Figura 6 - Representação dos esforços internos em estruturas planas.....	26
Figura 7 - Diagramas ou Linhas de Estado dos esforços solicitantes internos.....	27
Figura 8- Composição das estruturas reticulares	28
Figura 9 - Composição das estruturas reticulares	28
Figura 10 - Ação de um carregamento sobre uma estrutura.....	30
Figura 11 - Representação dos tipos de carregamentos distribuídos mais comuns em estruturas reticulares.....	30
Figura 12 - Equações do equilíbrio estático	31
Figura 13- Exemplo de vigas.....	33
Figura 14- Exemplos de estruturas hiperestáticas	35
Figura 15- Viga contínua com três vãos e carregamento uniformemente distribuído no primeiro vão	35
Figura 16- Problema auxiliar.....	36
Figura 17- Solicitação externa isolada no problema	36
Figura 18- Força (Hiperestático) X1 isolado no problema.....	37
Figura 19- Força (Hiperestático) X2 isolado no problema.....	38
Figura 20- Combinação de diagramas de momentos fletores para o cálculo do termo de carga δ_{10}	39
Figura 21- Combinação de diagramas de momentos fletores para o cálculo do termo de carga δ_{20}	40
Figura 22- Combinação de diagramas de momentos fletores para o cálculo do coeficiente de flexibilidade δ_{11}	40
Figura 23- Combinação de diagramas de momentos fletores para o cálculo dos coeficientes de flexibilidade δ_{12} e δ_{21}	41

Figura 24- Combinação de diagramas de momentos fletores para o cálculo do coeficiente de flexibilidade δ_{22}	41
Figura 25 - Blocos em python	43
Figura 26 - Exemplo de variáveis do tipo strings.....	44
Figura 27 - Representação de uma lista.....	46
Figura 28 - Representação de uma lista.....	46
Figura 29 - Caixa preta recebendo mensagens	49
Figura 30 - Esquema didático de hierarquia das classes	49
Figura 31 - Exemplo de código para geração de arquivo html.....	50
Figura 32 - Hierarquia das classes implementadas.....	51
Figura 33 - Classe problema estrutural dentro do Visual Studio Code	52
Figura 34 - Dados de entradas e variáveis.....	53
Figura 35 - Código de armazenamento para problemas analisados	53
Figura 36- Parte do código para caracterizar o problema.....	54
Figura 37- Código para inserir vinculações no problema real.....	55
Figura 38-Código para inserir vinculações no problema auxiliar	55
Figura 39-Código para inserir vinculações no problemas virtuais	56
Figura 40 - Código da função geradora do DLC	57
Figura 41 - Código da função calcular reações	58
Figura 42 - Parte do código referente a desenhar seção	59
Figura 43 – Parte do código gerador do arquivo HTML.....	60
Figura 44 – Viga hiperestática com carga uniforme - ftool.....	61
Figura 45- Exemplo 01 - Problema Estrutural Real	62
Figura 46- Exemplo 01 - Problema Auxiliar.....	62
Figura 47- Exemplo 01 - Problema 0	63
Figura 48- Exemplo 01 - Problema 1	63
Figura 49- Exemplo 01 - Problema 2	64
Figura 50- DCL - exemplo 01 - Problema 0.....	65
Figura 51- Cálculo das reações de apoio - exemplo 01 - problema 0	65
Figura 52- DCL - exemplo 01 - problema 1	66
Figura 53- Cálculo das reações de apoio - exemplo 01 - problema 1	66
Figura 54- DCL - exemplo 01 - problema 2.....	67
Figura 55- Cálculo das reações de apoio - exemplo 01 - problema 2	67
Figura 56- Seccionamento local da barra 1 - exemplo 01 - Problema 0	68

Figura 57- Função dos esforços internos da barra 1 - exemplo 01 - Problema 0	68
Figura 58- Seccionamento local da barra 1 - exemplo 01 - Problema 1	69
Figura 59- Função dos esforços internos da barra 1 - exemplo 01 - Problema 1	69
Figura 60- Seccionamento local da barra 1 - exemplo 01 - Problema 2	70
Figura 61- Função dos esforços internos da barra 1 - exemplo 01 - Problema 2	70
Figura 62- Seccionamento local da barra 2 - exemplo 01 - Problema 0	71
Figura 63- Função dos esforços internos da barra 2 - exemplo 01 - Problema 0	71
Figura 64- Seccionamento local da barra 3 - exemplo 01 - Problema 0	72
Figura 65- Função dos esforços internos da barra 3 - exemplo 01 - Problema 0	72
Figura 66- Seccionamento local da barra 2 - exemplo 01 - Problema 1	73
Figura 67- Função dos esforços internos da barra 2 - exemplo 01 - Problema 1	73
Figura 68- Seccionamento local da barra 3 - exemplo 01 - Problema 1	74
Figura 69- Função dos esforços internos da barra 3 - exemplo 01 - Problema 1	74
Figura 70- Seccionamento local da barra 2 - exemplo 01 - Problema 2	75
Figura 71- Função dos esforços internos da barra 2 - exemplo 01 - Problema 2	75
Figura 72- Seccionamento local da barra 3 - exemplo 01 - Problema 2	76
Figura 73- Função dos esforços internos da barra 3 - exemplo 01 - Problema 2	76
Figura 74- Diagrama de esforço cortante – Exemplo 01 - Problema 0	77
Figura 75- Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema 0.....	77
Figura 76- Diagrama de esforço cortante – Exemplo 01 - Problema 1	78
Figura 77- Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema 1.....	78
Figura 78- Diagrama de esforço cortante – Exemplo 01 - Problema 2	79
Figura 79- Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema 2.....	79
Figura 80- Diagrama de momento fletor – Exemplo 01 - Problema 0.....	80
Figura 81- Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema 0.....	80
Figura 82- Diagrama de momento fletor – Exemplo 01 - Problema 1	81
Figura 83- Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema 1.....	81
Figura 84- Diagrama de momento fletor – Exemplo 01 - Problema 2	82
Figura 85- Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema 2.....	82
Figura 86- Exemplo 1 - Problema real	83
Figura 87- DCL - Exemplo 01 - Problema real.....	83
Figura 88- Cálculos das reações de apoio - exemplo 01 - Problema real.....	84
Figura 89-- Diagrama de esforço cortante – Exemplo 01 - Problema real.....	84
Figura 90- - Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema Real	85

Figura 91- - Diagrama de momento fletor – Exemplo 01 - Problema real.....	85
Figura 92- - Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema Real	85
Figura 93 - Planta baixa prédio analisado	86
Figura 94 – Prédio analisado	87
Figura 95 - Áreas de influência da laje nas vigas	88
Figura 96 – Viga hiperestática sob influências das cargas.	88
Figura 97- Exemplo 02 - Problema Estrutural	89
Figura 98- Exemplo 02 - Problema Auxiliar	89
Figura 99- Exemplo 02 - Problema 0	90
Figura 100- Exemplo 02 - Problema 1	90
Figura 101- Exemplo 02 - Problema 2	91
Figura 102- Exemplo 02 - Problema 3	91
Figura 103- DCL - exemplo 02 - Problema 0.....	92
Figura 104- Cálculo das reações de apoio - exemplo 02 - problema 0	92
Figura 105- DCL - exemplo 02 - problema 1	93
Figura 106- Cálculo das reações de apoio - exemplo 02 - problema 1	93
Figura 107- DCL - exemplo 02 - problema 2.....	94
Figura 108- Cálculo das reações de apoio - exemplo 02 - problema 2	94
Figura 109- DCL - exemplo 02 - problema 3.....	95
Figura 110- Cálculo das reações de apoio - exemplo 02 - problema 3	95
Figura 111- Diagrama de momento fletor – Exemplo 02 - Problema 0.....	96
Figura 112- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 0.....	96
Figura 113- Diagrama de momento fletor – Exemplo 02 - Problema 1	97
Figura 114- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 1.....	97
Figura 115- Diagrama de momento fletor – Exemplo 02 - Problema 2.....	98
Figura 116- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 2.....	98
Figura 117- Diagrama de momento fletor – Exemplo 02 - Problema 3	99
Figura 118- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 3.....	99
Figura 119- Exemplo 1 - Problema real	100
Figura 120- DCL - Exemplo 02 - Problema real	100
Figura 121- Cálculos das reações de apoio - exemplo 02 - Problema real.....	101
Figura 122-- Diagrama de esforço cortante – Exemplo 02 - Problema real.....	101
Figura 123- - Diagrama de esforço cortante/ FTOOL - Exemplo 02 - Problema Real	102
Figura 124- - Diagrama de momento fletor – Exemplo 02 - Problema real.....	102

Figura 125- - Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema Real 102

LISTA DE QUADROS

Quadro 1 - Classificação das ações	23
Quadro 2 - Cargas em modelos estruturais.....	29
Quadro 3 - Operações com números	44
Quadro 4 - Operadores relacionais	45
Quadro 5 - Operadores lógicos	45
Quadro 6 - Conceitos essenciais da Programação Orientada a Objetos	47

SUMÁRIO

1	INTRODUÇÃO	15
1.1	CONSIDERAÇÕES INICIAIS	15
1.2	OBJETIVOS	16
1.2.1	<i>Objetivos Gerais.....</i>	<i>16</i>
1.2.2	<i>Objetivos Específicos</i>	<i>16</i>
1.3	PROBLEMÁTICA E JUSTIFICATIVA.....	16
1.4	ESTRUTURA DO TRABALHO.....	17
2	REVISÃO DA LITERATURA	18
2.1	ABORDAGEM SOBRE A ENGENHARIA ESTRUTURAL	18
2.2	ANÁLISE ESTRUTURAL	18
2.3	ESTRUTURAS E SEUS ELEMENTOS.....	19
2.3.1	<i>Definição de estrutura.....</i>	<i>19</i>
2.3.2	<i>Tipos de sistemas estruturais</i>	<i>20</i>
2.4	AÇÕES ATUANTES NAS ESTRUTURAS.....	22
2.4.1	<i>Esforços solicitantes externos</i>	<i>23</i>
2.4.1.1	<i>Ações externas ativas</i>	<i>23</i>
2.4.1.2	<i>Ações externas reativas</i>	<i>24</i>
2.4.2	<i>Esforços solicitantes internos.....</i>	<i>25</i>
2.4.2.1	<i>Esforços internos em estruturas planas</i>	<i>26</i>
2.4.2.2	<i>Diagramas dos esforços solicitantes internos.....</i>	<i>27</i>
2.5	ESTRUTURAS RETICULARES	27
2.5.1	<i>Cargas em estruturas reticulares.....</i>	<i>29</i>
2.5.1.1	<i>Cargas concentradas.....</i>	<i>29</i>
2.5.1.2	<i>Cargas distribuídas</i>	<i>30</i>
2.5.2	<i>Equilíbrio estático.....</i>	<i>30</i>
2.5.3	<i>Estabilidade e estaticidade de estruturas planas.....</i>	<i>31</i>
2.6	VIGAS.....	32
2.6.1	<i>Tipos de vigas.....</i>	<i>33</i>
2.6.1.1	<i>Viga simplesmente apoiada ou viga simples</i>	<i>33</i>
2.6.1.2	<i>Viga engastada ou viga em balanço</i>	<i>33</i>
2.6.1.3	<i>Viga em balanço ou com apoios fixos e rolete</i>	<i>33</i>
2.7	VIGAS HIPERESTÁTICAS.....	34

2.7.1	<i>Determinação do grau de hiperestaticidade.....</i>	34
2.7.2	<i>Método das Forças para analisar estruturas estaticamente indeterminadas.....</i>	35
2.7.2.1	<i>Análise de viga continua</i>	35
2.8	LINGUAGEM PYTHON	42
2.8.1	<i>Apresentando o Python</i>	42
2.8.2	<i>Conceitos básicos.....</i>	42
2.8.3	<i>Variáveis.....</i>	43
2.8.3.1	<i>Variáveis string</i>	44
2.8.3.2	<i>Variáveis numéricas</i>	44
2.8.3.3	<i>Variáveis do tipo lógico</i>	45
2.8.3.4	<i>Lista.....</i>	46
2.8.3.5	<i>Tupla</i>	46
2.8.4	<i>Orientação a objetos</i>	46
2.8.4.1	<i>Encapsulamento</i>	48
2.8.4.2	<i>Herança</i>	49
2.8.4.3	<i>Polimorfismo.....</i>	50
2.8.5	<i>Arquivos – Geração HTML.....</i>	50
3	METODOLOGIA	51
3.1	CLASSES IMPLEMENTADAS.....	51
3.2	DETERMINAÇÃO DO PROBLEMA ESTRUTURAL	52
3.3	RESOLUÇÃO DO PROBLEMA ESTRUTURAL	53
3.3.1	<i>Características do problema estrutural</i>	54
3.3.2	<i>Vinculações dos problemas.....</i>	54
3.3.3	<i>Resolvendo problemas auxiliares.....</i>	56
4	RESULTADOS.....	61
4.1	EXEMPLO 01 – VIGA HIPERESTATICA GRAU DOIS	61
4.2	EXEMPLO 02 – VIGA HIPERESTATICA EM PREDIO UNIFAMILIAR	86
5	CONCLUSÃO	103
	REFERÊNCIAS	104

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

Com o desenvolvimento tecnológico e a necessidade de maior otimização do tempo, há uma necessidade de obter informações rápidas e precisas, principalmente na construção civil, no qual é um dos responsáveis do desenvolvimento urbano de cidades e países.

A construção civil está dividida em várias áreas, como por exemplo, pavimentação, estruturas, saneamento entre outras, na área de estruturas o responsável é o engenheiro civil, este deve dominar com precisão o dimensionamento de elementos estruturais (viga, pilar e laje) de uma construção, de modo que os mesmos consigam receber, resistir e transmitir os esforços solicitantes a estrutura (esqueleto) de forma segura, dando aos seus futuros usuários conforto e comodidade.

De acordo com Branchier (2017), é relevante a 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. Os desenvolvimentos 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).

Com as informações dispostas, este trabalho procurou com o auxílio de um software desenvolvido em linguagem Python, resolver problemas na área de estruturas estaticamente indeterminadas, mais precisamente problemas hiperestáticos em vigas contínuas sujeitos a diversas combinações de carregamentos, força 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 em disciplinas como Mecânica das Estruturas I e Mecânica das Estruturas II.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

Desenvolver um programa capaz de apresentar ao seu usuário um relatório com todos os dados apresentados, utilizados para calcular e analisar vigas hiperestáticas, como, cálculos das reações de apoios, diagrama de corpo livre, equações dos esforços normais, cortantes e momento fletor obtido a partir dos métodos das seções, e seus respectivos gráficos.

1.2.2 Objetivos Específicos

- Demonstrar o problema real, problema auxiliar e problemas virtuais;
- Desenvolver um relatório de cálculo contendo todas as equações para as reações nos apoios, seções, esforços normais, cortantes e momentos fletores;
- Apresentar os gráficos de esforço normal, momento fletor e esforço cortante;
- Validar o software por meio de exemplos de aplicações comparado a o software Ftool.

1.3 PROBLEMÁTICA E JUSTIFICATIVA

A elaboração de projeto e execução de estruturas é um campo da Engenharia Civil no qual muitos engenheiros se especializam. Estes são os chamados *engenheiros estruturais*. 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 vigas hiperestáticas.

Com o decorrer da graduação em Engenharia Civil, o discente é abordado por problemas relacionados a análises estruturais, que em sua maioria, possui um nível de complexidade, necessitando de um bom tempo para resolver o problema, ou até mesmo de difícil solução sem 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 uma viga, 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.

Com isso, será possível o desenvolvimento de um programa que traga a solução dos esforços ao seu usuário com um relatório oriundo da análise de vigas?

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 estaticamente indeterminadas dando enfoque nas estruturas denominadas por vigas hiperestáticas. Como consequência delas, são apresentadas as reações e esforços internos que nelas atuam quando submetidas a carregamentos. Ainda nessa seção é apresentada 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 vigas: 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 de vigas hiperestáticas, comparando os resultados obtidos com os apresentados pelo programa FTOOL.

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 ESTRUTURAL

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 (MARTHA, 2010). Com o passar dos séculos, diversos estudiosos, 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).

Assim podemos notar que a engenharia estrutural tem enfoque no cálculo e determinação de padrões de estruturas adequados as construções reais de obras e edificações para sempre buscar a melhora na eficiência e durabilidade.

2.2 ANÁLISE ESTRUTURAL

Conforme Soriano (2013), as estruturas são sistemas físicos formados por elementos conectados e deformáveis capazes de receber, suportar e transmitir esforços. Assim, uma estrutura ao ser construída, os seus componentes necessitam ser dimensionados para não apresentarem deformações e vibrações excessivas que prejudiquem o uso e a estética da mesma. Os princípios físicos e matemáticos envolvidos na criação da engenharia estrutural são utilizados nesta etapa (MARTHA, 2010).

Dentre esses componentes podemos Alencar as vigas, onde as mesmas podem ser submetidas a ação de diversos tipos de carregamentos. Por exemplo, quando uma carga é aplicada em uma área muito pequena, pode ser informado como uma carga concentrada ou pontual, que não passa de uma força simples (HIBBELER, 2010).

Há casos em que o carregamento está distribuído pelo eixo da viga, essas cargas são representadas como carregamentos distribuídos. Esse tipo de força é medida por sua

intensidade. Outra forma de carregamento bastante conhecido são os binários, ilustrados por momentos que atuam sobre o balanço da viga (HIBBELER, 2010).

Segundo Martha (2007), a análise estrutural moderna trabalha em quatro níveis de abstração (Figura 1), sendo que o primeiro nível remete à estrutura física, isto é, a real, o segundo nível corresponde a estrutura real com o acréscimo da idealização de seu comportamento, o terceiro nível representa o modelo discreto inserido de diversos parâmetros ao modelo estrutural e o quarto nível é a conversão do modelo discreto em modelo computacional através da implementação em softwares.

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



FONTE: Martha (2010)

2.3 ESTRUTURAS E SEUS ELEMENTOS

2.3.1 Definição de estrutura

De acordo com La Rovere e Moraes (2005) entende-se por estrutura a parte da construção responsável pela estabilidade e pela resistência a ações externas. De maneira geral estrutura submetida a ações externas deve possuir a capacidade de respeitar os requisitos de segurança quanto à ruptura dos materiais utilizados como também quanto à estabilidade global ou parcial de todos seus elementos; além disso deve demonstrar bom desempenho estrutural, no que diz respeito a deformações e durabilidade, de acordo com o fim e vida útil para a qual foi projetada.

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).

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**

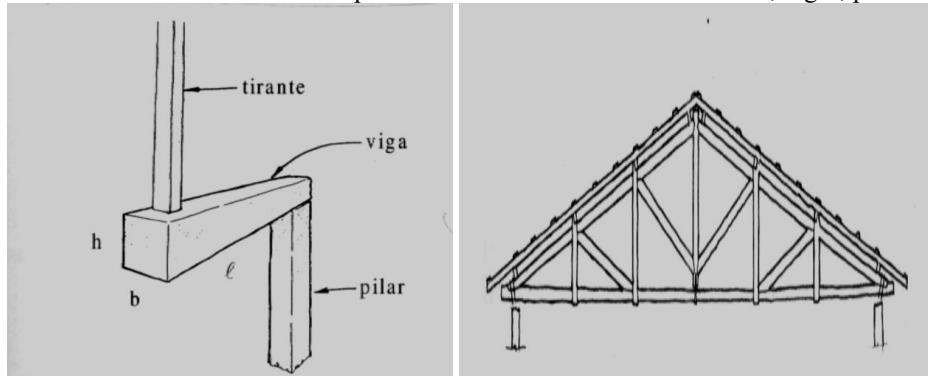
Para ALMEIDA (2009) estruturas unidimensionais podem ser denominadas de estruturas reticulares, que são aquelas compostas por elementos unidimensionais, ou seja, em que o comprimento prevalece sobre as outras duas dimensões. Exemplo: barras, vigas e pilares.

Segundo REBELLO (2000) barra tem o comprimento de uma das dimensões maior que as outras. É um elemento estrutural muito aproveitado para suspender cargas, como um cabo, para aguentar cargas como um pilar, ou vencer vãos, como as vigas. A agregação de barras pode formar sistemas estruturais complexos, em formato de treliças, pórticos e grelhas. Ver figura 1.

As vigas são elementos unidimensionais, mais encontradas de forma horizontal, onde uma de suas funções é vencer os vãos entre os pilares, servindo de apoio para as lajes, alvenarias de tijolos e algumas vezes a outras vigas. É solicitado predominantemente à flexão. Esses elementos são dimensionados para absorver esforços de cisalhamento graças seus carregamentos e de esforços de torção se suportarem platibandas ou pertencerem a estruturas tridimensionais (KAEFER, 2000).

Os pilares são unidimensionais, encontrados verticalmente, asseguram o vão vertical do pé-direito dando apoio às vigas e é exigido para compressão (KAEFER, 2000).

Figura 1 - Estruturas reticulares formadas por elementos unidimensionais: Barras, vigas, pilares e Treliças;



Fonte: Rebello (2000)

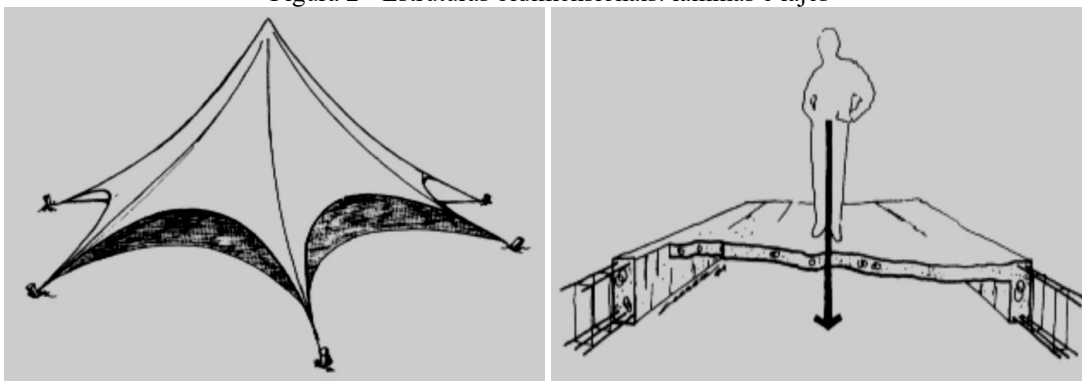
- Bidimensionais

Conforme Almeida (2009), estruturas bidimensionais são aquelas em que duas de suas dimensões prevalecem em relação a terceira. Tendo como exemplo lâminas e lajes.

Nas lâminas o seu comprimento e largura sobressaem em relação a espessura. Alguns exemplos de lâminas são as lonas e lajes. As lâminas que apresentam propriedades análogas às lonas são conhecidas por membranas, são muito finas e oferecem resistência ao seu plano. As membranas tendem a aceitar as formas dos carregamentos que atuam nelas. As lâminas que proporcionam atributos iguais às lajes são chamadas de placas. Possuem uma alta rigidez e apresentam a capacidade de vencer vãos (REBELLO, 2000)

Outro tipo de elemento estrutural são as lajes, que por sua vez, apresentam estruturas bidimensionais, volta e meia horizontais. Configura os pisos dos cômodos. É projetada para contrapor de forma direta às cargas verticais que iram agir sobre o piso, e é solicitado predominantemente à flexão. Nas suas aplicações mais simples, as lajes aliviam as cargas verticais atuantes no piso e o seu peso em vigas de apoio (KAEFER, 2000). Ver exemplo de estruturas bidimensionais (Figura 2).

Figura 2 - Estruturas bidimensionais: lâminas e lajes



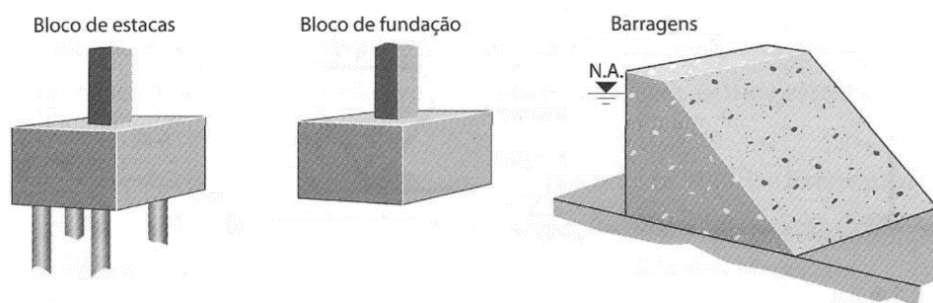
Fonte: Rebello (2000)

- 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.

Sendo considerado segundo KAEFER (2000) como elementos complementares que além dos sistemas básicos há outros elementos como os de fundação, sapatas, radiers e blocos sobre estacas. São responsáveis por distribuir a carga total da estrutura para o solo. Existe também os elementos complementares como escadas, caixas de água e muros de arrimos. Ver figura 3

Figura 3 - Estruturas tridimensionais



Fonte: Almeida (2009)

2.4 AÇÕES ATUANTES NAS ESTRUTURAS

É de extrema importância no início de qualquer obra de estruturas, estar ciente de todas as forças que estarão dispostas nos elementos estruturais na construção, pois, com essas

informações será possível determinar da maneira correta o tipo de material a ser adotado assim como os apoios que necessitarão ser utilizados, entre outras medidas que devem ser adotadas.

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 (SORIANO, 2010). 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

Ações que agem sob as estruturas de maneira externa serão divididas em ativas e reativas, onde as consideradas ativas, são provocadas por agentes do meio externo e podem ser subdividido quanto a posição, à duração, à forma de aplicação e à variação com o tempo (ALMEIDA, 2009).

De acordo com a posição elas se subdividem em fixas e móveis. As fixas são carregamentos que não variam sua posição, enquanto as móveis são o caso contrário (ALMEIDA, 2009).

Enquanto na aplicação as cargas serão distribuídas em 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). Onde posteriormente será visto a representação das forças na estrutura.

2.4.1.2 Ações externas reativas

Estes 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).

A classificação dos vínculos vem a partir do número de deslocamentos impedidos:

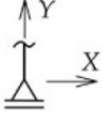
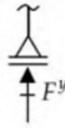
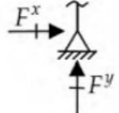
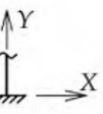
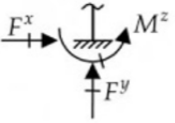
- **Apoio simples (do primeiro gênero):**
 - **Impede** a translação em uma das direções.
 - **Permite** a translação na direção perpendicular à impedida.
 - **Permite** a rotação em torno do eixo Z.

- **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.

Na figura seguinte será apresentada, os vínculos possíveis, para os modelos estruturais planos de vigas e treliças, esse primeiro sendo o objeto de estudo deste trabalho, os apoios associados a estas direções de deslocamento e as formas simplificadas de serem representados.

Figura 4 - Representação dos apoios em modelos planos de estruturas

Apoios	Símbolos	Restrições em deslocamentos e rotações	Reações de apoio
Simples do 1º gênero vertical		$\Delta^y = 0$	
Simples do 2º gênero		$\Delta^x = 0$ $\Delta^y = 0$	
Engaste (3º gênero)		$\Delta^x = 0$ $\Delta^y = 0$ $\theta = 0$	

Fonte: Martha (2010)

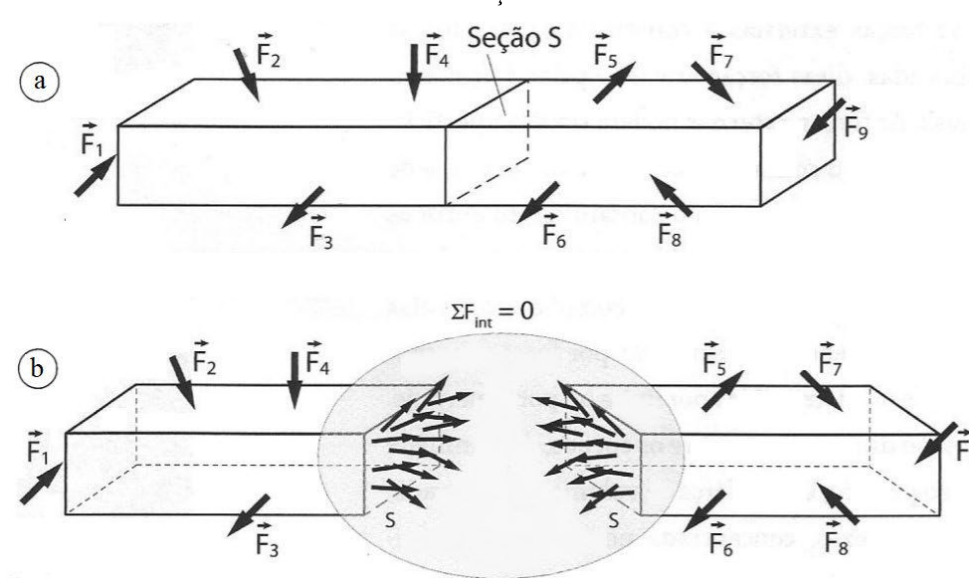
2.4.2 Esforços solicitantes internos

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 5. 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 5a. 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 5b (ALMEIDA, 2009).

Figura 5 - Representação dos esforços internos de um elemento estrutural quando submetido a um sistema de forças



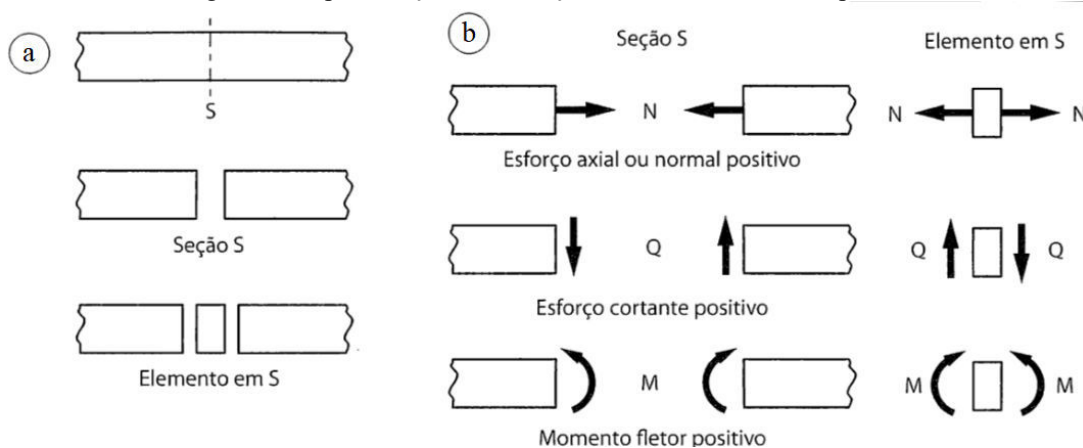
Fonte: Almeida (2009)

2.4.2.1 Esforços internos em estruturas planas

Quando as forças atuadas em uma estrutura pertencem a um mesmo plano. Assim a estrutura pode ser estudada e resolvida segundo um modelo plano apresentado 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 6a 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 6b indica os sentidos positivos desses esforços.

Figura 6 - Representação dos esforços internos em estruturas planas



Fonte: Adaptado de Almeida (2009)

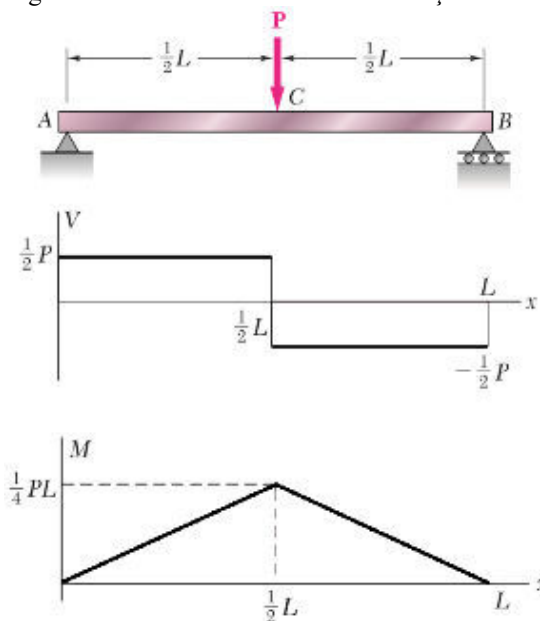
2.4.2.2 Diagramas dos esforços solicitantes internos

São gráficos que representam os 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).

Nas literaturas para questões de cálculos o diagrama dos esforços normais é identificado simplificada por DEN, o diagrama dos esforços cortantes receber a identificação de DEC e o diagrama de momento fletor é denominado por DMF.

A figura 7 apresenta um exemplo dos diagramas dos esforços internos em uma viga plana bi apoiada com carregamento pontual vertical.

Figura 7 - Diagramas ou Linhas de Estado dos esforços solicitantes internos



Fonte: Beer (2011)

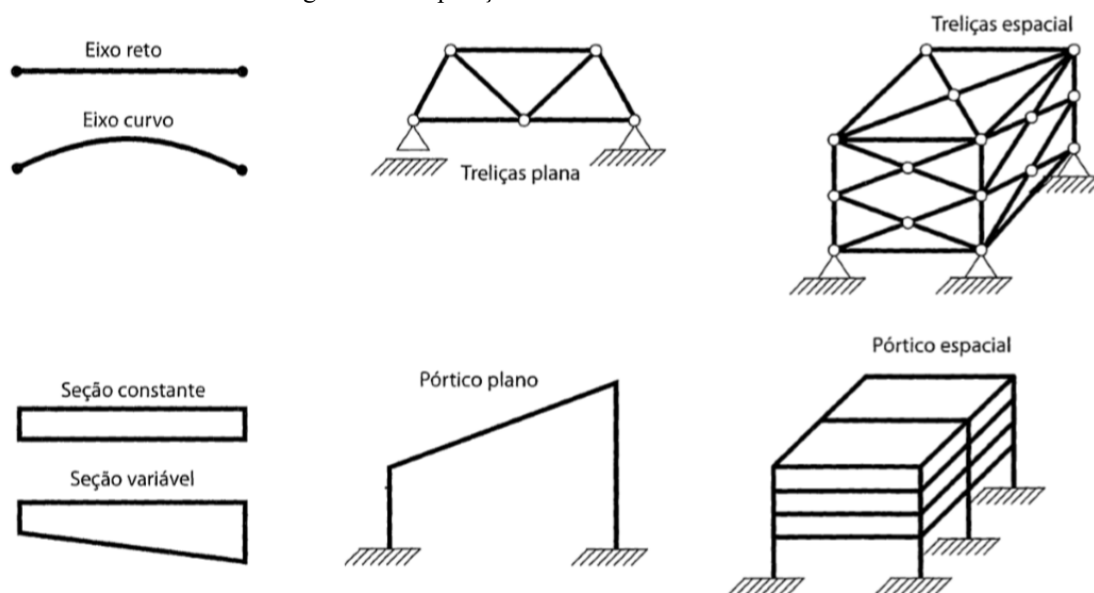
2.5 ESTRUTURAS RETICULARES

De acordo com a literatura vista, as estruturas unidimensionais ou reticulares, são formadas por elementos ou barras, com dimensões distintas, sendo, o comprimento, muito maior que a 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).

Para continuar os cálculos devemos analisar e caracterizar 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 8 e 9 seguinte apresenta um resumo do que foi discutido acima e apresenta as estruturas que podem ser formadas a partir dessas configurações.

Figura 8- Composição das estruturas reticulares



Fonte: Almeida (2009)

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	

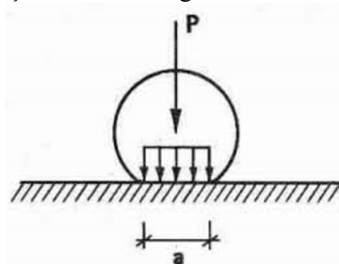
Fonte: Almeida (2009)

2.5.1.1 Cargas concentradas

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

Para efeito de demonstração, a figural 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)

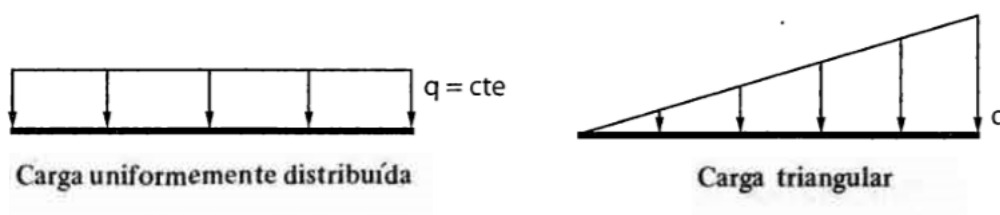
2.5.1.2 Cargas distribuídas

Além das forças pontuais, as forças e momentos podem ser representados de forma simplificada, demonstrada distribuída ao longo de um comprimento. Sendo que, nessa parte, 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).

Na figura seguinte será demonstrado cargas uniformemente distribuídas e cargas triangulares.

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



Fonte: Sussekind (1981)

2.5.2 Equilíbrio estático

Para que um corpo, sob influência de um sistema de forças, permaneça em equilíbrio, é imprescindível que elas não provoquem nenhuma deformação de rotação nem translação a este corpo, isto é, a resultante das forças e dos momentos dever ser nulos.

Quando falamos em estruturas, os apoios são os elementos que suportam e impedem os deslocamentos quando as forças ativas são submetidas à estrutura, eles geram forças e momentos reativos, que, junto com as ações ativas formam um sistema de forças em equilíbrio (ALMEIDA, 2009).

De acordo com a literatura podemos, a partir desses conceitos, formalizar equações matemáticas que são fornecidas as equações do equilíbrio estático para uma estrutura espacial nas direções x e y, conforme ilustra a figura 12.

Figura 12 - Equações do equilíbrio estático

$$\sum F_x = 0$$

$$\sum F_y = 0$$

$$\sum M = 0$$

Fonte: Vieira e Torres (2018)

2.5.3 Estabilidade e estaticidade de estruturas planas

Para determinar a estabilidade das estruturas, divididas em estáveis e instáveis, deve analisar o equilíbrio das forças ativas com as forças reativas, mantendo-a estática, quando as forças estão em equilíbrio essas são descritas como estruturas estáveis, já as instáveis ocorrem quando as forças reativas (os apoios) forem em número menor que as forças ativas, com isso, não são capazes de manter a estrutura em equilíbrio e por esse motivo estão propícias a se moverem (Almeida 2009).

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

Na primeira classificação temos a estrutura hipostáticas, essas possuem 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.

As estruturas isostáticas possuem a quantidade necessária de vínculos para que a estrutura permaneça em equilíbrio, impedindo qualquer movimentação por esforços

submetidos. Isto quer dizer que o número de reações de apoio a determinar é igual ao número de equações de equilíbrio.

No último caso temos as estruturas hiperestáticas, que essas, possuem uma quantidade de apoios maior que o necessário para impedir possíveis movimentos da estrutura, dessa forma, com a quantidade de equações menor que o número de incógnitas, será necessário, além das equações de equilíbrio, adicionar novas equações, a fim de tornar o problema possível.

Para este trabalho somente interessam as estruturas classificadas em hiperestática, uma vez que para serem analisadas devesse criar posterior ao problema real, um problema auxiliar, que, a partir dele vai gerar problemas virtuais e nesta fase bastam aplicar as equações do equilíbrio estático e um sistema simples para obter as forças necessárias para solução do problema.

2.6 VIGAS

De acordo com Martha (2010) viga é um elemento estrutural unifilar, isto é, tem um eixo bem definido. Define-se viga como um modelo estrutural cujas barras estão todas em um mesmo eixo (que pode ser inclinado ou curvo).

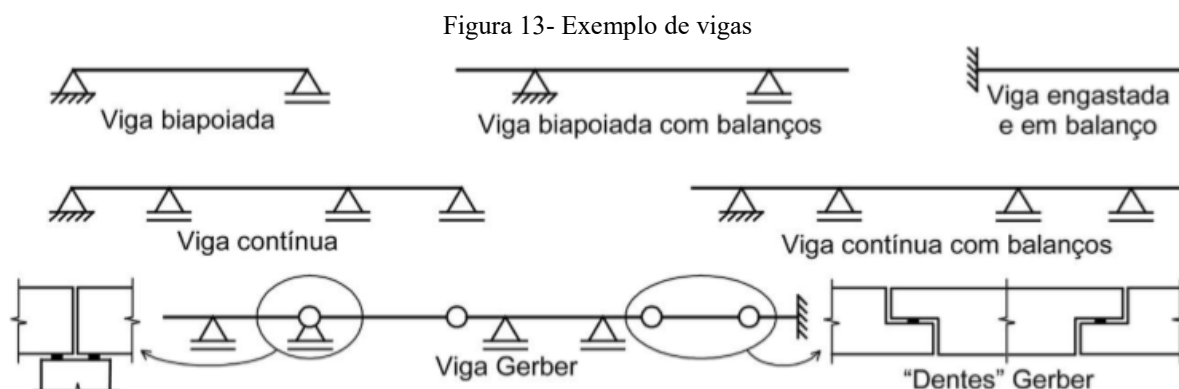
Nas palavras de Soriano (2010) viga é um modelo constituído de barras dispostas em uma mesma linha reta horizontal, sob ações que o solicita no plano vertical, de maneira que desenvolva momento fletor de vetor representativo normal a esse plano, esforço cortante vertical e, eventualmente, esforço normal. Esse último esforço ocorre quando há força externa não vertical. A distância entre dois apoios consecutivos é chamada de vão e o trecho compreendido entre esses apoios recebe o nome de tramo.

Vigas são muito importantes e utilizadas em pontes, viadutos, passarelas e em construções em geral, assim, muitas das vezes, destinadas a sustentação de outros elementos estruturais, em distribuição de esforços verticais. No caso das edificações, as vigas, além de seus pesos, recebem forças das lajes e as transmitem as colunas (Soriano, 2010).

O modelo de viga pode representar um elemento estrutural que, na estrutura real, tem ligações simples com outros elementos, como é o caso de uma viga de ponte, ou pode corresponder a um elemento estrutural que, embora esteja fortemente conectado a outros elementos, para fins de análise tem o comportamento idealizado isoladamente ao longo de um eixo. É comum analisar de forma isolada vigas de edifícios, que estão conectadas a pilares e a vigas transversais. (Martha, 2010)

2.6.1 Tipos de vigas

Um dos elementos de maior estima em projetos estruturais são as vigas. Elas são imprescindíveis em obras de todos os portes onde têm o objetivo de enviar os esforços verticais recebidos da laje para o pilar ou para transmitir uma carga concentrada, caso sirva de apoio a um pilar. Ver na figura 13 exemplo de vigas e em seguida a descrição das vigas apresentadas.



2.6.1.1 Viga simplesmente apoiada ou viga simples

As vigas podem ser rotuladas pelo modo como estão apoiadas. Por exemplo, uma viga com um apoio de pino, ou articulação, em uma extremidade e um apoio de rolete na outra é habitualmente conhecida como uma viga simplesmente apoiada ou viga simples (GERE, 2004).

2.6.1.2 Viga engastada ou viga em balanço

Uma viga engastada e em balanço se caracteriza por ter uma extremidade livre (balanço), e a outra extremidade engastada. Balanços podem ocorrer em outros tipos de viga, como biapoiada, contínua e Gerber. (Martha, 2010)

2.6.1.3 Viga em balanço ou com apoios fixos e rolete

Há também as vigas em balanço que apresentam um apoio fixo em uma extremidade e um apoio móvel em um determinado ponto do eixo da barra, com a peça se desdobrando após o suporte móvel. Para esse caso a parte depois do apoio móvel se torna análoga à viga em balanço, exceto que o eixo poderá girar no ponto onde se encontra o suporte móvel (GERE, 2004).

2.7 VIGAS HIPERESTÁTICAS

Como já foi descrito, esse tipo de viga será o modelo analisado no trabalho em questão. Usaremos o método das forças para resolver os problemas dispostos, nesse método inicialmente será necessário determinar o grau de hiperestaticidade da viga, bem como as equações de compatibilidade dos deslocamentos, fazendo com que assim retire-se os apoios excedentes e chegue à solução do problema.

2.7.1 Determinação do grau de hiperestaticidade

Segundo La Rovere e Moraes (2005) o grau de hiperestaticidade de uma estrutura pode ser externo ou interno. O grau de hiperestaticidade externo (ge) é dado por:

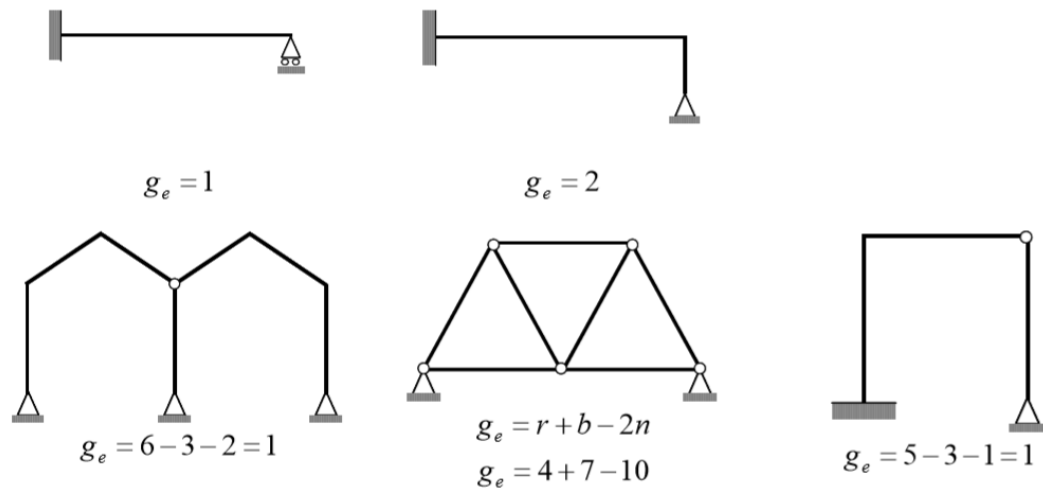
$$ge = r - e - nr$$

Sendo r o número de reações, e o número de equações da estática e nr o número de equações provenientes de rótulas. Este último é expresso por:

$$nr = b - l$$

Sendo b igual ao número de barras ligadas à rótula. O grau de hiperestaticidade interno (gi) é igual ao número de esforços internos necessários ao traçado de diagramas, conhecidas as reações. Na figura a seguir será apresentado exemplos de como calcular o grau de hiperestaticidade.

Figura 14- Exemplos de estruturas hiperestáticas



Fonte: La Rovere e Moraes (2005)

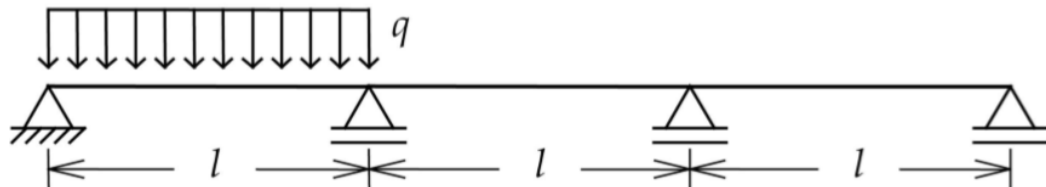
2.7.2 Método das Forças para analisar estruturas estaticamente indeterminadas

2.7.2.1 Análise de viga contínua

Para chegar ao problema auxiliar é necessário a eliminação de alguns vínculos de apoios ligados diretamente ao grau de hiperestaticidade do problema real (Martha, 2010).

Tomaremos como exemplo a figura 15 como uma viga contínua, com três vãos e uma força uniformemente distribuída abrangendo o vão da esquerda. A rigidez à flexão da viga, EI , é fornecida. Pede-se o diagrama de momentos fletores da estrutura. Nesse cálculo, não são considerados efeitos axiais (mesmo porque não existem esforços axiais na viga contínua) ou efeitos de cisalhamento na energia de deformação (Martha, 2010).

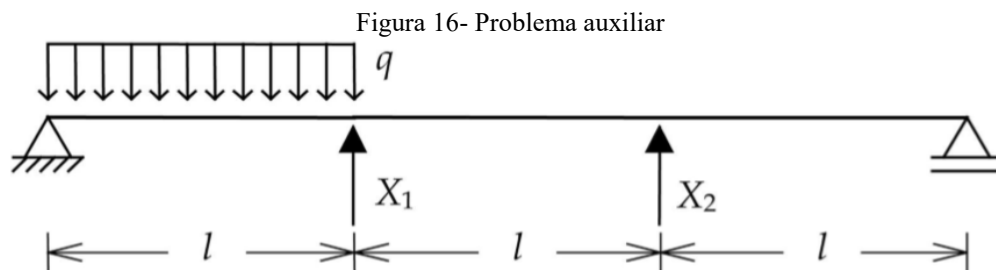
Figura 15- Viga contínua com três vãos e carregamento uniformemente distribuído no primeiro vão



Fonte: Martha (2010).

A estrutura da figura 15 apresenta uma viga de grau de hiperestaticidade $g = 2$. Para a resolução pelo método das forças, eliminaremos os vínculos externos (vínculos de apoio) para gerar o problema auxiliar necessário para solução.

Na figura a seguir demonstra a retirada dos apoios centrais e aplicação de duas forças (os hiperestáticos X_1 e X_2) serão as forças relacionadas aos vínculos.



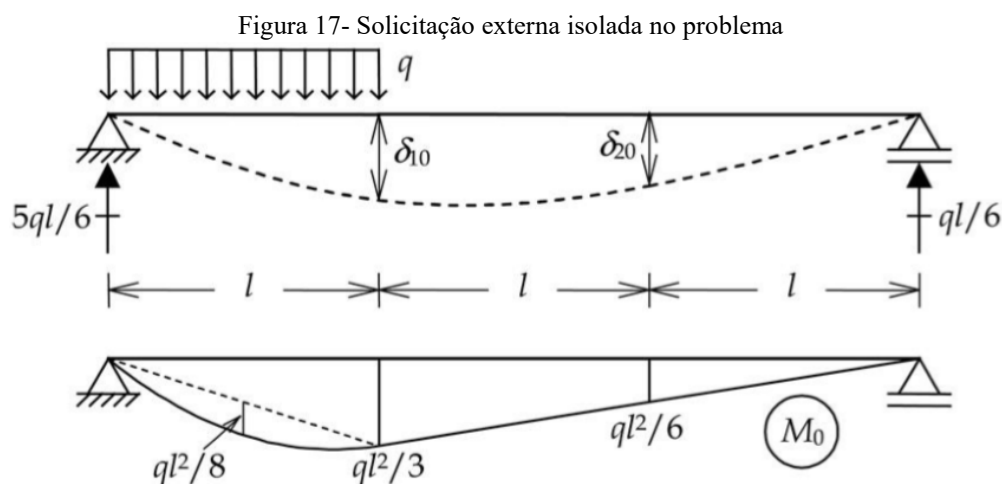
Fonte: Martha (2010).

Para chegarmos à solução é necessário determinar os valores que as reações de apoio X_1 e X_2 devem ter para que, juntamente com o carregamento atuante, os deslocamentos verticais dos pontos dos apoios eliminados sejam nulos. Com esses dados as condições de compatibilidade externas restabelecidas e eliminadas com a criação do problema auxiliar (Martha, 2010).

A partir do grau de hiperestaticidade é determinado a quantidade de casos necessários para a resolução do problema, nesse caso temos a estrutura original (problema real) é duas vezes hiperestática, fazendo com que assim gere três casos básicos, como mostrado a seguir.

Caso (0) – Solicitação externa (carregamento) isolada no problema, nesse caso somente a solicitação externa atua no problema e os valores dos hiperestáticos são nulos ($X_1 = 0$ e $X_2 = 0$).

A Figura mostra a configuração deformada do caso (0), indicando os termos de carga δ_{10} e δ_{20} e o diagrama de momentos fletores M_0 para esse caso (Martha, 2010).



Fonte: Martha (2010).

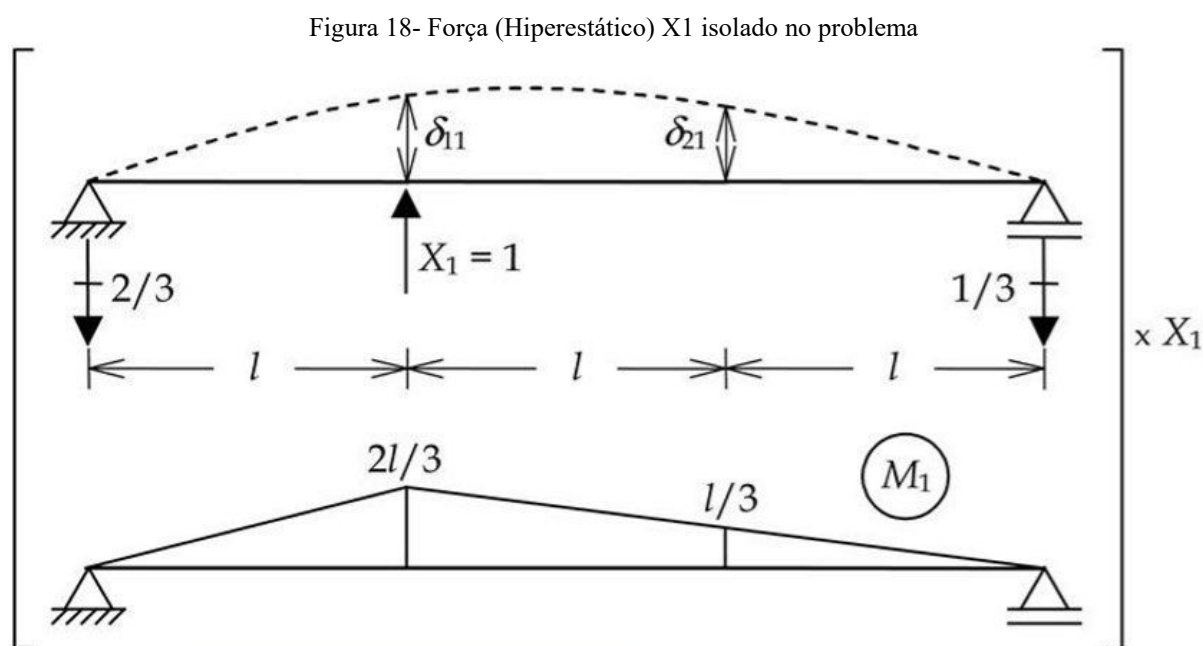
Os termos de carga no caso (0) têm a seguinte interpretação física:

δ_{10} → deslocamento vertical no ponto do apoio eliminado associado a X_1 provocado pelo carregamento externo no caso (0);

δ_{20} → deslocamento vertical no ponto do apoio eliminado associado a X_2 provocado pelo carregamento externo no caso (0).

Caso (1) – Força (Hiperestático) X_1 isolado no SP

Nesse caso, somente o hiperestático X_1 atua no problema, sem a solicitação externa e com $X_2 = 0$. Como o valor do hiperestático X_1 não é conhecido, coloca-se X_1 em evidência no caso (1), considerando-se seu efeito unitário multiplicado externamente pela incógnita X_1 , como indicado na Figura 18. A configuração deformada e o diagrama de momentos fletores do caso (1) são mostrados na figura, na qual os coeficientes de flexibilidade δ_{11} e δ_{21} estão indicados. Por definição, o diagrama de momentos fletores M_1 é para $X_1 = 1$ (Martha, 2010).



Fonte: Martha (2010).

Os coeficientes de flexibilidade no caso (1) são interpretados fisicamente como:

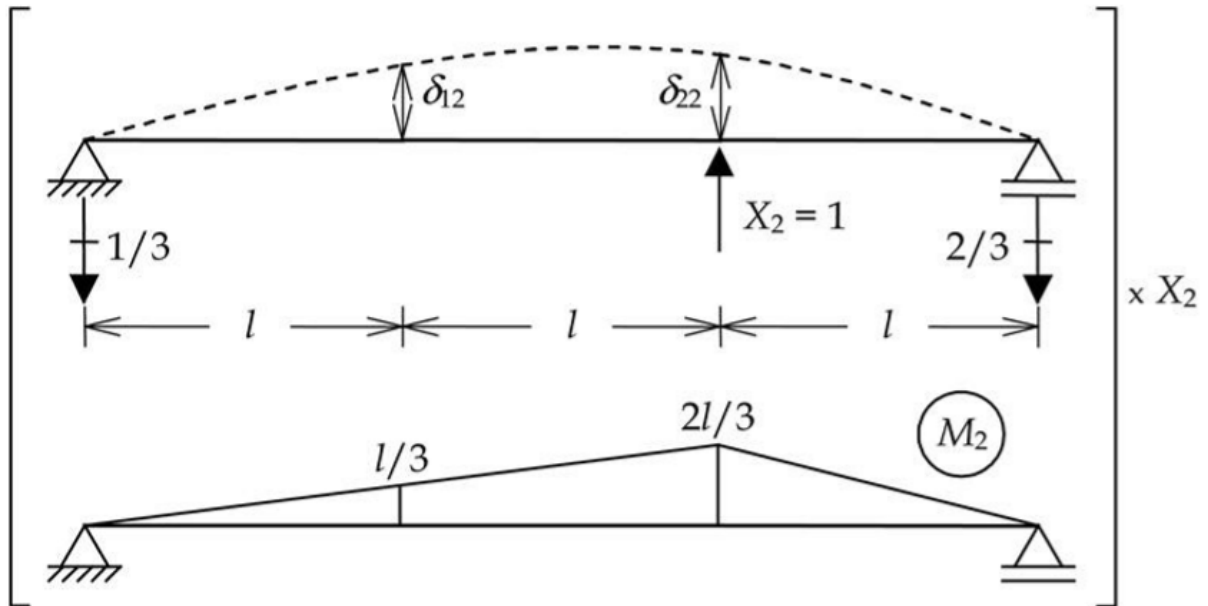
δ_{11} → deslocamento vertical no ponto do apoio eliminado associado a X_1 provocado por $X_1 = 1$ no caso (1);

δ_{21} → deslocamento vertical no ponto do apoio eliminado associado a X_2 provocado por $X_1 = 1$ no caso (1).

Caso (2) – Hiperestático X2 isolado no SP

Nesse caso, somente o hiperestático X2 atua no SP, sem a solicitação externa e com $X_1 = 0$. Analogamente ao caso (1), coloca-se X2 em evidência no caso (2). A configuração deformada e o diagrama de momentos fletores M_2 (para $X_2 = 1$) do caso (2) são mostrados na Figura 19, na qual os coeficientes de flexibilidade δ_{12} e δ_{22} estão indicados (Martha, 2010).

Figura 19- Força (Hiperestático) X2 isolado no problema



Fonte: Martha (2010).

Os coeficientes de flexibilidade no caso (2) têm a seguinte interpretação física:

δ_{12} → deslocamento vertical no ponto do apoio eliminado associado a X_1 provocado por $X_2 = 1$ no caso (2);

δ_{22} → deslocamento vertical no ponto do apoio eliminado associado a X_2 provocado por $X_2 = 1$ no caso (2).

Restabelecimento das condições de compatibilidade

Com a superposição dos três casos auxiliares criados, as condições de compatibilidade serão restabelecidas. O objetivo é restabelecer as condições impostas pelos apoios eliminados, isto é, determina-se que, na superposição, os deslocamentos verticais finais dos pontos dos apoios sejam nulos:

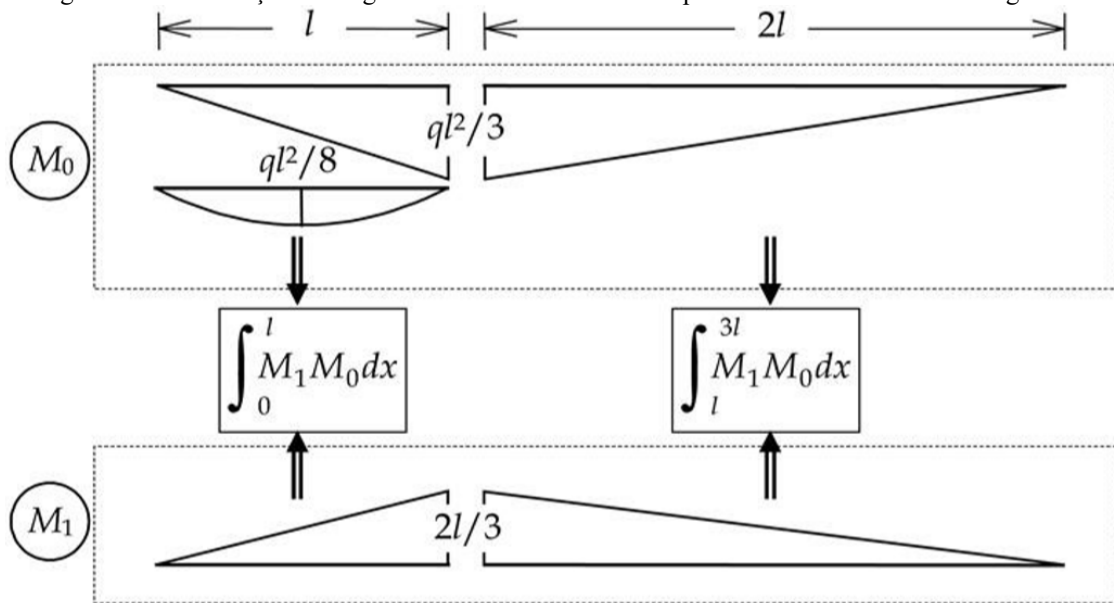
$$\begin{Bmatrix} \delta_{10} \\ \delta_{20} \end{Bmatrix} + \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

No presente exemplo da viga contínua com três vãos, para o problema adotado, os deslocamentos a serem calculados são os deslocamentos verticais nos pontos dos apoios eliminados para a criação do problema. Portanto, as cargas virtuais adotadas são forças unitárias aplicadas nesses pontos. Conforme observado anteriormente, esses sistemas correspondem justamente aos casos (1) e (2) para os hiperestáticos X_1 e X_2 com valores unitários. Dessa forma, os sistemas reais de deformação são os casos (0), (1) e (2), e os sistemas de forças virtuais são os casos (1) e (2) com $X_1 = 1$ e $X_2 = 1$, respectivamente (Martha, 2010).

Para determinar os valores de X_1 e X_2 é necessário calcular os deslocamentos nos problemas (0), (1) e (2) e as funções em cada seção das barras, esses dados são necessários para entrar na equação dos deslocamentos e obter os seus resultados, esse processo se vê necessário em cada ponto dos vínculos retirados afim de aplicar na formula matricial. As equações e figuras demonstradas respectivamente a seguir serão utilizadas para calcular os deslocamentos, e chegar ao resultado dos valores de X_1 e X_2 .

$$\delta_{10} = \frac{1}{EI} * \int_0^{3L} M_1 M_0 dx$$

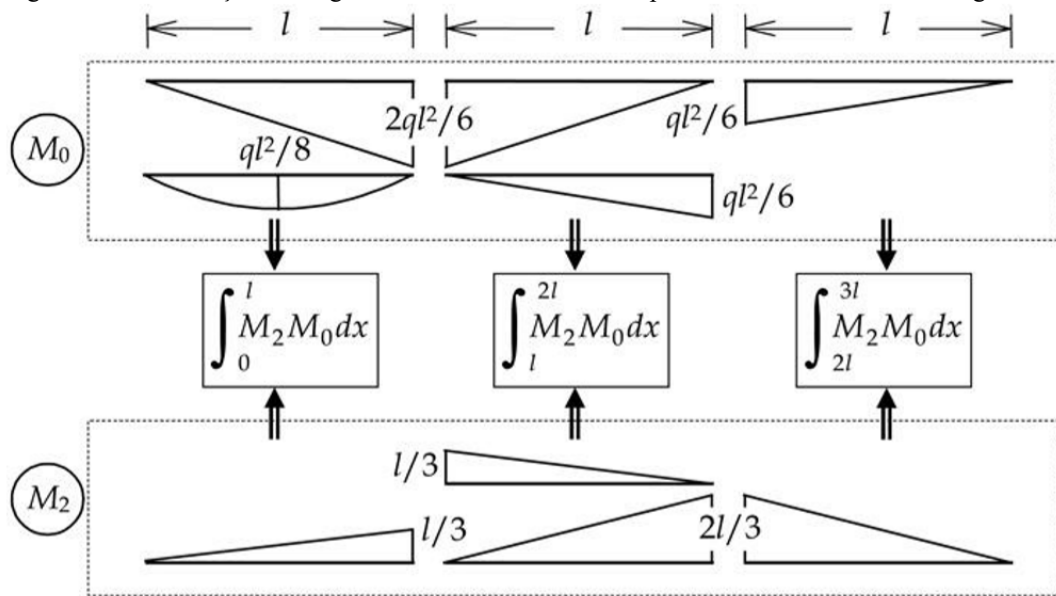
Figura 20- Combinação de diagramas de momentos fletores para o cálculo do termo de carga δ_{10}



Fonte: Martha (2010).

$$\delta_{20} = \frac{1}{EI} * \int_0^{3L} M_2 M_0 dx$$

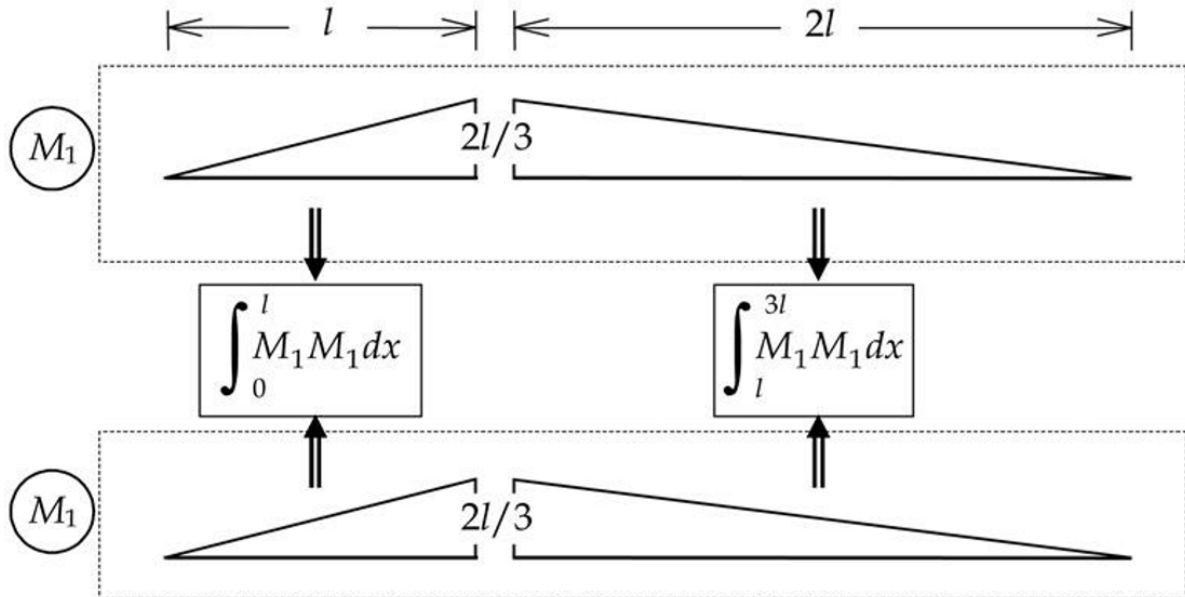
Figura 21- Combinação de diagramas de momentos fletores para o cálculo do termo de carga δ_{20}



Fonte: Martha (2010).

$$\delta_{11} = \frac{1}{EI} * \int_0^{3L} M_1 M_1 dx$$

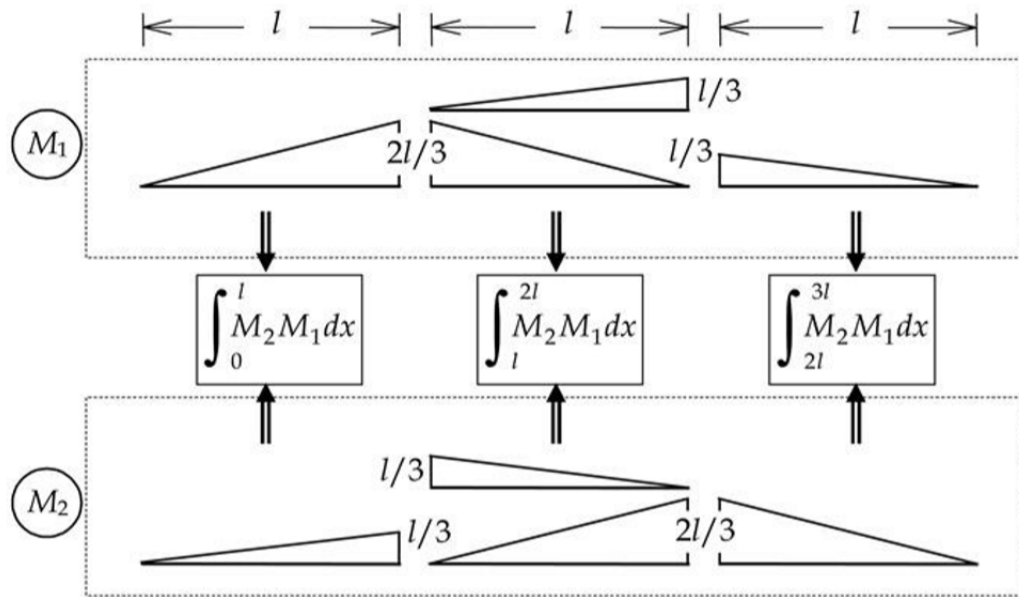
Figura 22- Combinação de diagramas de momentos fletores para o cálculo do coeficiente de flexibilidade δ_{11}



Fonte: Martha (2010).

$$\delta_{12} = \delta_{21} = \frac{1}{EI} * \int_0^{3L} M_1 M_2 dx$$

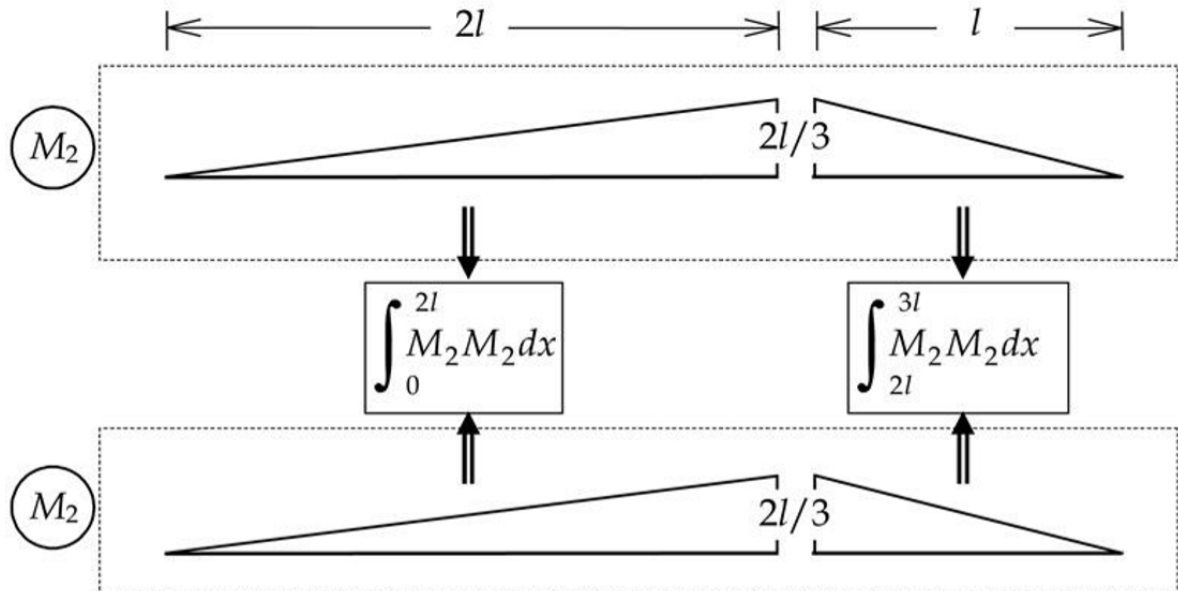
Figura 23- Combinação de diagramas de momentos fletores para o cálculo dos coeficientes de flexibilidade δ_{12} e δ_{21}



Fonte: Martha (2010).

$$\delta_{22} = \frac{1}{EI} * \int_0^{3L} M_2 M_2 dx$$

Figura 24- Combinação de diagramas de momentos fletores para o cálculo do coeficiente de flexibilidade δ_{22}



Fonte: Martha (2010).

2.8 LINGUAGEM PYTHON

2.8.1 Apresentando o Python

Linguagem criada pelo holandês Guido Van Rossum em 1991, onde foi gerada a partir de uma linguagem na época chamada de ABC, que nos dias atuais está em uso constante, pois, esse tipo de ferramenta robusta é capaz de realizar tarefas complexas com fácil manipulação e didática simples, proporcionando um melhor aprendizado.

Segundo Borges (2010), Python é uma linguagem de programação de alto nível, interpretada, imperativa, e orientada a objetos, formando assim um programa com uma sintaxe clara e concisa, favorecendo a legibilidade e a tornando mais produtiva.

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.

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).

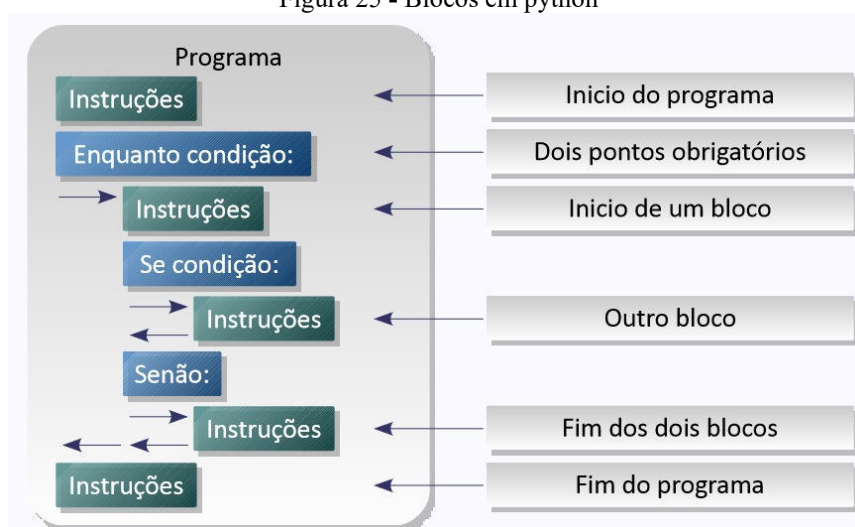
2.8.2 Conceitos básicos

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 25 (BORGES, 2010).

Figura 25 - Blocos em python



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).

2.8.3 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).

2.8.3.1 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 26.

Figura 26 - Exemplo de variáveis do tipo strings

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

Fonte: Autoria Própria (2019)

2.8.3.2 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)

2.8.3.3 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)

2.8.3.4 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 27 - Representação de uma lista

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

Fonte: Borges (2010)

2.8.3.5 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 28, sendo os parênteses opcionais.

Figura 28 - Representação de uma lista

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

Fonte: Borges (2010)

2.8.4 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, a fim 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.

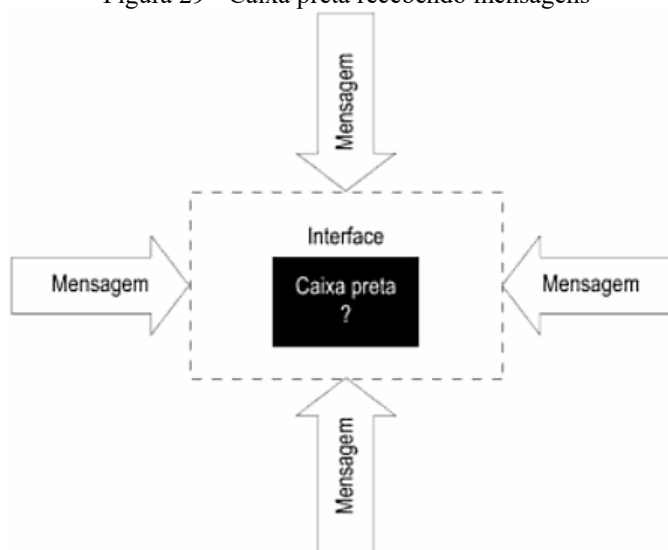
Fonte: Chagas e Oliveira (2009)

2.8.4.1 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 30). 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 29 - Caixa preta recebendo mensagens

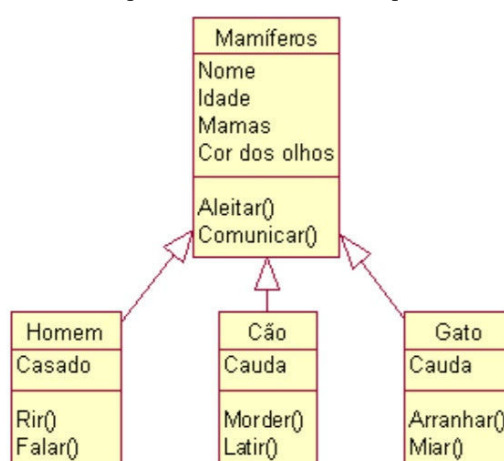


Fonte: Goetten e Winck (2006)

2.8.4.2 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 30 - Esquema didático de hierarquia das classes



Fonte: Leite e Rahal (2006)

A figura 31 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.

2.8.4.3 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.

2.8.5 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 31 - 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)

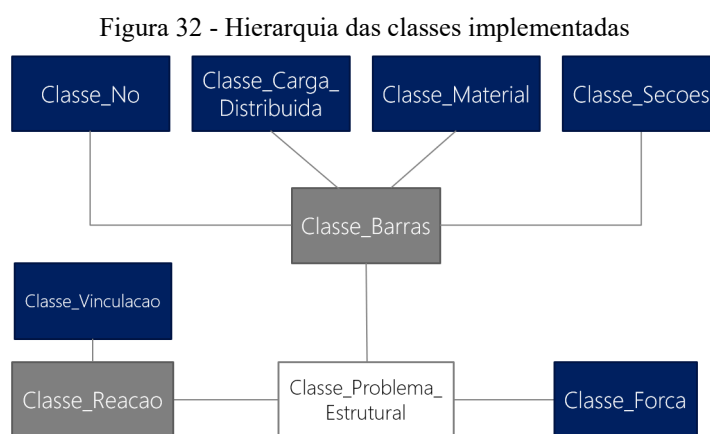
3 METODOLOGIA

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

Em seguida, utilizando a linguagem Python e algumas das suas bibliotecas (Matplotlib, Numpy) realizou-se a implementação de elementos em duas dimensões para análise estrutural de vigas hiperestáticas. Tais implementações foram realizadas no ambiente de desenvolvimento Visual Studio Code, 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 33). 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 34 ilustra o Visual Studio Code com a implementação em Python da classe problema estrutural, onde é possível verificar todas as outras classes sendo importadas para dentro dela e proporcionando as operações dentro do código.

Figura 33 - Classe problema estrutural dentro do Visual Studio Code

```

from Classe_Forca import Forca
from Classe_Materiais import Material
from Classe_No import No
from Classe_Reacao import Reacao
from Classe_Secoos import Secoos
from Classe_Vinculacao import Vinculacao
from Classe_Barras import Barras
from Classe_Carga_Distribuida import Carga_Distribuida
import math
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Circle, Wedge, Polygon
from matplotlib.collections import PatchCollection
from mailbox import MMDFMessage
from signal import signal
from sympy import solve, Symbol
from mailcap import show
from numpy import sign, subtract, mat
import mpl_toolkits.axisartist.floating_axes as floating_axes
from matplotlib.transforms import Affine2D
from builtins import eval
from math import asin

```

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:

- Entre com o comprimento do vão;
- Entre com o valor da carga distribuída do vão;
- Entre com o valor força concentrada do vão;
- Entre com a posição da força concentrada do vão;

A partir das informações fornecidas, o programa vai armazenar as informações fornecidas nas suas respectivas variáveis para posteriormente resolver o problema (equação dos esforços internos e seus respectivos diagramas). A figura 34, apresenta o código disposto pelo usuário para fazer as perguntas e armazenar os dados.

Figura 34 - Dados de entradas e variáveis

```

1  # -*- coding: utf8 -*-
2  from Classe_Problema_Estrutural import Problema_Estrutural
3  from sympy import *
4  from sympy.integrals import Integral
5  from sympy.parsing.sympy_parser import parse_expr
6  import numpy as np
7
8  nvaos = int(input('Entre com o numero de vaos:'))
9  comp = []
10 carg = []
11 forcas = []
12 pos_forca = []
13 for i in range(nvaos):
14     l = float(input('Entre com o comprimento do vao %d:' % (i+1)))
15     comp.append(l)
16     q = float(input('Entre com o valor da forza distribuida do vao %d:' % (i+1)))
17     carg.append(q)
18     p = float(input('Entre com o valor da forza concentrada do vao %d:' % (i+1)))
19     forcas.append(p)
20     d = float(input('Entre com a posicao da forza concentrada do vao %d:' % (i+1)))
21     pos_forca.append(d)

```

Fonte: Autoria Própria (2019)

3.3 RESOLUÇÃO DO PROBLEMA ESTRUTURAL

Com os dados de entrada fornecidos, o programa inicia sua análise para ver o grau de hiperestaticidade, com isso, define-se o problema estrutural real, problema auxiliar e os demais problemas ligados ao grau de hiperestaticidade. A figura 35 apresenta a função que corresponde aos problemas a serem analisados.

Figura 35 - Código de armazenamento para problemas analisados

```

problema = Problema_Estrutural()
problema_aux = Problema_Estrutural()
problema_0 = Problema_Estrutural()
problemas = []
problema_real = Problema_Estrutural()

for i in range(nvaos-1):
    problemas.append(Problema_Estrutural())

nnos = 1
x_nos = [0]
for vao in range(nvaos):
    if pos_forca[vao] != 0:
        x_nos.append(x_nos[nnos-1]+pos_forca[vao])
        nnos = nnos + 1
x_nos.append(x_nos[nnos-1]+comp[vao]-pos_forca[vao])
nnos = nnos + 1

```

Fonte: Autoria Própria (2019)

3.3.1 Características do problema estrutural

A partir dos parâmetros armazenados anteriormente, o programa irá criar todas as características para resolver o problema, como, quantidade e posição de nós, material e seção, quantidade de barras, cargas distribuídas e forças concentradas.

Figura 36- Parte do código para caracterizar o problema

```
# Criando os Nós
for i in range(nnos):
    letra = letras[i]
    xno = x_nos[i]
    problema.adicionar_no(id = letra, x = xno, y = 0)
    problema_aux.adicionar_no(id = letra, x = xno, y = 0)
    problema_0.adicionar_no(id = letra, x = xno, y = 0)
    for j in range(nvaos-1):
        problemas[j].adicionar_no(id = letra, x = xno, y = 0)
    problema_real.adicionar_no(id = letra, x = xno, y = 0)

# Criando Material e Seção
problema.adicionar_material(id = 'Mat01', young = 1.0, poison = 0.1, alfa = 1.0, peso_esp = 1.0, tensao_adm = 1.0)
problema_aux.adicionar_material(id = 'Mat01', young = 1.0, poison = 0.1, alfa = 1.0, peso_esp = 1.0, tensao_adm = 1.0)
problema_0.adicionar_material(id = 'Mat01', young = 1.0, poison = 0.1, alfa = 1.0, peso_esp = 1.0, tensao_adm = 1.0)
problema_real.adicionar_material(id = 'Mat01', young = 1.0, poison = 0.1, alfa = 1.0, peso_esp = 1.0, tensao_adm = 1.0)
for j in range(nvaos-1):
    problemas[j].adicionar_material(id = 'Mat01', young = 1.0, poison = 0.1, alfa = 1.0, peso_esp = 1.0, tensao_adm = 1.0)

problema.adicionar_secao(id = 'Sec01', area = 1.0, inercia = 1.0)
problema_aux.adicionar_secao(id = 'Sec01', area = 1.0, inercia = 1.0)
problema_0.adicionar_secao(id = 'Sec01', area = 1.0, inercia = 1.0)
for j in range(nvaos-1):
    problemas[j].adicionar_secao(id = 'Sec01', area = 1.0, inercia = 1.0)
problema_real.adicionar_secao(id = 'Sec01', area = 1.0, inercia = 1.0)

# Criando as Barras
for i in range(nnos-1):
    ltr_noi = letras[i]
    ltr_nof = letras[i+1]
    problema.adicionar_barra(ltr_noi, ltr_nof, 'Mat01', 'Sec01')
    problema_aux.adicionar_barra(ltr_noi, ltr_nof, 'Mat01', 'Sec01')
    problema_0.adicionar_barra(ltr_noi, ltr_nof, 'Mat01', 'Sec01')
    for j in range(nvaos-1):
        problemas[j].adicionar_barra(ltr_noi, ltr_nof, 'Mat01', 'Sec01')
    problema_real.adicionar_barra(ltr_noi, ltr_nof, 'Mat01', 'Sec01')
```

Fonte: Autoria Própria (2019)

3.3.2 Vinculações dos problemas

Para solução do problema é necessário a determinação dos vínculos existentes em cada problema disposto. Figura 37, 38 e 39 mostra o código que insere as vinculações nos problemas.

Figura 37- Código para inserir vinculações no problema real

```

nnos=0
problema.adicionar_vinculacao(id = 'Vic01', rest_x = 1, rest_y = 1, rest_r = 0)
problema.adicionar_vinculacao(id = 'Vic02', rest_x = 0, rest_y = 1, rest_r = 0)
for i in range(nvaos):
    if i == 0:
        problema.atribuir_vinculacao(id_no = 'A', id_vinculacao = 'Vic01')
        if pos_forca[i] != 0:
            nnos = nnos + 1
    elif i == (nvaos-1):
        id = letras[nnos]
        problema.atribuir_vinculacao(id_no = id, id_vinculacao = 'Vic02')
        if pos_forca[i] != 0:
            nnos = nnos + 1
        id = letras[nnos+1]
        problema.atribuir_vinculacao(id_no = id, id_vinculacao = 'Vic02')
    else:
        id = letras[nnos]
        problema.atribuir_vinculacao(id_no = id, id_vinculacao = 'Vic02')
        if pos_forca[i] != 0:
            nnos = nnos + 1
    nnos = nnos + 1

```

Fonte: Autoria Própria (2019)

Figura 38-Código para inserir vinculações no problema auxiliar

```

nnos=0
problema_aux.adicionar_vinculacao(id = 'Vic01', rest_x = 1, rest_y = 1, rest_r = 0)
problema_aux.adicionar_vinculacao(id = 'Vic02', rest_x = 0, rest_y = 1, rest_r = 0)
problema_aux.adicionar_forca(id = 'Fn', fx = 0.0, fy = 999 , m = 0.0)
for i in range(nvaos):
    if i == 0:
        problema_aux.atribuir_vinculacao(id_no = 'A', id_vinculacao = 'Vic01')
        if pos_forca[i] != 0:
            nnos = nnos + 1
    elif i == (nvaos-1):
        id = letras[nnos]
        problema_aux.atribuir_forca(id_no = id, id_forca = 'Fn')
        if pos_forca[i] != 0:
            nnos = nnos + 1
        id = letras[nnos+1]
        problema_aux.atribuir_vinculacao(id_no = id, id_vinculacao = 'Vic02')
    else:
        id = letras[nnos]
        problema_aux.atribuir_forca(id_no = id, id_forca = 'Fn')
        if pos_forca[i] != 0:
            nnos = nnos + 1
    nnos = nnos + 1

```

Fonte: Autoria Própria (2019)

Figura 39-Código para inserir vinculações no problemas virtuais

```

problema_0.adicionar_vinculacao(id = 'Vic01', rest_x = 1, rest_y = 1, rest_r = 0)
problema_0.adicionar_vinculacao(id = 'Vic02', rest_x = 0, rest_y = 1, rest_r = 0)
problema_0.atribuir_vinculacao(id_no = 'A', id_vinculacao = 'Vic01')
id = letras[nnos]
problema_0.atribuir_vinculacao(id_no = id, id_vinculacao = 'Vic02')

nnos=0
for i in range(nvaos-1):
    problemas[i].adicionar_vinculacao(id = 'Vic01', rest_x = 1, rest_y = 1, rest_r = 0)
    problemas[i].adicionar_vinculacao(id = 'Vic02', rest_x = 0, rest_y = 1, rest_r = 0)
    problemas[i].adicionar_forca(id = 'Fn', fx = 0.0, fy = 1, m = 0.0)

for i in range(nvaos):
    if i == 0:
        for j in range(nvaos-1):
            problemas[j].atribuir_vinculacao(id_no = 'A', id_vinculacao = 'Vic01')
        if pos_forca[i] != 0:
            nnos = nnos + 1
    elif i == (nvaos-1):
        id = letras[nnos]
        problemas[i-1].atribuir_forca(id_no = id, id_forca = 'Fn')
        if pos_forca[i] != 0:
            nnos = nnos + 1
        id = letras[nnos+1]
        for j in range(nvaos-1):
            problemas[j].atribuir_vinculacao(id_no = id, id_vinculacao = 'Vic02')
    else:
        id = letras[nnos]
        problemas[i-1].atribuir_forca(id_no = id, id_forca = 'Fn')
        if pos_forca[i] != 0:
            nnos = nnos + 1
    nnos = nnos + 1

```

Fonte: Autoria Própria (2019)

3.3.3 Resolvendo problemas auxiliares

De acordo com os dados de entrada que determinam o grau de hiperestaticidade, o programa agora tem todos os problemas desenhados, gerando assim, o problema real hiperestático, o problema auxiliar, e os problemas virtuais isostáticos, a partir desses problemas virtuais vem a resolução do problema real. O primeiro passo é gerar o DCL dos problemas virtuais para seguir com os cálculos. A figura 40 mostra parte do código para gerar o DCL.

Figura 40 - Código da função geradora do DLC

```

def desenhar_dcl(self, arq = 'G01.png'):
    x_max = self.barras[len(self.barras)-1].no_f.x + 1
    fig, ax = plt.subplots()
    fig.set_size_inches(19, 8)
    vmax = 0.0

    for barra in self.barras:

        if barra.carga_distribuida != None:
            vaux = barra.carga_distribuida.fyi

            if abs(vmax) < abs(vaux):
                vmax = vaux

            vaux = barra.carga_distribuida.fyf
            if abs(vmax) < abs(vaux):
                vmax = vaux

        barra.desenhar_barra(ax)
    for id, no in self.nos.items():
        no.desenhar(ax, id)

    for id, no in self.nos.items():
        no.desenhar_forca(ax)

    for barra in self.barras:
        barra.desenhar_carga_distribuida_vertical(ax, vmax)

    for barra in self.barras:
        barra.desenhar_carga_distribuida_horizontal(ax)

    for id, no in self.nos.items():
        no.desenhar_reacoes(ax, id, x_max + 1)

    ax.set_xlim(-2, 32)
    ax.set_ylim(-2, 8)
    #ax.set_aspect('equal')

```

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

$$\Sigma F_x = 0$$

$$\Sigma F_y = 0$$

$$\Sigma M_o = 0$$

Tomando o ponto inicial “A” como o escolhido para a realização do cálculo do momento. A figura 41 expõe trechos do código escrito.

Figura 41 - Código da função calcular reações

```
def desenhar_calcular_reacoes(self, arq = 'H01.png'):
    fs = 0.5
    sfh = ''
    sfv = ''
    sm = ''
    simbolos = []

    for id, no in self.nos.items():
        if no.vinculacao:
            if no.vinculacao.rest_x:
                sfh = sfh + str('+H%s' % id)
                simbolos.append('H%s' % id)

            if no.vinculacao.rest_y:
                sfv = sfv + str('+V%s' % id)
                dist = no.x - self.nos['A'].x
                sm = sm + str('+V%s * %.5f' % (id,dist))
                simbolos.append('V%s' % id)

            if no.vinculacao.rest_r:
                sm = sm + str('-M%s' % id)
                simbolos.append('M%s' % id)

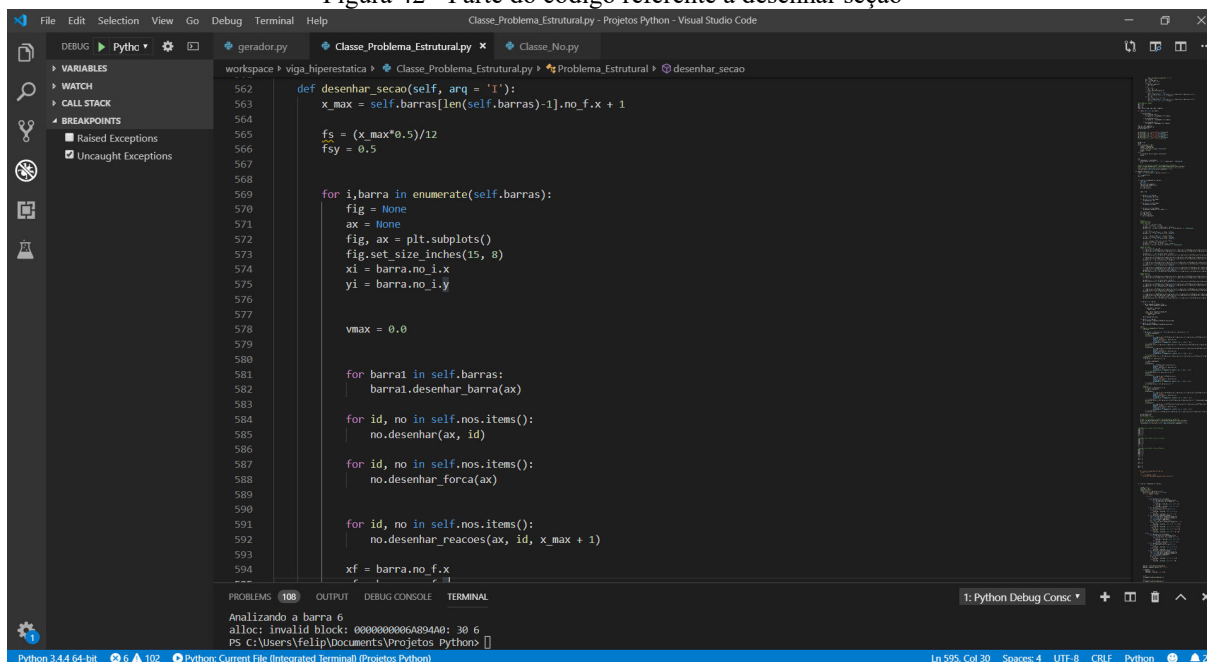
    for id, no in self.nos.items():
        if no.forca:
            if no.forca.fx > 0:
                dist = no.y - self.nos['A'].y
                sfh = sfh + str('+.5f' % no.forca.fx)
                if dist > 0:
                    sm = sm + str('-(.5f * %.5f)' % (no.forca.fx,dist))
                elif dist < 0:
                    sm = sm + str('+(.5f * %.5f)' % (-no.forca.fx,dist))

            elif no.forca.fx < 0:
                dist = no.y - self.nos['A'].y
```

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 da viga e sua posição. Utilizando o método das seções para determinar as funções necessárias para cálculos dos gráficos o eixo do objeto 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 e dos trechos que deverão ser seccionados para a resolução de suas equações de esforços internos. Figura 42 descreve parte do código “desenhar Seção”.

Figura 42 - Parte do código referente a desenhar seção



Fonte: Autoria Própria (2019)

Posterior aos seccionamentos realizados ao longo da viga, 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.

No fim da resolução dos problemas virtuais 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.

A partir desses números e com funções matriciais pode-se descobrir as forças (F_n) correspondentes ao número de apoios retirados, tornando a viga isostática, para resolução do problema real, nessa parte o programa repete o processo para resolver o problema real com as novas cargas encontradas e assim chegando a solução do problema.

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 43 pode-se visualizar parte do código responsável por gerar o arquivo do relatório.

Figura 43 – Parte do código gerador do arquivo HTML

```

2702     def gerar_relatorio_vc(self, nvaos = 3):
2703         ntrx = len(self.barras)
2704         k = 0
2705         texto = '<p><strong>PROBLEMA ESTRUTURAL:</strong></p>'
2706         texto = texto + '<p><strong></strong></p>'
2707         texto = texto + '<p><strong>PROBLEMA AUXILIAR:</strong></p>'
2708         texto = texto + '<p><strong></strong></p>'
2709         texto = texto + '<p><strong>PROBLEMA 0:</strong></p>'
2710         texto = texto + '<p><strong></strong></p>'
2711         for i in range(nvaos-1):
2712             texto = texto + '<p><strong>PROBLEMA %d:</strong></p>' % (i+1)
2713             texto = texto + '<p><strong></strong></p>' % (i+4)
2714         for i in range(nvaos):
2715             texto = texto + '<p><strong>RESOLUCAO PROBLEMA %d:</strong></p>' % (i)
2716             texto = texto + '<p><strong></strong></p>' % (i+3)
2717             texto = texto + '<p><strong>DCL:</strong></p>'
2718             texto = texto + '<p><strong></strong></p>' % (i+3)
2719             texto = texto + '<p><strong>CALCULO DAS REACOES:</strong></p>'
2720             texto = texto + '<p><strong></strong></p>' % (i+3)
2721             for j in range(len(self.barras)):
2722                 texto = texto + '<p><strong>* TRECHO 0%d: %5f m &nbsp;&nbsp;&nbsp;&nbsp;& X &nbsp;&nbsp;&nbsp;&nbsp;& X &nbsp;&nbsp;&nbsp;&nbsp;& %5f m</strong></p>' \
2723                     % (j+1, self.barras[j].no_i.x, self.barras[j].no_f.x)
2724                 texto = texto + '<p><strong></strong></p>' % (k, (j+1))
2725                 texto = texto + '<p><strong></strong></p>' % (k, (j+intrx+1))
2726                 k = k + 1
2727             texto = texto + '<p><strong>DIAGRAMAS DE ESFOR&cedil;OS SOLICITANTES:</strong></p>'
2728             texto = texto + '<p><strong></strong></p>' % (i, (2+2*ntrx))
2729             texto = texto + '<p><strong></strong></p>' % (i, (3+2*ntrx))
2730             texto = texto + '<p><strong></strong></p>' % (i, (4+2*ntrx))
2731         texto = texto + '<p><strong>PROBLEMA ESTRUTURAL:</strong></p>'
2732         texto = texto + '<p><strong></strong></p>'
2733         texto = texto + '<p><strong>DCL:</strong></p>'
2734         texto = texto + '<p><strong></strong></p>'
2735         texto = texto + '<p><strong>CALCULO DAS REACOES:</strong></p>'
2736         texto = texto + '<p><strong></strong></p>'
2737         for j in range(len(self.barras)):

```

Fonte: Autoria Própria (2019)

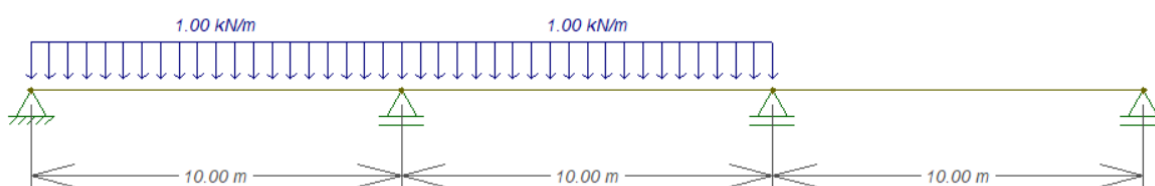
4 RESULTADOS

Para efetuar a validação do programa desenvolvido, serão executados dois exemplos. O primeiro está presente em conteúdo acadêmico, enquanto o terceiro é uma simulação baseada em parâmetros de uma estrutura real.

4.1 EXEMPLO 01 – VIGA HIPERESTÁTICA GRAU DOIS

O primeiro exemplo é um exemplo montado em um programa de cálculo estrutural (ftool). Esse exemplo apresenta um carregamento retangular distribuído de cima para baixo sobre o elemento horizontal da estrutura com uma carga de 1 kN/m , as dimensões se encontram na figura 44 abaixo.

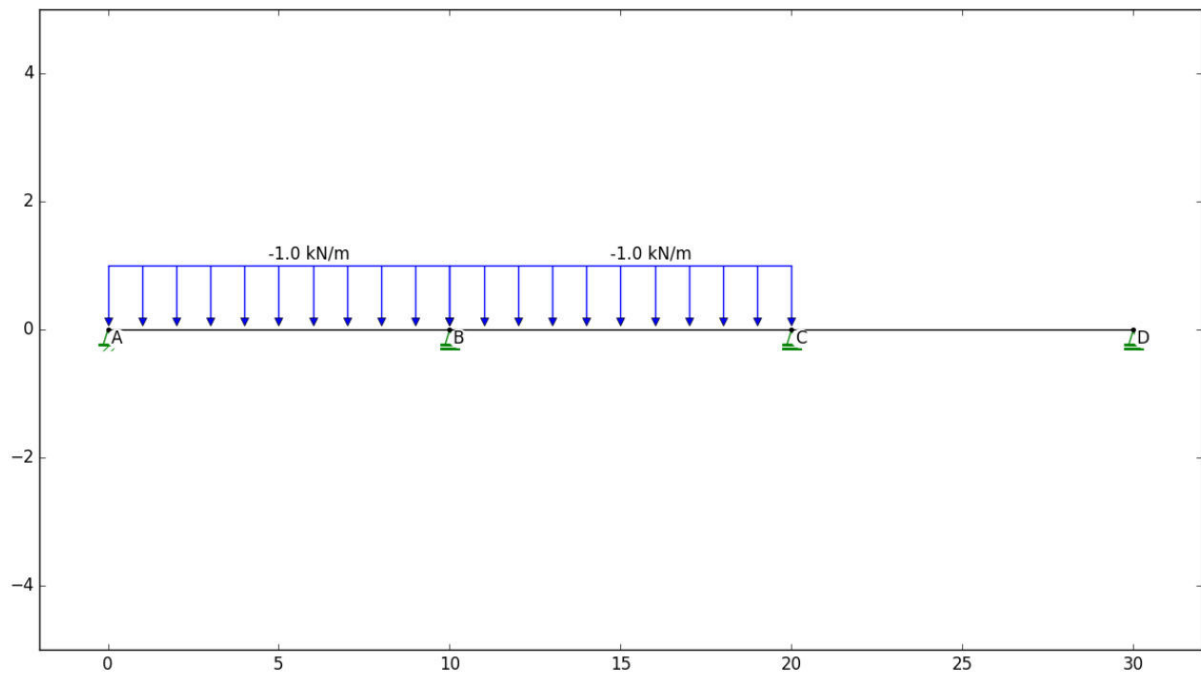
Figura 44 – Viga hiperestática com carga uniforme - ftool



Fonte: Ftool(2019)

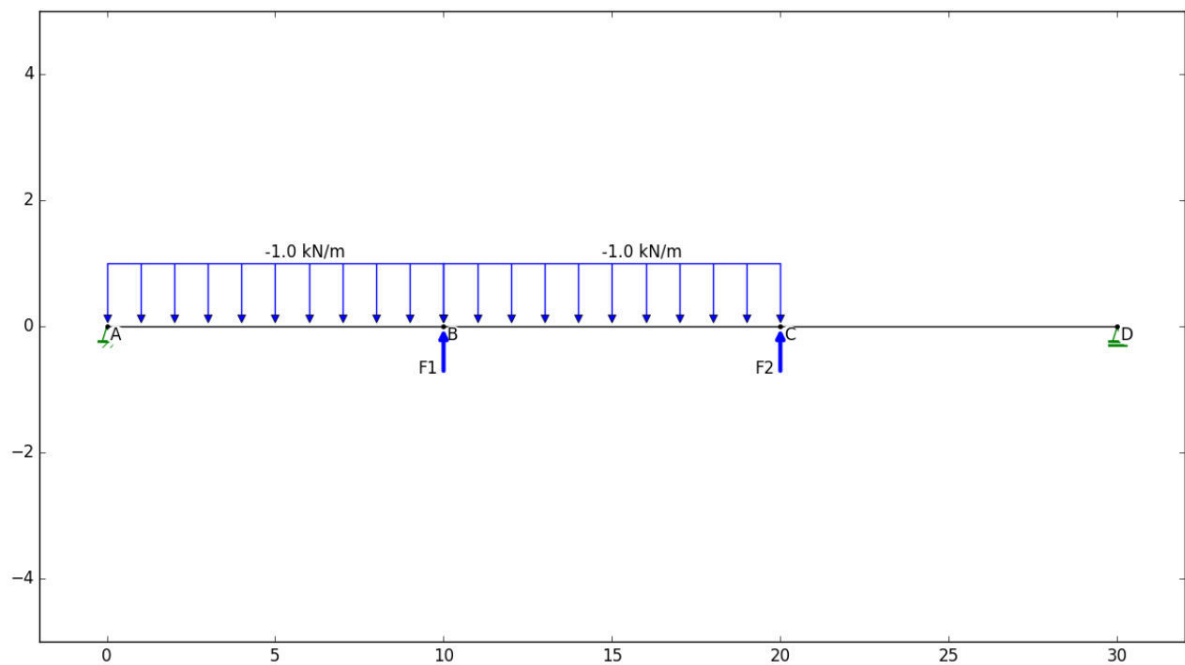
Para resolver o problema inicialmente é necessário apresentar o desenho do problema real, problema auxiliar e problemas virtuais ligados ao grau de hiperestaticidade. Esses problemas são gerados a partir dos dados fornecidos pelo usuário (ver figuras 45, 46, 47, 48 e 49).

Figura 45- Exemplo 01 - Problema Estrutural Real



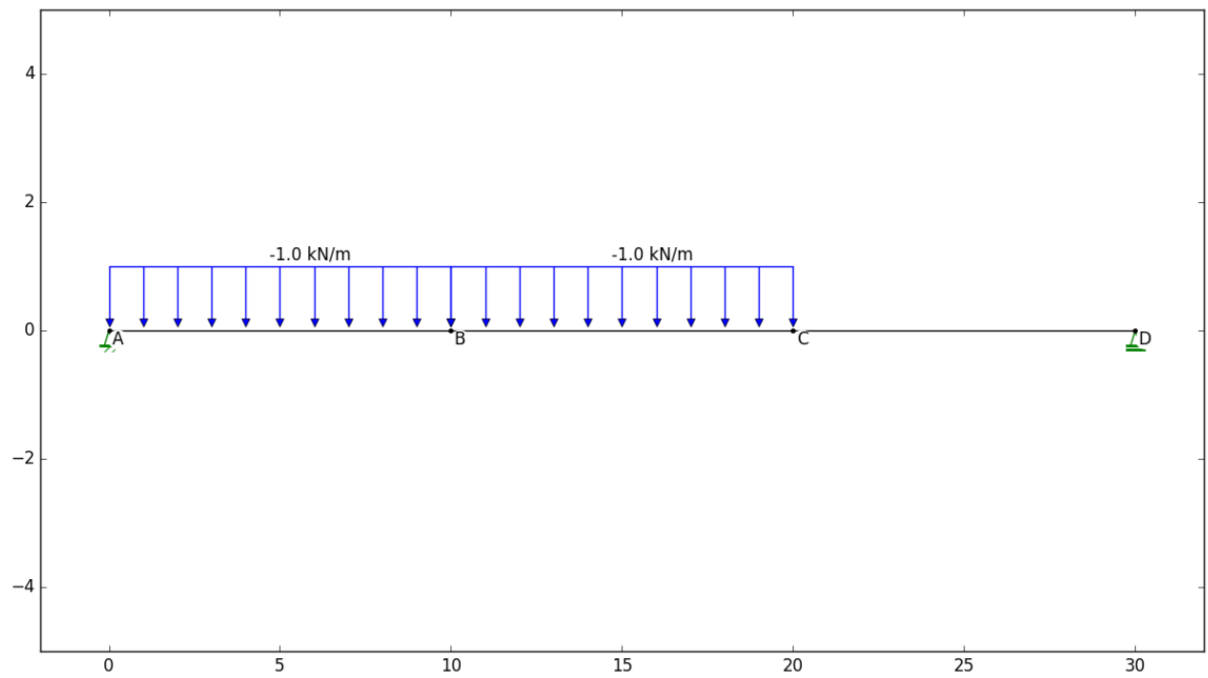
Fonte: Autoria Própria (2019)

Figura 46- Exemplo 01 - Problema Auxiliar



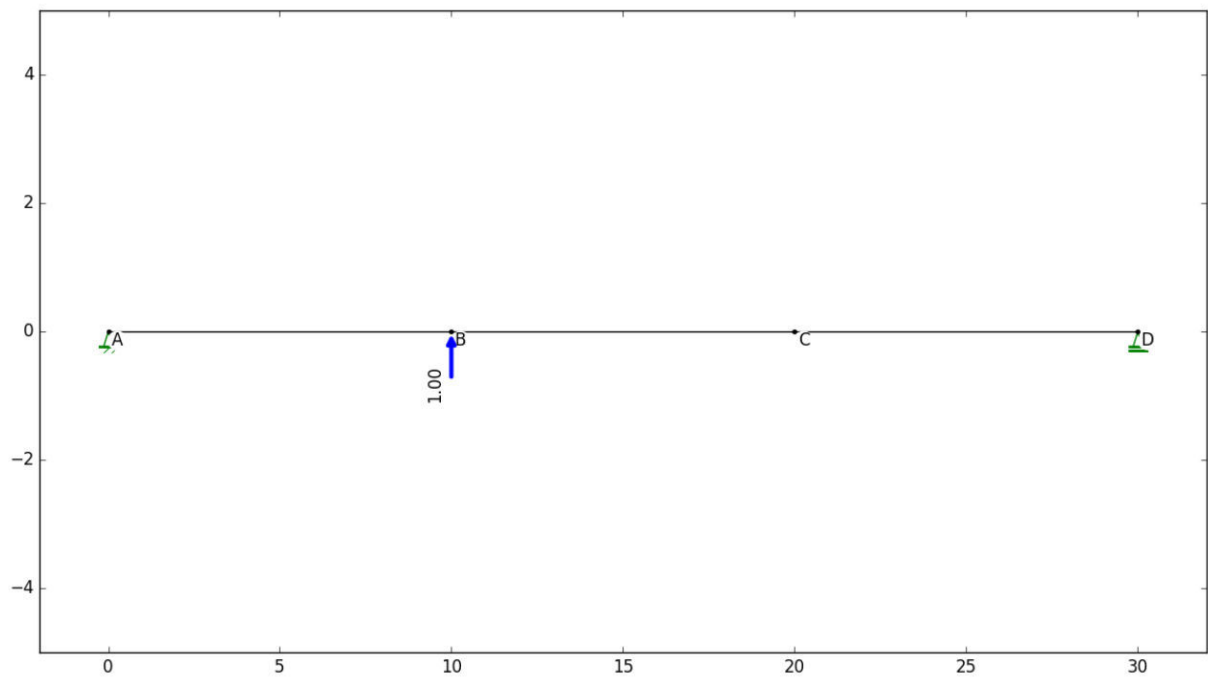
Fonte: Autoria Própria (2019)

Figura 47- Exemplo 01 - Problema 0



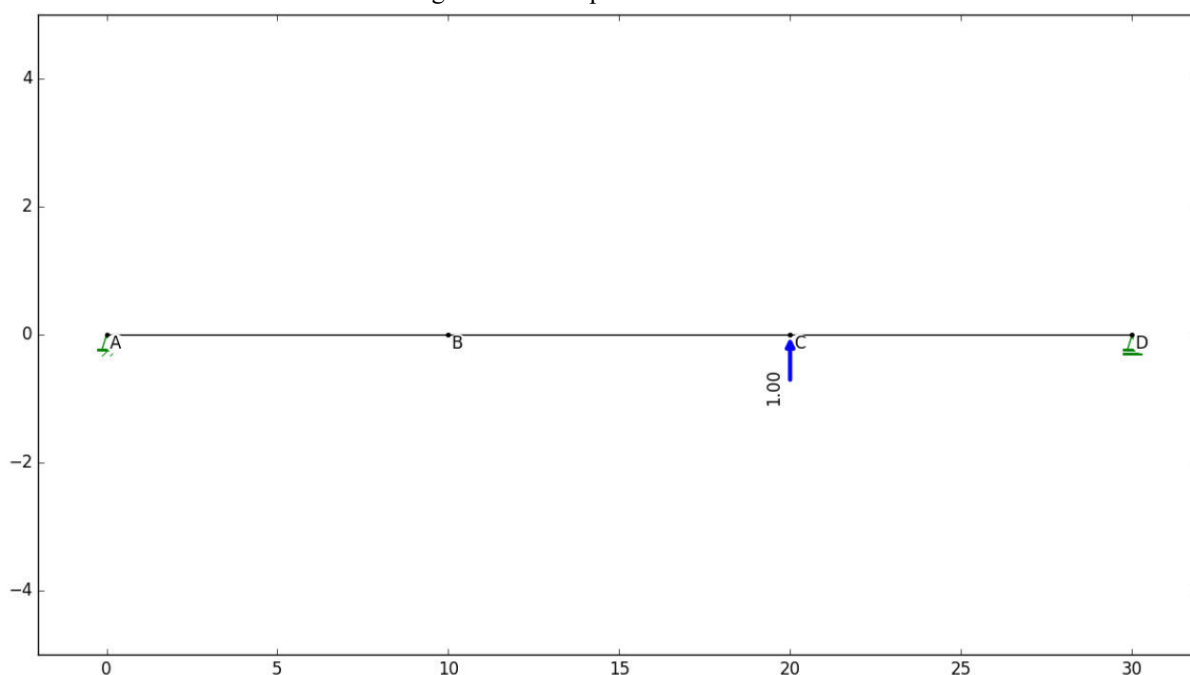
Fonte: Autoria Própria (2019)

Figura 48- Exemplo 01 - Problema 1



Fonte: Autoria Própria (2019)

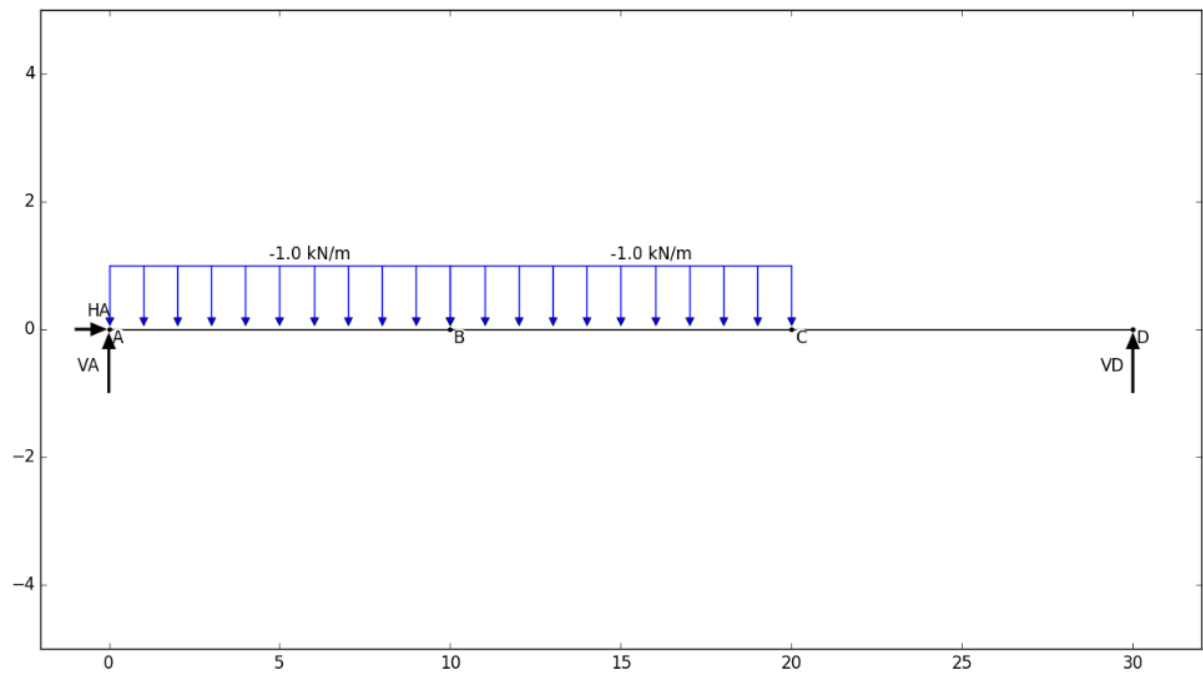
Figura 49- Exemplo 01 - Problema 2



Fonte: Autoria Própria (2019)

Em seguida é gerado o DCL dos problemas virtuais, onde, eles que serão responsáveis para resolver o problema real posteriormente, exibindo todas as forças ativas e reativas atuantes na viga. Tendo representado o DCL (figura 50, 52 e 54). É possível, agora, aplicando-se as equações de equilíbrio estático para determinar os valores de cada reação do problema (ver figura 51, 53 e 55).

Figura 50- DCL - exemplo 01 - Problema 0



Fonte: Autoria Própria (2019)

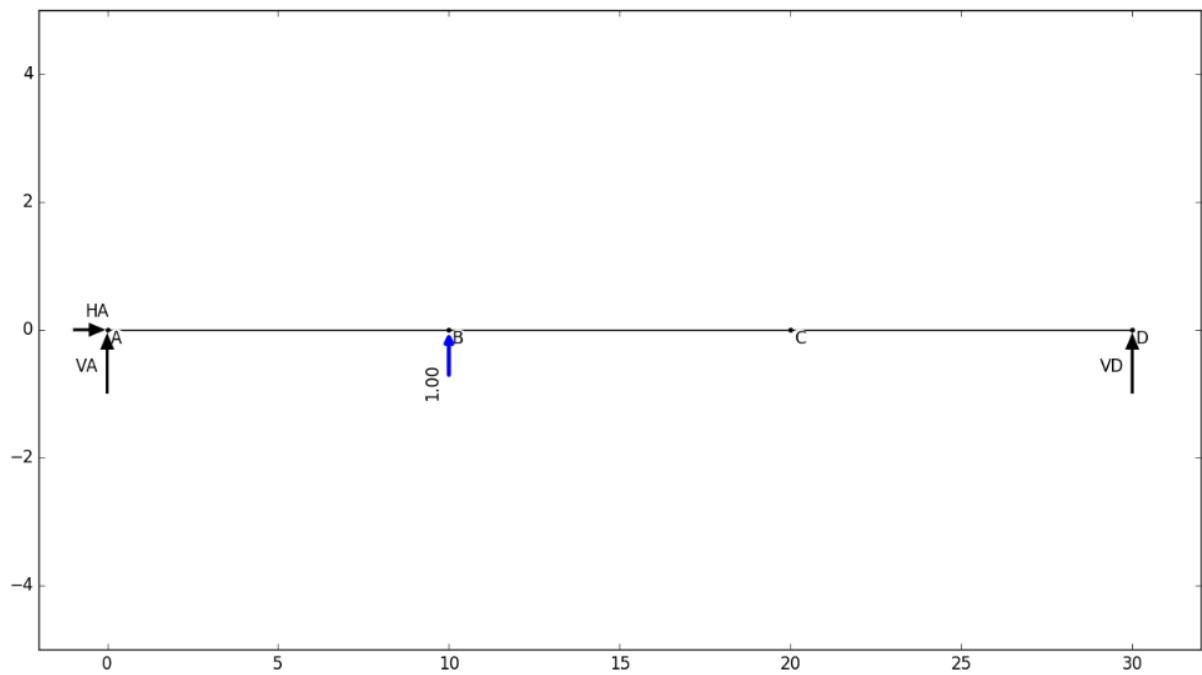
Figura 51- Cálculo das reações de apoio - exemplo 01 - problema 0

$$\begin{aligned}
 \sum F_H &= 0 \\
 +H_A &= 0 \\
 \sum F_V &= 0 \\
 +V_D + V_A - 10.00000 - 10.00000 &= 0 \\
 \sum M_A &= 0 \\
 +V_D * 30.00000 + V_A * 0.00000 - (10.00000 * 5.00000) - (10.00000 * 15.00000) &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_A &= 0.00000 \\
 V_D &= 6.66667 \\
 V_A &= 13.33333
 \end{aligned}$$

Fonte: Autoria Própria (2019)

Figura 52- DCL - exemplo 01 - problema 1



Fonte: Autoria Própria (2019)

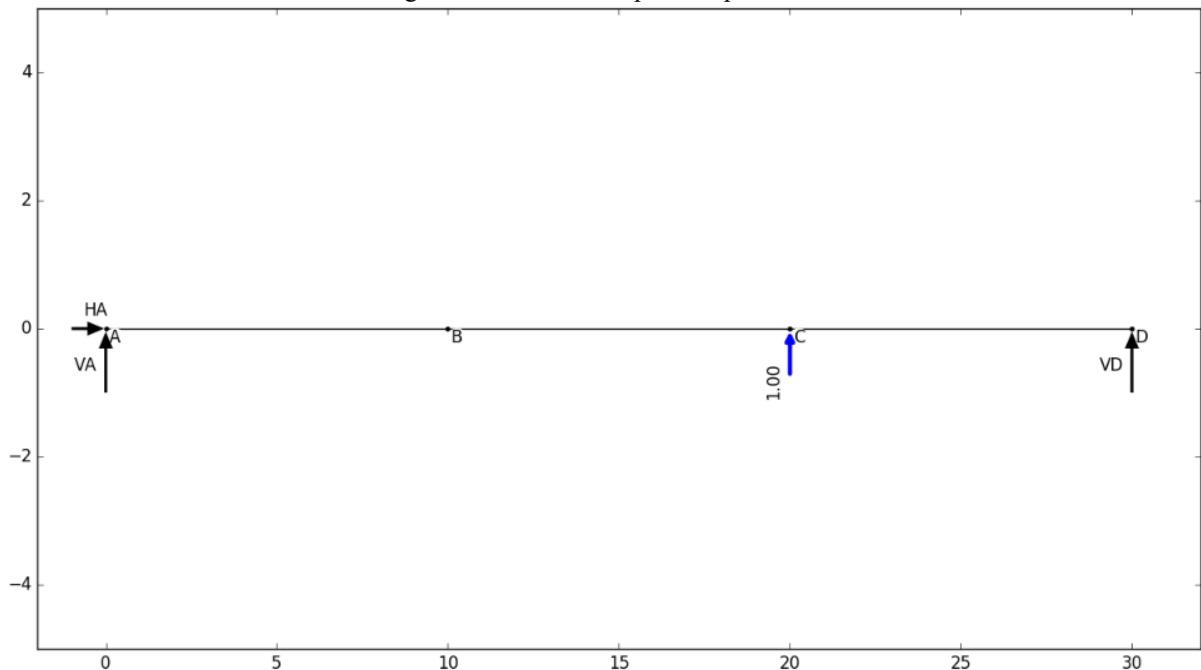
Figura 53- Cálculo das reações de apoio - exemplo 01 - problema 1

$$\begin{aligned} \sum F_H &= 0 \\ +HA &= 0 \\ \sum F_V &= 0 \\ +VD + VA + 1.00000 &= 0 \\ \sum M_A &= 0 \\ +VD * 30.00000 + VA * 0.00000 + (1.00000 * 10.00000) &= 0 \end{aligned}$$

$$\begin{aligned} HA &= 0.00000 \\ VD &= -0.33333 \\ VA &= -0.66667 \end{aligned}$$

Fonte: Autoria Própria (2019)

Figura 54- DCL - exemplo 01 - problema 2



Fonte: Autoria Própria (2019)

Figura 55- Cálculo das reações de apoio - exemplo 01 - problema 2

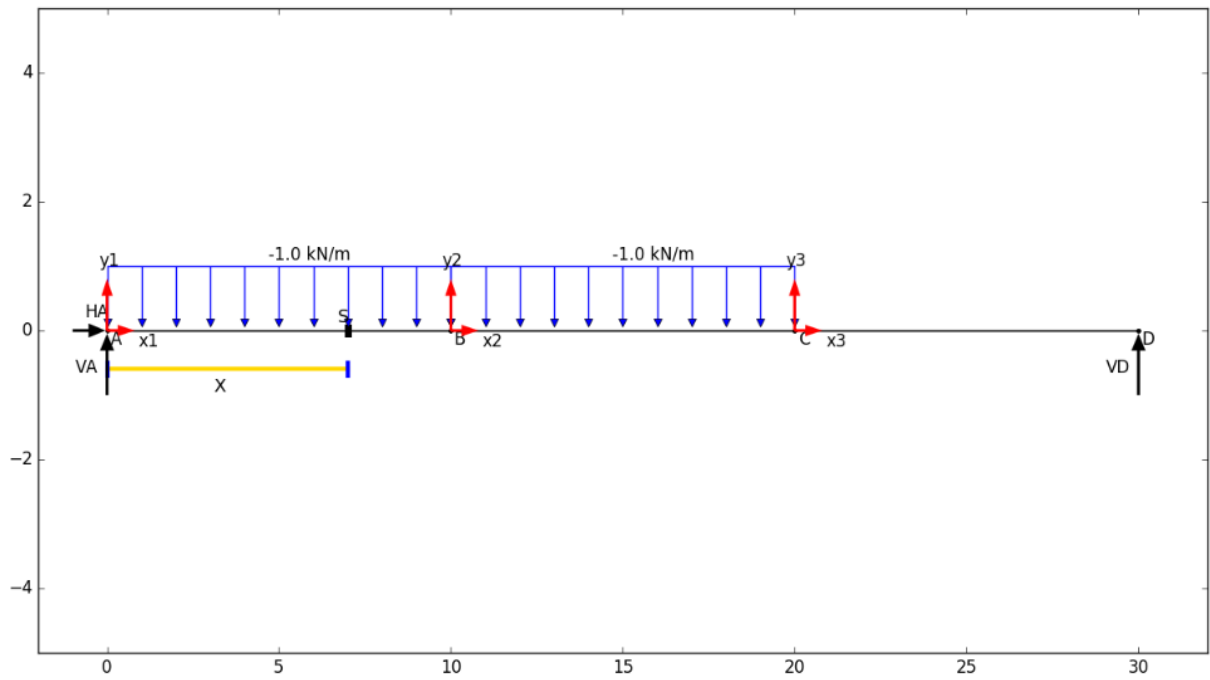
$$\begin{aligned}
 \sum F_H &= 0 \\
 +H_A &= 0 \\
 \sum F_V &= 0 \\
 +V_D + V_A + 1.00000 &= 0 \\
 \sum M_A &= 0 \\
 +V_D * 30.00000 + V_A * 0.00000 + (1.00000 * 20.00000) &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_A &= 0.00000 \\
 V_D &= -0.66667 \\
 V_A &= -0.33333
 \end{aligned}$$

Fonte: Autoria Própria (2019)

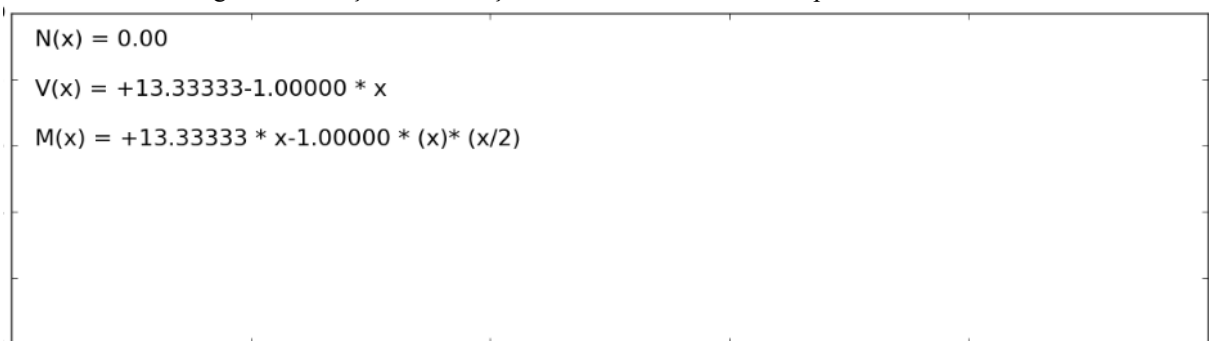
Na sequência, para determinação das equações dos esforços normais de todos os problemas virtuais, foi executada uma análise local de cada elemento, aplicando-se o método das seções (figura 56, 58 e 60). Para as equações ver figura 57, 59 e 61.

Figura 56- Seccionamento local da barra 1 - exemplo 01 - Problema 0



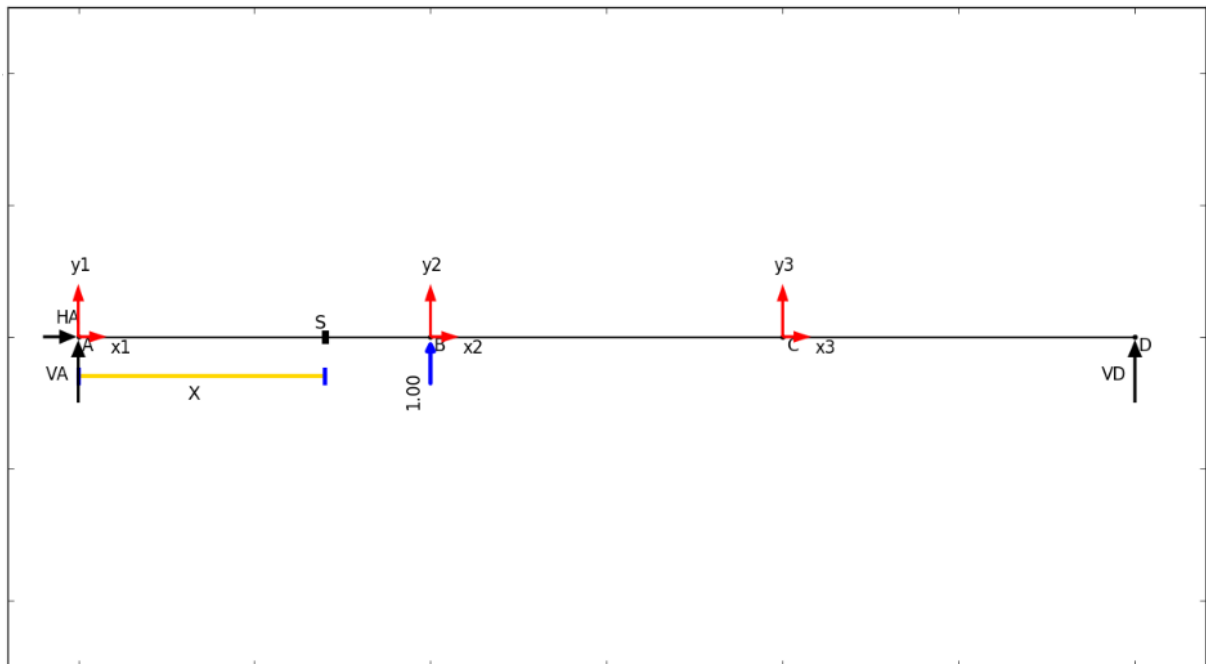
Fonte: Autoria Própria (2019)

Figura 57- Função dos esforços internos da barra 1 - exemplo 01 - Problema 0



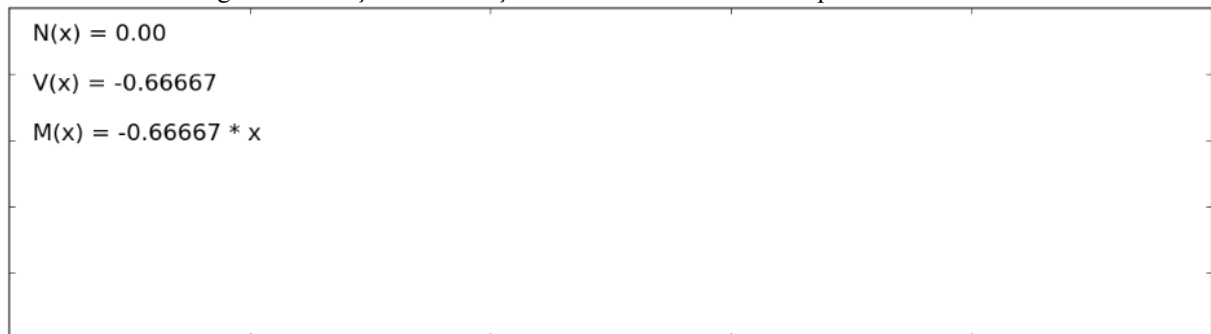
Fonte: Autoria Própria (2019)

Figura 58- Seccionamento local da barra 1 - exemplo 01 - Problema 1



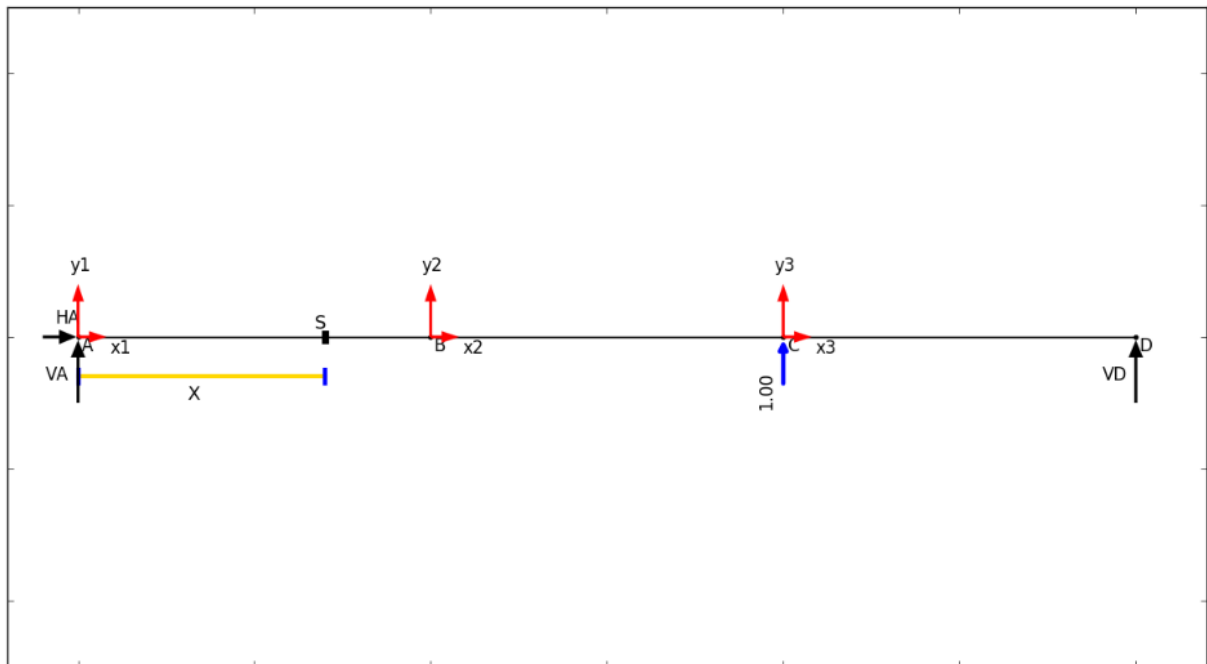
Fonte: Autoria Própria (2019)

Figura 59- Função dos esforços internos da barra 1 - exemplo 01 - Problema 1



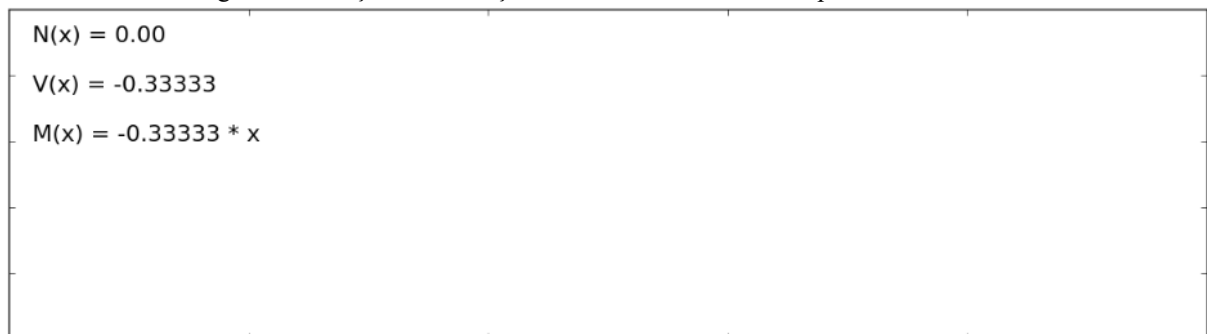
Fonte: Autoria Própria (2019)

Figura 60- Seccionamento local da barra 1 - exemplo 01 - Problema 2



Fonte: Autoria Própria (2019)

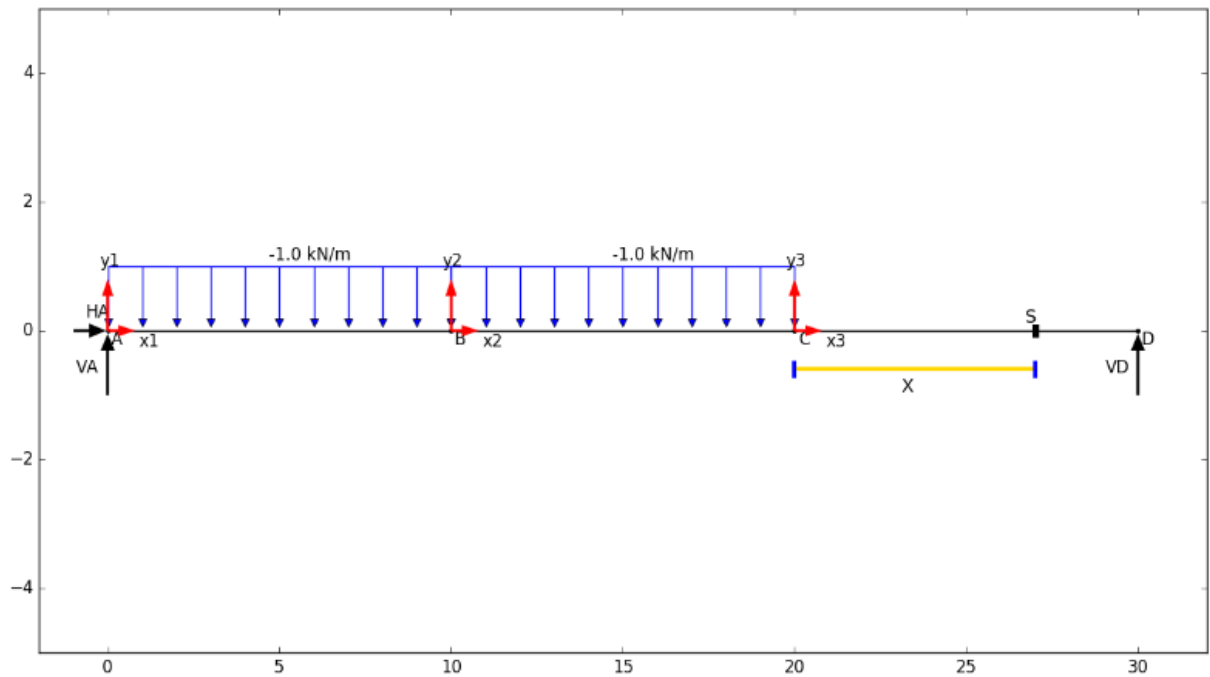
Figura 61- Função dos esforços internos da barra 1 - exemplo 01 - Problema 2



Fonte: Autoria Própria (2019)

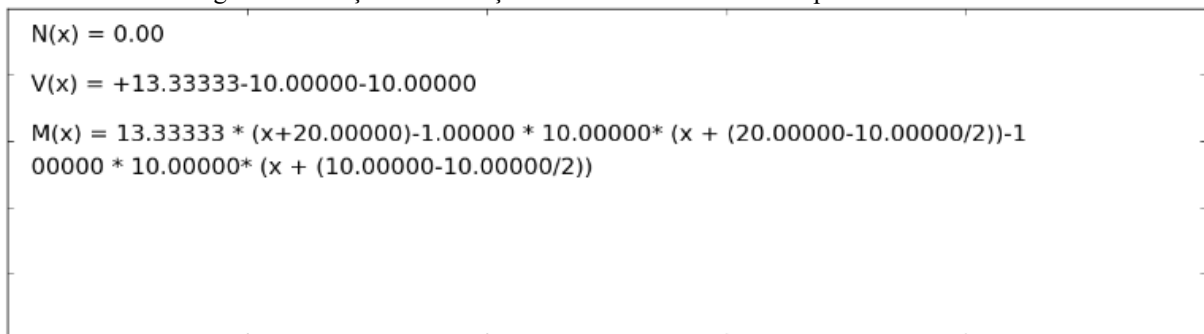
De forma semelhante, os demais elementos foram analisados, observar nas figuras 62, 64, 66, 68, 70 e 72. Na sequência de cada figura será apresentada suas respectivas funções dos esforços internos ao longo do trecho da barra (ver figuras 63, 65, 67, 69, 71 e 73).

Figura 64- Seccionamento local da barra 3 - exemplo 01 - Problema 0



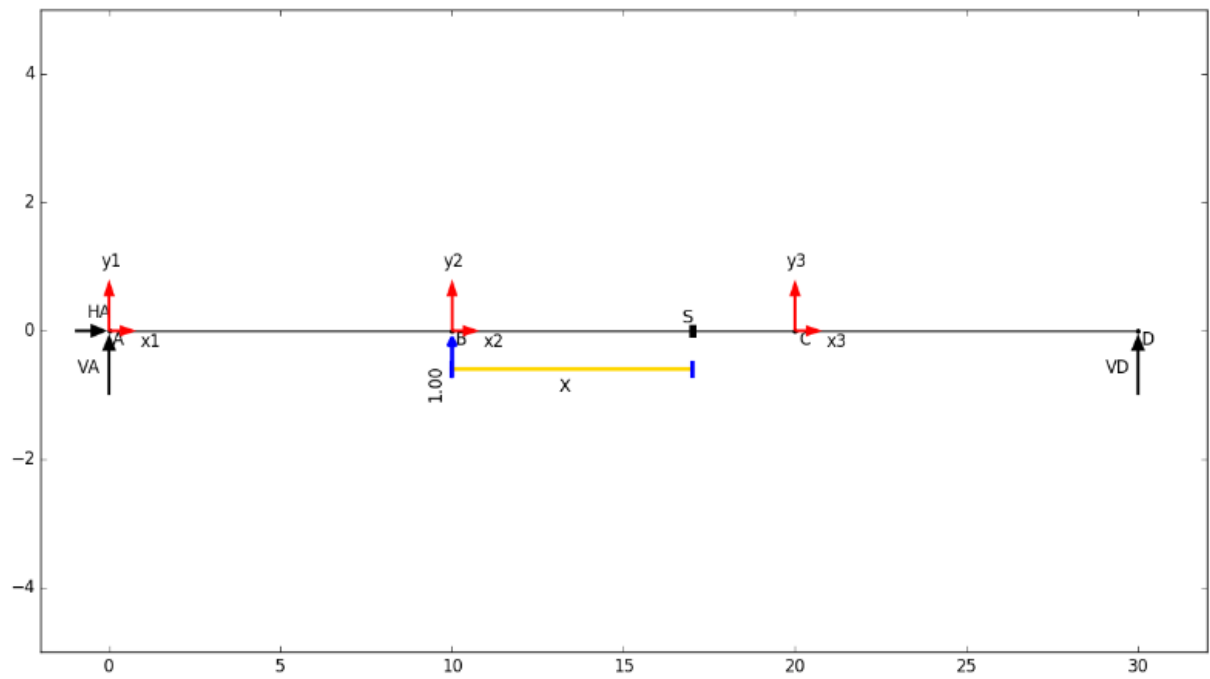
Fonte: Autoria Própria (2019)

Figura 65- Função dos esforços internos da barra 3 - exemplo 01 - Problema 0



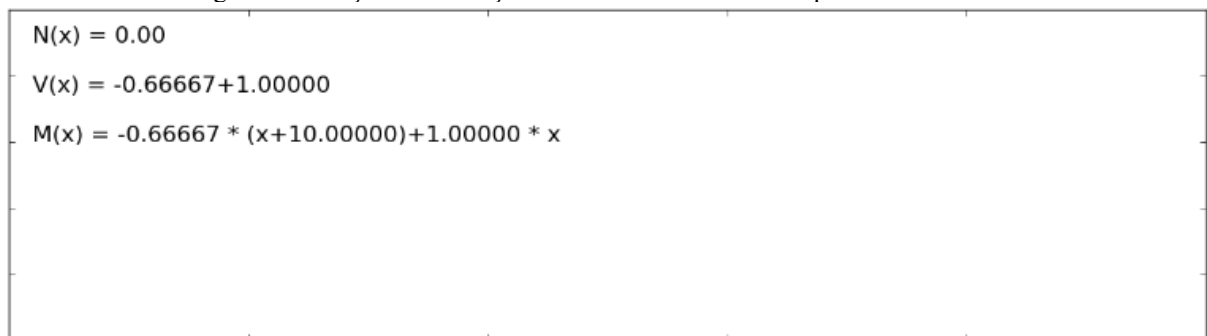
Fonte: Autoria Própria (2019)

Figura 66- Seccionamento local da barra 2 - exemplo 01 - Problema 1



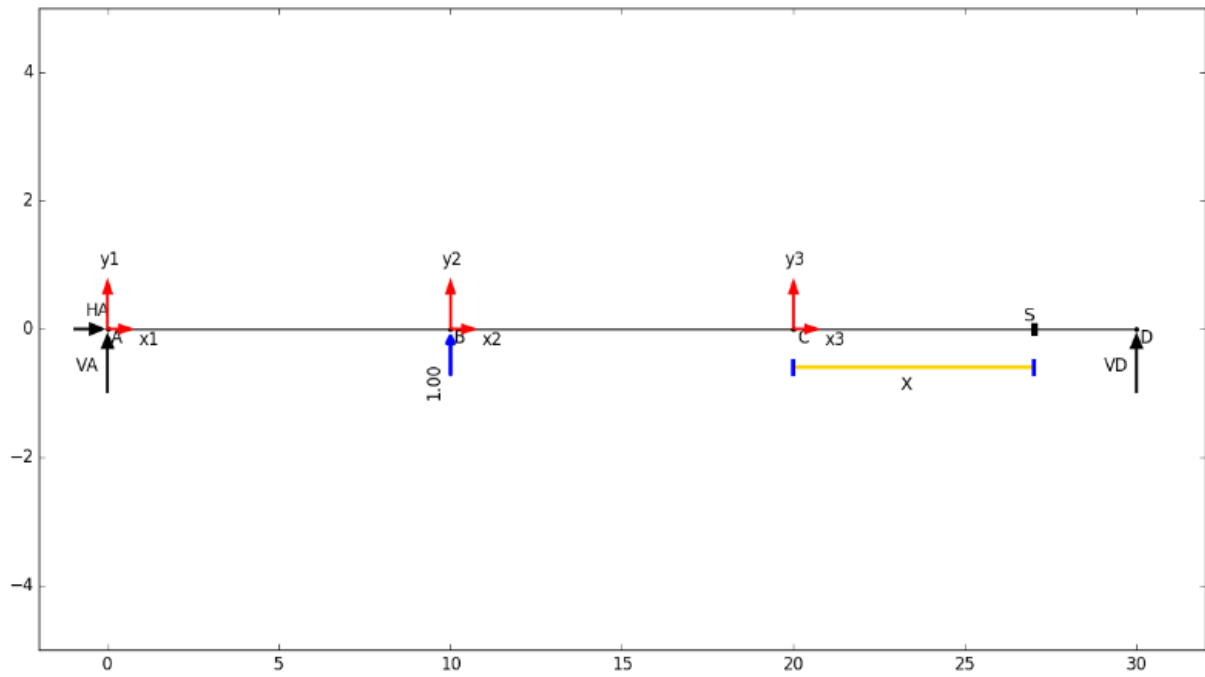
Fonte: Autoria Própria (2019)

Figura 67- Função dos esforços internos da barra 2 - exemplo 01 - Problema 1



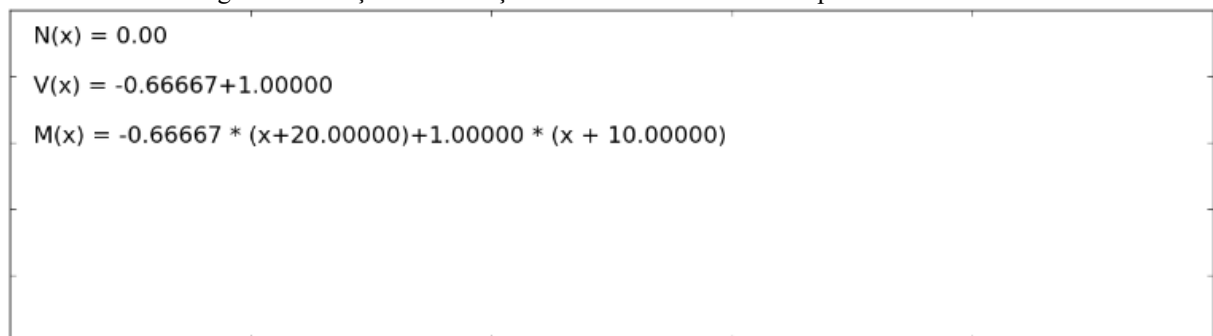
Fonte: Autoria Própria (2019)

Figura 68- Seccionamento local da barra 3 - exemplo 01 - Problema 1



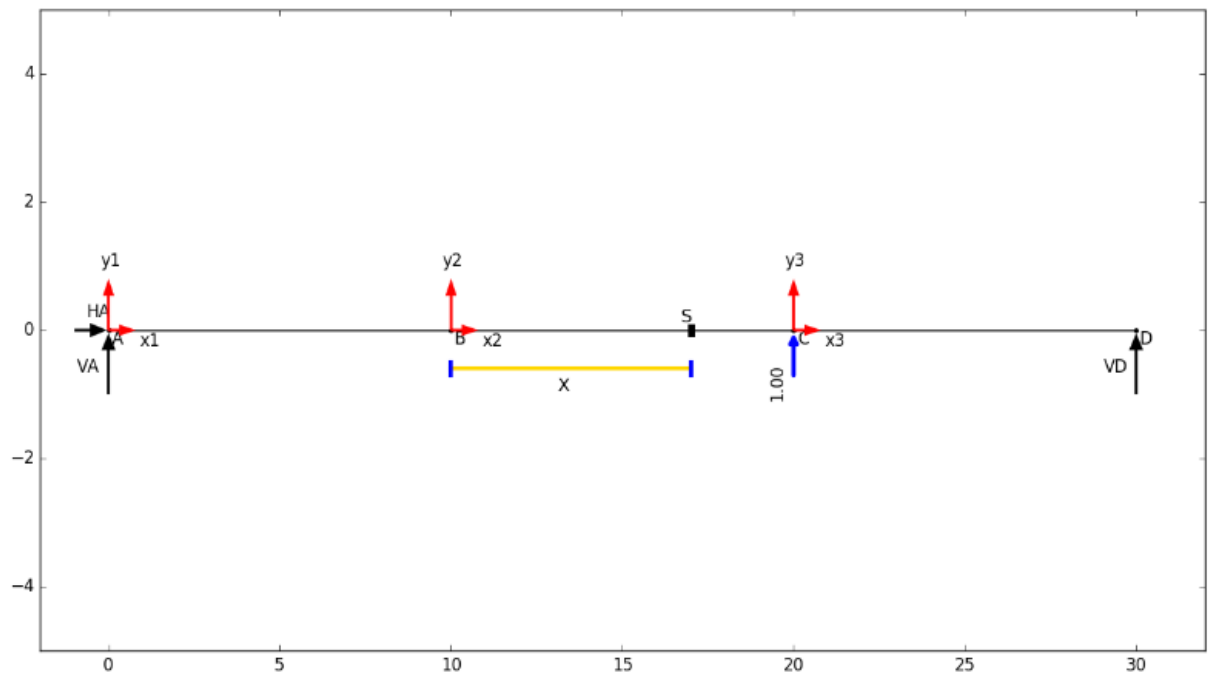
Fonte: Autoria Própria (2019)

Figura 69- Função dos esforços internos da barra 3 - exemplo 01 - Problema 1



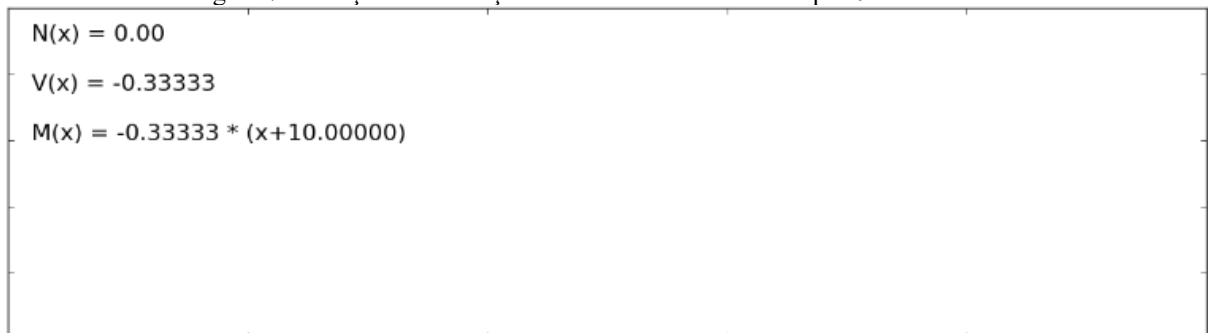
Fonte: Autoria Própria (2019)

Figura 70- Seccionamento local da barra 2 - exemplo 01 - Problema 2



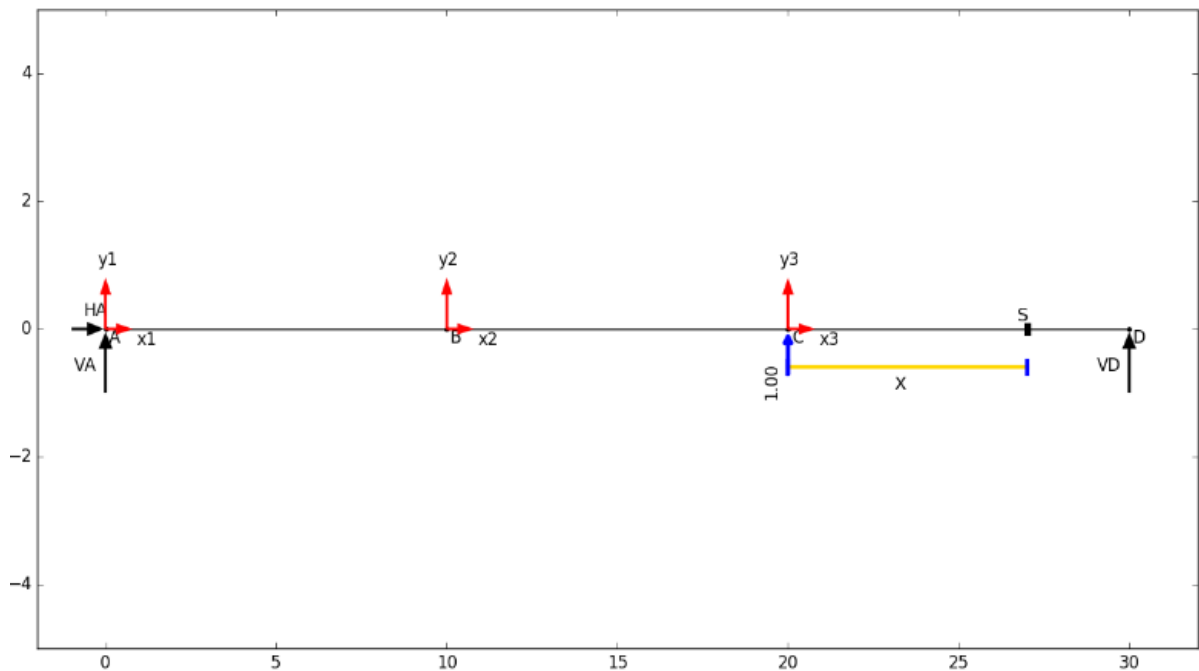
Fonte: Autoria Própria (2019)

Figura 71- Função dos esforços internos da barra 2 - exemplo 01 - Problema 2



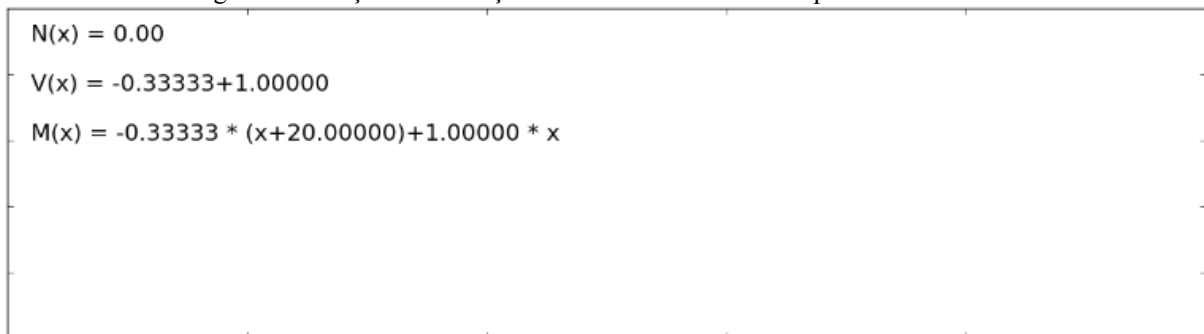
Fonte: Autoria Própria (2019)

Figura 72- Seccionamento local da barra 3 - exemplo 01 - Problema 2



Fonte: Autoria Própria (2019)

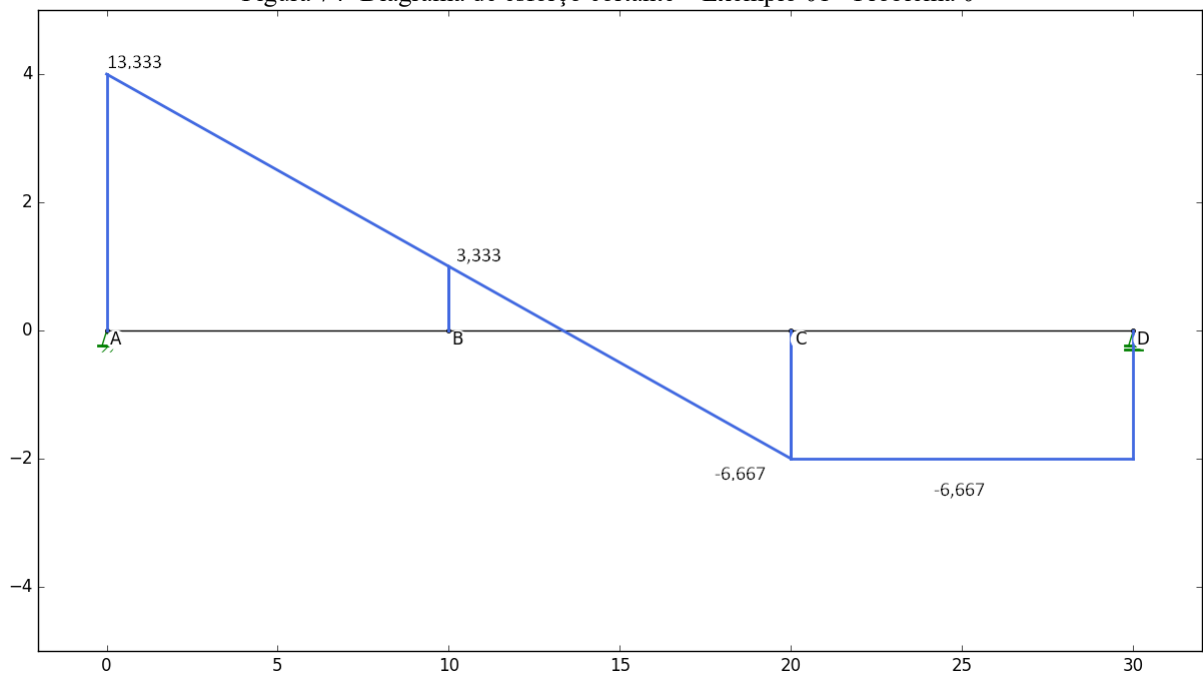
Figura 73- Função dos esforços internos da barra 3 - exemplo 01 - Problema 2



Fonte: Autoria Própria (2019)

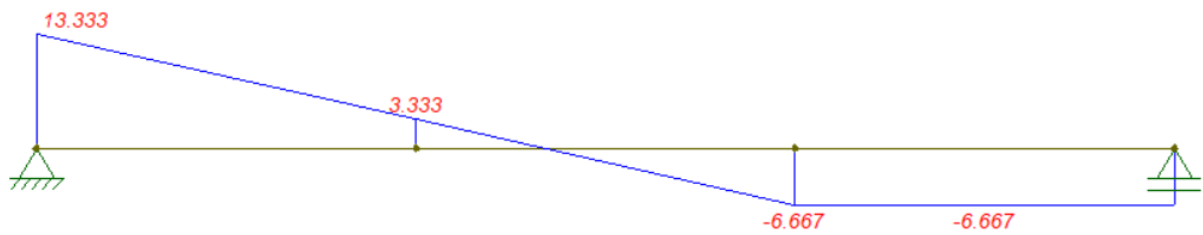
Após apresentação dos resultados em forma de equações, os mesmos são plotados em gráficos, todos os problemas virtuais, os quais serão comparados com os obtidos pelo Ftool. Os primeiros resultados são os diagramas de esforço normal, porém quando as vigas contínuas estão submetidas a forças verticais, elas não apresentam esforços normais, então, vamos seguir para as representações dos gráficos de esforço cortante e momento fletor. Cada gráfico gerado no programa será comparado com seu respectivo traçado gerado pelo Ftool. Representado nas figuras a seguir.

Figura 74- Diagrama de esforço cortante – Exemplo 01 - Problema 0



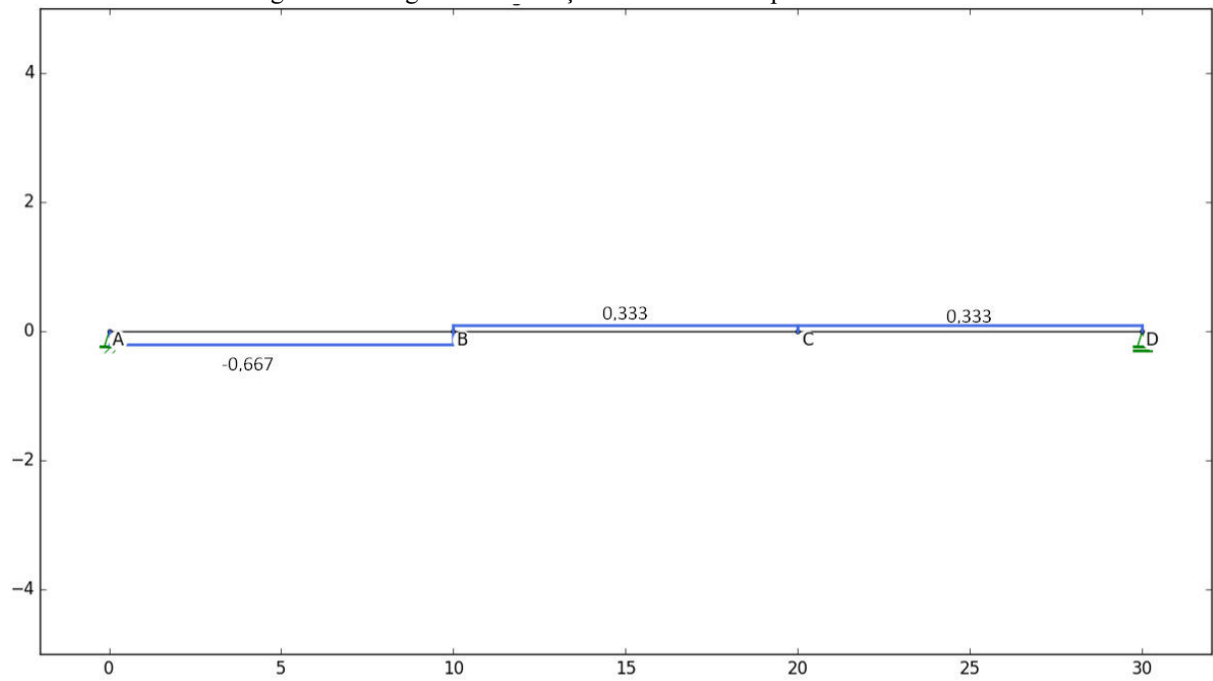
Fonte: Autoria Própria (2019)

Figura 75- Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema 0



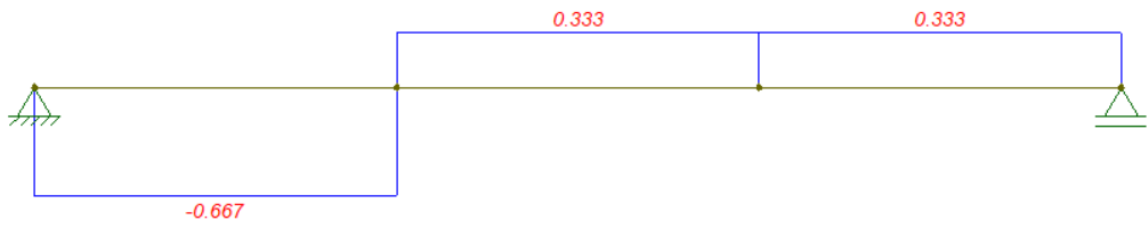
Fonte: Ftool (2019)

Figura 76- Diagrama de esforço cortante – Exemplo 01 - Problema 1



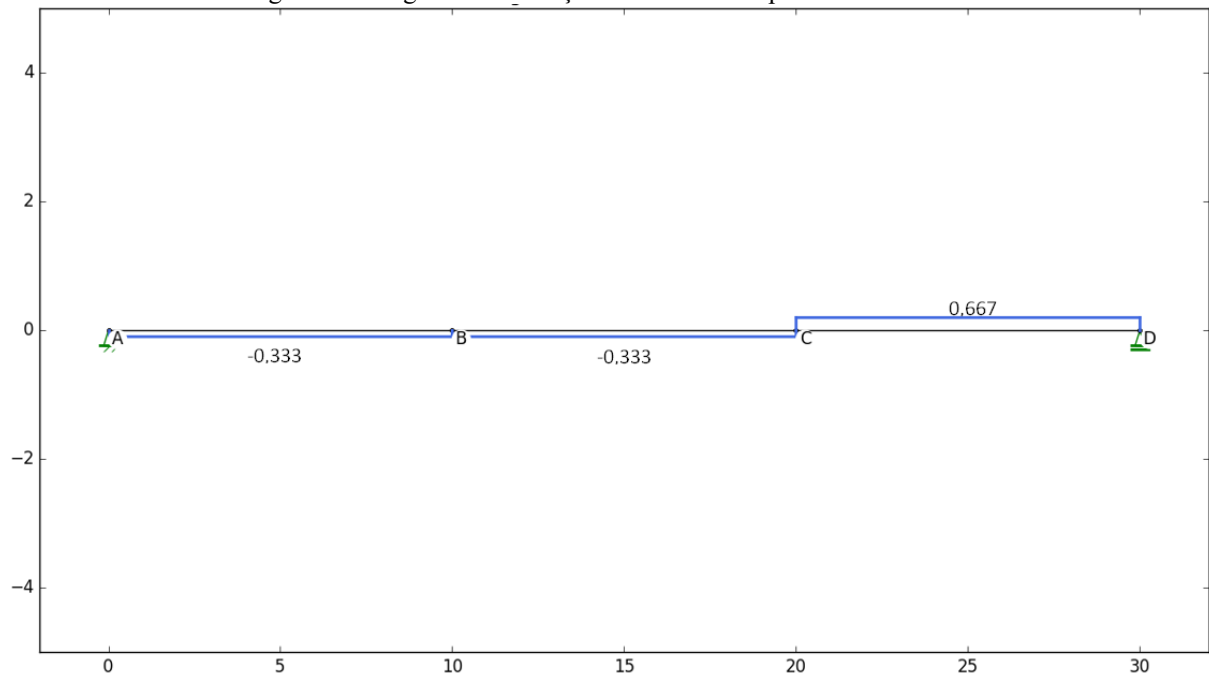
Fonte: Autoria Própria (2019)

Figura 77- Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema 1



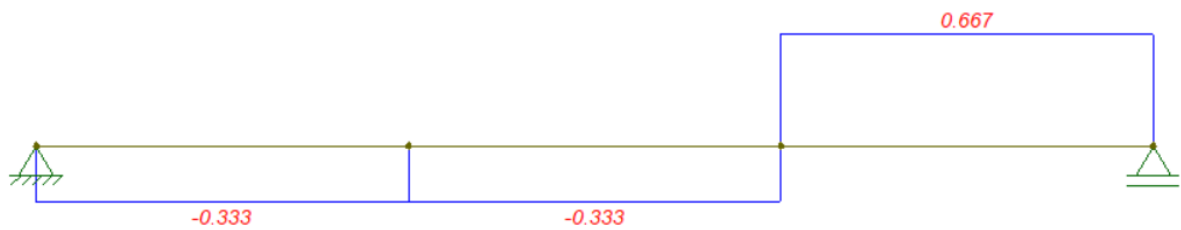
Fonte: Ftool (2019)

Figura 78- Diagrama de esforço cortante – Exemplo 01 - Problema 2



Fonte: Autoria Própria (2019)

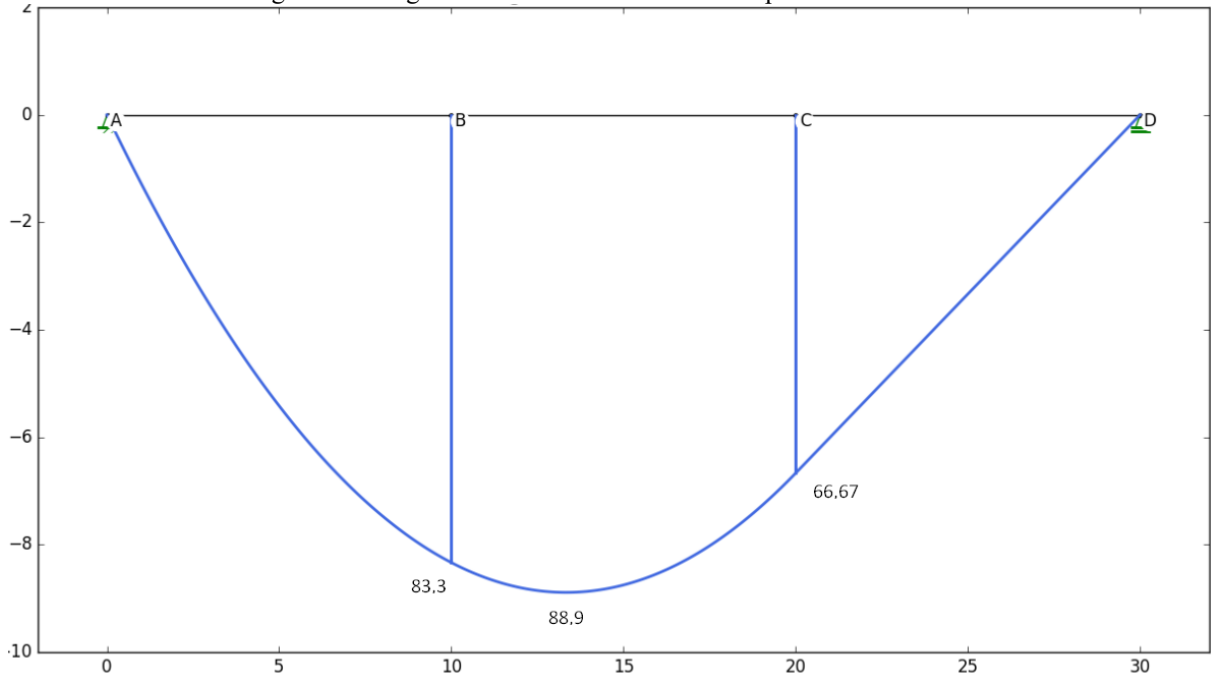
Figura 79- Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema 2



Fonte: Ftool (2019)

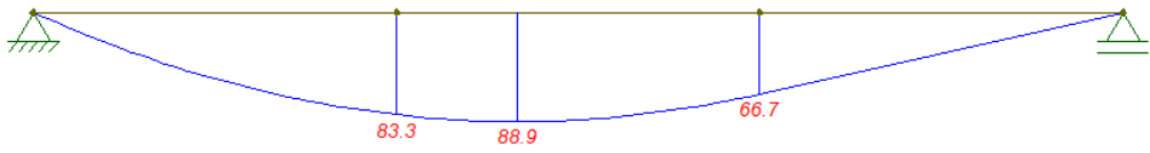
A seguir são demonstrados os diagramas de momento fletor.

Figura 80- Diagrama de momento fletor – Exemplo 01 - Problema 0



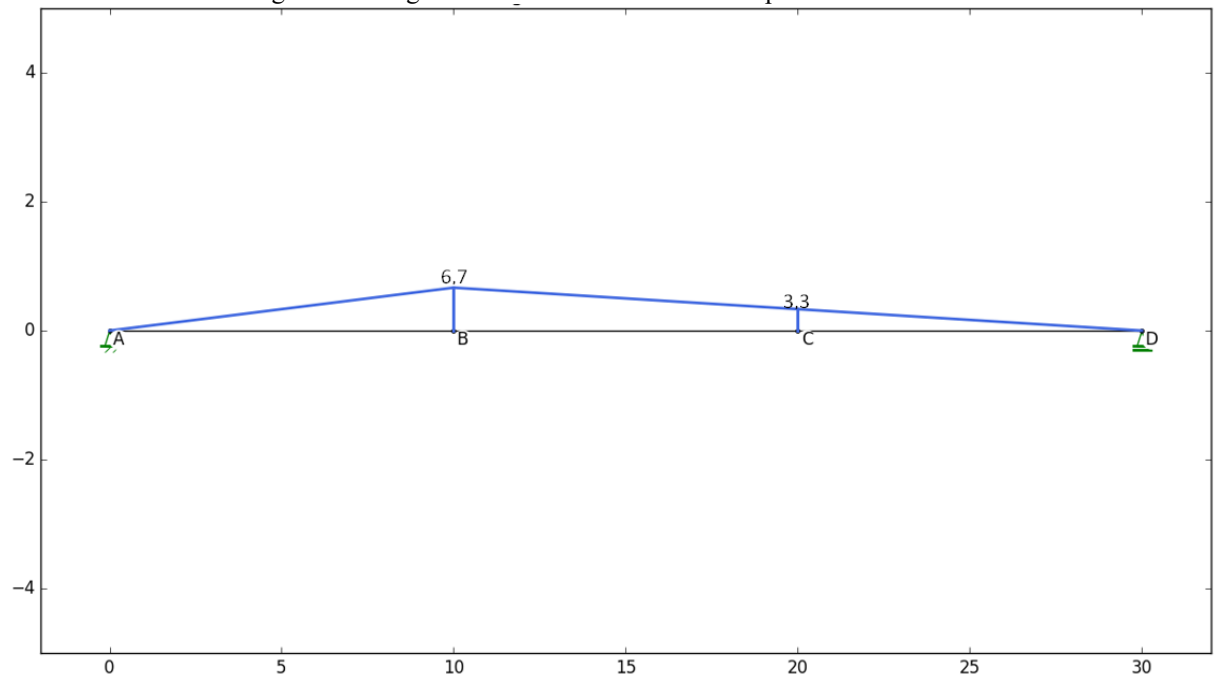
Fonte: Autoria Própria (2019)

Figura 81- Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema 0



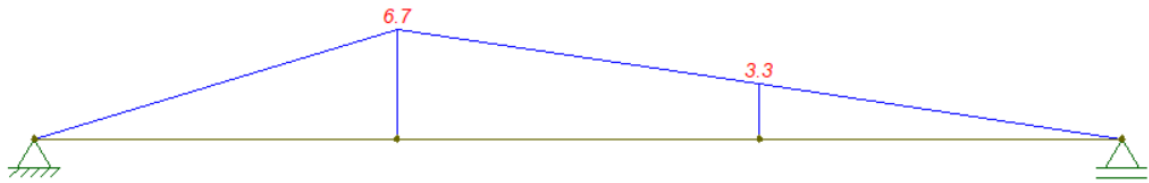
Fonte: Ftool (2019)

Figura 82- Diagrama de momento fletor – Exemplo 01 - Problema 1



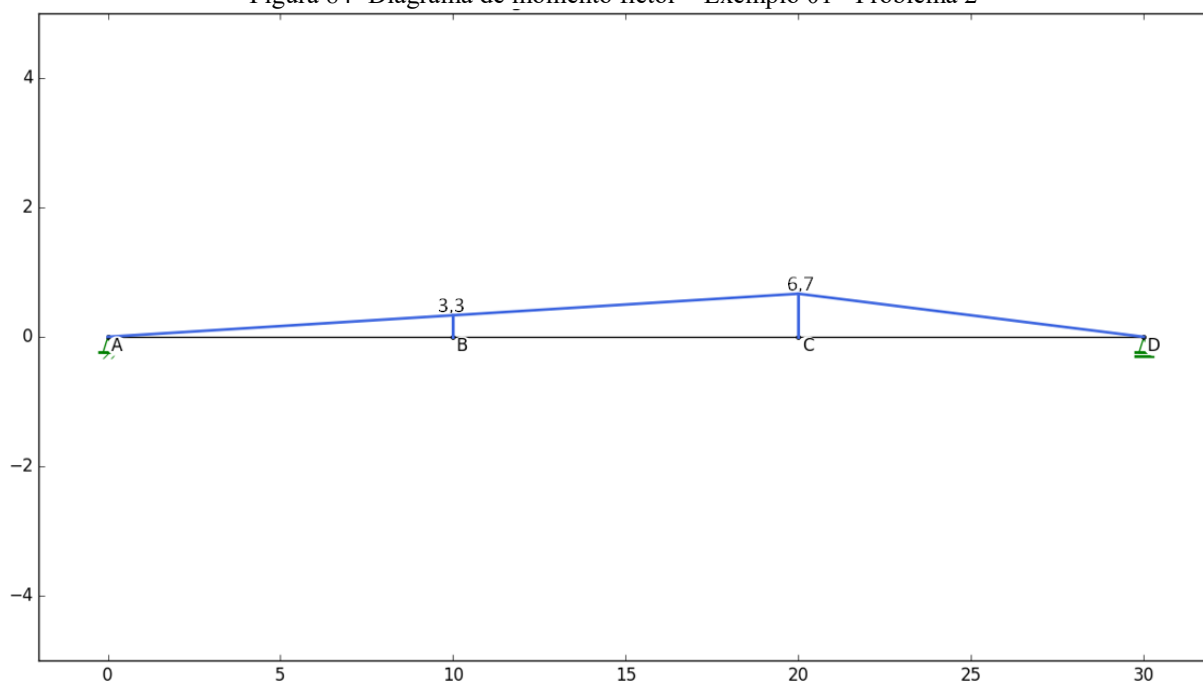
Fonte: Autoria Própria (2019)

Figura 83- Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema 1



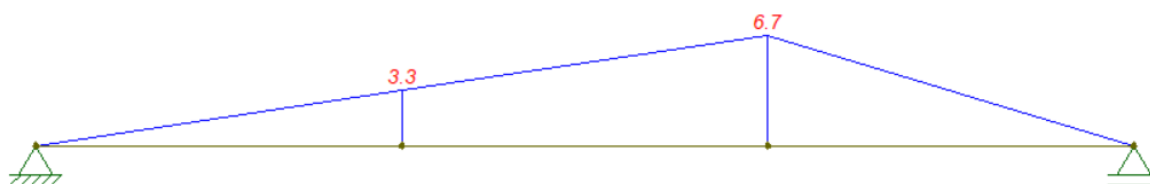
Fonte: Ftool (2019)

Figura 84- Diagrama de momento fletor – Exemplo 01 - Problema 2



Fonte: Autoria Própria (2019)

Figura 85- Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema 2

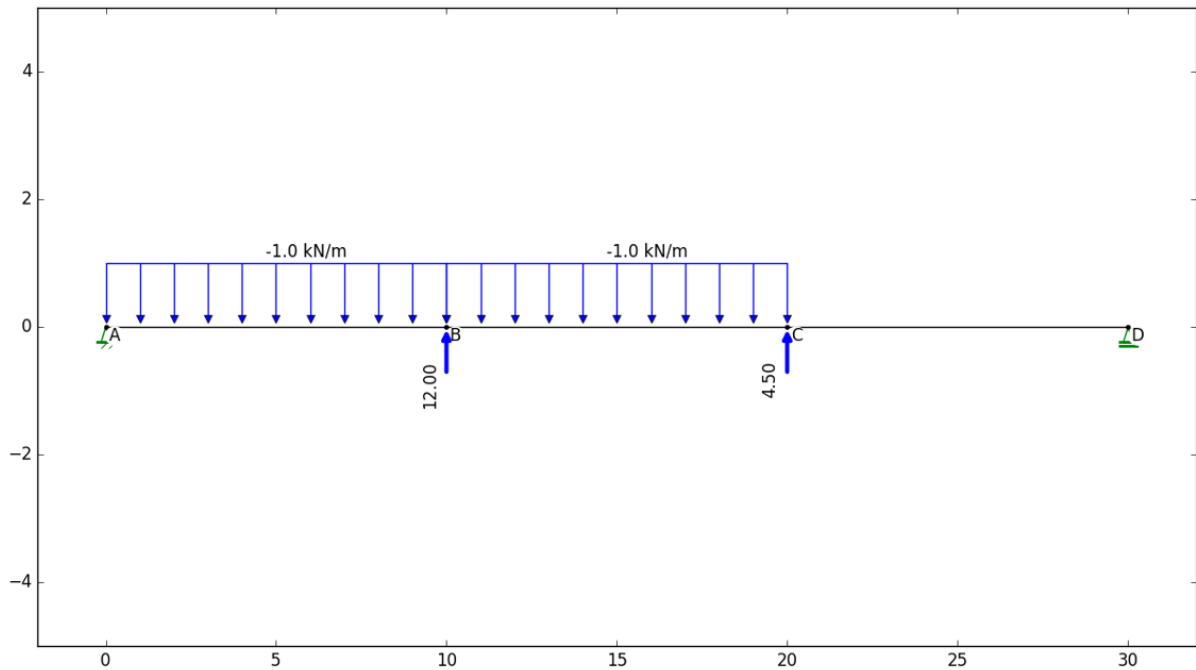


Fonte: Ftool (2019)

Os seus resultados saíram como o esperado, apenas apresentando uma diferença na escala de visualização.

De acordo com os resultados dos problemas virtuais, utilizando a lógica da resolução pelo processo das forças e funções matriciais determinam o valor das forças nos locais dos apoios retirados, agora se tornando uma viga isostática. (Ver figura 93)

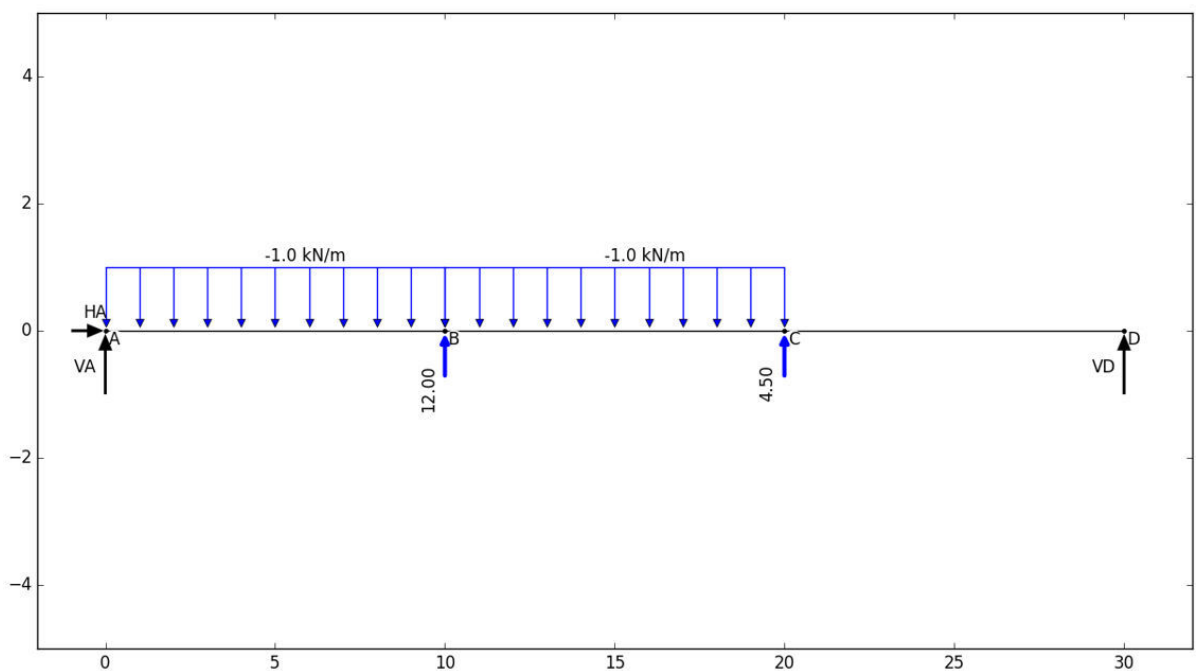
Figura 86- Exemplo 1 - Problema real



Fonte: Autoria Própria (2019)

Com as novas forças nos vínculos que haviam sido retirados, agora podemos resolver o problema através das equações de equilíbrio. Ver figuras 87 e 88 para verificar o DCL e os cálculos das reações de apoio.

Figura 87- DCL - Exemplo 01 - Problema real



Fonte: Autoria Própria (2019)

Figura 88- Cálculos das reações de apoio - exemplo 01 - Problema real

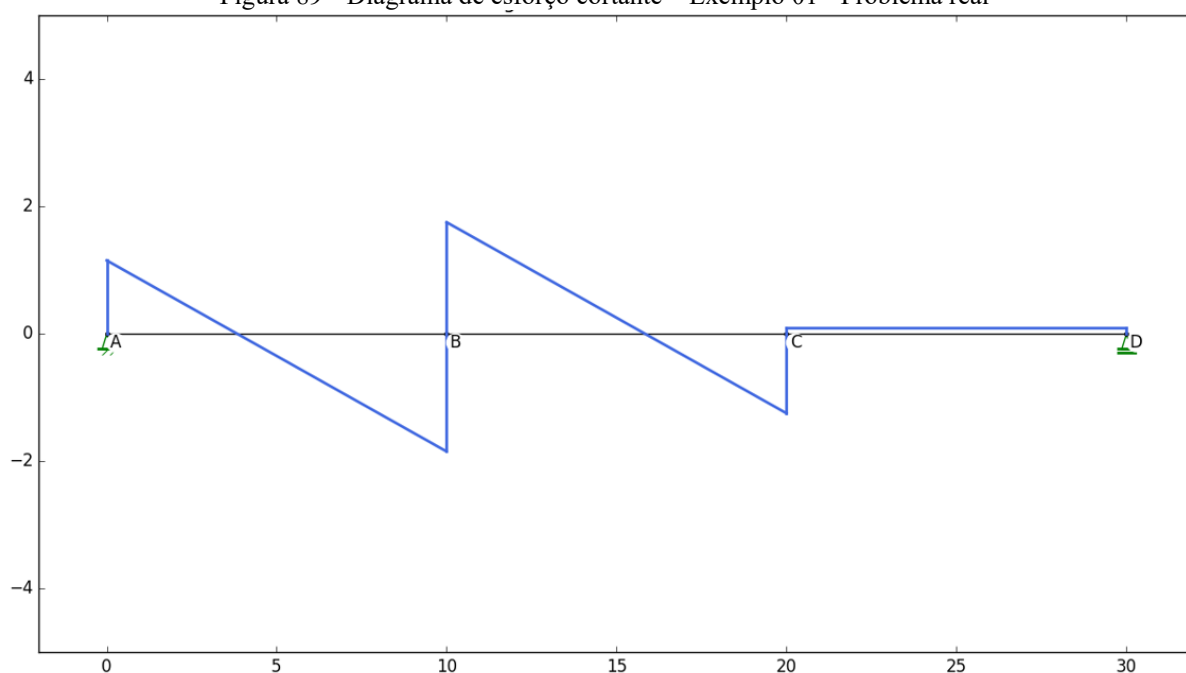
$$\begin{aligned} \sum F_H &= 0 \\ +H_A &= 0 \\ \sum F_V &= 0 \\ +V_A + V_D + 12.00002 + 4.49987 - 10.00000 - 10.00000 &= 0 \\ \sum M_A &= 0 \\ +V_A * 0.00000 + V_D * 30.00000 + (12.00002 * 10.00000) + (4.49987 * 20.00000) - (10.00000 \\ &\quad * 5.00000) - (10.00000 * 15.00000) \end{aligned}$$

$H_A = 0.00000$
 $V_A = 3.83336$
 $V_D = -0.33325$

Fonte: Autoria Própria (2019)

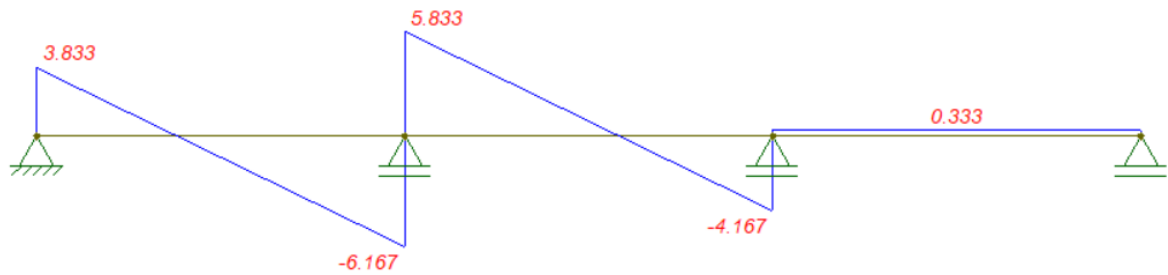
Como já conhecemos todo o processo para obtenção das funções de todas as seções, seguimos com o procedimento no exemplo em questão o programa se mostrou eficaz para execução desta função. Então como sabemos que os gráficos de esforços normais serão nulos, vamos seguir para a comparação dos demais gráficos com os modelos gerados pelo Ftool. Ver figuras a seguir com a representação do programa e o modelo gerado pelo Ftool respectivamente.

Figura 89-- Diagrama de esforço cortante – Exemplo 01 - Problema real



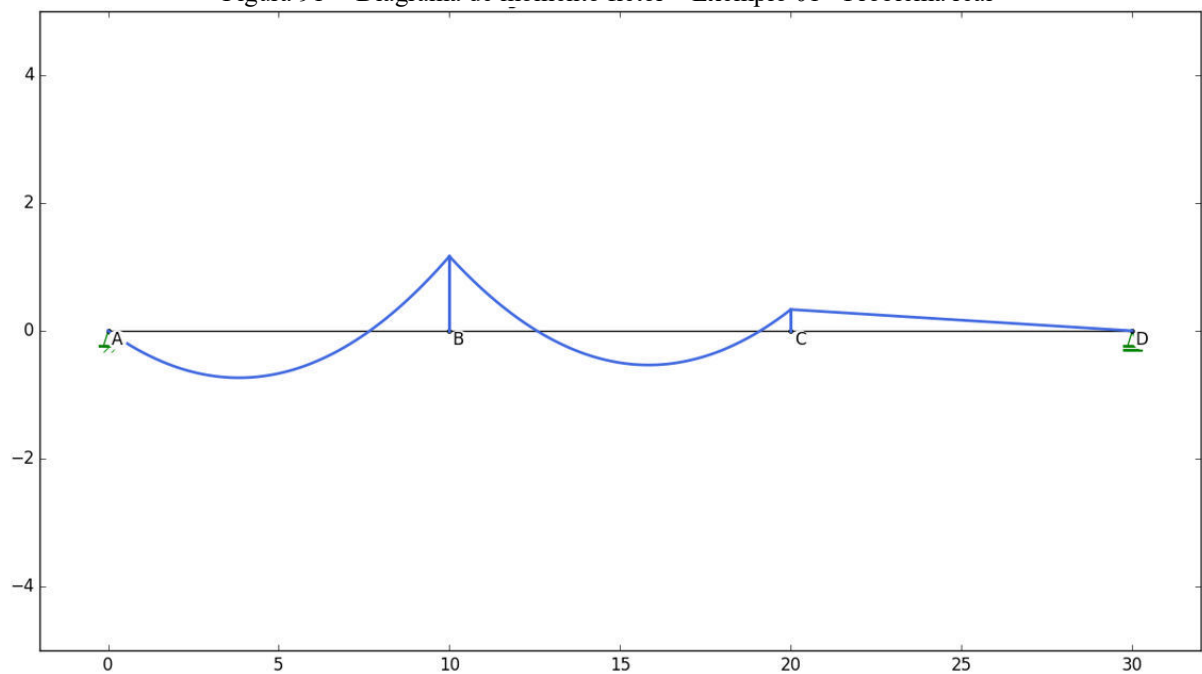
Fonte: Autoria Própria (2019)

Figura 90- - Diagrama de esforço cortante/ FTOOL - Exemplo 01 - Problema Real



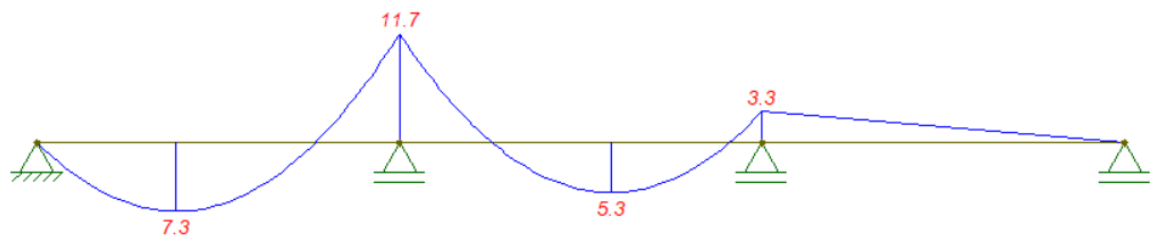
Fonte: Ftool (2019)

Figura 91- - Diagrama de momento fletor – Exemplo 01 - Problema real



Fonte: Autoria Própria (2019)

Figura 92- - Diagrama de momento fletor/ FTOOL - Exemplo 01 - Problema Real

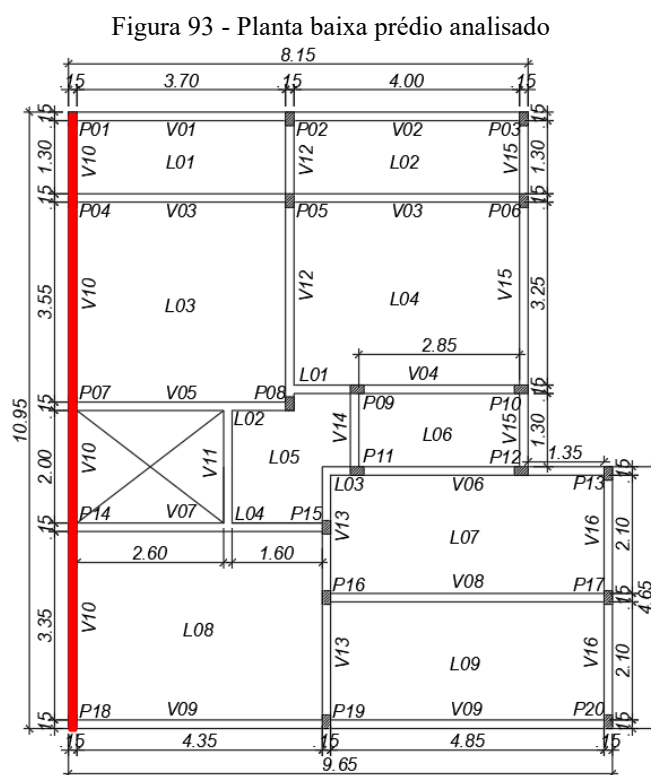


Fonte: Ftool (2019)

Como podemos ver o programa funciona perfeitamente, e o resultado foi como esperávamos, vamos seguir para o próximo exemplo trazendo um modelo de nossa realidade.

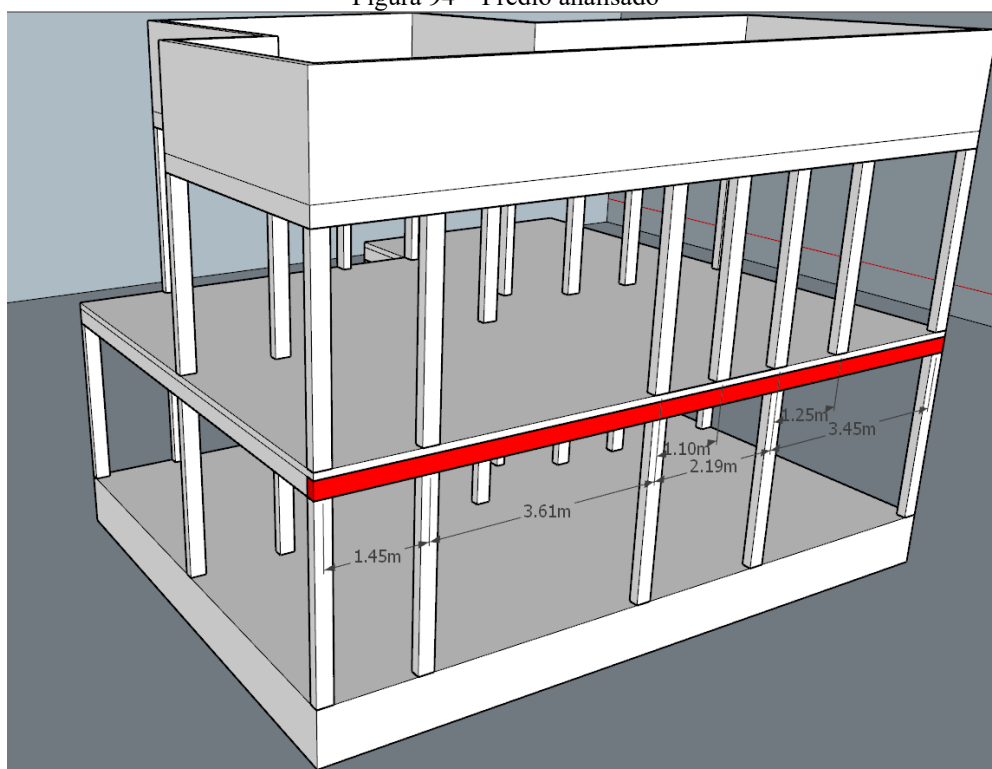
4.2 EXEMPLO 02 – VIGA HIPERESTATICA EM PREDIO UNIFAMILIAR

O segundo exemplo, baseia-se em analisar uma viga específica de um prédio unifamiliar. Para tanto, foi estipulado suas dimensões (largura e altura) conforme presentes em projeto, ela possui característica de estar dividida em 4 vãos e com isso um grau de hiperestaticidade $g=3$, vamos agora analisar a viga destacada (Figura 93 e 94).



Fonte: AutoCAD (2019)

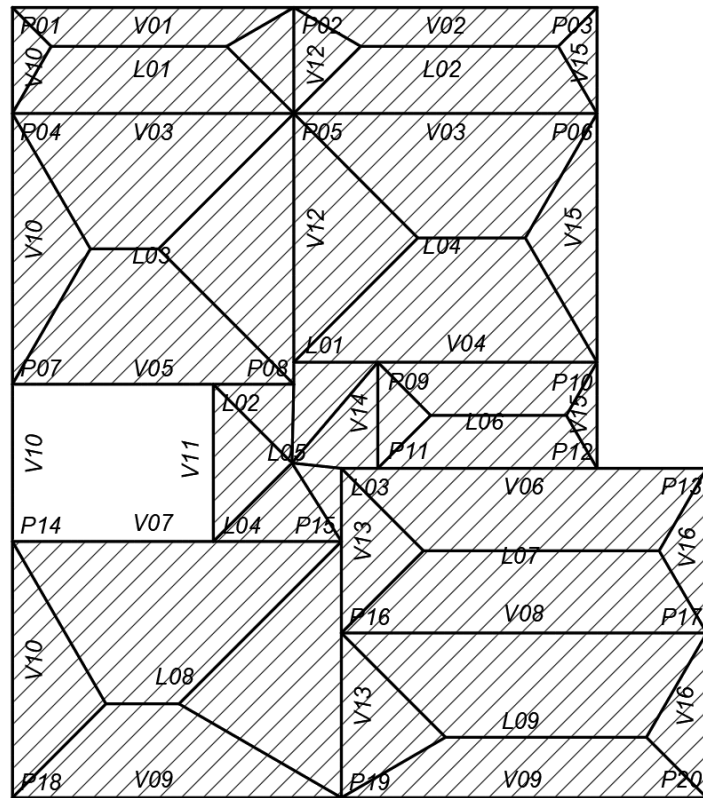
Figura 94 – Prédio analisado



Fonte: SKETCHUP (2019)

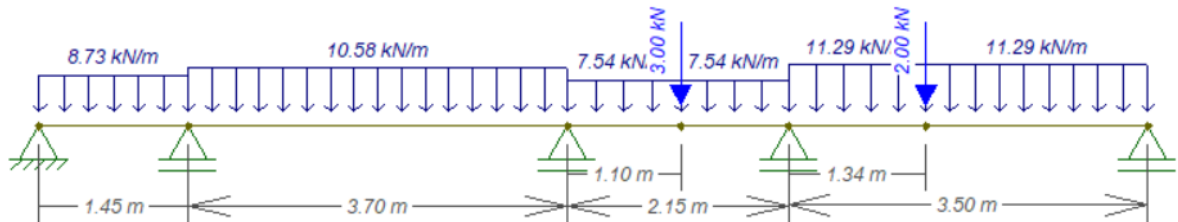
Para a modelagem dessa estrutura, algumas considerações foram feitas. Nesse caso, será analisado a influência das lajes na viga e cargas pontuais dos pilares superiores sob a viga, primeiro analisaremos as influências das lajes sob a viga (ver figura 95).

Figura 95 - Áreas de influência da laje nas vigas



De acordo com as áreas de influência das lajes na viga, peso próprio e cargas pontuais de pilares superiores, surgirá um problema de viga hiperestática de grau dois conforme demonstra figura 96.

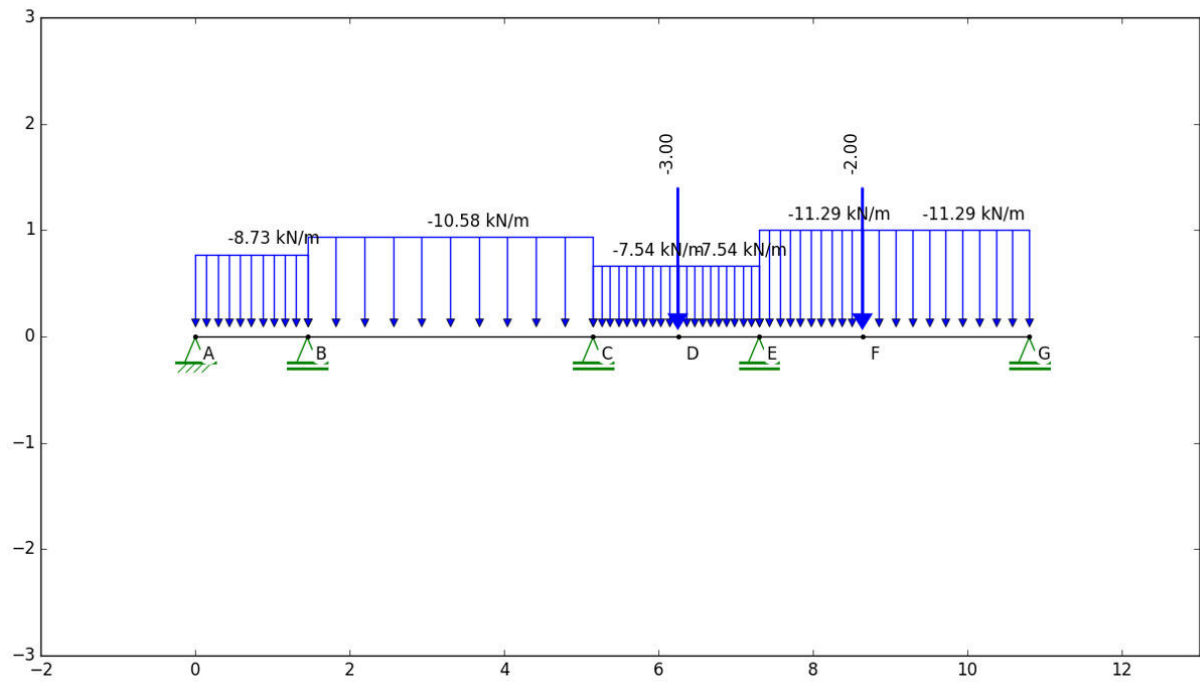
Figura 96 – Viga hiperestática sob influências das cargas.



Fonte: Ftool (2019)

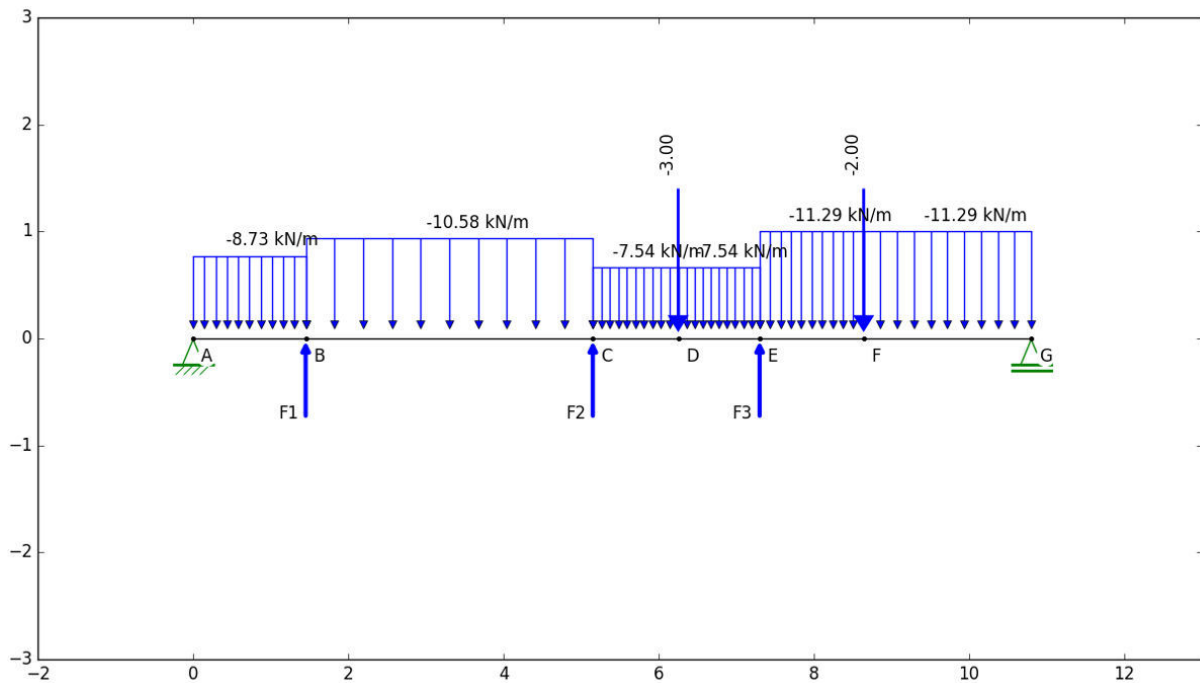
De forma análoga aos exemplos anteriores, de início, são introduzidos os dados de entrada, a fim de modelar o problema em questão. Dessa forma, a figura 97, 98, 99, 100, 101 e 102 ilustra o problema real gerado, problema auxiliar e problemas virtuais necessários para resolver o problema.

Figura 97- Exemplo 02 - Problema Estrutural



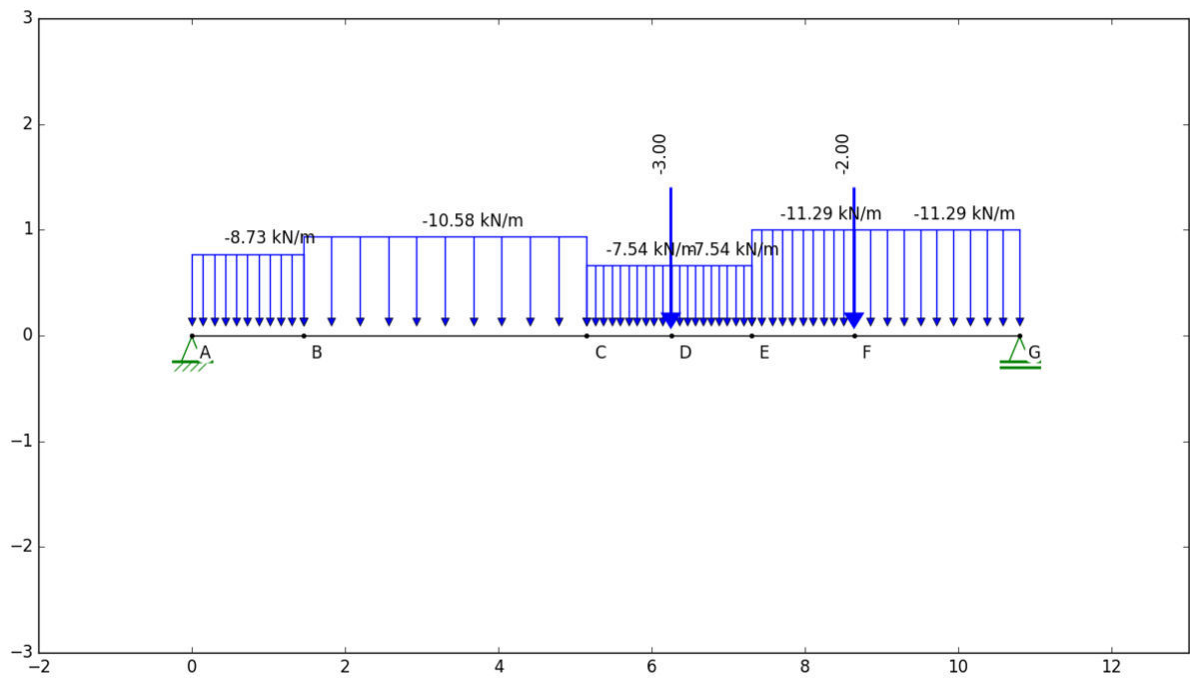
Fonte: Autoria Própria (2019)

Figura 98- Exemplo 02 - Problema Auxiliar



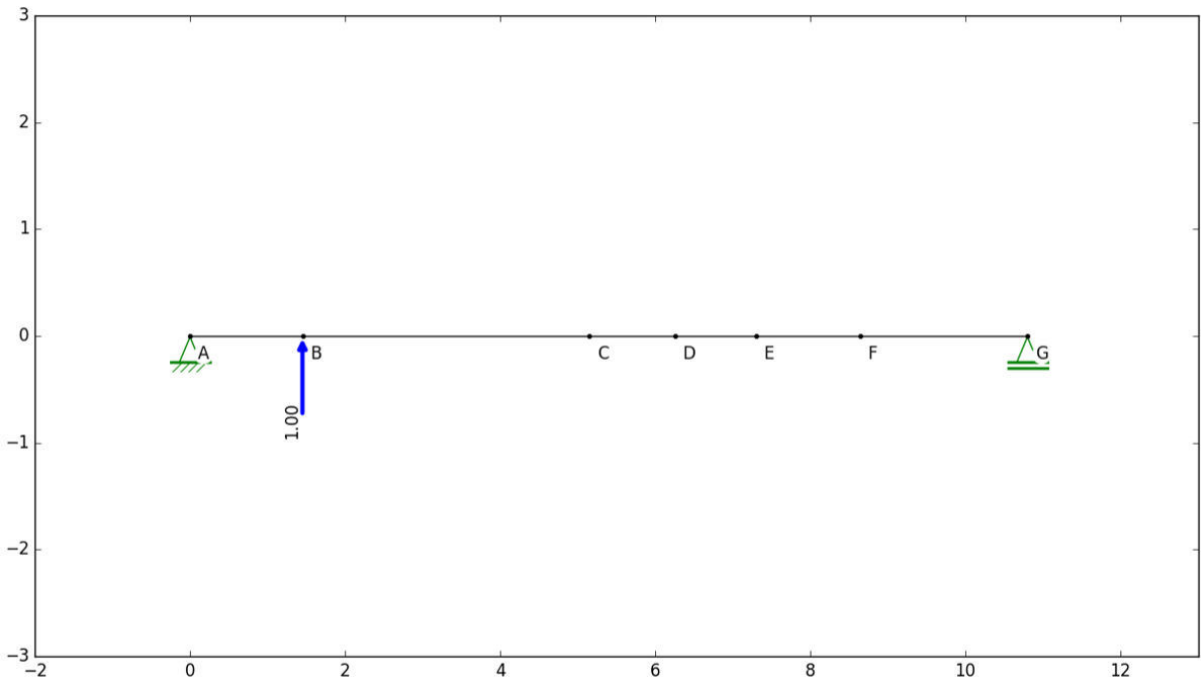
Fonte: Autoria Própria (2019)

Figura 99- Exemplo 02 - Problema 0



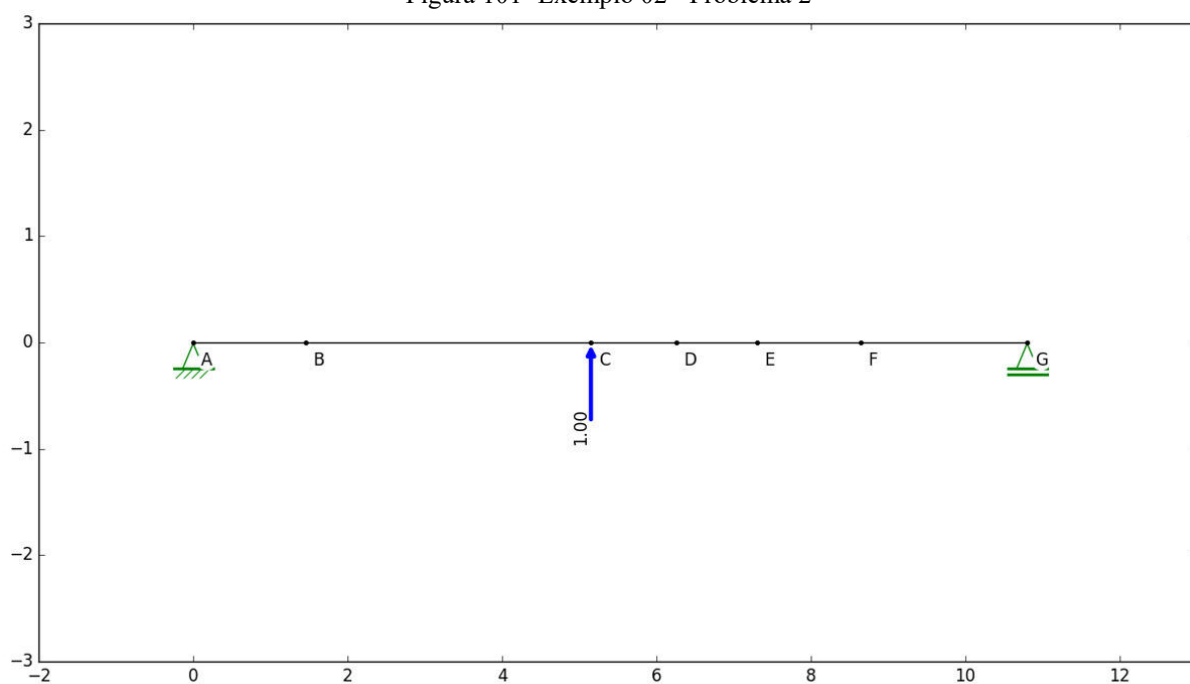
Fonte: Autoria Própria (2019)

Figura 100- Exemplo 02 - Problema 1



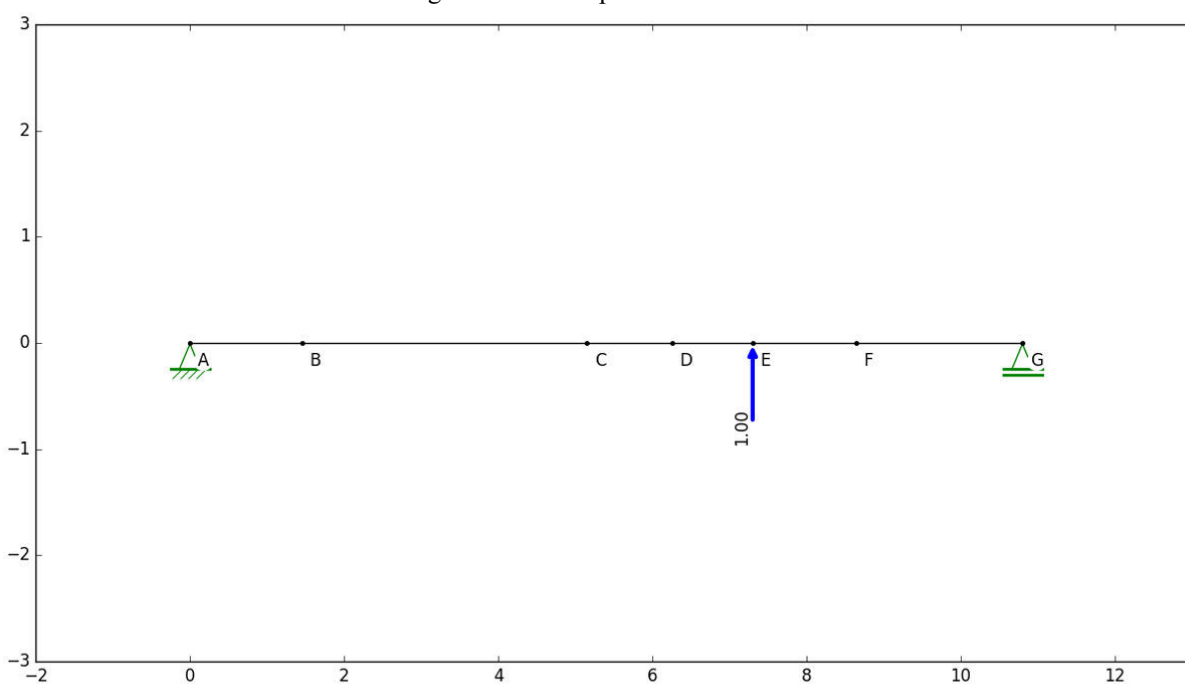
Fonte: Autoria Própria (2019)

Figura 101- Exemplo 02 - Problema 2



Fonte: Autoria Própria (2019)

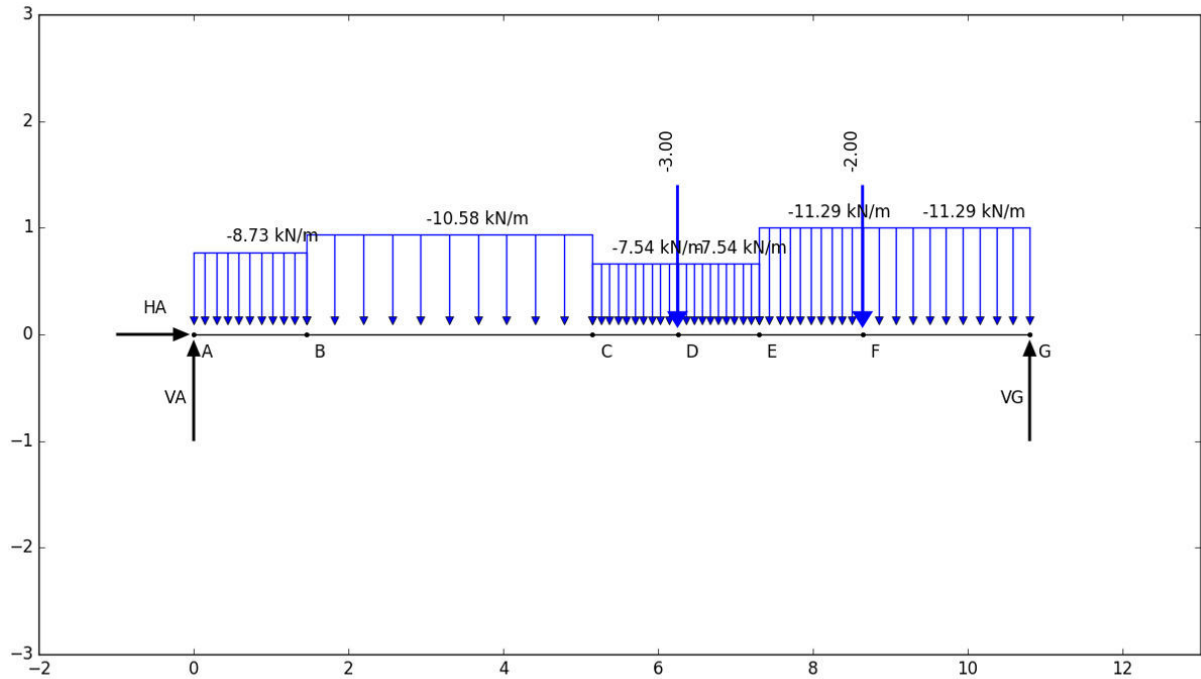
Figura 102- Exemplo 02 - Problema 3



Fonte: Autoria Própria (2019)

As figuras posteriores demonstram o DCL, seguido dos cálculos das reações de apoio para cada problema em questão.

Figura 103- DCL - exemplo 02 - Problema 0



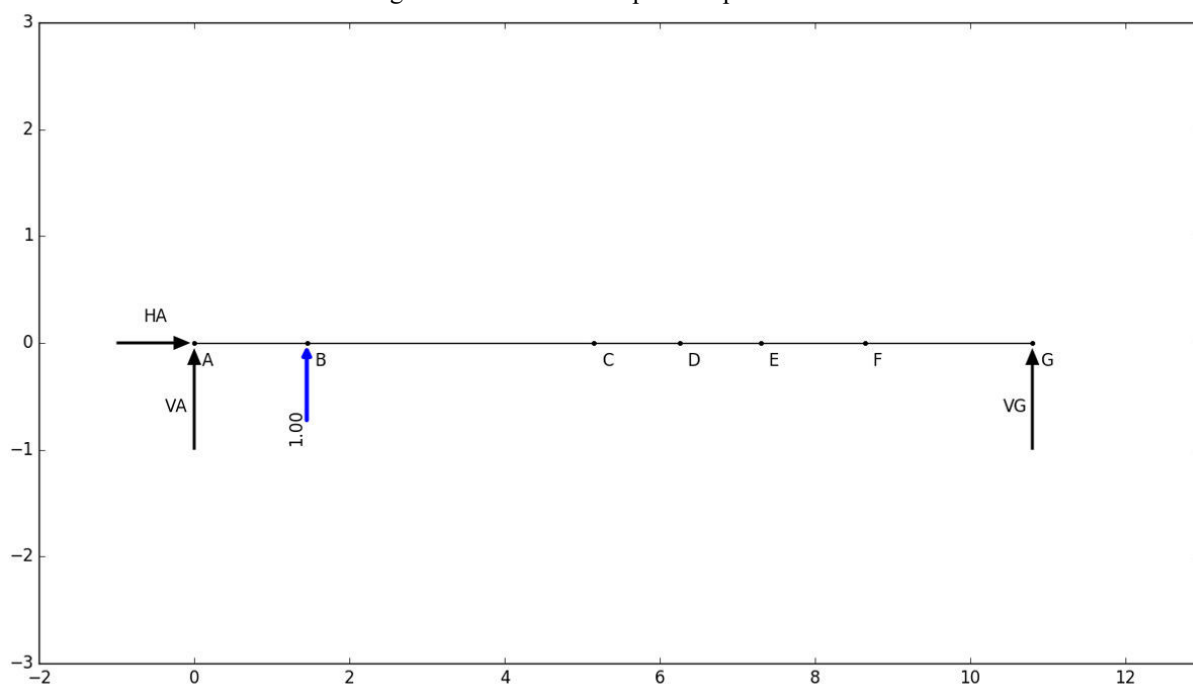
Fonte: Autoria Própria (2019)

Figura 104- Cálculo das reações de apoio - exemplo 02 - problema 0

$$\begin{aligned}
 \sum F_H &= 0 \\
 +H_A &= 0 \\
 \sum F_V &= 0 \\
 +V_G + V_A + (-3.00000) + (-2.00000) - 12.65850 - 39.14600 - 8.29400 - 7.91700 - 15.12860 - 24.38640 &= 0 \\
 \sum M_A &= 0 \\
 +V_G * 10.80000 + V_A * 0.00000 + (-3.00000 * 6.25000) + (-2.00000 * 8.64000) - (12.65850 \\
 &\quad 0.72500) - (39.14600 * 3.30000) - (8.29400 * 5.70000) - (7.91700 * 6.77500) - (15.12860 \\
 &\quad * 7.97000) - (24.38640 * 9.72000) \\
 H_A &= 0.00000 \\
 V_G &= 58.60310 \\
 V_A &= 53.92740
 \end{aligned}$$

Fonte: Autoria Própria (2019)

Figura 105- DCL - exemplo 02 - problema 1



Fonte: Autoria Própria (2019)

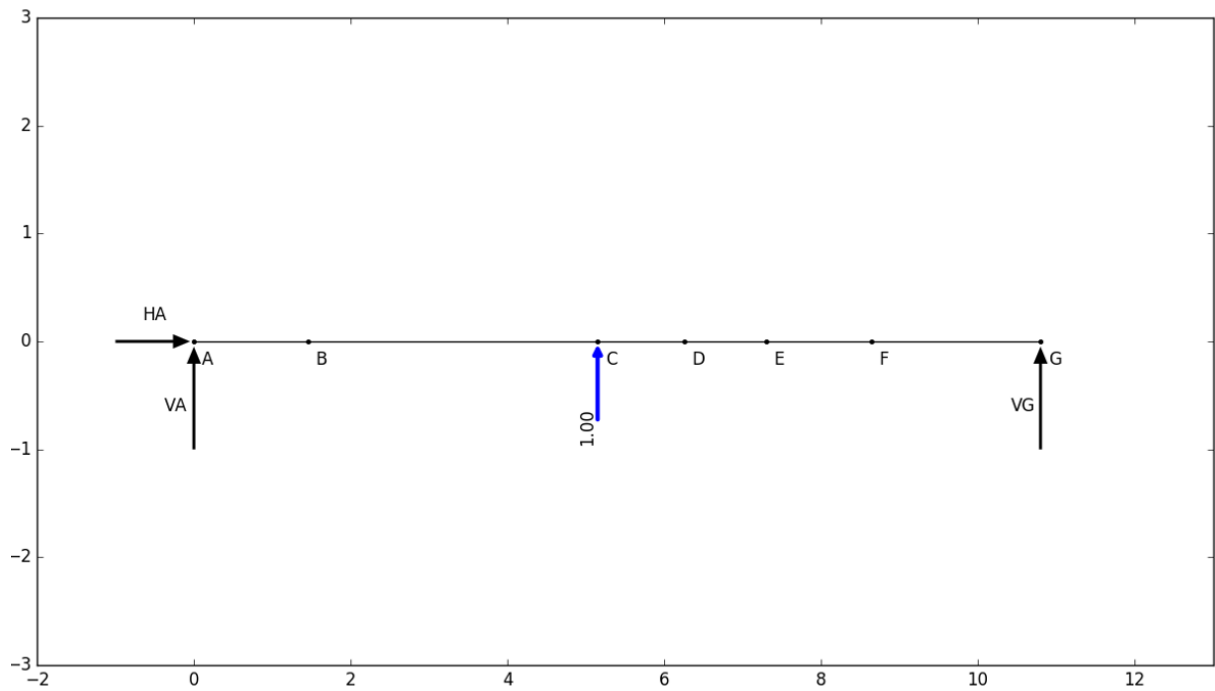
Figura 106- Cálculo das reações de apoio - exemplo 02 - problema 1

$$\begin{aligned} \sum F_H &= 0 \\ +H_A &= 0 \\ \sum F_V &= 0 \\ +V_G + V_A + 1.00000 &= 0 \\ \sum M_A &= 0 \\ +V_G * 10.80000 + V_A * 0.00000 + (1.00000 * 1.45000) &= 0 \end{aligned}$$

$$\begin{aligned} H_A &= 0.00000 \\ V_G &= -0.13426 \\ V_A &= -0.86574 \end{aligned}$$

Fonte: Autoria Própria (2019)

Figura 107- DCL - exemplo 02 - problema 2



Fonte: Autoria Própria (2019)

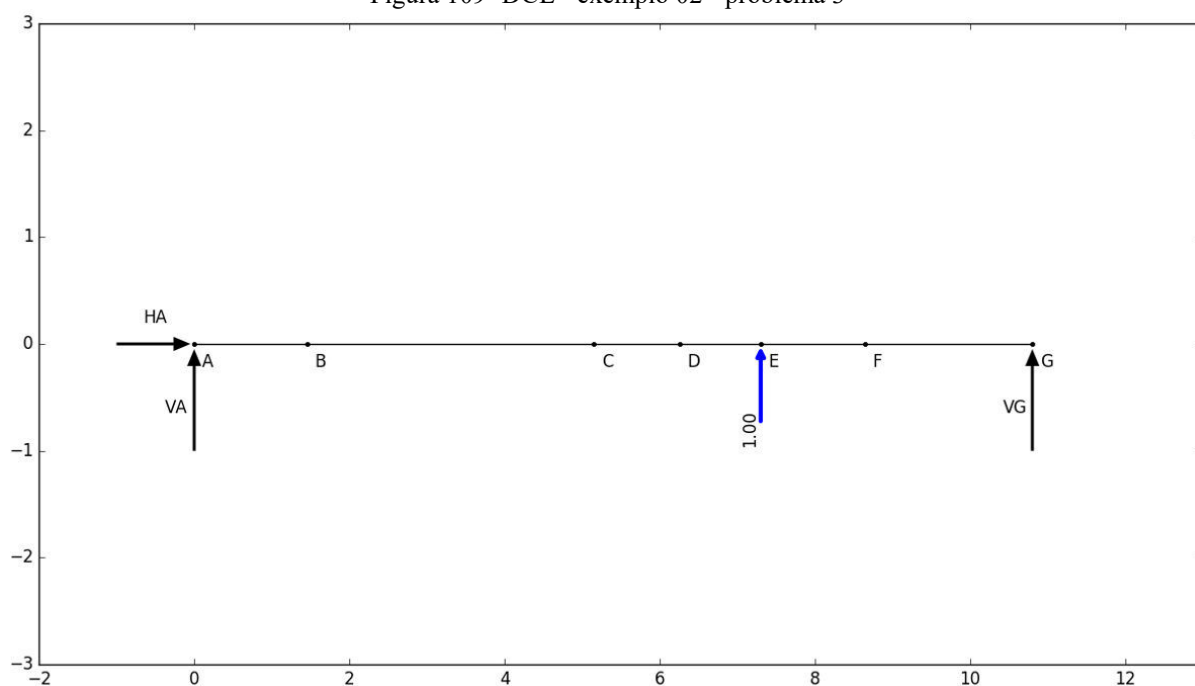
Figura 108- Cálculo das reações de apoio - exemplo 02 - problema 2

$$\begin{aligned} \sum F_H &= 0 \\ +H_A &= 0 \\ \sum F_V &= 0 \\ +V_G + V_A + 1.00000 &= 0 \\ \sum M_A &= 0 \\ +V_G * 10.80000 + V_A * 0.00000 + (1.00000 * 5.15000) &= 0 \end{aligned}$$

$$\begin{aligned} H_A &= 0.00000 \\ V_G &= -0.47685 \\ V_A &= -0.52315 \end{aligned}$$

Fonte: Autoria Própria (2019)

Figura 109- DCL - exemplo 02 - problema 3



Fonte: Autoria Própria (2019)

Figura 110- Cálculo das reações de apoio - exemplo 02 - problema 3

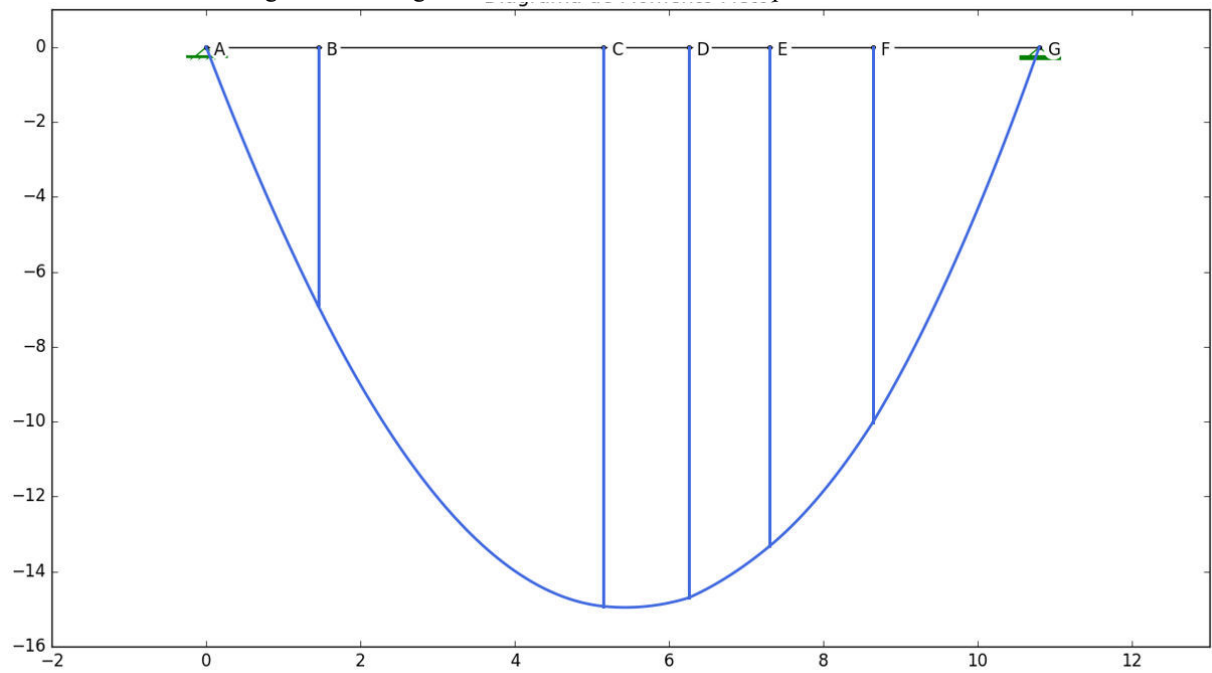
$$\begin{aligned}
 \sum F_H &= 0 \\
 +HA &= 0 \\
 \sum F_V &= 0 \\
 +VG + VA + 1.00000 &= 0 \\
 \sum M_A &= 0 \\
 +VG * 10.80000 + VA * 0.00000 + (1.00000 * 7.30000) &= 0
 \end{aligned}$$

$$\begin{aligned}
 HA &= 0.00000 \\
 VG &= -0.67593 \\
 VA &= -0.32407
 \end{aligned}$$

Fonte: Autoria Própria (2019)

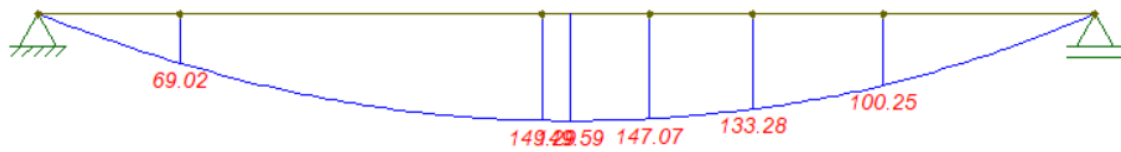
Para a leitura não se tornar desgastante e repetitiva, bem como o exemplo anterior demonstrou plenamente a capacidade e eficiência para descrever e apresentar as equações dos esforços internos e gráficos dos esforços. Com isso, agora, vamos apresentar apenas os diagramas de momento fletor e os comparar com os gerados no software Ftool.

Figura 111- Diagrama de momento fletor – Exemplo 02 - Problema 0



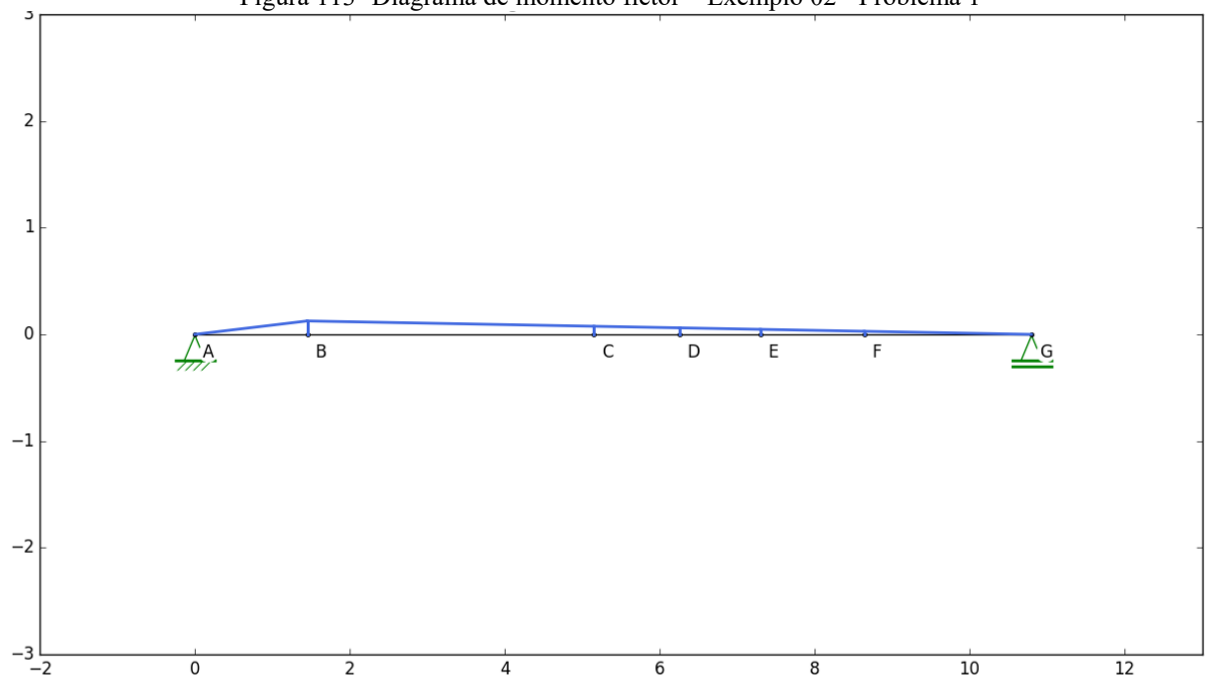
Fonte: Autoria Própria (2019)

Figura 112- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 0



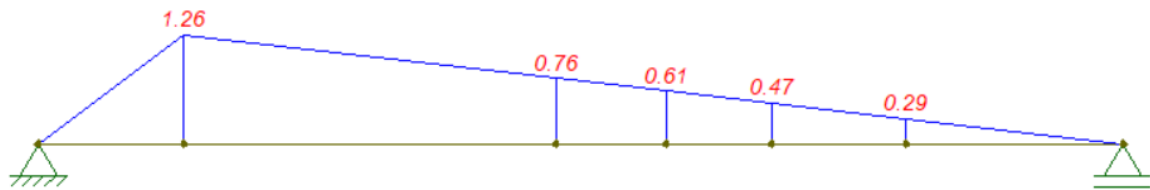
Fonte: Ftool (2019)

Figura 113- Diagrama de momento fletor – Exemplo 02 - Problema 1



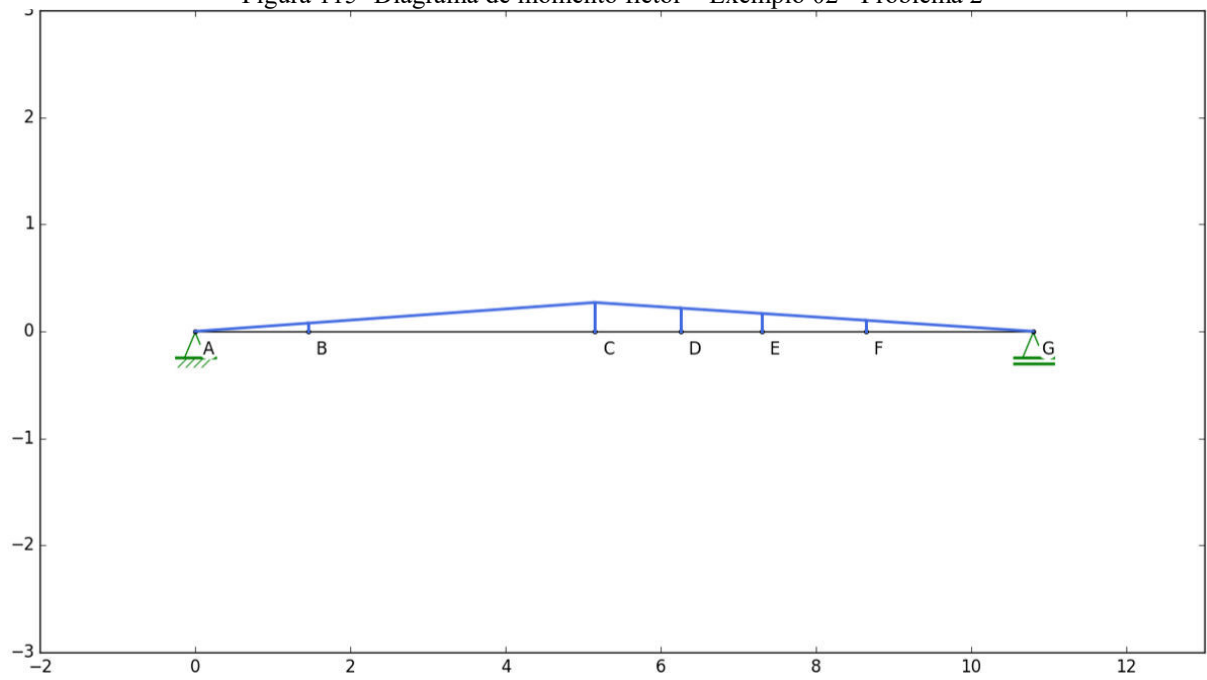
Fonte: Autoria Própria (2019)

Figura 114- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 1



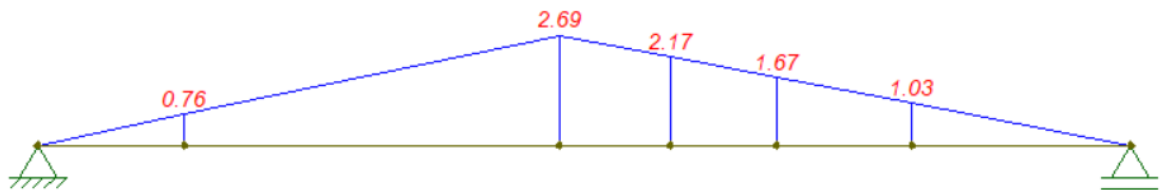
Fonte: Ftool (2019)

Figura 115- Diagrama de momento fletor – Exemplo 02 - Problema 2



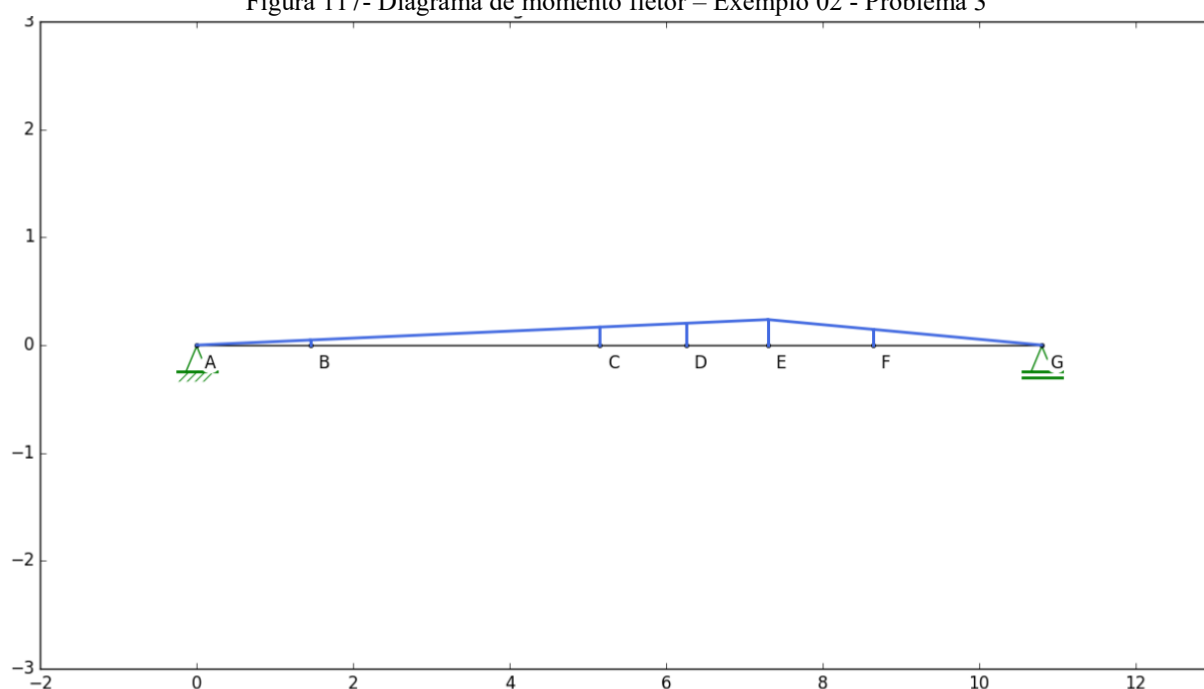
Fonte: Autoria Própria (2019)

Figura 116- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 2



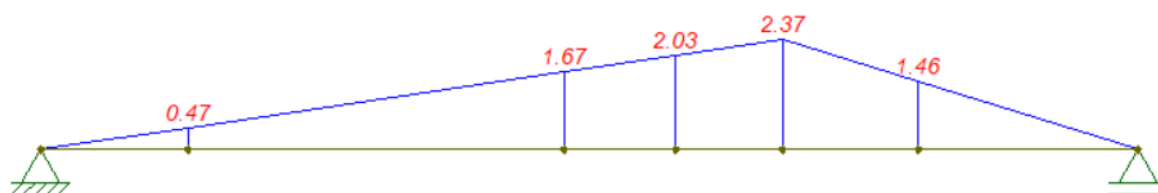
Fonte: Ftool (2019)

Figura 117- Diagrama de momento fletor – Exemplo 02 - Problema 3



Fonte: Autoria Própria (2019)

Figura 118- Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema 3

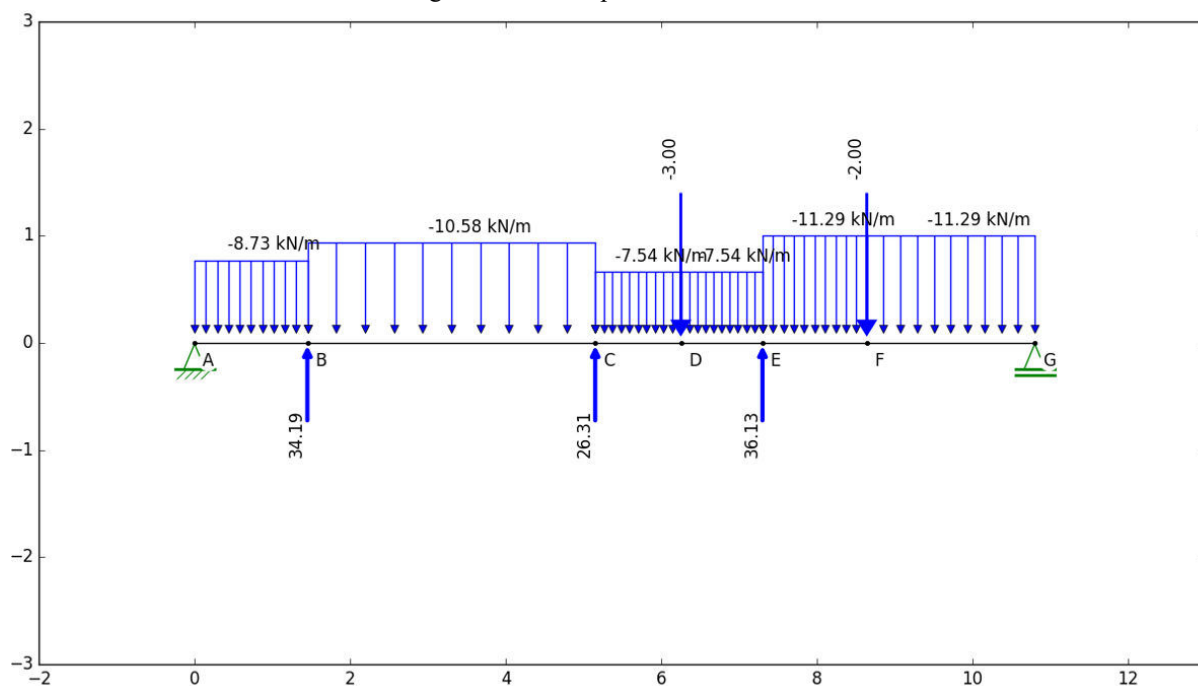


Fonte: Ftool (2019)

Posteriormente no resultado final analisaremos a consistência da informação para realmente validar o programa comparado a gráficos gerados no Ftool.

Com os dados obtidos com os problemas acima, agora podemos definir os valores de F1 e F2 e redesenhar o problema real de forma estaticamente determinada. Ver figura a seguir.

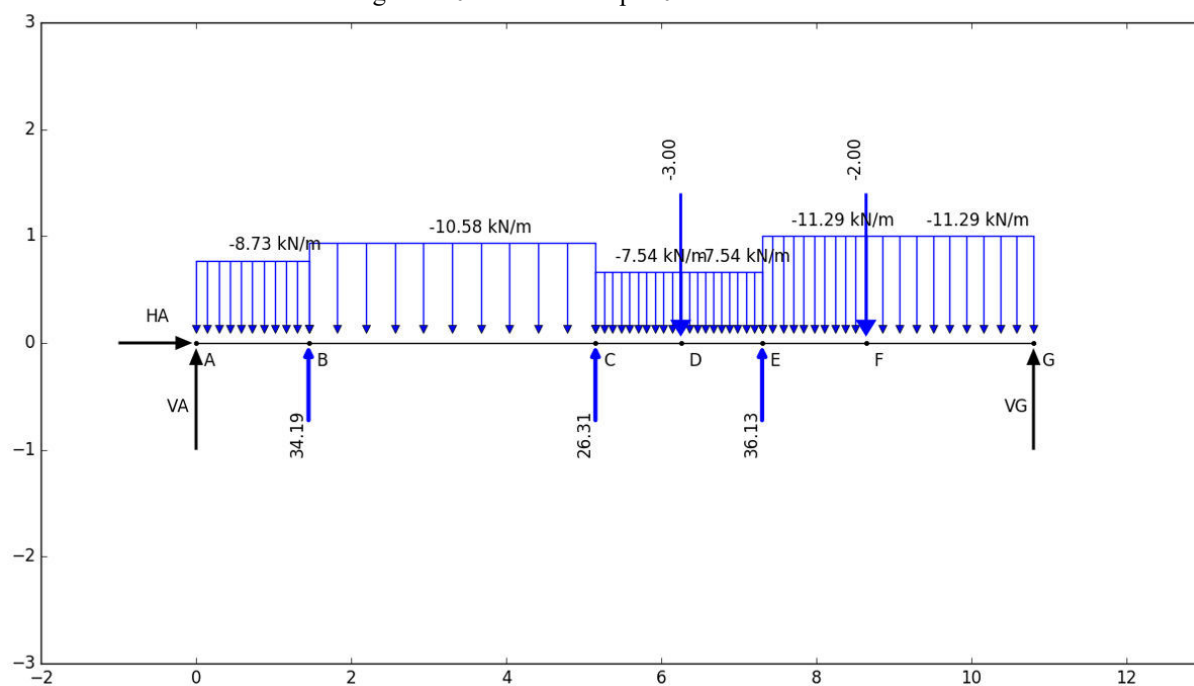
Figura 119- Exemplo 1 - Problema real



Fonte: Autoria Própria (2019)

Com as novas forças nos vínculos que haviam sido retirados, vamos seguir com a resolução do problema real. Ver figuras 120 e 121 para verificar o DCL e os cálculos das reações de apoio.

Figura 120- DCL - Exemplo 02 - Problema real



Fonte: Autoria Própria (2019)

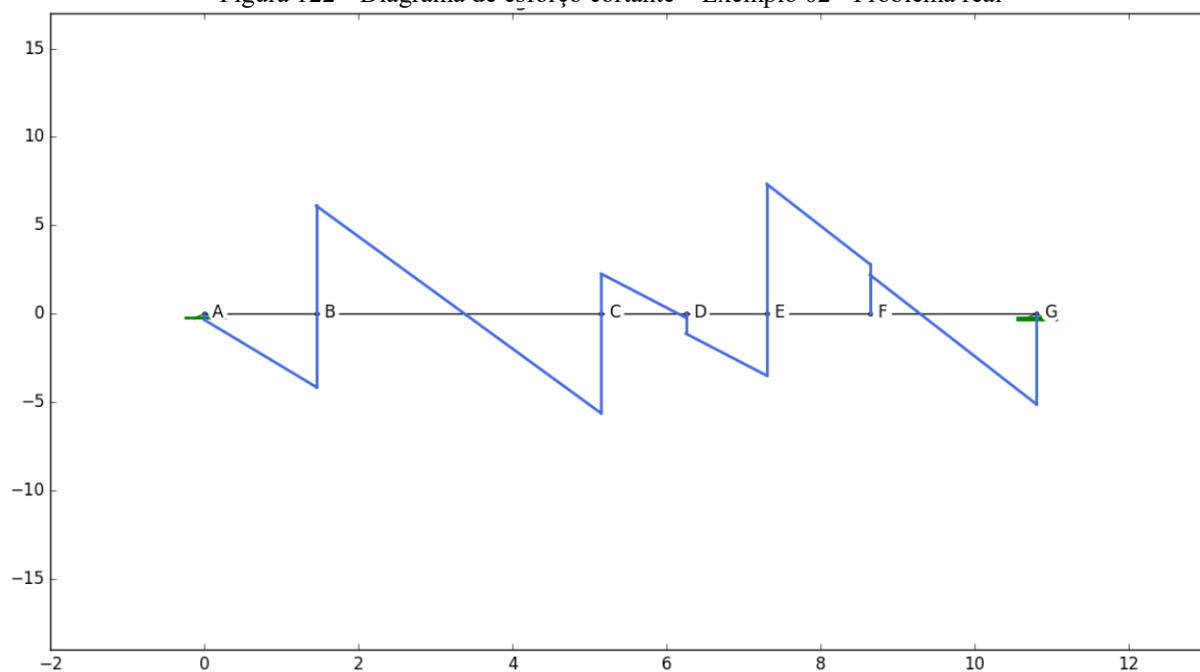
Figura 121- Cálculos das reações de apoio - exemplo 02 - Problema real

$$\begin{aligned}
 \sum F_H &= 0 \\
 +H_A &= 0 \\
 \sum F_V &= 0 \\
 +V_G + V_A + 34.19312 + 26.31031 + (-3.00000) + (-2.00000) + 36.12868 - 12.65850 - 39.14600 - 8.29400 - 7.1700 - 15.12860 - 24.38640 \\
 \sum M_A &= 0 \\
 +V_G * 10.80000 + V_A * 0.00000 + (34.19312 * 1.45000) + (26.31031 * 5.15000) + (-3.00000 * 6.25000) + (-2.00000 * 8.64000) + (36.12868 * 7.30000) - (12.65850 * 0.72500) - (39.14600 * 3.30000) - (8.29400 * 5.70000) - (7.91700 * 6.77500) - (15.12860 * 7.97000) - (24.38640 * 9.72000) \\
 V_A &= -1.14753 \\
 V_G &= 17.04592 \\
 H_A &= 0.00000
 \end{aligned}$$

Fonte: Autoria Própria (2019)

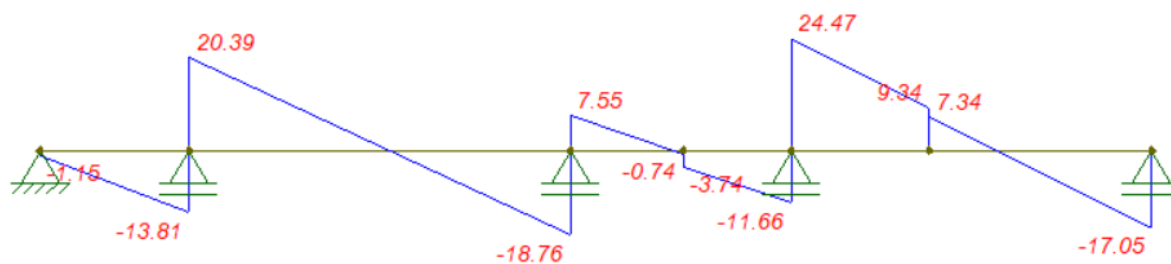
Como já foi dito e feito em exemplo anterior, vamos prosseguir para a comparação dos gráficos gerados com os gráficos do software Ftool. Figuras a seguir mostram diagramas de esforço normal, esforço cortante e momento fletor.

Figura 122-- Diagrama de esforço cortante – Exemplo 02 - Problema real



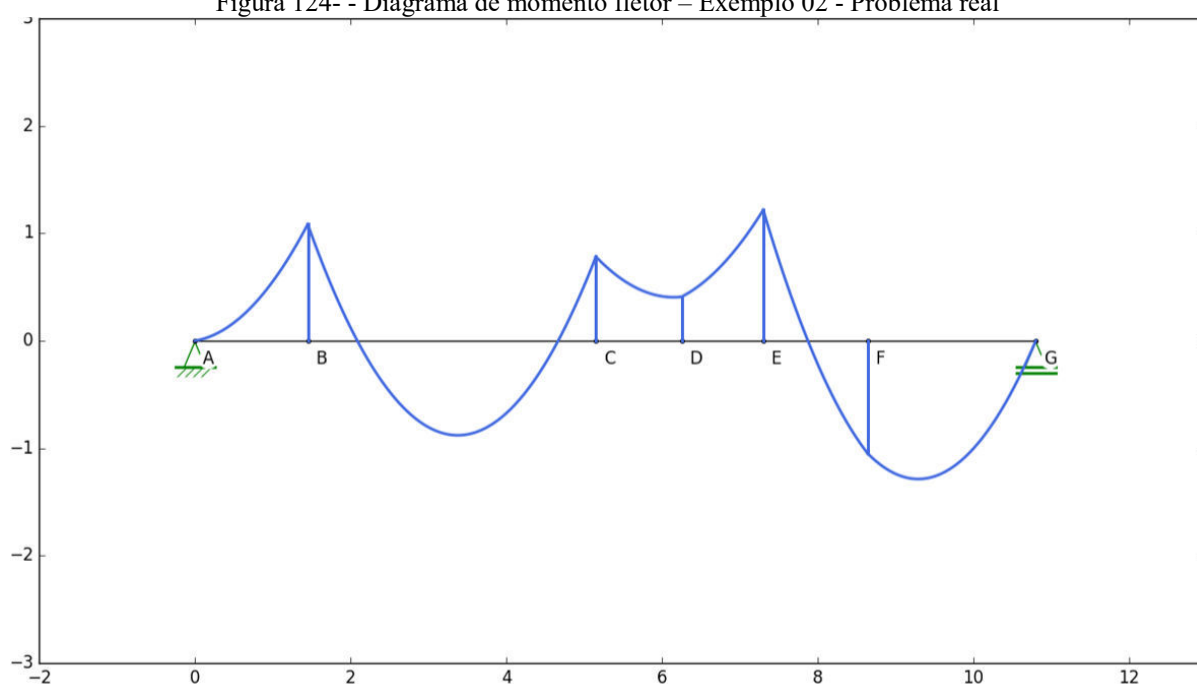
Fonte: Autoria Própria (2019)

Figura 123- - Diagrama de esforço cortante/ FTOOL - Exemplo 02 - Problema Real



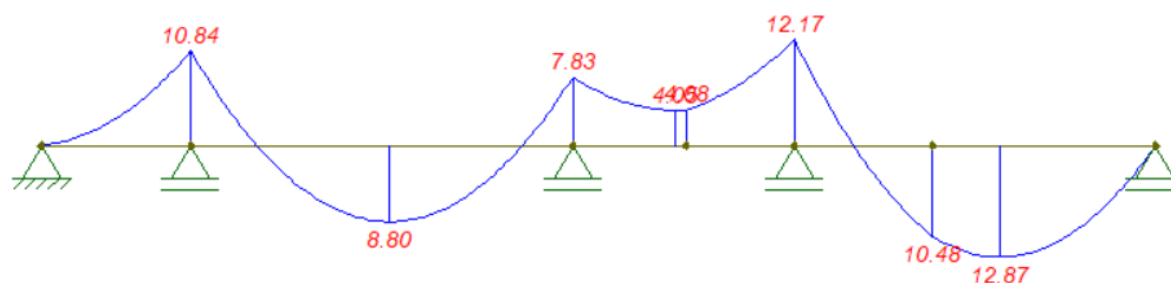
Fonte: Ftool (2019)

Figura 124- - Diagrama de momento fletor – Exemplo 02 - Problema real



Fonte: Autoria Própria (2019)

Figura 125- - Diagrama de momento fletor/ FTOOL - Exemplo 02 - Problema Real



Fonte: Ftool (2019)

A partir da análise de comparação dos gráficos do Ftool, chegamos ao resultado esperado de forma clara e precisa.

5 CONCLUSÃO

Com o trabalho apresentado, podemos concluir que o programa elaborado pode ser uma ferramenta de grande auxílio na resolução de problemas de vigas hiperestáticas, pois o mesmo gera um relatório detalhado e simplificados das questões envolvidas, fazendo com que, apesar, de ser um programa complexo, pois apresentou resultados semelhante e precisos comparados ao Ftool, porem de forma didático e simples, fornece soluções rápidas otimizando o tempo na resolução e análise dos resultados.

Possui diferencial na questão de gerar relatórios com todos os passos para resolução do problema em questão, de forma ilustrativa e detalhada. Com a facilidade para utilizar o software, ele poderá ser usado tanto pelos professores quanto os alunos, fazendo com que assim possam reduzir a análise de problemas específicos.

Para realização de futuros trabalhos ligados a esse tema, pode-se incluir a análise de vigas hiperestáticas, pelo método dos deslocamentos, 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.

Outra implantação pode ser o desenvolvimento de um relatório para o cálculo de vigas armadas, demonstrando um passo a passo de todo o dimensionamento, desde obtenção de suas cargas até obtenção de sua área de aço.

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.
- 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.
- LAROVERE, Henriette Lebre; MORAES, Poliana Dias de. **ANÁLISE ESTRUTURAL II**. Florianópolis: Ufsc, 2005.
- 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.

MCCOMARC, Jack C. **ANALISE ESTRUTURAL: Usando Metodos Clássicos e Métodos Matriciais**. 4. ed. Rio de Janeiro: Ltc, 2009. 480 p. Tradução e revisão técnica Amir Kurban.

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

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.

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.

VIEIRA, Rodrigo Cuberos; TORRES, Ana Paula Vedoato. **Estruturas Hiperestáticas**. Londrina: Editora e Distribuidora Educacional S.a., 2018.