

UNIVERSIDADE DE RIBEIRÃO PRETO
CURSO DE ENGENHARIA CIVIL

PEDRO SARTORI DIAS DOS REIS

SOFTWARE EDUCACIONAL PARA VERIFICAÇÃO DE ESFORÇOS EM
ESTRUTURAS

RIBEIRÃO PRETO
2023

PEDRO SARTORI DIAS DOS REIS

SOFTWARE EDUCACIONAL PARA VERIFICAÇÃO DE ESFORÇOS EM
ESTRUTURAS

Monografia apresentada à Universidade de
Ribeirão Preto UNAERP, como requisito para
a obtenção do título de Bacharel em Engenharia
Civil.

Orientadora: Prof.^a Dra. Tamiris Luiza Soares
Lanini

Ribeirão Preto
2023

PEDRO SARTORI DIAS DOS REIS
SOFTWARE EDUCACIONAL PARA VERIFICAÇÃO DE ESFORÇOS EM
ESTRUTURAS

Monografia apresentada à Universidade de
Ribeirão Preto UNAERP, como requisito para
a obtenção do título de Bacharel em Engenharia
Civil.

Aprovação: ____/____/____

Prof.^a Dra. Tamiris Luiza Soares Lanini
Universidade de Ribeirão Preto
Orientadora

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por me conceder força e sabedoria ao longo desta jornada acadêmica. À minha família, em especial aos meus pais, pela constante inspiração, apoio incondicional e sacrifícios feitos para a minha educação.

À minha dedicada orientadora, Tamiris, expresso minha profunda gratidão por sua orientação, paciência e incentivo, que foram fundamentais para o desenvolvimento deste trabalho.

À professora Aline, responsável pela disciplina de TCC, e a todos que, de alguma forma, compartilharam comigo esse percurso acadêmico, o meu sincero obrigado. Este trabalho é fruto não apenas do meu esforço, mas do apoio e colaboração de muitos que estiveram ao meu lado.

RESUMO

No contexto pedagógico da Engenharia Civil, o presente trabalho tem como propósito o desenvolvimento de um *software* que sirva como ferramenta de apoio para estudantes, proporcionando uma compreensão aprofundada dos processos e métodos de cálculo de estruturas isostáticas e hiperestáticas. A pesquisa realiza uma revisão abrangente da literatura, contemplando temas essenciais de análise estrutural, resistência dos materiais e programação. Com uma abordagem centrada na facilitação da compreensão de métodos de análise de estruturas em um espaço tridimensional, o *software* é projetado para tornar as relações estruturais mais acessíveis e intuitivas, contribuindo assim para uma formação acadêmica mais eficaz e completa.

Palavras-chave: *Software* educacional. Análise de estruturas. Modelagem tridimensional.

ABSTRACT

In the pedagogical context of Civil Engineering, this work aims to develop a software that serves as a support tool for students, providing an in-depth understanding of the processes and methods for calculating both statically determinate and indeterminate structures. The research encompasses a comprehensive literature review, covering essential topics in structural analysis, materials strength, and programming. With an approach focused on facilitating the understanding of structural analysis methods in a three-dimensional space, the software is designed to make structural relationships more accessible and intuitive, thereby contributing to a more effective and comprehensive academic education.

Keywords: Educational software. Structural analysis. Three-dimensional modeling.

LISTA DE FIGURAS

Figura 1 - Viga suportando carga pontual	11
Figura 2 – Representação simbólica de um apoio móvel	17
Figura 3 - Representação simbólica de um apoio fixo	17
Figura 4 - Representação simbólica do engaste.....	18
Figura 5 - Forças agindo em uma viga	19
Figura 6 - Braço de alavanca	20
Figura 7 - Momentos aplicados em uma viga.....	20
Figura 8 – Tensão normal.....	21
Figura 9 - Tensão cisalhante.....	22
Figura 10 – Força normal de compressão.....	23
Figura 11 – Força normal de tração.....	23
Figura 12 – Força cortante.....	24
Figura 13 – Momento fletor.....	24
Figura 14 – Carga distribuída	25
Figura 15 – Força resultante de uma carga distribuída.....	26
Figura 16 – Translação de estrutura hipostática	28
Figura 17 – Rotação de estrutura hipostática	28
Figura 18 – Estruturas isostáticas	29
Figura 19 – Viga hiperestática.....	31
Figura 20 - Sistemas principais e redundantes	34
Figura 21 – Deformações, rotações e coeficientes de flexibilidade	36
Figura 22 – Pontos na reta r.....	37
Figura 23 – Interface do <i>Visual Studio Community</i>	49
Figura 24 – Interface do <i>GIMP</i>	50
Figura 25 – Conversão de posições tridimensionais em bidimensionais	51
Figura 26 – Rotação dos pontos	52
Figura 27 – Viga isostática estudada	62
Figura 28 – Aplicando o método das seções	63
Figura 29 - Diagrama de força cortante.....	64
Figura 30 - Diagrama de momento fletor	64
Figura 31 – Diagrama de força normal.....	65

Figura 32 – Viga hiperestática estudada.....	65
Figura 33 – Deslocamentos e rotações dos sistemas isostáticos	66
Figura 34 – Esforços externos dos sistemas isostáticos	67
Figura 35 – Momentos fletores dos sistemas isostáticos	68
Figura 36 – Reações de apoio da viga hiperestática.....	69
Figura 37 – Aplicação do método das seções na viga hiperestática.....	70
Figura 38 – Modelo da viga isostática estudada.....	71
Figura 39 – Apoios da viga isostática estudada.....	72
Figura 40 – Inserção da carga distribuída na viga isostática	72
Figura 41 – Inserção das forças pontuais na viga isostática	73
Figura 42 – Diagrama de momento fletor da viga isostática.....	73
Figura 43 – Diagrama de força cortante da viga isostática.....	74
Figura 44 – Modelo da viga hiperestática estudada	74
Figura 45 – Apoios da viga hiperestática estudada	75
Figura 46 – Inserção dos carregamentos na viga hiperestática	76
Figura 47 – Equações de momento fletor da viga hiperestática.....	76
Figura 48 – Equações de força cortante da viga hiperestática.....	77
Figura 49 – Resultado do sistema principal.....	78
Figura 50 – Resultado do sistema unitário 1	79
Figura 51 – Resultado do sistema unitário 2	80
Figura 52 – Interface do aplicativo.....	81
Figura 53 – Cargas provenientes de outras estruturas	82
Figura 54 – Apoios provenientes de outras estruturas.....	82

SUMÁRIO

1	INTRODUÇÃO	11
1.1	A DIFICULDADE EM SE CALCULAR ESTRUTURAS HIPERESTÁTICAS	12
1.2	O SOFTWARE COMO FACILIDADOR NOS ESTUDOS	13
1.3	OBJETIVOS	13
1.3.1	Objetivos Específicos	14
1.4	JUSTIFICATIVA	14
2	REVISÃO DA LITERATURA.....	16
2.1	APOIOS E VÍNCULOS	16
2.2	FORÇAS.....	18
2.3	MOMENTO.....	19
2.4	TENSÃO.....	21
2.5	ESFORÇOS INTERNOS SOLICITANTES	22
2.6	CARGAS DISTRIBUÍDAS	25
2.7	CONDIÇÕES DE EQUILÍBRIO	26
2.8	ESTRUTURAS HIPOSTÁTICAS, ISOSTÁTICAS E HIPERESTÁTICAS	27
2.9	ESTRUTURAS HIPOSTÁTICAS	27
2.10	ESTRUTURAS ISOSTÁTICAS	28
2.11	OBTENÇÃO DE ESFORÇOS EM ESTRUTURAS ISOSTÁTICAS.....	29
2.12	DIAGRAMAS DE ESFORÇOS INTERNOS.....	30
2.13	ESTRUTURAS HIPERESTÁTICAS	30
2.14	PRINCÍPIO DOS TRABALHOS VIRTUAIS	32
2.15	MÉTODO DA FORÇA	34
2.16	EQUAÇÕES PARAMÉTRICAS DA RETA.....	37
2.17	ESCALONAMENTO OU ELIMINAÇÃO GAUSSIANA.....	38
2.18	A LINGUAGEM DE PROGRAMAÇÃO C#	39
2.18.1	A Plataforma .NET (<i>dot NET</i>).....	40
2.18.2	Windows Forms.....	40
2.19	ORIENTAÇÃO A OBJETOS	40
2.19.1	Classes, Objetos, Propriedades e Métodos	41
2.19.2	Encapsulamento	42

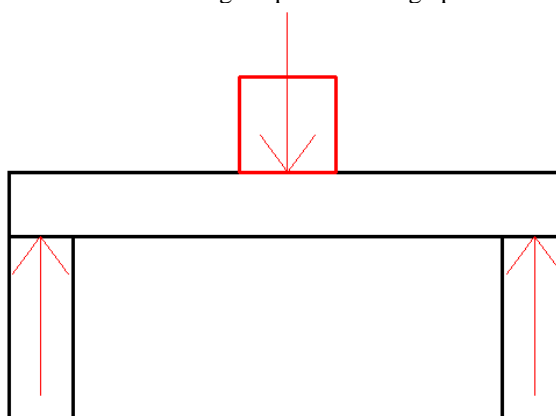
2.19.3	Herança	42
2.19.4	Polimorfismo	43
2.20	SOFTWARES EXISTENTES	43
2.21	ESTADO DA ARTE	45
3	METODOLOGIA.....	48
3.1	FERRAMENTAS COMPUTACIONAIS UTILIZADAS NO DESENVOLVIMENTO.....	49
3.2	CRIAÇÃO DO SOFTWARE.....	50
3.2.1	Criação de Classes Utilitárias	52
3.2.1.1	Classe para cálculos gerais	53
3.2.1.2	Classe para as equações paramétricas de reta.....	53
3.2.1.3	Classes para as equações de Momento Fletor, Força Cortante e Normal.....	54
3.2.1.4	Classes que representam os esforços externos.....	54
3.2.2	Criação dos Apoios.....	54
3.2.3	Criação das Estruturas	55
3.2.3.1	Classe auxiliar para as estruturas	56
3.2.4	Gerenciador de ambiente	56
3.2.4.1	Métodos para atualização do ambiente	57
3.2.4.2	Métodos para a interação com o usuário	57
3.2.5	Criação dos Formulários.....	58
3.2.5.1	Formulário principal.....	59
3.2.5.2	Formulários para inserção de cargas	59
3.2.5.3	Formulário para a modificação do tipo de apoio	60
3.2.5.4	Formulário para a modificação da relação entre as estruturas.....	60
3.2.5.5	Formulário para a obtenção dos valores no ponto	60
3.2.6	Mecanismos de Salvamento e Carregamento de Arquivos	60
3.2.7	Aplicação de Testes	61
4	ESTUDO DE CASO	62
4.1	ESTUDO DE CASO 1: OBTENÇÃO DOS ESFORÇOS INTERNOS EM UMA VIGA ISOSTÁTICA	62
4.2	ESTUDO DE CASO 2: OBTENÇÃO DE ESFORÇOS INTERNOS EM UMA VIGA HIPERESTÁTICA	65
5	RESULTADOS	71
5.1	RESULTADOS DO ESTUDO DE CASO 1	71

5.2	RESULTADOS DO ESTUDO DE CASO 2.....	74
5.3	OUTROS RESULTADOS PERTINENTES	80
6	CONCLUSÃO.....	83
	REFERÊNCIAS.....	85

1 INTRODUÇÃO

Segundo Botelho (1998) a determinação de esforços em estruturas tem como objetivo analisar o que ocorre em uma estrutura quando submetida a ações, tanto externas, como cargas provenientes de uso, ocupação, vento e demais, quanto intrínsecas a esta, como seu peso próprio. Com isso é possível realizar o dimensionamento de estruturas de forma que esta resista aos esforços solicitantes sobre ela. Tomando como exemplo uma viga apoiada sobre dois pilares (Figura 1), em que sobre ela se encontra uma carga pontual que age como um esforço ativo sobre a estrutura, tem-se que os pilares exercem um esforço reativo na viga fazendo com que esta esteja em equilíbrio, ou seja, imóvel.

FIGURA 1 - Viga suportando carga pontual



Fonte: Elaborado pelo autor, 2023

Estes esforços ativos e reativos fazem com que internamente à estrutura ajam esforços internos solicitantes, sendo eles as forças normais, que comprimem e tracionam a peça, forças tangenciais cortantes, momento fletor e momento torsor. Os esforços internos solicitantes por sua vez geram esforços internos resistentes, sendo eles as tensões de tração, compressão e cisalhamento. Estes últimos, a depender de sua intensidade podem comprometer a natureza do material de modo a leva-lo à sua degeneração e rompimento, causando assim o colapso da estrutura (Martha, 2013).

Dado isso, o estudo acerca de como agem estes esforços nas estruturas é de extrema importância, já que ele possibilita a escolha correta dos materiais, seu dimensionamento e disposição de modo a garantir a segurança, viabilidade econômica e estética.

Por este motivo, o cálculo dos esforços internos de uma estrutura é de suma importância na análise estrutural, e sua obtenção pode ainda ser muito trabalhosa a depender do tipo da

estrutura, isso porque as estruturas podem ser classificadas conforme o seu grau de hiperestaticidade, ou seja, seu grau de indeterminação estática, que expressa a diferença entre o número de valores desconhecidos em um sistema estático e o número de equações existentes para solucioná-lo. As estruturas que apresentam grau de hiperestaticidade nulo podem ter seus esforços internos encontrados mais facilmente, já que o número de incógnitas possibilita a resolução do problema estático por meio das equações disponíveis, estas estruturas são chamadas de isostáticas. Já as estruturas que possuem um grau de hiperestaticidade maior que zero apresentam uma maior dificuldade na obtenção dos esforços internos, já que seu número de incógnitas não possibilita a resolução de seu problema estático apenas utilizando-se as equações de equilíbrio disponíveis, sendo necessária a utilização de outros métodos para tal, estas estruturas são chamadas de hiperestáticas (Martha, 2013).

1.1 A DIFICULDADE EM SE CALCULAR ESTRUTURAS HIPERESTÁTICAS

Como pontuado anteriormente, as estruturas hiperestáticas possuem mais incógnitas do que equações de equilíbrio, deste modo, as equações existentes são insuficientes para obter-se os valores das reações de apoio, por isso, para obtê-los usa-se o Método da Força, que é um método de aplicação do Princípio dos Trabalhos Virtuais utilizado para a obtenção de deslocamentos e rotações em pontos da estrutura.

A utilização deste método para a resolução de sistemas hiperestáticos é feita pela retirada de reações nos apoios, de modo a separar a estrutura entre uma estrutura isostática, e outras n estruturas redundantes para cada uma das reações retiradas da estrutura original.

Com base nisso, devem ser encontrados os deslocamentos e/ou rotações nos pontos em que foram retiradas as reações, tanto da estrutura isostática gerada a partir da retirada das reações, quanto das estruturas redundantes geradas para cada reação que foi retirada.

O objetivo do processo é gerar equações de compatibilidade de modo que a somatória dos deslocamentos e/ou rotações em cada um dos pontos seja igual a zero, de modo a formar um sistema de n equações resolvível, onde as incógnitas são os valores das reações retiradas da viga original.

Deste modo, resolvendo-se as equações e encontrando os valores das reações removidas a viga original possuirá incógnitas suficientes para serem encontradas com as três equações de equilíbrio.

No entanto, este é um processo trabalhoso, e a depender do nível de hiperestaticidade da estrutura podem ser geradas inúmeras estruturas redundantes, e, por conseguinte um grande número de equações de compatibilidade tornando inviável a resolução manual do problema.

Um outro grande problema da utilização deste método é o fato de que a determinação de cada um dos deslocamentos envolve a obtenção das equações dos esforços, principalmente as equações de momento fletor, no caso das vigas, que a depender da viga e da disposição e natureza das cargas, podem existir em grande quantidade e complexidade.

1.2 O SOFTWARE COMO FACILIDADOR NOS ESTUDOS

Devido a esta grande complexidade os estudantes podem enfrentar dificuldades em realizar estes processos de cálculo, sendo uma das principais dificuldades o fato de não conseguirem confirmar se os passos intermediários de cálculo estão corretos.

Isso pode ser um grande obstáculo, já que pequenos erros acumulados podem comprometer o resultado final, além de serem difíceis de serem detectados em processos de cálculo que muitas vezes podem ser longos e repetitivos.

Nesse contexto, um *software* que realiza os cálculos de forma automatizada e que dê *feedbacks* que possam auxiliar os estudantes durante as etapas intermediárias de cálculo pode desempenhar um papel fundamental no aprendizado, fornecendo o suporte necessário e dando confiança ao estudante para que possa prosseguir com seus cálculos.

Uma outra vantagem na utilização de *softwares* é que, ao oferecer os *feedbacks* de forma automatizada, o estudante pode focar mais na compreensão dos conceitos dos cálculos, em vez de se preocuparem demasiadamente com os processos repetitivos.

1.3 OBJETIVOS

Diante de tais considerações este estudo tem como objetivo desenvolver um *software* de análise estrutural tridimensional que sirva como uma ferramenta educacional de apoio para estudantes de Engenharia Civil, fornecendo uma solução prática e acessível para as dificuldades enfrentadas pelos estudantes ao lidar com os cálculos complexos envolvendo a análise de estruturas.

1.3.1 Objetivos Específicos

De modo a cumprir este propósito foram estabelecidos os seguintes objetivos específicos

- Realizar um estudo da literatura relacionada à análise estrutural e resistência dos materiais, buscando métodos utilizados para resolver problemas referentes a cálculo de estruturas isostáticas e hiperestáticas.
- Definir as funcionalidades e características que o *software* deverá ter, incluindo a interface interativa, a criação de modelos tridimensionais discretizados, a capacidade de interação entre as estruturas, como identificação automática de apoios e de transmissão de esforços e a geração de resultados, como diagramas de esforço cortante, deformação e momento fletor, bem como as reações dos apoios e equações de momento fletor.
- Projetar uma interface amigável e intuitiva, de modo a possibilitar a fácil utilização do *software* por parte dos estudantes.
- Desenvolver uma arquitetura de *software* funcional, implementando as funcionalidades descritas na fase de definição, utilizando os métodos estudados durante a fase de revisão da literatura.
- Realizar testes para validar as implementações do *software* por meio de exemplos e estudos de caso, de modo a verificar a precisão dos resultados e a usabilidade.
- Apresentar os resultados obtidos no desenvolvimento do *software* por meio de exemplos e estudos de caso, evidenciando como ele pode efetivamente servir como uma ferramenta de auxílio nos estudos de estruturas.

1.4 JUSTIFICATIVA

O *software* tem como propósito simplificar o processo de cálculo, permitindo que os estudantes obtenham *feedbacks* visuais e compreendam de forma mais eficaz os conceitos fundamentais das disciplinas de estruturas. Por meio do *software*, os estudantes poderão criar modelos tridimensionais das estruturas, aplicar cargas, estabelecer relações de transmissão das cargas entre as estruturas e visualizar os resultados detalhados, como reações de apoio e diagramas de esforço cortante, deformação e momento fletor. Isso possibilitará que os

estudantes possam realizar seus estudos com mais segurança durante as etapas da realização dos cálculos, bem como melhorará a compreensão dos problemas conceituais das disciplinas.

2 REVISÃO DA LITERATURA

Na revisão da literatura deste trabalho, foram realizadas pesquisas abrangentes em várias áreas interdisciplinares, abordando conceitos essenciais relacionados à análise de estruturas, resistência de materiais e programação em C#. O objetivo principal dessa busca bibliográfica foi aprofundar o entendimento das teorias e práticas que fundamentam a análise de estruturas tridimensionais, bem como explorar os princípios da resistência dos materiais aplicados a projetos estruturais.

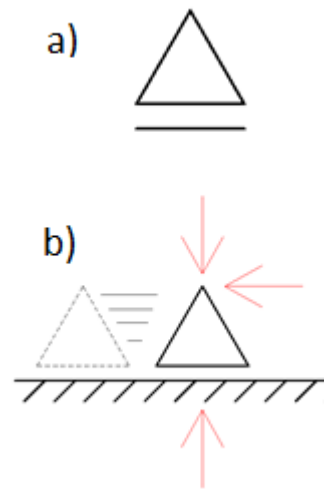
Além disso, investigou-se a linguagem de programação C# e suas aplicações no desenvolvimento de *software* voltado para a engenharia civil. A revisão da literatura oferecerá uma base sólida para o desenvolvimento do *software* proposto e a análise crítica das metodologias empregadas neste trabalho.

2.1 APOIOS E VÍNCULOS

Segundo Botelho (1998) Os apoios ou vínculos são elementos nas estruturas que fornecem restrições ao movimento e à deformação.

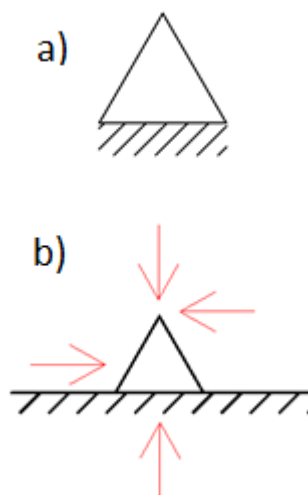
Eles podem ser classificados de três formas a depender de sua natureza e grau de restrição de movimento, sendo elas Apoios Móveis Apoios Fixos, e Engaste.

Os apoios móveis (figura 2a) são apoios que restringem o movimento do corpo na direção normal à superfície em que ele está apoiado, permitindo o movimento na direção tangencial à superfície, como mostra a figura 2b.

FIGURA 2 – Representação simbólica de um apoio móvel

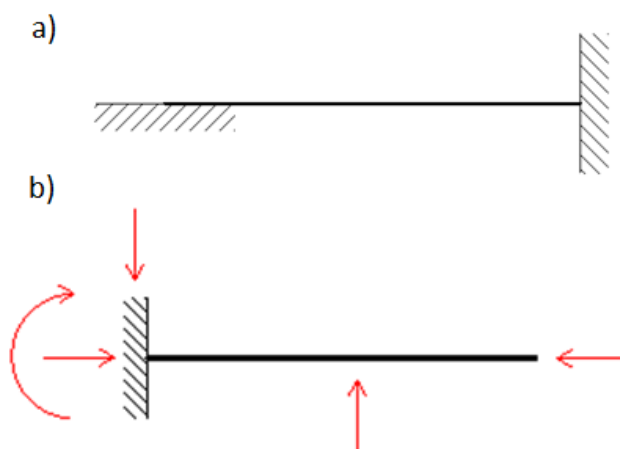
Fonte: elaborado pelo autor, 2023

Os apoios fixos (figura 3a) são apoios que, além de restringir a movimentação do corpo na direção normal, também restringem na direção tangencial à superfície de apoio, como mostra a figura 3b.

FIGURA 3 - Representação simbólica de um apoio fixo

Fonte: Elaborado pelo autor, 2023

O engaste (figura 4a) é um tipo de apoio que, além de impedir a movimentação de translação em qualquer direção, ainda impede que o corpo sofra rotação, o que faz com que ele gere resistência a qualquer esforço de momento no ponto, como mostra a figura 4b.

FIGURA 4 - Representação simbólica do engaste

Fonte: Elaborado pelo autor, 2023

2.2 FORÇAS

Segundo Feodosiev (1977) as forças podem ser consideradas como uma grandeza que expressa a interação mecânica dos corpos. Ao considerar uma estrutura separada das outras partes do sistema, as forças externas são aquelas que podem ser substituídas sobre as ações das partes sobre a estrutura.

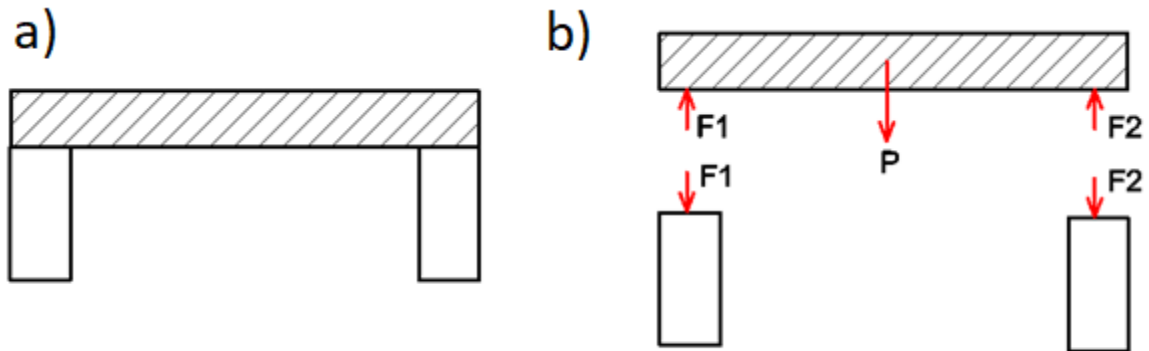
Elas podem ser separadas entre volumétricas e superficiais, sendo as primeiras as forças que agem em todas as partículas de um corpo, como exemplos pode-se citar a força peso que se dá pela relação entre a massa de um corpo e o campo gravitacional em que ele está inserido, a força magnética que é resultante da relação entre a carga de um corpo e o campo magnético. Na figura 5 é possível verificar a força peso, uma força volumétrica, expressa por sua resultante (P), agindo em uma viga.

Já as forças superficiais são aplicadas na superfície de um corpo, e são resultado das ações de outros em contato com o corpo analisado. Na figura 5 as forças F_1 e F_2 são consideradas forças superficiais, já que são resultantes da interação da viga com as superfícies das seções dos dois pilares abaixo dela.

Segundo Hibbeler (2013), se a área de aplicação ou o volume de aplicação da carga for suficientemente pequeno comparado às demais dimensões da estrutura, ela pode ser entendida como uma carga concentrada, aplicada em um único ponto da peça. E se ela for aplicada ao

longo de uma área estreita ela pode ser vista como uma carga distribuída (verificar capítulo 2.6).

FIGURA 5 - Forças agindo em uma viga



Fonte: Elaborado pelo autor, 2023

As forças ativas são aquelas que resultam da aplicação direta de cargas sobre uma estrutura, como o peso próprio, força magnética, força do vento e forças provenientes da interação de outros corpos na estrutura, como o peso de uma parede apoiada sobre uma viga, por exemplo e o peso próprio (P) da viga na figura 5b.

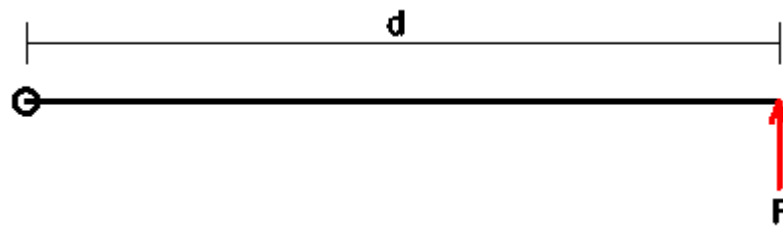
As forças reativas são aquelas que surgem nos apoios, resultantes da interação entre o elemento e outros corpos que servem como apoio para ele, estas forças surgem de modo a contrapor-se às forças ativas na estrutura a fim de que as condições de equilíbrio sejam mantidas. Na figura 5b, as forças F1 e F2 são consideradas forças reativas, já que estas surgem de modo a contrapor-se à força peso (P) que está agindo na viga.

2.3 MOMENTO

Segundo Melconian (1999) o momento (M), ou torque, expressa a tendência de uma força aplicada em um ponto causar uma rotação na peça, e pode ser calculado multiplicando-se o módulo da força (F) pelo braço de alavanca (d), como mostrado na figura 6.

$$M = F \cdot d$$

(1)

FIGURA 6 - Braço de alavanca

Fonte: Elaborado pelo autor, 2023

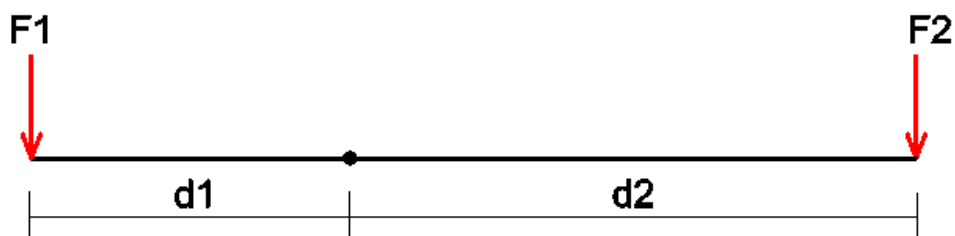
O momento resultante (MR) que age em uma peça é calculado pela somatória de todos os momentos que agem sobre a peça. Se a peça está em equilíbrio translacional e rotacional o momento resultante pode ser obtido tomando qualquer ponto como eixo de rotação, e seu valor será sempre nulo devido à oposição entre as forças ativas e reativas.

$$MR = \sum_{i=1}^n F_i \cdot d_i \quad (2)$$

Na equação: F_i são as forças perpendiculares ao eixo de rotação e d_i são as distâncias das forças até o eixo.

Como ilustração, o momento resultante da figura 7 pode ser calculado da seguinte forma:

$$MR = F_1 \cdot d_1 + F_2 \cdot d_2 \quad (3)$$

FIGURA 7 - Momentos aplicados em uma viga

Fonte: Elaborado pelo autor, 2023

2.4 TENSÃO

Segundo Hibbeler (2013), a tensão, representada aqui pela letra P, pode ser entendida como uma grandeza que mede a distribuição de uma força aplicada na superfície de aplicação. A tensão média (P_m) pode ser calculada dividindo-se a força resultante (ΔR) pela área de aplicação (ΔA).

$$P_m = \frac{\Delta R}{\Delta A} \quad (4)$$

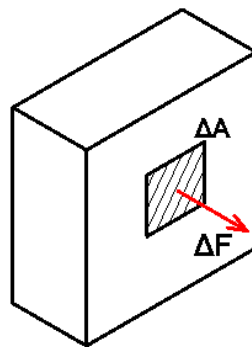
As tensões que agem em uma estrutura podem ser classificadas como tensão normal e tensão cisalhante

A tensão normal é a tensão que é causada por uma força que atua na direção normal à superfície de aplicação, como mostra a figura 8, e é expressa pela letra grega σ (sigma)

$$\sigma_{média} = \frac{\Delta F}{\Delta A} \quad (5)$$

Onde ΔF é a força resultante perpendicular à área de aplicação e ΔA é a área de aplicação da força.

FIGURA 8 – Tensão Normal



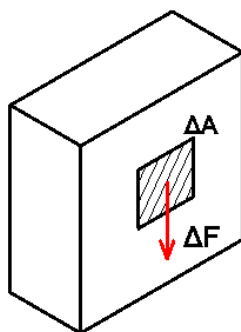
Fonte: Elaborado pelo autor, 2023

Segundo Botelho (1998) a tensão de cisalhamento é a tensão que ocorre devido à aplicação de uma força resultante aplicada paralelamente ao plano da área de aplicação, como mostra a figura 9, e é representada pela letra grega τ (tau).

$$\tau_{média} = \frac{\Delta F}{\Delta A} \quad (6)$$

Onde ΔF é a força resultante paralela à área de aplicação e ΔA é a área de aplicação da força.

FIGURA 9 - Tensão cisalhante



Fonte: Elaborado pelo autor, 2023

2.5 ESFORÇOS INTERNOS SOLICITANTES

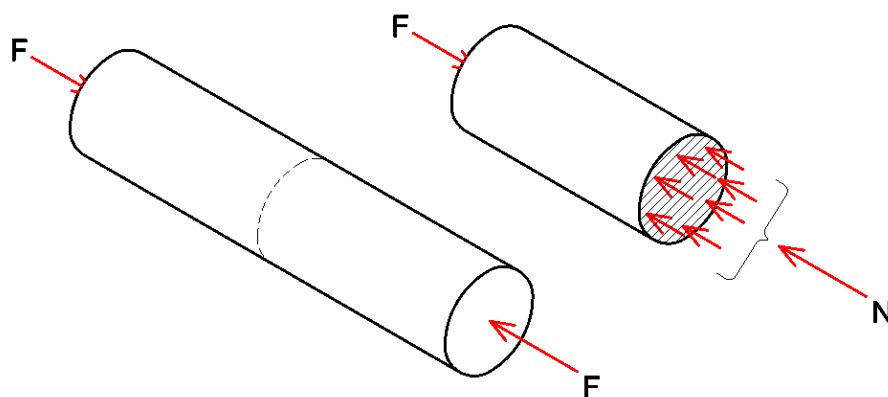
Esforços internos solicitantes são as forças que ocorrem internamente a uma peça como resposta às forças externas ativas e reativas e momentos.

Segundo Botelho (1998) elas são um efeito intermediário que ocorre nas estruturas em função da aplicação destas cargas externas, e são responsáveis por gerar os esforços internos resistentes, que são as tensões normais de tração e compressão e as tensões de cisalhamento que são a causa de possíveis colapsos na estrutura.

Os esforços internos podem ser divididos em forças normais de tração e compressão, força cortante, momento fletor e momento torsor.

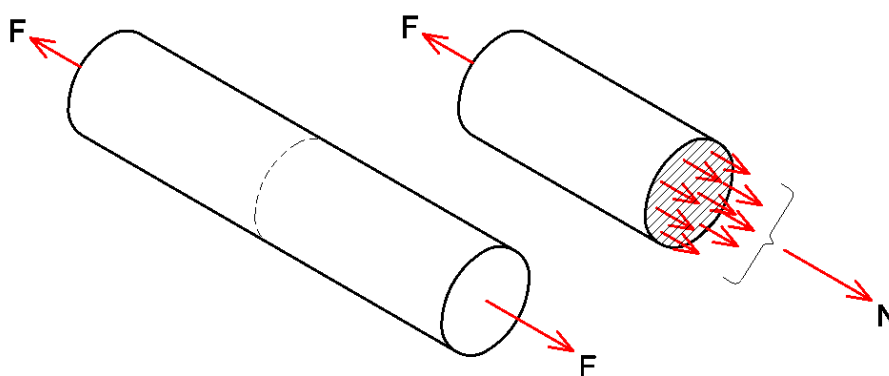
As forças normais de tração e compressão são forças que surgem internamente à estrutura em resposta a forças externas aplicadas na direção normal em relação à seção transversal da peça.

Segundo Feodosiev (1977), a única diferença entre a força normal de compressão e a força normal de tração é o sentido desta força, que no caso da compressão ela se dirige para dentro da peça, enquanto que no caso da tração ela se dirige para fora da mesma, como mostra a figura 10 que representa a força normal de compressão.

FIGURA 10 – Força normal de compressão

Fonte: Elaborado pelo autor, 2023

E a figura 11 que representa a força normal de tração. As forças normais de tração e compressão são expressas pela letra **N**.

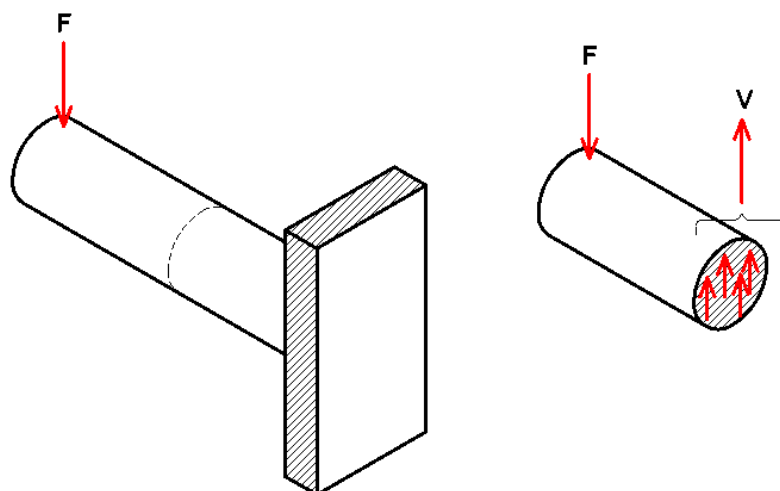
FIGURA 11 – Força normal de tração

Fonte: Elaborado pelo autor, 2023

As forças internas cortantes são as forças que surgem internamente à estrutura em resposta a forças externas aplicadas paralelamente à seção transversal da peça, e elas têm a mesma direção da força externa. A força interna cortante é representada pela letra **V**.

A força cortante pode ser considerada positiva ou negativa, por convenção ela é positiva quando tende a provocar a rotação da peça no sentido horário, e negativa quando tende a provocar a rotação no sentido anti-horário.

FIGURA 12 – Força cortante

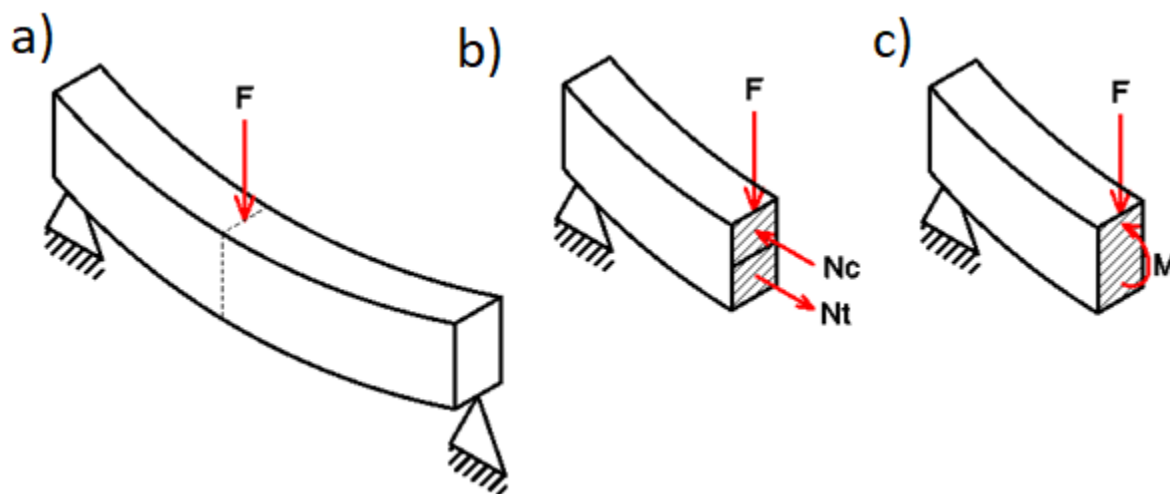


Fonte: Elaborado pelo autor, 2023

Segundo Miranda (2002), o momento fletor pode ser entendido como o momento resultante de todos os momentos que são aplicados na estrutura em relação a um ponto, podendo ser calculado para qualquer ponto da estrutura.

Para exemplificar a atuação do momento fletor (M) em uma estrutura, tomar-se-á uma viga biapoiada submetida a uma força F vertical com sentido para baixo. Quando a força F externa atua em uma estrutura verticalmente com sentido para baixo, a estrutura tende a se deformar, como consequência disso aparecem forças internas normais de tração (N_t) na parte inferior e de compressão (N_c) na parte superior, como ilustra a figura 13.

FIGURA 13 – Momento fletor



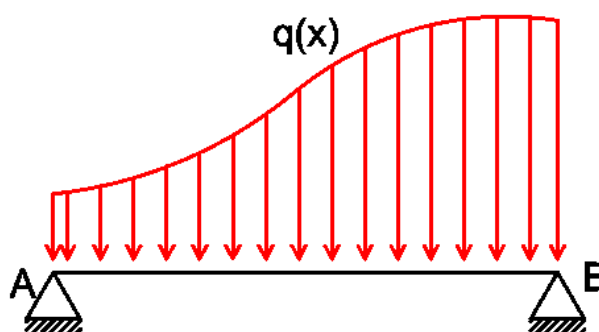
Fonte: Elaborado pelo autor, 2023

Assim como a força cortante, o momento fletor também pode ser positivo ou negativo. Segundo Melconian (1999) ele é, por convenção, positivo quando as fibras inferiores da peça são tracionadas, e negativo quando as partes superiores são tracionadas.

2.6 CARGAS DISTRIBUÍDAS

Foi estudado no capítulo 2.2 que as forças volumétricas são aquelas que atuam sobre todos os pontos do objeto e as forças superficiais são aquelas que atuam ao longo de uma superfície. No entanto, se o comprimento de aplicação da força for consideravelmente superior às demais dimensões da estrutura, considera-se que estas forças atuam de forma linear, ao longo de todo o comprimento de uma estrutura (Hibbeler, 2004), como mostra a figura 14, onde a carga distribuída é expressa em uma função de x , $q(x)$, sendo x a distância na direção paralela ao eixo da viga.

FIGURA 14 – Carga distribuída



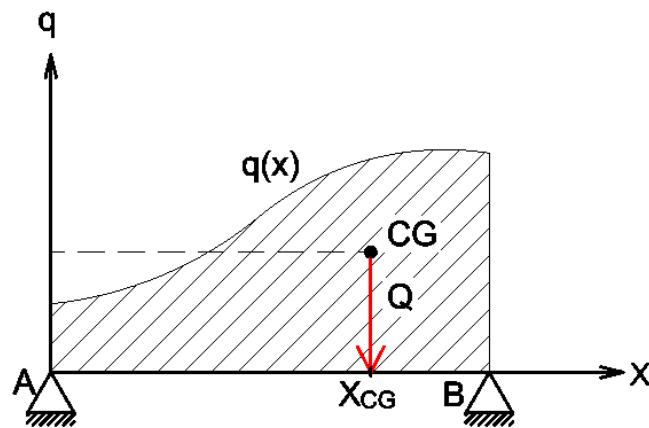
Fonte: Elaborado pelo autor, 2023

A força resultante Q de uma carga distribuída pode ser obtida pela integral da função da carga distribuída em função de x definida entre **A** e **B**.

$$Q = \int_A^B q(x) \cdot dx \quad (7)$$

E seu ponto de aplicação se dará na componente em x do centro geométrico da área abaixo do gráfico de $q(x)$ entre **A** e **B**, na mesma direção e sentido da carga distribuída como mostra a figura 15.

FIGURA 15 – Força resultante de uma carga distribuída



Fonte: Elaborado pelo autor, 2023

2.7 CONDIÇÕES DE EQUILÍBRIO

Para que um corpo esteja em equilíbrio é necessário que ele não sofra translação quando aplicadas forças externas, e nem rotação quando aplicados momentos externos. Para que isso ocorra, as forças reativas devem ser tais que contraponham os esforços ativos de forma que as forças resultantes na estrutura sejam nulas e gerem momentos tais que a resultante dos momentos na estrutura seja nula em relação a um ponto qualquer, esteja este ponto dentro do corpo ou fora dele. Segundo Hibbeler (2004), essas condições podem ser descritas pelas seguintes equações.

$$\sum F = 0 \quad (8)$$

$$\sum M = 0 \quad (9)$$

Onde $\sum F$ é a soma de todas as forças externas atuantes e $\sum M$ é a soma de todos os momentos atuantes na estrutura.

Como na engenharia normalmente utiliza-se um sistema coplanar de forças, a resultante de $\sum F$ pode ser decomposta em duas componentes, x e y, resultando em três condições de equilíbrio.

$$\sum F_x = 0 \quad (10)$$

$$\sum F_y = 0 \quad (11)$$

$$\sum M_o = 0 \quad (12)$$

Onde $\sum F_x$ é a somatória das decomposições das forças no eixo x, $\sum F_y$ é a somatória das decomposições das forças no eixo y e $\sum M_o$ é a somatória dos momentos no plano de estudo em relação ao ponto o.

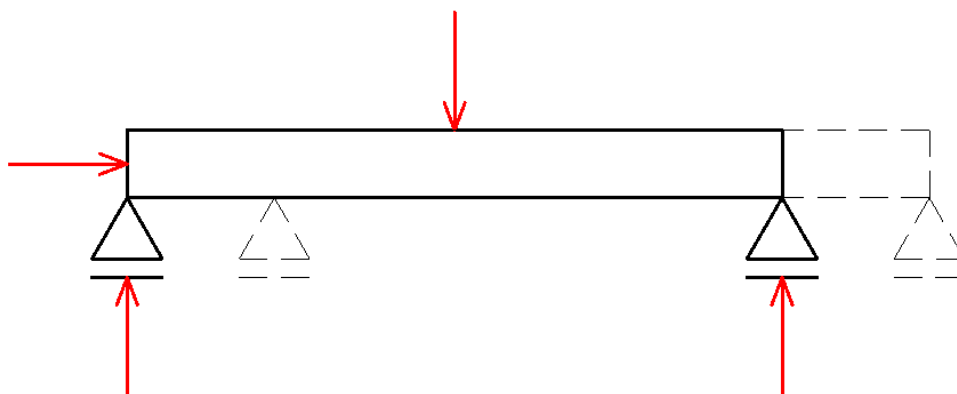
2.8 ESTRUTURAS HIPOSTÁTICAS, ISOSTÁTICAS E HIPERESTÁTICAS

Uma estrutura pode ser hipostática, isostática ou hiperestática, nomenclaturas que remetem ao grau de estabilidade em estruturas. As estruturas hipostáticas são instáveis, enquanto as isostáticas e hiperestáticas são estáveis, a diferença entre estas duas últimas reside no fato de que, no caso das estruturas isostáticas é possível determinar os esforços internos utilizando as três equações de equilíbrio estudadas anteriormente, enquanto que para determiná-los em uma estrutura hiperestática as equações não são suficientes, sendo necessária a utilização de outros métodos, como o método da força (Hibbeler, 2004).

2.9 ESTRUTURAS HIPOSTÁTICAS

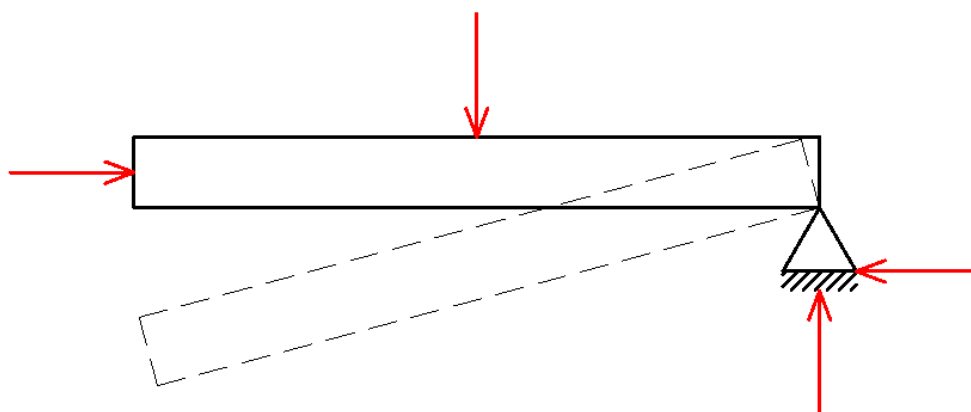
Segundo Botelho (1998), uma estrutura classifica-se como hipostática quando seus apoios não restringem os seus movimentos de translação e rotação, ou seja, quando alguma das três equações de equilíbrio estudadas anteriormente não é satisfeita. Neste caso, o número de vínculos não é suficiente para fornecer estabilidade à estrutura, e ela é considerada instável.

Um exemplo é a viga da figura 16, onde $\sum F_y = 0$, $\sum M_o = 0$, mas $\sum F_x \neq 0$, o que causa uma aceleração horizontal na estrutura.

FIGURA 16 – Translação de estrutura hipostática

Fonte: Elaborado pelo autor, 2022

E também a viga da figura 17, onde $\sum F_x = 0$, $\sum F_y = 0$, mas $\sum M_o \neq 0$, o que causa uma aceleração angular na peça.

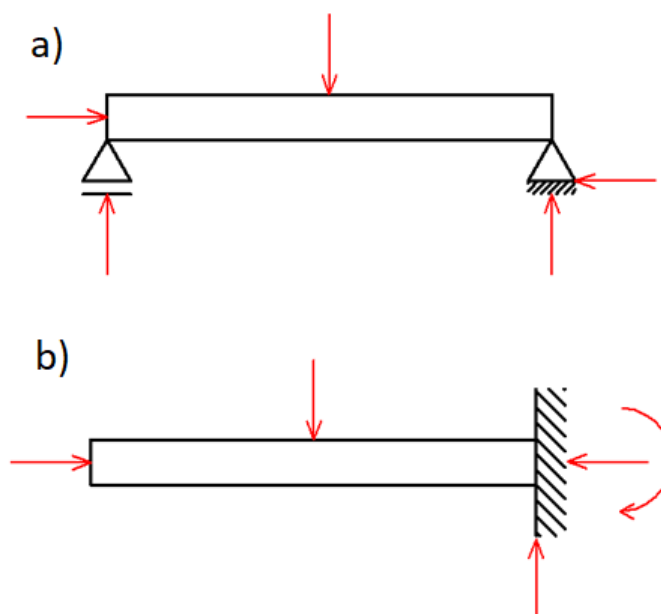
FIGURA 17 – Rotação de estrutura hipostática

Fonte: Elaborado pelo autor, 2022

2.10 ESTRUTURAS ISOSTÁTICAS

Segundo Feodosiev (1977) uma estrutura classifica-se como isostática quando o número o número de vínculos é tal que fornece estabilidade translacional e rotacional à estrutura e permite a determinação dos esforços utilizando-se as três equações de equilíbrio.

Na figura 18 é possível ver os dois exemplos apresentados anteriormente nas figuras 16 e 17, no entanto foram trocados os apoios de modo a torná-las isostáticas. É possível notar agora que todas as equações de equilíbrio são satisfeitas e não há mais a possibilidade de translação ou rotação nas vigas.

FIGURA 18 – Estruturas isostáticas

Fonte: Elaborado pelo autor, 2023

Neste caso a determinação destas reações nos apoios é obtida de forma simples, utilizando as equações citadas anteriormente. A obtenção das reações é crucial para a determinação dos esforços nas estruturas.

2.11 OBTENÇÃO DE ESFORÇOS EM ESTRUTURAS ISOSTÁTICAS

Como já mencionado anteriormente, o processo para a obtenção dos esforços internos em estruturas isostáticas é bastante simples, e segundo Hibbeler (2013), é possível utilizar o método das seções aplicando alguns passos para a obtenção dos resultados esperados.

Para a aplicação do método das seções para um segmento do corpo é necessário primeiramente, desenhar o diagrama de corpo livre de todo o elemento e calcular as reações nos apoios utilizando as equações de equilíbrio.

Em seguida deve-se desenhar um diagrama de corpo livre traçando um corte imaginário na seção em que se deseja obter os esforços internos, mantendo as cargas originais intactas. Assim deve-se também desenhar as resultantes das cargas internas que se deseja obter e por fim calculá-las utilizando as equações de equilíbrio.

2.12 DIAGRAMAS DE ESFORÇOS INTERNOS

Segundo Martha (2010), os diagramas de esforços internos são gráficos que caracterizam a variação dos valores dos esforços ao longo do eixo da estrutura, e são traçados com base nas equações dos esforços internos.

Os diagramas de esforços internos fornecem uma informação visual de quais são os esforços máximos nas estruturas e onde eles ocorrem, permitindo aos projetistas saber quais são as regiões mais propensas ao rompimento (Miranda, 2002).

2.13 ESTRUTURAS HIPERESTÁTICAS

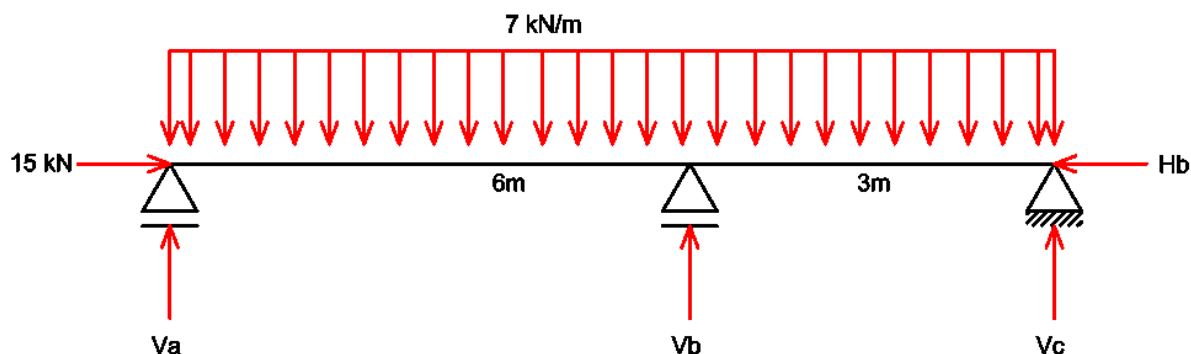
Assim como nas estruturas isostáticas, nas estruturas hiperestáticas os vínculos fornecem estabilidade translacional e rotacional à estrutura de modo que as condições de equilíbrio sejam satisfeitas, no entanto não é possível obter todas as reações nos apoios apenas utilizando as três equações de equilíbrio (Melconian, 1999), já que o número de incógnitas é superior ao número de equações, o que impossibilita a resolução das incógnitas por meio de um sistema de equações.

As estruturas hiperestáticas são classificadas conforme seu grau de hiperestaticidade (g), que para pórticos planos pode ser obtida pela seguinte equação:

$$g = [\text{número de incógnitas}] - [\text{número de equações de equilíbrio}] \quad (13)$$

Segundo Martha (2010), um grau de hiperestaticidade inferior a zero é uma condição suficiente para que o modelo seja hipostático, um grau nulo é uma condição necessária para que ele seja isostático, e um grau de hiperestaticidade superior a zero é uma condição necessária para que o modelo seja hiperestático.

FIGURA 19 – Viga hiperestática



Fonte: Elaborado pelo autor, 2023

Um exemplo de estrutura hiperestática é a viga da figura 19, onde há três vínculos e quatro forças reativas, o grau de hiperstaticidade pode ser obtido pela equação supramencionada:

$$g = 4 - 3 = 1$$

Aplicando o método das seções para a obtenção das reações nos apoios, adotando-se o eixo de rotação como a extremidade esquerda, tem-se:

$$\sum M_o = 0 \therefore 6V_b + 9V_c - \left(\int_0^9 7 \cdot dx \right) \cdot 4,5 = 0$$

$$\sum F_y = 0 \therefore V_a + V_b + V_c - \int_0^9 7 \cdot dx = 0$$

$$\sum F_x = 0 \therefore H_b - 15 = 0$$

Como é possível perceber, a única reação possível de ser encontrada pelo sistema de equações é **Hb**, com valor de 15kN, as demais são impossíveis de ser determinadas utilizando apenas as equações de equilíbrio.

Por este motivo, é preciso a utilização de outros métodos para a obtenção das forças reativas das estruturas hiperestáticas, um dos métodos clássicos para a resolução deste problema é o método da força, que será estudado posteriormente.

2.14 PRINCÍPIO DOS TRABALHOS VIRTUAIS

O Princípio dos Trabalhos Virtuais é um método de análise que se baseia no princípio da conservação de energia, tendo sido desenvolvido por John Bernoulli em 1717, e é utilizado para encontrar o deslocamento ou rotação em um ponto da estrutura. (Hibbeler, 2013)

Quando um corpo é estático, ela deve satisfazer as equações de equilíbrio, ou seja, as forças externas e as tensões internas devem estar em equilíbrio entre si, e também deve satisfazer às equações de compatibilidade, o que quer dizer que os deslocamentos externos da estrutura devem ser compatíveis com as deformações internas (Martha, 2010).

O Princípio dos Trabalhos Virtuais une estes dois sistemas por meio do conceito de trabalho, ou seja, para que uma estrutura permaneça estática, o trabalho das forças externas deve ser compatível com a energia de deformação interna, resultando na seguinte equação:

$$\sum F \cdot \Delta = \int \sigma \cdot d\varepsilon \quad (14)$$

Onde \mathbf{F} representa as forças externas, Δ os deslocamentos externos, σ as tensões internas e ε as deformações internas.

Na equação, a expressão $\sum \mathbf{F} \cdot \Delta$ representa os trabalhos externos e $\int \sigma \cdot d\varepsilon$ a energia de deformação interna (Martha, 2010).

No Princípio dos Trabalhos Virtuais, ou PTV, para a obtenção dos deslocamentos ou rotações em um ponto, utiliza-se o sistema estrutural real, e um sistema auxiliar virtual, onde aplica-se uma força, ou momento, virtual (imaginária) \mathbf{F}' unitária na mesma direção em que se procura encontrar o deslocamento.

Ao se aplicar a carga virtual \mathbf{F}' e a peça sofrer as ações das cargas reais, o ponto de aplicação da carga virtual sofrerá um deslocamento real Δ que deve ser compatível com a deformação virtual $d\varepsilon$. Deste modo, a força virtual \mathbf{F}' realizará um trabalho virtual externo e a tensão virtual σ realizará um trabalho virtual interno, utilizando a equação 1:

$$F' \cdot \Delta = \int \sigma \cdot d\varepsilon \quad (15)$$

Onde Δ é o deslocamento no ponto de aplicação da carga virtual. Como a força F' é unitária:

$$1. \Delta = \int \sigma \cdot d\varepsilon \quad (16)$$

Dado este fato a equação do trabalho virtual pode ser descrita pela seguinte expressão (Hibbeler, 2013):

$$1. \Delta = \int_{x1}^{x2} \frac{N \cdot n}{EA} dx + \int_{x1}^{x2} \frac{M \cdot m}{EI} dx + \int_{x1}^{x2} X \frac{v \cdot V}{GA} dx \quad (17)$$

Onde a primeira integral é o trabalho virtual interno realizado pelos esforços normais, a segunda é o trabalho virtual interno realizado pelo momento fletor, e a terceira é aquele realizado pelos esforços cortantes.

Como em barras longas o efeito da força cortante é pequeno em comparação às demais, seu trabalho virtual interno é desprezado, considerando-se apenas as duas primeiras integrais (Martha, 2010), resultando na seguinte expressão:

$$1. \Delta = \int_{x1}^{x2} \frac{N \cdot n}{EA} dx + \int_{x1}^{x2} \frac{M \cdot m}{EI} dx \quad (18)$$

E finalmente, segundo Hibbeler (2013), quando o método é aplicado a vigas, por serem longas e esbeltas é conveniente utilizar apenas a energia de deformação virtual causada pelo momento fletor, já que este tem muito mais influência no deslocamento do que os outros esforços neste tipo de estrutura, portanto:

$$1. \Delta = \int_{x1}^{x2} \frac{M \cdot m}{EI} dx \quad (19)$$

Onde **x1** é o ponto de início da seção de análise, **x2** é o ponto final da seção de análise, **M** é o momento fletor na viga, **m** é o momento fletor virtual provocado pela carga unitária virtual, **E** é o módulo de elasticidade do material, e **I** é o momento de inércia da área da seção transversal da peça calculado em torno da linha neutra, e:

$$1. \theta = \int_{x_1}^{x_2} \frac{M \cdot m_{\theta}}{EI} dx \quad (20)$$

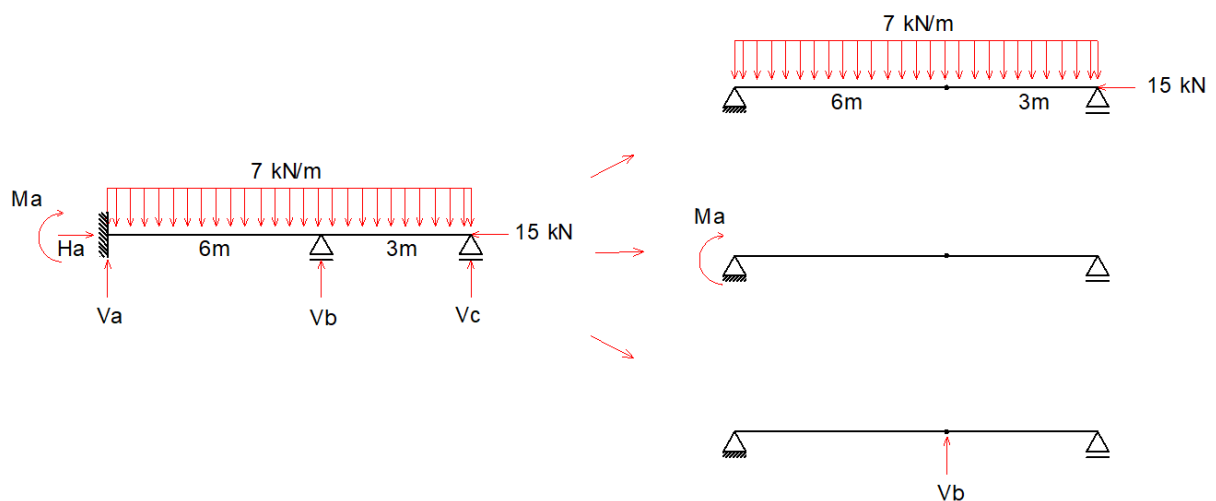
Onde θ é a rotação no ponto e m_{θ} é o momento fletor virtual provocado pelo momento unitário no ponto.

2.15 MÉTODO DA FORÇA

Como visto nos capítulos anteriores, não é possível encontrar os valores das reações dos apoios nas vigas hiperestáticas utilizando apenas as três equações de equilíbrio pois o número de incógnitas é inferior ao número de equações, por isso, para tal, utiliza-se o método da força.

Este método consiste na superposição de soluções básicas isostáticas, como expressa Martha (2010), onde o sistema hiperestático inicial é liberado das ligações adicionais, dando origem a um sistema principal isostático resultante da retirada dos vínculos adicionais e a outros sistemas básicos redundantes, onde atuam apenas as reações retiradas do sistema original (Feodosiev, 1997), como mostra a figura 20, onde no lado esquerdo tem-se o sistema hiperestático inicial, que é subdividido nos três sistemas da direita, em que na parte superior tem-se o sistema principal, onde foram retiradas as reações **Ma** e **Vb**, e logo abaixo tem-se os sistemas redundantes, onde atuam apenas as reações retiradas do sistema inicial.

FIGURA 20 - Sistemas principais e redundantes



Fonte: Elaborado pelo autor, 2023

Deste modo a partir do sistema principal é possível, aplicando-se o princípio dos trabalhos virtuais, encontrar os deslocamentos e rotações nos pontos onde foram retirados os apoios no sistema principal, sendo estes expressos por Δ_i para os deslocamentos e θ_i para as rotações, onde i é a identificação do ponto (Hibbeler, 2004).

Já no caso dos sistemas redundantes não é possível encontrar os deslocamentos gerados pelas reações, no entanto é possível encontrar os deslocamentos (ou rotações) causados por uma força (ou momento) unitária, expressos por f_{ij} para os deslocamentos e α_{ij} para as rotações, de modo que:

$$f_{ij} \cdot R_j = \Delta_{ij} \quad (21)$$

$$\alpha_{ij} \cdot R_j = \theta_{ij} \quad (22)$$

Onde f_{ij} é o deslocamento unitário no ponto i causado por uma força (ou momento) unitária no ponto j , R_j é a reação no ponto j , Δ_{ij} é o deslocamento no ponto i causado por R_j , α_{ij} é a rotação no ponto i causada por uma força (ou momento) unitária no ponto j e θ_{ij} é a rotação no ponto i causada por R_j . Estes deslocamentos e rotações unitários são chamados de coeficientes de flexibilidade (Martha, 2010).

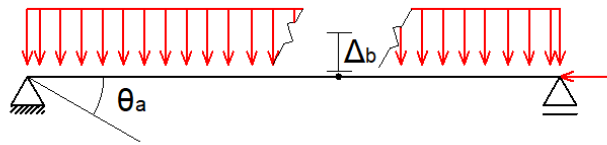
Sabendo-se disso, é possível estabelecer as chamadas equações de compatibilidade, sendo elas:

$$\Delta_i + \sum f_{ij} \cdot R_j = 0 \quad (23)$$

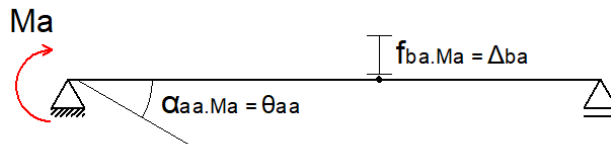
$$\theta_i + \sum \alpha_{ij} \cdot R_j = 0 \quad (24)$$

Deste modo, após encontrados os valores de todos os deslocamentos e rotações, tem-se um sistema linear onde as incógnitas serão as reações nos apoios retiradas do sistema hiperestático inicial.

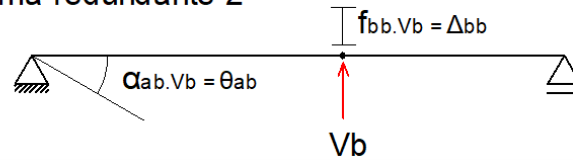
FIGURA 21 – Deformações, rotações e coeficientes de flexibilidade
Sistema principal



Sistema redundante 1



Sistema redundante 2



Fonte: Elaborado pelo autor, 2023

Tomando como exemplo a figura 21 tem-se que:

- θ_a é a rotação no ponto **a** no sistema principal.
- Δ_b é o deslocamento no ponto **b** no sistema principal.
- α_{aa} é a rotação unitária no ponto **a** causada por um momento unitário no ponto **a** do sistema redundante 1.
- α_{ab} é a rotação no ponto **a** causada por uma força unitária no ponto **b** do sistema redundante 2.
- f_{ba} é o deslocamento no ponto **b** causado por um momento unitário no ponto **a** do sistema redundante 1.
- f_{bb} é o deslocamento no ponto **b** causado por uma força unitária no ponto **b** do sistema redundante 2.
- θ_{aa} é a rotação no ponto **a** causada por M_a no sistema redundante 1.
- θ_{ab} é a rotação no ponto **a** causada por V_b no sistema redundante 2.
- Δ_{ba} é o deslocamento no ponto **b** causado por M_a no sistema redundante 1.
- Δ_{bb} é o deslocamento no ponto **b** causado por V_b no sistema redundante 2.

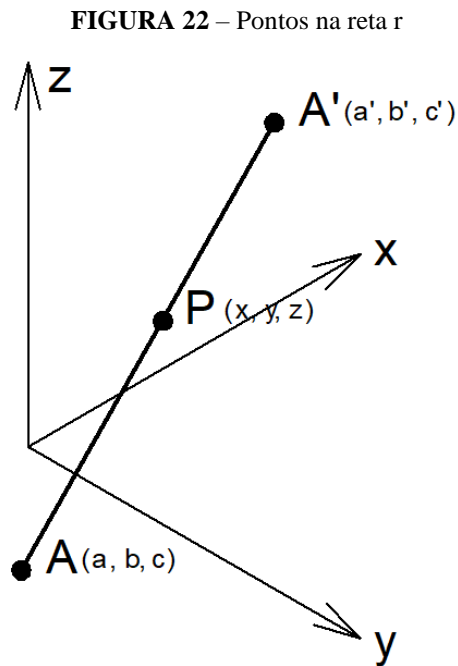
Para o exemplo em questão tem-se as seguintes equações de compatibilidade:

$$\theta_a + \alpha_{aa} \cdot M_a + \alpha_{ab} \cdot V_b = 0 \quad (25)$$

$$\Delta_b + f_{ba} \cdot M_a + f_{bb} \cdot V_b = 0 \quad (26)$$

2.16 EQUAÇÕES PARAMÉTRICAS DA RETA

Segundo Lima (2014), sendo r , uma reta do espaço que passa pelos pontos $A = (a, b, c)$, e $A' = (a', b', c')$, onde no plano também existe um ponto $P = (x, y, z)$, como mostra a figura 22.



Fonte: Elaborado pelo autor, 2023

Tem-se que o ponto P pertence à reta r apenas se:

$$x = a + \lambda(a' - a) \quad (27)$$

$$y = b + \lambda(b' - b) \quad (28)$$

$$z = c + \lambda(c' - c) \quad (29)$$

Onde $\lambda \in \mathbb{R}$.

São estas as equações paramétricas da reta. Dado isso, é possível afirmar que duas retas se interseccionam quando ambas possuem um ponto em comum. Este conceito será utilizado para verificar se duas estruturas, como vigas e pilares, se interseccionam, possibilitando a identificação automática de apoios e transferência de cargas.

2.17 ESCALONAMENTO OU ELIMINAÇÃO GAUSSIANA

De acordo com Lima (2014), o escalonamento é o método mais simples e eficaz para a resolução de sistemas lineares.

Tendo-se o seguinte sistema linear:

$$a1.x + b1.y + c1.z = d1$$

$$a2.x + b2.y + c2.z = d2$$

$$a3.x + b3.y + c3.z = d3$$

Tem-se que m é a matriz do sistema, e M é a matriz aumentada do sistema, onde:

$$m = \begin{bmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{bmatrix}$$

$$M = \begin{bmatrix} a1 & b1 & c1 & d1 \\ a2 & b2 & c2 & d2 \\ a3 & b3 & c3 & d3 \end{bmatrix}$$

O processo de escalonamento consiste na realização de operações elementares sobre a matriz aumentada, de modo que todos elementos nulos de cada linha da matriz escalonada estejam à esquerda dos elementos não-nulos. Estas operações são a troca da ordem das equações do sistema ou a substituição de uma linha da matriz aumentada pela sua soma com um múltiplo de outra linha do sistema, ou a substituição de uma linha pela soma de um múltiplo da mesma linha com outra linha da matriz aumentada.

As matrizes escalonadas m' e M' podem ser representadas do seguinte modo:

$$m' = \begin{bmatrix} a1' & b1' & c1' \\ 0 & b2' & c2' \\ 0 & 0 & c3' \end{bmatrix}$$

$$M' = \begin{bmatrix} a1' & b1' & c1' & d1' \\ 0 & b2' & c2' & d2' \\ 0 & 0 & c3' & d3' \end{bmatrix}$$

O que resulta em um sistema linear que pode ser facilmente resolvido:

$$a1'.x + b1'.y + c1'.z = d1'$$

$$b2'.y + c2'.z = d2'$$

$$c3'.z = d3'$$

2.18 A LINGUAGEM DE PROGRAMAÇÃO C#

C# é uma linguagem de programação desenvolvida pela Microsoft de tipagem forte e orientada a objetos, sendo uma linguagem com origem em linguagens derivadas de C, além disso a linguagem permite a criação de *softwares* executados no .NET (Microsoft, 2023).

Por ser fortemente tipada, os tipos de variáveis, parâmetros, e valores retornados de funções, que em C# são chamadas de “métodos”, têm seus tipos de dados especificados no código o que oferece diversas vantagens, como a segurança de tipo, que evita erros de tipagem, como somar um número inteiro com um caractere de texto, por exemplo, oferecem melhor legibilidade e manutenção, já que permite que o programador saiba qual é o tipo de dado que está lidando, além de apresentar um melhor desempenho, já que não há uma sobrecarga de verificação dinâmica em tempo de execução, como ocorre em linguagens de tipagem dinâmica, como *Python* e *JavaScript*, por exemplo (Skeet, 2019).

Como C# é uma linguagem orientada a objetos, é possível criar tipos de dados personalizados e definir seus comportamentos e atributos (Microsoft, 2023), um exemplo disto é a criação de um tipo de dados para as Vigas, onde é possível definir seus atributos, como comprimento, esforços internos e também seus comportamentos, como a sua aparência gráfica na interface e o modo quem que ela deve ter seus esforços internos calculados.

Por estes motivos a linguagem C# é excelente para o desenvolvimento de *softwares* que envolvem uma maior complexidade, já que possibilita uma fácil identificação de erros, mais eficiência na execução e a criação de tipos de dados mais complexos.

2.18.1 A Plataforma .NET (*dot NET*)

O .Net, ou “*dot NET*”, é uma plataforma de desenvolvimento desenvolvida pela Microsoft, de código aberto, ou seja, tem seu código fonte disponibilizado de forma pública, e que permite a criação de diferentes tipos de aplicações, como *desktop* (aplicações para o sistema operacional), *WEB*, jogos, Internet das Coisas e dispositivos móveis.

O .Net permite que qualquer aplicação escrita nas linguagens C#, F# ou *Visual Basic* sejam executadas de forma nativa em sistemas operacionais compatíveis (Microsoft, 2023).

2.18.2 Windows Forms

O *Windows Forms* é uma plataforma de desenvolvimento, parte da .NET, que dá suporte à criação de interfaces de usuário para aplicativos de área de trabalho (*Desktop*) e traz um conjunto de bibliotecas que facilitam a criação de formulários, como controles de formulário, tarefas de salvar e carregar arquivos externos e ferramentas para *design* de formulários (Skeet, 2019).

A plataforma permite também a customização dos controles de formulário, permitindo a inclusão de imagens externas, além de permitir a inclusão de elementos gráficos, como linhas e círculos, por meio do GDI+, uma parte do sistema operacional do Microsoft que permite a criação de gráficos bidimensionais de vetores, imagens e texto. Com estas tecnologias é possível criar aplicativos para área de trabalhos (*Desktop*) modernas, amigáveis e interativas utilizando a linguagem C# (Microsoft, 2023).

2.19 ORIENTAÇÃO A OBJETOS

Como já mencionado no capítulo anterior o C# é uma linguagem de programação orientada a objetos. Segundo Ricarte (2001) a orientação a objetos é um paradigma de programação que tem sua organização pautada em unidades chamadas de objetos. Conceitualmente os objetos são instâncias de classes, entidades que encapsulam informações acerca dos objetos e de que forma eles serão manipulados, como atributos, propriedades e métodos (funções definidas internamente às classes).

Este paradigma possibilita a criação de tipos de objetos (classes), a proteção dos atributos e funcionalidades internas do objeto (encapsulamento), a reutilização de código

(herança) e a definição das funcionalidades de cada objeto durante a execução do aplicativo (polimorfismo). Estas características são consideradas os três pilares da orientação a objetos (Leite et al. 2016).

2.19.1 Classes, Objetos, Propriedades e Métodos

Classes:

As classes são uma forma de definir as características e comportamentos de todos os objetos que são criados a partir dela. Elas são estruturas que servem como um molde para a criação de objetos, sendo uma descrição abstrata do que virão a ser os objetos instanciados a partir dela (Leite et al. 2016).

Um exemplo é a definição de uma classe que representa uma viga, onde são definidas todas as características e comportamentos que uma viga possui, a partir deste conceito abstrato podem ser instanciados objetos que seguem estas características.

Objetos:

Segundo Ricarte (2001), um objeto é a instanciação de uma classe, ou seja, é a representação concreta do conceito definido pela classe. Um exemplo deste conceito é a criação de vários objetos instanciados a partir da classe viga, onde cada um dos objetos possuirá todas as características e comportamentos definidos na classe, mas cada instância terá seus valores específicos para cada uma dessas propriedades.

Propriedades:

As propriedades são características ou atributos de um objeto, elas se assemelham às variáveis, mas são associadas a objetos específicos de uma classe. Elas representam dados ou estados que descrevem o objeto, e podem ter valores diferentes para cada instância (Leite et al. 2016).

Uma viga, por exemplo, possui comprimento, apoios, forças externas e localização no espaço, todas essas características são propriedades do objeto instanciado a partir da classe viga.

Métodos:

Os métodos são funções ou procedimentos que pertencem a uma classe específica. Eles definem o comportamento ou as ações que os objetos dessa classe podem realizar. Os métodos

executam as operações que serão fornecidas pelas classes e podem ainda manipular as propriedades de um objeto (Leite et al. 2016). Eles podem, ou não, receber dados como parâmetro, e retornar valores. Em suma, métodos são porções de código que manipulam dados e propriedades dos objetos e executam ações Ricarte (2001).

Um objeto do tipo viga, por exemplo, pode ter um método que recebe como parâmetro uma posição em seu eixo, executa um algoritmo de cálculo de seus esforços internos e fornece o valor do momento fletor obtido naquele ponto.

2.19.2 Encapsulamento

O encapsulamento é a premissa de que cada objeto deve manter oculta toda a informação utilizada para sua manipulação, de modo que apenas o mínimo necessário para a sua utilização seja acessado por outros componentes. Com o encapsulamento, é possível que o usuário possa manipular o objeto sem preocupar-se com lógicas e detalhes internos, trabalhando a um “nível mais alto de abstração” (Ricarte, 2001), o que simplifica o processo de programação e desenvolvimento dos aplicativos.

Um exemplo deste conceito é a definição de uma classe que representa uma viga, e todo o processo de cálculo para a obtenção de seus esforços internos seja ocultada em um determinado método (função definida internamente à classe). Deste modo, é possível instanciar vários objetos do tipo “viga” e fazê-las calcular seus próprios esforços internos apenas executando o método em questão para cada uma delas, sem precisar se preocupar com a lógica interna de cálculo.

2.19.3 Herança

Segundo PET-Tele (2009) a herança é um poderoso mecanismo das linguagens orientadas a objetos pois permite o reuso do código, ela permite que métodos e atributos definidos em uma classe sejam reutilizados em outras classes, neste relacionamento as classes das quais o código é herdado são chamadas de superclasses (ou classes mães) e as classes que herdam o código são chamadas de subclasses (ou classes filhas).

As subclasses que estendem as superclasses herdam todas as características, métodos, propriedades e atributos definidos nas classes “mães”, possibilitando a reutilização de código, e também podem sobrescrever as características se necessário, além de definir características

adicionais, o que faz com que as classes “filhas” sejam mais específicas que as classes “mães” (Ricarte, 2001).

Um exemplo deste mecanismo é a criação de uma classe que representa uma estrutura linear discretizada, apelidada de “*LinearStructure*”, que define um método que calcula a equação de reta da estrutura. Se forem criadas duas classes, uma representando um pilar, apelidada de “*Column*”, e outra representando uma viga, apelidada de “*Beam*”, e estas classes estenderem a classe “*LinearStructure*”, as duas classes filhas também herdarão a lógica de cálculo de suas equações de reta, eliminando a necessidade de que a lógica seja implementada novamente para cada uma das duas classes.

2.19.4 Polimorfismo

O polimorfismo é uma característica da orientação a objetos que permite que uma subclasse possa invocar métodos definidos em uma superclasse mas com um comportamento diferente, ou seja, que os métodos definidos na classe mãe possam ser sobrescritos nas classes filhas, de modo que estas apresentem características únicas específicas para as classes (PET-Tele, 2009).

Um exemplo deste mecanismo é a criação de uma superclasse que define um apoio genérico e possui um método que executará um código que fará a representação gráfica do apoio na interface de usuário, onde serão criadas três classes que representarão os três tipos de apoio abordados no capítulo 2.1, os apoios fixos, móveis e engaste, que herdarão da superclasse.

Naturalmente estas classes filhas herdarão o método que desenha os apoios, no entanto é possível sobrescrever este método para cada uma das classes herdadas, de modo que cada uma desenhe na interface a representação gráfica correspondente ao seu tipo de apoio. Deste modo, ao executar o mesmo método em uma iteração pelos apoios instanciados a partir das classes filhas, cada um dos objetos será desenhado na interface com sua própria representação gráfica de apoio.

2.20 SOFTWARES EXISTENTES

Este capítulo destina-se a apresentar uma visão abrangente dos programas disponíveis no mercado relacionados à análise estrutural. Este segmento é fundamental para contextualizar o ambiente tecnológico atual, fornecendo informações sobre as ferramentas já estabelecidas na

área. A análise desses *softwares* será essencial para compreender como a proposta em desenvolvimento se diferencia e atende às necessidades específicas dos estudantes e iniciantes na engenharia.

Um dos *softwares* existentes no mercado é o SAP2000, um *software* amplamente reconhecido no campo da engenharia civil e análise estrutural. Desenvolvido pela *Computers and Structures, Inc.* (CSI), esse *software* é uma ferramenta poderosa para análise e projeto de estruturas. O SAP2000 oferece funcionalidades avançadas para a análise de sistemas estruturais complexos, incluindo elementos finitos, elementos de casca e sólidos, além de opções para análise sísmica e dinâmica. É amplamente utilizado na indústria de engenharia civil e é apreciado por sua capacidade de lidar com uma ampla gama de problemas estruturais. O SAP2000 é uma escolha para engenheiros que buscam uma solução abrangente para análise e projeto estrutural (CSI AMERICA, 2023).

Outro *software* existente no mercado é o *Robot Structural Analysis*, desenvolvido pela Autodesk. Este programa oferece um conjunto abrangente de recursos para análise e dimensionamento de estruturas, adequado para edifícios e pontes. O *Robot Structural Analysis* permite que os engenheiros realizem análises estáticas e dinâmicas, considerando cargas diversas, como cargas de vento, neve e sísmicas. Além disso, ele é integrado ao *software* BIM (*Building Information Modeling*) da Autodesk, o que facilita a colaboração e a transferência de dados entre o projeto estrutural e o projeto arquitetônico. Com sua interface intuitiva e recursos avançados, o *Robot Structural Analysis* é uma opção para profissionais que trabalham no projeto de estruturas civis complexas (AUTODESK, 2023).

Já o Ftool é um programa voltado para a análise estrutural de sistemas planos. Ele foi desenvolvido na PUC-Rio pelo professor Luiz Fernando Martha, em um projeto de pesquisa integrado com o apoio do CNPq, e é conhecido por sua abordagem simplificada e sua ênfase na prototipagem eficiente de estruturas. Inicialmente desenvolvido com foco no uso educacional em sala de aula, o Ftool evoluiu para se tornar uma ferramenta frequentemente empregada em projetos profissionais de engenharia. O Ftool inclui vários casos de carga, combinações de casos, seções transversais e propriedades, entre outras funcionalidades (FTOOL, 2023).

Segundo Martha (2002), professor do Departamento de Engenharia Civil da PUC-Rio e idealizador e responsável pelo *software*, a integração entre as fases de pré-processamento, análise numérica e pós-processamento, juntamente com a ênfase na facilidade de uso, capacidade de modelagem e versatilidade na visualização de resultados, são características que distinguem o Ftool dos programas comerciais de análise por elementos finitos.

2.21 ESTADO DA ARTE

Durante a pesquisa realizada para este trabalho, foi essencial explorar outras fontes e recursos relevantes para a análise de estruturas, e um dos *softwares* que emergiu durante essa pesquisa foi o SAP2000. Reconhecido por sua ampla utilização na indústria de engenharia e construção, o SAP2000 destaca-se como uma ferramenta profissional de análise estrutural de alta complexidade.

O SAP2000 é um *software* profissional amplamente utilizado por engenheiros e profissionais da área de engenharia civil e estrutural para análise e projeto de estruturas complexas e projetos reais. Ele é projetado para uso profissional em projetos de engenharia de grande escala.

Embora seja uma ferramenta extremamente poderosa, ele apresenta algumas diferenças notáveis em relação à proposta do *software* que está sendo desenvolvido. Enquanto o SAP2000 é projetado principalmente para atender às necessidades de engenheiros profissionais e projetos de engenharia complexos, a proposta deste projeto concentra-se em proporcionar uma abordagem mais acessível e educacional para a análise de estruturas tridimensionais.

Ele também oferece uma ampla gama de recursos avançados, como análise não linear, análise dinâmica e design de elementos finitos, que podem ser excessivamente complexos para estudantes e iniciantes. Em contraste, este projeto visa simplificar o processo de análise de estruturas, tornando-o mais acessível e compreensível, com foco no público estudantil em processo de graduação.

Além disso o SAP2000 é um *software* comercial e pago. Ele é desenvolvido pela empresa *Computers and Structures, Inc.* (CSI) e é licenciado com base em assinaturas anuais ou de longo prazo, e o custo da licença pode variar dependendo da versão e dos recursos específicos incluídos, o que pode dificultar o acesso de estudantes ao *software*.

Um outro *software* estudado é o *Robot Structural Analysis*, desenvolvido pela *Autodesk*. Ele é uma solução profissional amplamente adotada por engenheiros e profissionais da área de engenharia civil e estrutural. É projetado para atender às necessidades de projetos reais e complexos de engenharia, oferecendo recursos avançados. No entanto, vale ressaltar que apresenta algumas distinções notáveis em relação à proposta do *software* em desenvolvimento.

Enquanto o *Robot Structural Analysis* é uma ferramenta altamente sofisticada, com recursos para análise avançada, dimensionamento e projeto de estruturas complexas, o *software* em desenvolvimento se concentra em proporcionar uma abordagem educacional e acessível à análise de estruturas tridimensionais. O projeto tem como público-alvo principal estudantes e

iniciantes na área de engenharia, visando tornar o processo de análise de estruturas mais compreensível e acessível a esse grupo.

Adicionalmente, o *Robot Structural Analysis* é um *software* comercial e pago. Ele é licenciado por meio de assinaturas anuais ou de longo prazo, com custos que podem variar dependendo da versão e dos recursos específicos incluídos. Esse modelo de licenciamento pode representar um desafio para estudantes e indivíduos que buscam acessar a ferramenta, em comparação com a proposta de desenvolvimento do *software*, que visa oferecer uma solução mais acessível, especialmente voltada para fins educacionais.

Além do SAP2000 e do *Robot Structural Analysis*, também foi estudado o *software* Ftool. O Ftool é um programa de análise estrutural de pórticos planos, que tem como objetivo a prototipagem de estruturas. O programa foi inicialmente desenvolvido para uso em sala de aula, mas evoluiu para uma ferramenta frequentemente utilizada em projetos estruturais profissionais. O Ftool permite a criação e manipulação de elementos estruturais, e a análise estrutural.

Em comparação com o *software* em desenvolvimento, o Ftool apresenta distinções notáveis. O Ftool é estruturado para a análise de estruturas bidimensionais, em contraste com o *software* em questão, que tem seu foco na análise de estruturas tridimensionais, proporcionando uma compreensão mais abrangente das interações estruturais em três dimensões.

Uma distinção importante reside na capacidade de fornecimento de equações de esforços internos. O Ftool não oferece essa funcionalidade de maneira direta, demandando cálculos adicionais para obter essas informações. O *software* em desenvolvimento, por outro lado, visa fornecer essas equações de esforços internos de forma integrada, simplificando o processo de análise para os usuários.

Outra diferença essencial está relacionada às reações nos apoios. Enquanto o Ftool não fornece diretamente essas reações, exigindo cálculos adicionais por parte do usuário, o *software* proposto tem como objetivo incluir essa funcionalidade para tornar o processo de cálculo das etapas intermediárias mais acessível ao estudante.

Além disso, o Ftool requer informações adicionais, como dados sobre materiais e seções, o que pode adicionar complexidade ao processo de obtenção dos resultados. O *software* em desenvolvimento procura simplificar esse aspecto, tornando o fornecimento dessas informações mais intuitivo e eficiente.

O método de apresentação dos resultados também difere. O *Ftool* fornece valores absolutos de rotação e deformação, enquanto o *software* em desenvolvimento se concentra em

fornecer resultados em função módulo de elasticidade, inércia e área da seção, facilitando o entendimento das etapas intermediárias de cálculo.

3 METODOLOGIA

Neste estudo, o delineamento metodológico foi inicialmente caracterizado com base nas categorias de pesquisa definidas por Gerhardt e Silveira (2009), as quais incluem a abordagem, a natureza, os objetivos e os procedimentos da pesquisa.

A pesquisa utilizada neste trabalho é de natureza prática, já que visa o desenvolvimento de novas tecnologias que oferecem uma solução prática para a melhoria no aprendizado dos estudantes, tendo foco na aplicação direta do conhecimento teórico na criação de uma ferramenta útil. Ela também tem objetivo exploratório, já que propõe o desenvolvimento de um novo *software*, com recursos inovadores, onde serão exploradas novas maneiras de atingir os objetivos, além de sugerir novas maneiras de melhorar o processo de aprendizagem dos estudantes de engenharia civil, o que alinha a pesquisa com o objetivo exploratório.

Além disso ela pode ser classificada como uma pesquisa de abordagem quantitativa, já que o *software* a ser desenvolvido manipulará valores numéricos, executará cálculos e gerará resultados quantitativos, para isso também foi necessário realizar um levantamento bibliográfico dotado majoritariamente de elementos quantitativos, como métodos de cálculo de esforços e deformações. E quanto aos procedimentos técnicos ela é uma pesquisa-ação, pois busca a implementação de ações práticas destinadas a melhorar o aprendizado de estruturas.

A etapa de estudo deste trabalho foi conduzida realizando-se uma pesquisa bibliográfica abrangente, que incluiu a análise de materiais acadêmicos e técnicos relacionados à análise estrutural, resistência dos materiais e programação em linguagem C#.

O processo de pesquisa envolveu a utilização de uma variedade de fontes, incluindo livros, repositórios *online* e documentação técnica. As buscas foram conduzidas em bibliotecas acadêmicas, repositórios online, e recursos especializados, garantindo que as fontes utilizadas fossem confiáveis e atualizadas.

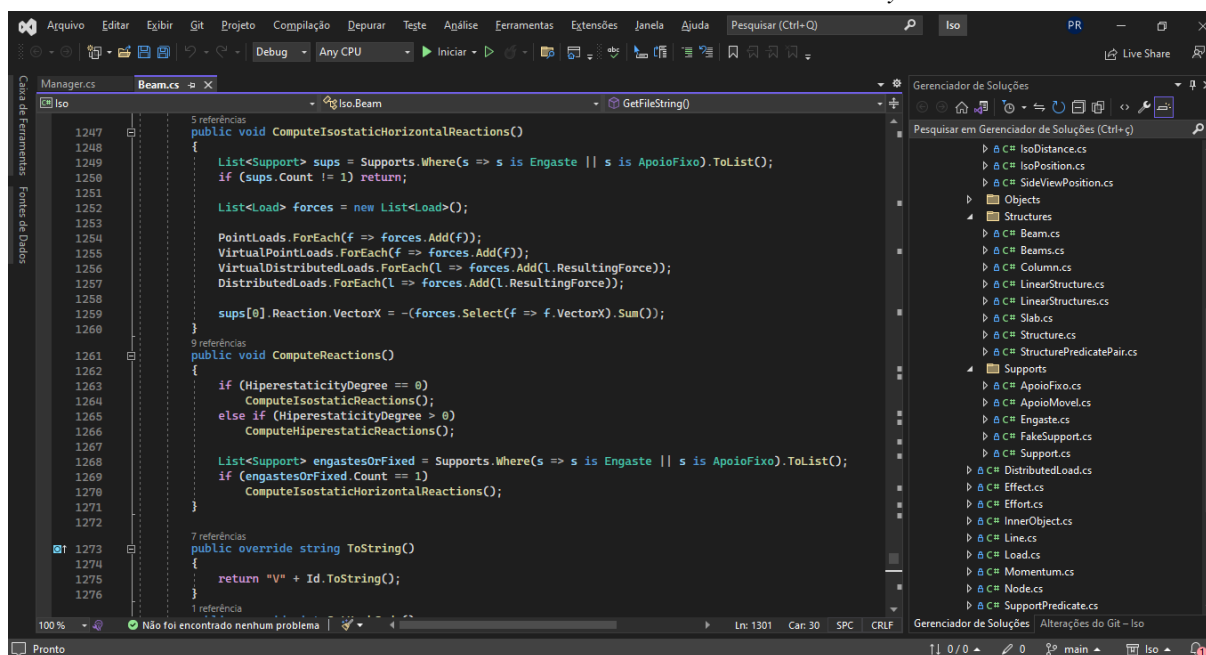
A seleção das fontes foi realizada de modo a considerar a pertinência do conteúdo, a qualidade das fontes e a contribuição para os tópicos abordados. A revisão da literatura e a pesquisa documental desempenharam um papel fundamental na identificação de lacunas no conhecimento existente e na definição de um quadro teórico que serviu de base para o desenvolvimento do *software*.

3.1 FERRAMENTAS COMPUTACIONAIS UTILIZADAS NO DESENVOLVIMENTO

A linguagem de programação escolhida para o desenvolvimento do *software* foi o C#, juntamente com o framework .NET, por ser uma linguagem bem estruturada, de tipagem forte e suporte à programação orientada a objetos, o que a torna uma linguagem ideal para desenvolvimento de aplicações volumosas e de maior complexidade, além de que o .NET fornece diversos recursos e bibliotecas para facilitar o desenvolvimento, como o *Windows Forms* e o GDI+, ferramentas já descritas nos capítulos anteriores.

O ambiente de desenvolvimento utilizado para a codificação foi o Visual Studio Community, uma suíte de desenvolvimento integrado (IDE) gratuita desenvolvida pela Microsoft.

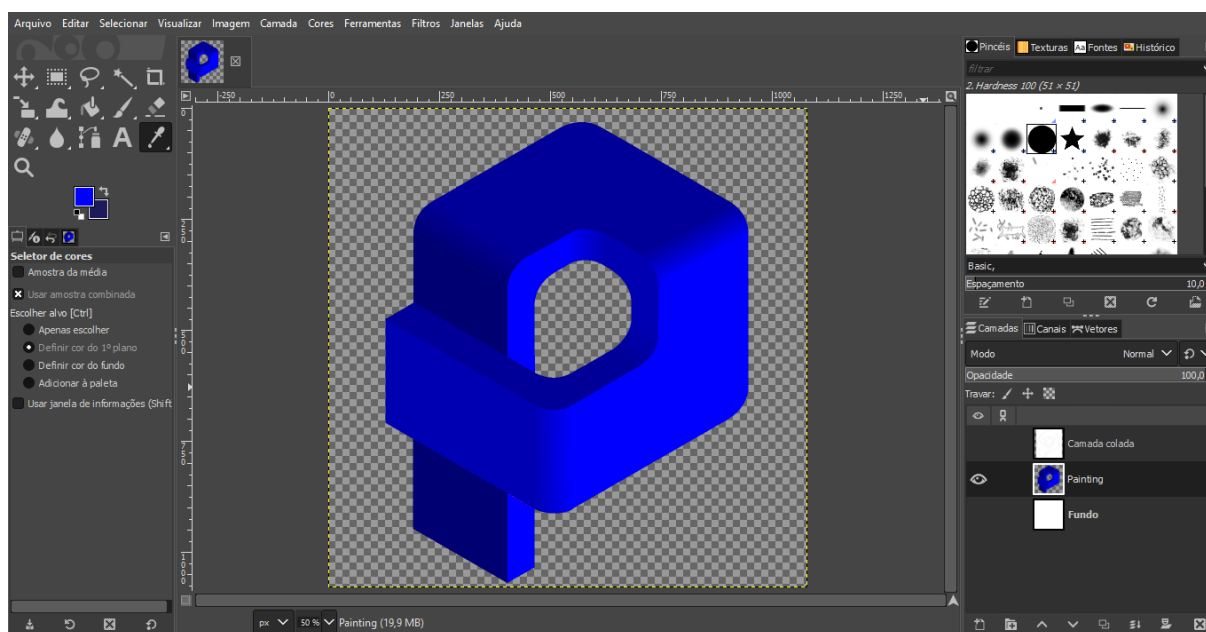
FIGURA 23 – Interface do *Visual Studio Community*



Fonte: Elaborado pelo autor, 2023

Para os elementos de arte e design do *software*, como a logomarca, os ícones de botões e outros elementos visuais, foi utilizado o *GIMP* (*GNU Image Manipulation Program*), um *software* gratuito e de código aberto utilizado para a edição de imagens, uma alternativa poderosa a programas comerciais como o *Adobe Photoshop*.

FIGURA 24 – Interface do GIMP



Fonte: Elaborado pelo autor, 2023

Em algumas etapas de teste do aplicativo foi utilizado o *software Ftool*, definido em sua página oficial como uma ferramenta para análise estruturas de estruturas planas. Com o aplicativo é possível obter resultados como diagramas de esforços internos e de deformações e rotações, e com alguns cálculos também é possível extrair informações sobre as reações nos apoios.

3.2 CRIAÇÃO DO SOFTWARE

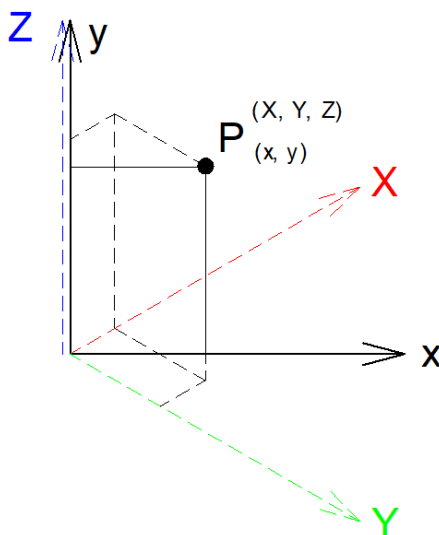
Como já mencionado no capítulo 2.18.2, a .NET possui o GDI+, uma biblioteca para a criação de elementos gráficos, no entanto a ferramenta possibilita a criação de desenhos bidimensionais, com coordenadas x e y , dadas em pixels que partem do canto superior esquerdo da tela e seguem nos sentidos direito e inferior da janela.

Para possibilitar a criação de elementos em um plano tridimensional foi criada uma classe especial chamada de *IsoPosition*, que abstrai o conceito de uma posição tridimensional, onde no momento da instanciação de um objeto desta classe é preciso fornecer os parâmetros x , y e z . Internamente à estrutura da classe foram criados mecanismos que convertem estas posições tridimensionais em posições bidimensionais que serão efetivamente utilizadas pelo GDI+ para imprimir o resultado final na tela.

A conversão destas posições é feita por meio de um esquema de vista isométrica, onde a representação gráfica do objeto tridimensional é projetada em um plano bidimensional. Desta

forma é possível traduzir uma posição tridimensional em outra de duas dimensões, como mostra a figura 25, onde o ponto P pode representar tanto a posição bidimensional (x, y) quanto a posição tridimensional isométrica (X, Y, Z) .

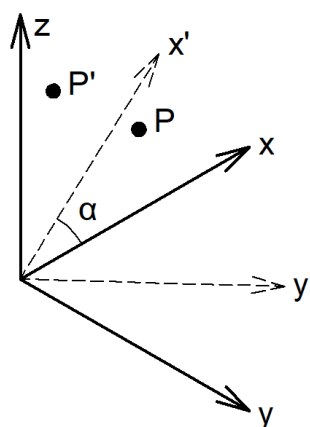
FIGURA 25 – Conversão de posições tridimensionais em bidimensionais



Fonte: Elaborado pelo autor, 2023

No entanto, apenas a conversão de dimensionalidade gera uma imagem isométrica estática, onde não é possível rotacionar os elementos, para contornar este problema foi utilizado um sistema de conversão dos pontos bidimensionais baseado em um ângulo de rotação, onde o campo dos eixos foi dividido em 8 partes, separadas por ângulos de 30° e 60° , e para cada uma das partes foram definidas duas funções, uma correlacionando o ângulo de rotação com a posição x , e uma correlacionando o ângulo de rotação com a posição y . Deste modo foi possível desenvolver um mecanismo em que o usuário pode alterar a posição da tela ao alterar o valor do ângulo de deslocamento do eixo bidimensional, e as posições x e y serão convertidas para uma posição correspondente que possa ser interpretada pelo GDI+.

O esquema é mostrado na figura 26, onde o eixo é rotacionado em um ângulo α e o ponto P passa a ser representado na tela na posição P'.

FIGURA 26 – Rotação dos pontos

Fonte: Elaborado pelo autor, 2023

Para tornar esta conversão possível foi necessário criar uma classe, apelidada de *AxisConverter*, utilizada pela classe *IsoPosition* para realizar a conversão dos pontos tridimensionais em pontos bidimensionais levando em conta o ângulo de rotação dos eixos. Esta classe possui mecanismos que convertem a posição tridimensional inicial em um objeto de outra classe, chamada de *BiPosition*, que representa uma posição bidimensional, que por sua vez converte os pontos em posições na tela do computador, dadas em pixels, que são interpretadas pelo GDI+.

A classe *BiPosition* também utiliza parâmetros de escala de *zoom* e deslocamentos horizontais e verticais para converter as posições, possibilitando que o usuário possa dar *zoom* nos elementos gráficos e também deslocá-los pela tela do computador.

3.2.1 Criação de Classes Utilitárias

Para viabilizar implementações subsequentes, foram criadas classes utilitárias, abrangendo funcionalidades que incluem cálculos de derivadas e integrais, equações de reta e outros recursos. Essas classes utilitárias desempenham um papel fundamental na expansão e no aprimoramento das capacidades do sistema, fornecendo uma base para a criação de cálculos e modelos mais avançados.

3.2.1.1 Classe para cálculos gerais

Foi criada uma classe específica, para a execução de cálculos gerais, que seriam utilizados mais adiante durante o desenvolvimento do *software*, esta classe possui métodos que solucionam sistemas lineares de equações de duas incógnitas, utilizando o método da substituição e sistemas lineares com múltiplas incógnitas, utilizando uma lógica de escalonamento. Como visto no capítulo de levantamento bibliográfico, a obtenção dos esforços internos em estruturas hiperestáticas envolve a solução de sistemas de equações de compatibilidade, os métodos em questão serão aplicados posteriormente para solucionar estes problemas.

Também foram criados métodos que calculam as integrais e derivadas de funções polinomiais, onde é fornecida uma matriz contendo os multiplicadores e expoentes e as funções retornam os multiplicadores e expoentes das integrais e derivadas da matriz. Estes métodos serão úteis para diversos cálculos, que como visto no capítulo de revisão bibliográfica, serão utilizados posteriormente.

Também estão presentes na classe métodos que obtêm o ponto de intersecção entre duas retas, para isso são fornecidas como parâmetro as equações paramétricas das retas e uma lógica interna é executada para a identificação do ponto em comum entre as duas retas utilizando os conceitos descritos no capítulo 2.16.

3.2.1.2 Classe para as equações paramétricas de reta

Foi criada uma classe que representa as equações paramétricas da reta, para instanciar um objeto desta classe deve-se passar como parâmetro duas posições tridimensionais. A classe possui métodos para a obtenção de pontos na reta, pontos de intersecção com outra equação de reta, realização de testes de pertencimento de um ponto à reta e testes para verificar se um ponto pertence a ela e a outra equação de reta simultaneamente.

3.2.1.3 Classes para as equações de Momento Fletor, Força Cortante e Normal

Foram criadas classes para a representação das equações de esforços internos, elas possuem mecanismos de obtenção de valores em um determinado ponto, além de valores máximos e mínimos da equação.

3.2.1.4 Classes que representam os esforços externos

Foram criadas classes que representam as cargas distribuídas, forças pontuais e momentos. Todos os objetos que são instanciados a partir destas classes devem receber no momento da instanciação a estrutura a qual pertencem e a posição em que estão sendo aplicadas na estrutura.

Elas possuem propriedades que indicam os seus valores em kN, posições no eixo da estrutura e posições no espaço tridimensional, além de possuírem métodos que definem seu design gráfico e o modo que serão desenhadas na tela do computador, além de possuírem mecanismos que definem de que modo elas interagirão com os usuários, como campos de seleção e estados de seleção e não-seleção.

A classe que representa a carga distribuída também possui mecanismos de fornecem a sua força resultante, ou seja, um objeto do tipo carga pontual que representa a força resultante de uma determinada carga distribuída.

3.2.2 Criação dos Apoios

Foram criadas classes que representam os apoios fixo, móvel e engaste. Como estas entidades possuem características em comum foi criada uma superclasse que abstrai o conceito de um suporte genérico, chamada *Support*.

A classe *Support* contém propriedades e métodos que são comuns para todos os outros apoios, como as posições tridimensionais no espaço, a posição no eixo da estrutura a qual pertence, as reações, que armazenam objetos do tipo carga pontual, e também ferramentas que permitem a interação com o usuário, como caixas de seleção e estado de selecionado e não-selecionado.

As classes que representam os apoios fixo, móvel e engaste são classes filhas da classe *Structure*, e por sua vez herdam todas as propriedades da classe mãe, além de sobrescreverem os métodos referentes à sua representação gráfica na tela do computador.

3.2.3 Criação das Estruturas

Foram criadas classes que representam as estruturas envolvidas, sendo elas as vigas e pilares, como estas entidades possuem diversas características em comum, foi criada uma superclasse que representa uma estrutura genérica, chamada *Structure*.

A classe *Structure* possui propriedades e métodos que definem as características e comportamento de uma estrutura, como a sua posição do espaço, os esforços externos que estão sendo aplicados nela, os apoios que ela possui e as equações de esforços internos.

As classes que representam as vigas e pilares herdam da classe *Structure*, e, portanto, também têm acesso às mesmas propriedades e métodos, as duas classes também sobrescrevem o método que define sua representação gráfica, possibilitando assim o polimorfismo, ou seja, que os dois tipos de estrutura tenham comportamentos diferentes quando impressos na tela do computador.

As classes que representam as estruturas possuem métodos que verificam os pontos de intersecção com outros objetos, lógicas que definem se ele está sendo apoiado ou se serve de apoio para outras estruturas utilizando os mecanismos já mencionados anteriormente. Elas também possuem mecanismos de inserção e remoção de apoios e esforços de acordo com as necessidades da execução do código.

Elas também executam internamente o cálculo das reações dos apoios, identificando o próprio grau de hiperestaticidade. Caso ela seja uma estrutura isostática, há a execução de uma lógica para o cálculo das reações utilizando o método das seções, caso ela seja uma estrutura hiperestática, a lógica anterior é reutilizada para o cálculo das reações dos sistemas isostáticos necessários pelo método da força, tendo as equações envolvendo integrais e resolução de sistemas lineares resolvidas utilizando as classes utilitárias mencionadas anteriormente.

Estas classes também possuem métodos que desenharam na tela do computador suas respectivas equações de esforços internos para cada um dos objetos instanciados a partir delas.

3.2.3.1 Classe auxiliar para as estruturas

Foi criada uma classe auxiliar para realizar operações nas estruturas, esta classe contém lógicas para a obtenção dos pontos que dividem as seções nas estruturas, elas são importantes para a aplicação dos algoritmos que obtém as equações de esforços internos das estruturas utilizando o método das seções, já que para cada trecho será necessária a aplicação do algoritmo para a obtenção das equações. São dois os métodos, um obtém os pontos de divisão das seções para uma estrutura, e outro que obtém os pontos de divisão de seção que existem em comum em duas estruturas, sendo este último usado na obtenção de deslocamentos e rotações utilizando o princípio dos trabalhos virtuais, já que tanto o sistema principal quanto o sistema virtual precisam ter as mesmas segmentações para a aplicação dos cálculos.

A classe também possui algoritmos para a obtenção das equações de esforços internos utilizando as equações de equilíbrio, a lógica utiliza os esforços solicitantes e reações de apoios já calculadas para a obtenção das equações.

Há também algoritmos para calcular as deformações nas estruturas em um ponto da estrutura, sendo dois algoritmos, um obtém as deformações em função da área da seção e do módulo de elasticidade, ou seja, a deformação é obtida por meio do cálculo do trabalho virtual gerado pelo esforço normal. O outro obtém as deformações em função do módulo de elasticidade e do momento de inércia da peça, utilizando o cálculo de trabalho virtual gerado pelo momento fletor. Também existem outros dois métodos semelhantes para a obtenção das rotações em um ponto, que seguem a mesma lógica dos primeiros, também utilizando o princípio dos trabalhos virtuais.

Além disso a classe também conta com métodos que retornam as deformações, rotações e esforços internos mais notórios para a representação gráfica, como valores máximos e mínimos e valores das extremidades. Estes valores serão utilizados nos diagramas de esforços internos e deformações.

3.2.4 Gerenciador de ambiente

O gerenciador de ambiente, apelidado de *Manager*, é uma classe criada para gerenciar todos os objetos e elementos em tempo de execução, apenas um objeto desta classe é instanciado no momento inicial da execução do *software*. Nessa instância são armazenadas

todas as estruturas, carregamentos, e demais elementos que venham a ser criados durante a execução do aplicativo.

A classe conta com métodos de atualização do ambiente, gestão de elementos selecionados, comunicação entre elementos, impressão dos elementos na tela do computador e inserção e remoção de elementos no ambiente.

3.2.4.1 Métodos para atualização do ambiente

Durante a execução do *software*, o usuário pode desenhar objetos e definir a relação entre eles, toda vez que o usuário executa uma ação durante a execução do *software*, o método de atualização do ambiente é chamado, durante o processo de atualização o gerenciador executa comandos para a geração de nós. Os nós são entidades que marcam os pontos importantes no ambiente tridimensional que possibilitam a interação do usuário com o aplicativo, eles marcam os pontos das extremidades e da metade de estruturas, pontos de intersecção entre elas e o ponto inicial do ambiente tridimensional.

Após gerados os nós, é executada a ação de construção dos elementos estruturais, neste método é realizada uma iteração por todas as estruturas presentes no ambiente, e para cada uma delas são chamados os métodos de identificação de intersecções, onde cada uma das estruturas reconhecem seus apoios e esforços internos baseando-se nas interações com as demais estruturas. Após todas as estruturas terem reconhecido seus apoios e esforços externos solicitantes, é chamado o método de cálculo das reações de apoio para cada uma das estruturas.

Após a construção das estruturas é chamado o método de desenho dos elementos na tela, para isso é feita uma iteração por todos os elementos, e para cada um deles é chamado o método de desenho da respectiva instância.

3.2.4.2 Métodos para a interação com o usuário

Conforme o usuário interage com o ambiente de desenho tridimensional, podem acontecer modificações tanto no ambiente de execução quanto no ambiente gráfico do aplicativo. Estas interações podem ser:

Modificação da vista: O usuário pode modificar a vista tridimensional durante a execução do aplicativo, como transladar a vista, rotacioná-la e dar *zoom*, quando estas ações são feitas, o gerenciador altera as propriedades que definem o deslocamento, *zoom* e ângulo de rotação da tela e executa a ação de desenhar todos os elementos novamente.

Cliques e seleção de elementos: Quando o usuário seleciona os elementos durante a execução do aplicativo o gerenciador é responsável por atualizar o estado de seleção de cada um dos objetos presentes no ambiente e redesenhá-los, já que o estado de seleção altera aspectos visuais dos elementos, e as propriedades que são fornecidas ao usuário. Se uma viga é selecionada, por exemplo, seu aspecto visual será alterado e suas propriedades, como diagramas de esforços internos e reações de apoio serão desenhados na tela.

Inserção e remoção de elementos: Quando o usuário desenha uma nova estrutura ou insere novas cargas em uma estrutura, o gerenciador age inserindo um novo objeto da estrutura no ambiente ou inserindo a nova carga em uma estrutura já existente, e executa novamente a ação de atualização, deste modo, todo o processo descrito no capítulo 3.2.4.1 é repetido. O mesmo acontece para a remoção de elementos.

Alteração na configuração das estruturas: Durante o uso do aplicativo o usuário pode optar por trocar o tipo de apoio em uma estrutura ou alterar o modo com que as estruturas se comportam, como por exemplo alterando a relação entre estrutura de apoio e estrutura apoiada, nestes casos, o gerenciador altera as propriedades dos elementos modificados e executa o processo de atualização novamente.

3.2.5 Criação dos Formulários

Para possibilitar a interação do usuário com o aplicativo, foram utilizadas as bibliotecas do *Windows Forms* para a criação de formulários. Os formulários contêm as vistas tridimensionais, controles de interação e elementos gráficos necessários para tornar a experiência de uso do aplicativo agradável. Também é nos formulários que acontece toda a lógica de interação do usuário com o gerenciador de ambiente.

3.2.5.1 Formulário principal

O formulário principal desempenha um papel central e multifuncional na interface do usuário, nele está presente a vista tridimensional do aplicativo, que possibilita a interação do usuário com os elementos do ambiente de execução.

Foi criada também uma barra lateral, onde está presente o quadro de visualização dos diagramas e controles de formulário que possibilitam o usuário de modificar estruturas, explorar suas características e manipular sua configuração com facilidade. A interatividade é aprimorada através de uma série de botões que concedem ao usuário o controle sobre o processo de design e análise.

Também foi adicionada uma barra superior, que abriga os botões de desenho de estruturas, botões que oferecem a capacidade de adicionar cargas distribuídas, forças pontuais e momentos às estruturas, permitindo uma manipulação completa das condições de carregamento. A barra superior também contém botões para salvamento e carregamento de arquivos, além de um botão que dá acesso ao formulário que lida com a obtenção de valores específicos para os pontos das estruturas.

3.2.5.2 Formulários para inserção de cargas

Foram criados formulários que permitem que o usuário insira esforços externos solicitantes na estrutura selecionada, sendo ativados quando o usuário clicar sobre os botões de inserção de cargas distribuídas e cargas pontuais.

O formulário para adicionar cargas distribuídas oferece campos dedicados para especificar o valor do carregamento em kN/m e a posição de incidência, permitindo ao usuário modelar de forma precisa como as cargas são distribuídas na estrutura. Da mesma forma, o formulário para adicionar cargas pontuais e momentos possibilita a inserção desses componentes de carregamento, garantindo que sua localização e magnitude sejam devidamente representadas na análise tridimensional da estrutura.

3.2.5.3 Formulário para a modificação do tipo de apoio

Este formulário foi programado para ser ativado quando o usuário seleciona um apoio da estrutura selecionada e clica sobre o botão de edição de suporte. Ele possui as caixas de marcação contendo as opções de apoio disponíveis, sendo elas Apoio Móvel, Fixo e Engaste. Quando o tipo de apoio é escolhido o gerenciador de ambiente é invocado para fazer as modificações na instância da estrutura selecionada.

3.2.5.4 Formulário para a modificação da relação entre as estruturas

Este formulário foi criado para permitir que o usuário altere as configurações de relação entre estruturas, ou seja, definir se uma estrutura serve de apoio para a outra, ou se a carga de uma estrutura será transferida para a outra. Neste formulário o usuário pode escolher entre as opções de a estrutura selecionada apoiar a outra, outra estrutura apoiar a viga selecionada, ou que a relação seja detectada automaticamente.

3.2.5.5 Formulário para a obtenção dos valores no ponto

Foi criado também um formulário dedicado à obtenção dos valores dos esforços internos, deformações e rotações em determinado ponto da estrutura. Nele, o usuário deve fornecer o ponto na estrutura em que deseja saber os valores e selecionar se deseja saber os valores dos esforços internos, deformações ou rotações. Dado isso, o formulário buscará a estrutura selecionada por meio do gerenciador de ambiente, e obterá os valores desejados por meio da instância do objeto.

3.2.6 Mecanismos de Salvamento e Carregamento de Arquivos

As entidades criadas no decorrer de uma execução de um aplicativo de área de trabalho existem apenas durante aquela execução, isso significa que quando o aplicativo é fechado todas as informações e modificações são perdidas. Dado isso, foram desenvolvidos mecanismos para salvar os dados de execução do aplicativo em arquivos de forma que eles possam ser carregados

em outras execuções. Para que isso fosse possível, foi criada uma notação em forma de texto que pudesse ser interpretada pelo aplicativo no momento inicial da execução de cada sessão do *software*.

Para isso, em todas as entidades, sejam elas estruturas, apoios ou cargas, foram adicionadas funcionalidades de transcrição de suas propriedades no formato de notação de texto e também de instanciar novos objetos a partir de um arquivo já existente. Deste modo, o arquivo de salvamento pode ser gerado e interpretado de forma recursiva.

Quando o usuário executa a ação de salvamento, o gerenciador de ambiente faz uma iteração por todas as instâncias de estruturas existentes no ambiente, que por sua vez fazem uma iteração por todos os apoios e esforços que contêm. Deste modo é gerado um arquivo contendo todas as informações daquela execução do aplicativo. O mesmo é feito no momento da leitura de um arquivo, e todas as entidades são geradas de forma recursiva e adicionadas ao ambiente de execução do aplicativo.

3.2.7 Aplicação de Testes

Durante todas as fases do desenvolvimento do *software*, adotou-se uma abordagem rigorosa de teste e validação para garantir a qualidade e precisão das funcionalidades implementadas. Esta estratégia de teste envolveu a verificação das etapas do processo de desenvolvimento, incluindo cálculos manuais, verificações das funcionalidades do ambiente gráfico e verificações de funcionalidades de componentes específicos do *software*, como comportamento dos controles de formulário.

Foram realizados cálculos manuais para determinar os resultados esperados em cenários específicos, e esses resultados foram comparados com os dados gerados pelo *software* e complementados por verificações utilizando a ferramenta Ftool. Essa abordagem permitiu identificar discrepâncias e assegurar que o *software* produzisse resultados precisos e consistentes em relação às fórmulas e teorias utilizadas no desenvolvimento.

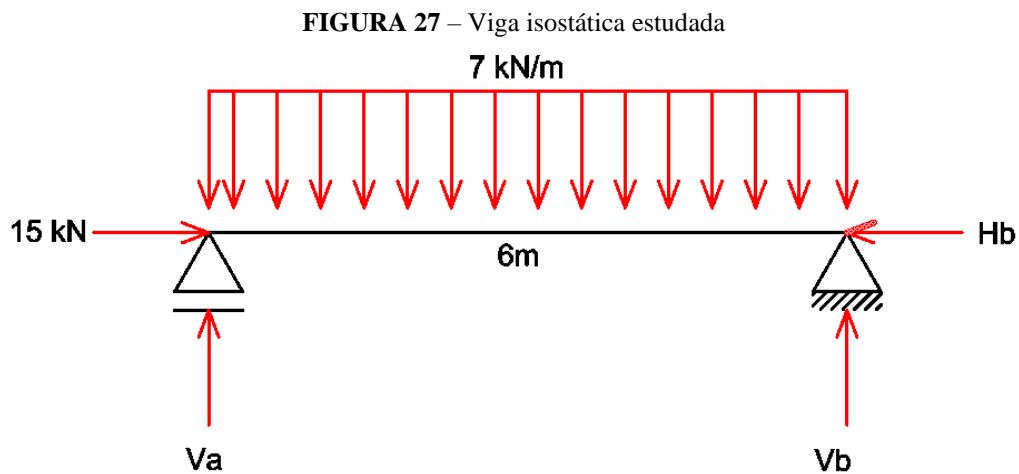
A construção do ambiente gráfico também foi submetida a uma série de testes. A funcionalidade de cada componente gráfico, como o comportamento da visualização tridimensional e os formulários de adição de cargas e momentos, foi verificada para garantir que todas as operações ocorressem conforme o esperado. Isso incluiu testes de entrada de dados, interação com os elementos da interface e a resposta do sistema às ações do usuário.

4 ESTUDO DE CASO

Neste capítulo serão feitos dois estudos de caso. No primeiro será resolvida uma viga isostática pelo método das seções, e no segundo será resolvida uma viga hiperestática pelo método da força. Os resultados deste estudo serão comparados aos resultados do aplicativo no capítulo de resultados.

4.1 ESTUDO DE CASO 1: OBTENÇÃO DOS ESFORÇOS INTERNOS EM UMA VIGA ISOSTÁTICA

Para exemplificar a aplicação do método das seções, será feito um estudo de caso, exemplificando a obtenção de esforços internos na viga da figura 27, que mostra o diagrama de corpo livre da viga biapoiada de 6 metros, onde **Va** é a reação vertical do apoio móvel da esquerda e **Vb** e **Hb** são as reações vertical e horizontal do apoio fixo da direita, respectivamente.



Fonte: Elaborado pelo autor, 2023

Seguindo os passos do método das seções descritos na revisão da literatura, a obtenção dos valores dos apoios é feita utilizando as equações de equilíbrio, para isso escolheu-se a extremidade esquerda da viga como eixo para o cálculo do momento resultante e tomando-se a variável **x** para representar a distância em relação à extremidade esquerda, tem-se:

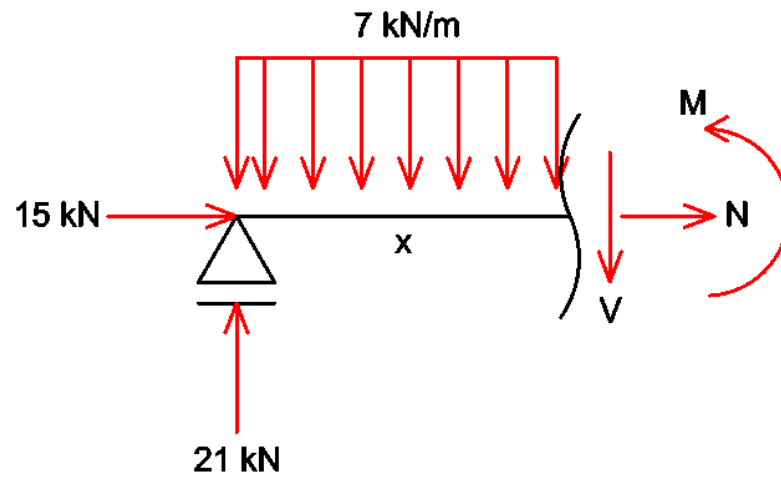
$$\sum M_o = 0 \therefore (Vb \cdot 6) - \left(\int_0^6 7 \, dx \right) \cdot 3 = 0$$

$$\sum Fy = 0 \therefore Va + Vb - \int_0^6 7 dx = 0$$

$$\sum Fx = 0 \therefore Hb - 15 = 0$$

Resolvendo-se as equações tem-se que $Va = 21 \text{ kN}$, $Vb = 21 \text{ kN}$ e $Hb = 15 \text{ kN}$. Em seguida, traça-se um corte imaginário a uma distância x da viga, e desenha-se o diagrama de corpo livre indicando os esforços internos, onde N é a força normal, V é a força cortante e M é o momento fletor, obedecendo as convenções descritas no capítulo 2.5.

FIGURA 28 – Aplicando o método das seções



Fonte: Elaborado pelo autor, 2023

Assim, é possível utilizar as equações de equilíbrio para obter-se os esforços internos no ponto genérico x .

$$\sum Mx = 0 \therefore M - 21x + \left(\int_0^x 7 \cdot dx \right) \cdot \frac{x}{2} = 0$$

$$\sum Fy = 0 \therefore V + \int_0^x 7 \cdot dx - 21 = 0$$

$$\sum Fx = 0 \therefore N + 15 = 0$$

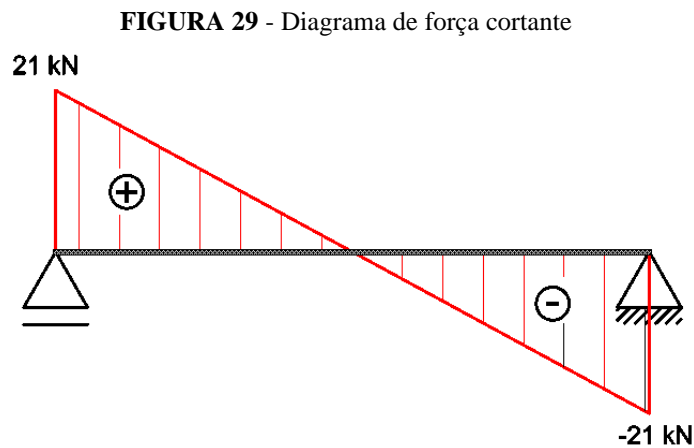
Deste modo, os valores dos esforços internos para um ponto qualquer x no segmento da viga são dados pelas equações:

$$M = -3,5x^2 + 21x$$

$$V = -7x + 21$$

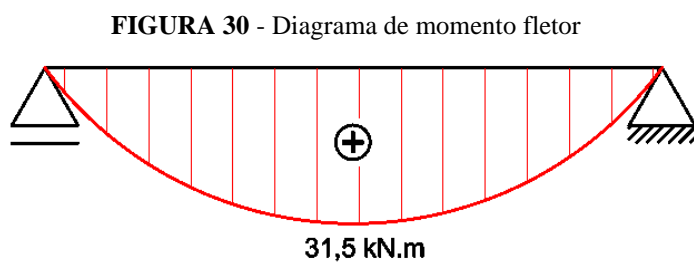
$$N = -15 \text{ kN}$$

Na figura 29 observa-se o diagrama de força cortante, traçado com base na equação de cortante.



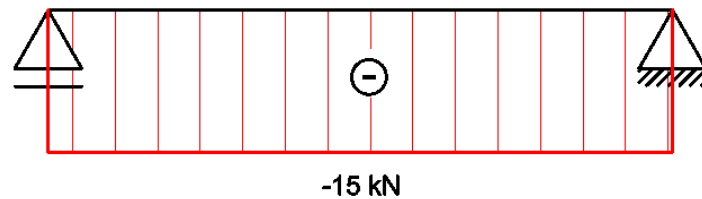
Fonte: Elaborado pelo autor, 2023

já o diagrama de momento fletor na figura 30 é desenhado com base na equação de momento fletor e, por convenção, desenha-se a parte positiva do gráfico abaixo das fibras inferiores da barra, e a parte negativa acima das fibras superiores.



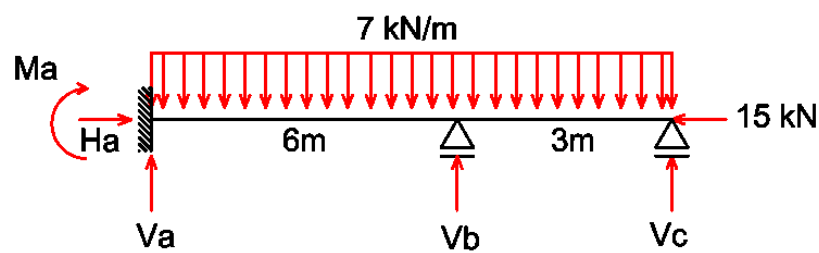
Fonte: Elaborado pelo autor, 2023

Por fim, na figura 31, tem-se o diagrama de força normal, também por convenção os esforços de tração são desenhados na parte superior da barra, e são tidos como positivos, e os de compressão são desenhados na parte superior, e tidos como negativos.

FIGURA 31 – diagrama de força normal

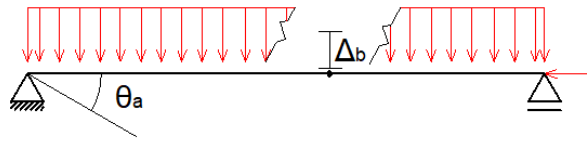
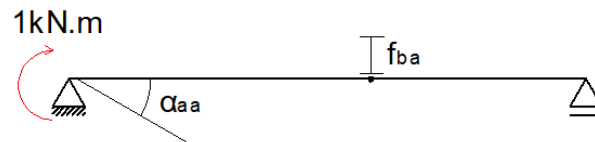
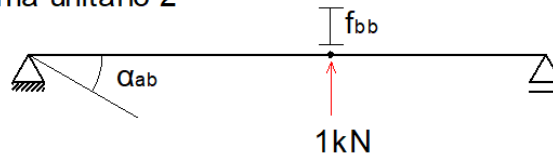
Fonte: Elaborado pelo autor, 2023

4.2 ESTUDO DE CASO 2: OBTENÇÃO DE ESFORÇOS INTERNOS EM UMA VIGA HIPERESTÁTICA

FIGURA 32 – Viga hiperestática estudada

Fonte: Elaborado pelo autor, 2023

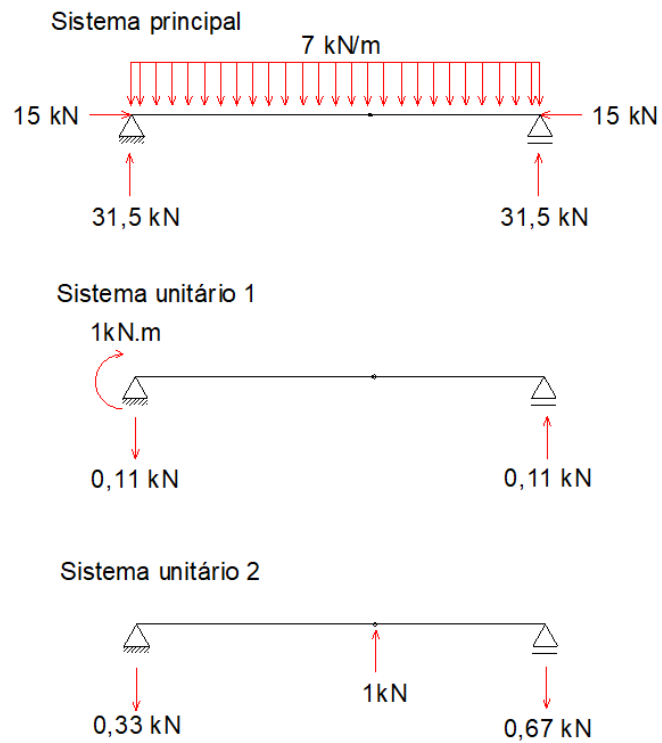
Para este estudo de caso será utilizada a viga apresentada na figura 32. Primeiramente deve-se retirar os vínculos de modo a formar um sistema principal e os sistemas redundantes isostáticos. Para que possam ser encontrados os coeficientes de flexibilidade devem ser aplicadas forças e momentos unitários nos sistemas redundantes, deste modo, sistemas isostáticos a serem calculados estão expressos na figura 33.

FIGURA 33 – Deslocamentos e rotações dos sistemas isostáticos**Sistema principal****Sistema unitário 1****Sistema unitário 2**

Fonte: Elaborado pelo autor, 2023

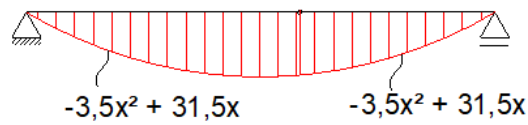
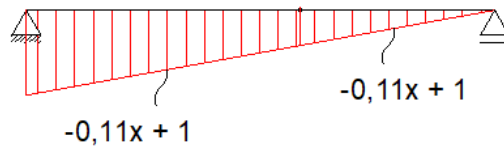
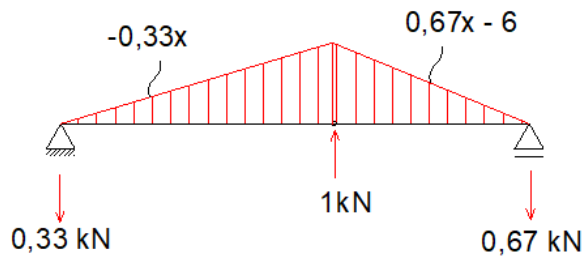
As reações e as equações de momento fletor para cada um destes sistemas podem ser encontradas utilizando as três equações de equilíbrio e o método das seções, explorados no capítulo 2.11.

Na figura 34 é possível ver os esforços externos nos três sistemas, incluindo as reações nos apoios calculadas utilizando-se as três equações de equilíbrio.

FIGURA 34 – Esforços externos dos sistemas isostáticos

Fonte: Elaborado pelo autor, 2023

E na figura 35 é possível ver os diagramas e equações de momento fletor, calculados por meio do método das seções para cada um dos segmentos dos sistemas.

FIGURA 35 – Momentos fletores dos sistemas isostáticos**Sistema principal****Sistema unitário 1****Sistema unitário 2****Fonte:** Elaborado pelo autor, 2023

O próximo passo é encontrar as deformações, rotações e coeficientes de flexibilidade utilizando o Princípio dos Trabalhos Virtuais, portanto tem-se que:

$$1. \theta_a = \int_{x_1}^{x_2} \frac{M \cdot m_\theta}{EI} dx = \int_0^9 \frac{(-3,5x^2 + 31,5x) \cdot (-0,11x + 1)}{EI} dx = \frac{212,63}{EI}$$

$$\begin{aligned} 1. \Delta_b &= \int_{x_1}^{x_2} \frac{M \cdot m}{EI} dx \\ &= \frac{1}{EI} \left(\int_0^6 (-3,5x^2 + 31,5x) \cdot (-0,33x) \cdot dx \right. \\ &\quad \left. + \int_6^9 (-3,5x^2 + 31,5x) \cdot (0,67x - 6) \cdot dx \right) = \frac{-519,69}{EI} \end{aligned}$$

$$1. \alpha_{aa} = \int_{x_1}^{x_2} \frac{M \cdot m_\theta}{EI} dx = \int_0^9 \frac{(-0,11x + 1)^2}{EI} dx = \frac{3}{EI}$$

$$1. f_{bb} = \int_{x_1}^{x_2} \frac{M \cdot m}{EI} dx = \frac{1}{EI} \left(\int_0^6 (-0,33x)^2 dx + \int_6^9 (0,67x - 6)^2 \cdot dx \right) = \frac{12}{EI}$$

$$1. \alpha_{ab} = 1. f_{ba} = \int_{x_1}^{x_2} \frac{M \cdot m}{EI} dx$$

$$= \frac{1}{EI} \left(\int_0^6 (-0,11x + 1) \cdot (-0,33x) \cdot dx + \int_6^9 (-0,11x + 1) \cdot (0,67x - 6) \cdot dx \right)$$

$$= \frac{-4}{EI}$$

Com os valores das deformações, rotações e coeficientes encontrados, agora é possível encontrar as reações da viga original utilizando as equações de compatibilidade:

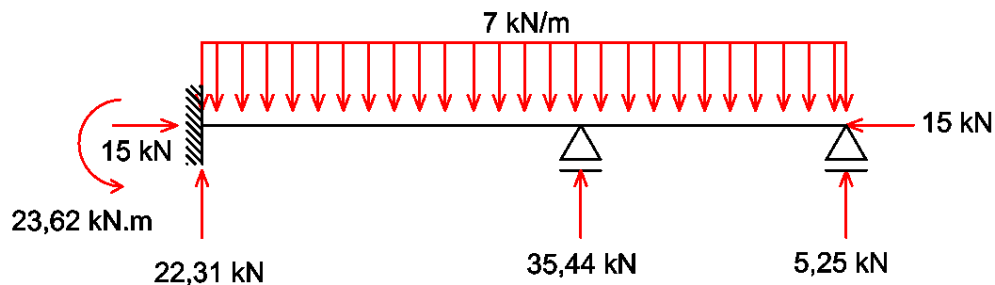
$$\frac{212,63}{EI} + \frac{3}{EI} \cdot M_a - \frac{4}{EI} \cdot V_b = 0$$

$$\frac{-519,69}{EI} - \frac{4}{EI} \cdot M_a + \frac{12}{EI} \cdot V_b = 0$$

Resolvendo o sistema de equações de compatibilidade tem-se que **M_a** vale **-23,62 kN.m**, ou seja, **23,62 kN.m** no sentido contrário ao sentido adotado para o cálculo, e **V_b** vale **35,44 kN** no mesmo sentido adotado para o cálculo.

Agora resta apenas encontrar os valores das reações **V_a** , **H_a** e **V_c** , que podem ser encontradas utilizando as três equações de equilíbrio, fazendo isso obtém-se os valores das reações restantes, sendo **V_a** igual a **22,31 kN**, **H_a** igual a **15 kN** e **V_c** igual a **5,25 kN**, como é possível observar na figura 36.

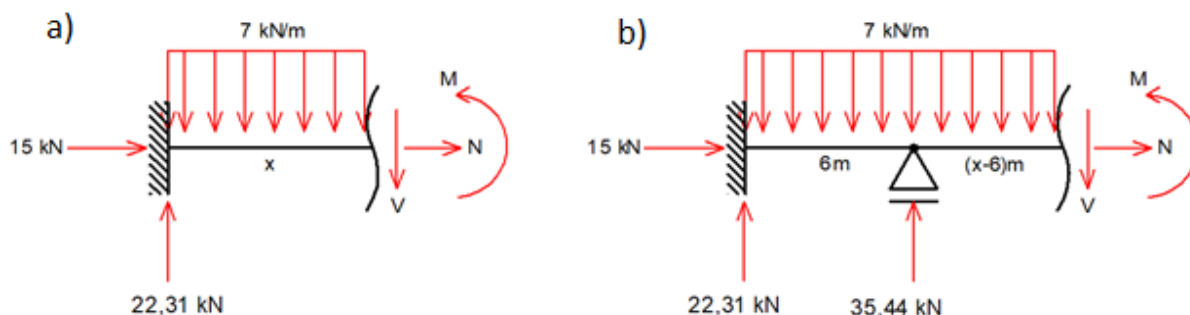
FIGURA 36 – Reações de apoio da viga hiperestática



Fonte: Elaborado pelo autor, 2023

Depois de encontradas as reações nos apoios, os esforços internos para os segmentos entre 0 e 6 metros e entre 6 e 0 metros podem ser encontrados por meio do método das seções, como mostra a figura 37, nos itens a e b respectivamente.

FIGURA 37 – Aplicação do método das seções na viga hiperestática



Fonte: Elaborado pelo autor, 2023

Finalmente, resolvendo-se as equações de equilíbrio, tem-se que, para o segmento de 0 a 6 metros:

$$M = -3,5x^2 + 22,31x - 23,62$$

$$V = -7x + 22,31$$

$$N = -15 \text{ kN}$$

E para o segmento de 6 a 9 metros:

$$M = -3,5x^2 + 57,75x - 236,26$$

$$V = -7x + 57,75$$

$$N = -15 \text{ kN}$$

Vale ressaltar que para a viga em questão, que possui grau de hiperestaticidade igual a 2, foi necessário trabalhar com três sistemas isostáticos e resolver o sistema com duas equações de compatibilidade, no entanto, conforme aumenta-se o grau de hiperestaticidade o número de sistemas isostáticos e equações de compatibilidade aumentam proporcionalmente, onde o número de sistemas isostáticos é igual ao grau de hiperestaticidade mais um, e o número de equações de compatibilidade é igual ao grau de hiperestaticidade (Luiz Fernando Martha), o que torna os cálculos de sistemas hiperestáticos com altos graus de hiperestaticidade muito mais trabalhosos.

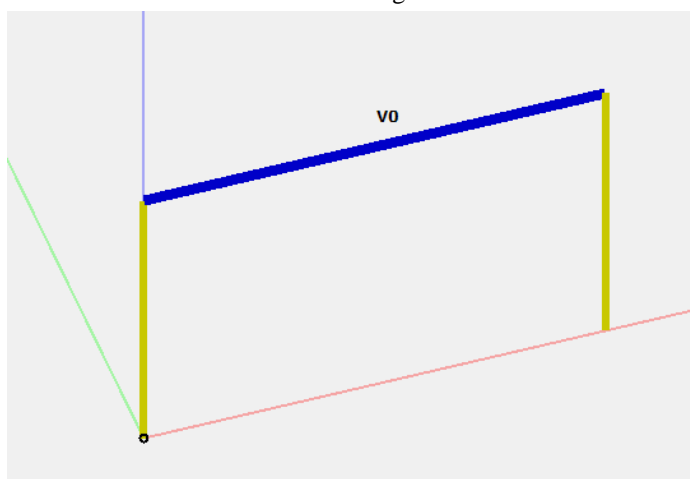
5 RESULTADOS

Neste capítulo serão apresentados os dados e análises obtidos por meio do aplicativo desenvolvido, os quais serão comparados com os resultados dos estudos de caso previamente realizados, descritos detalhadamente no capítulo 4. Esta seção permitirá a avaliação da eficácia e precisão do *software*, fornecendo uma base sólida para a análise comparativa entre os resultados obtidos com o aplicativo e as expectativas derivadas do estudo de caso.

5.1 RESULTADOS DO ESTUDO DE CASO 1

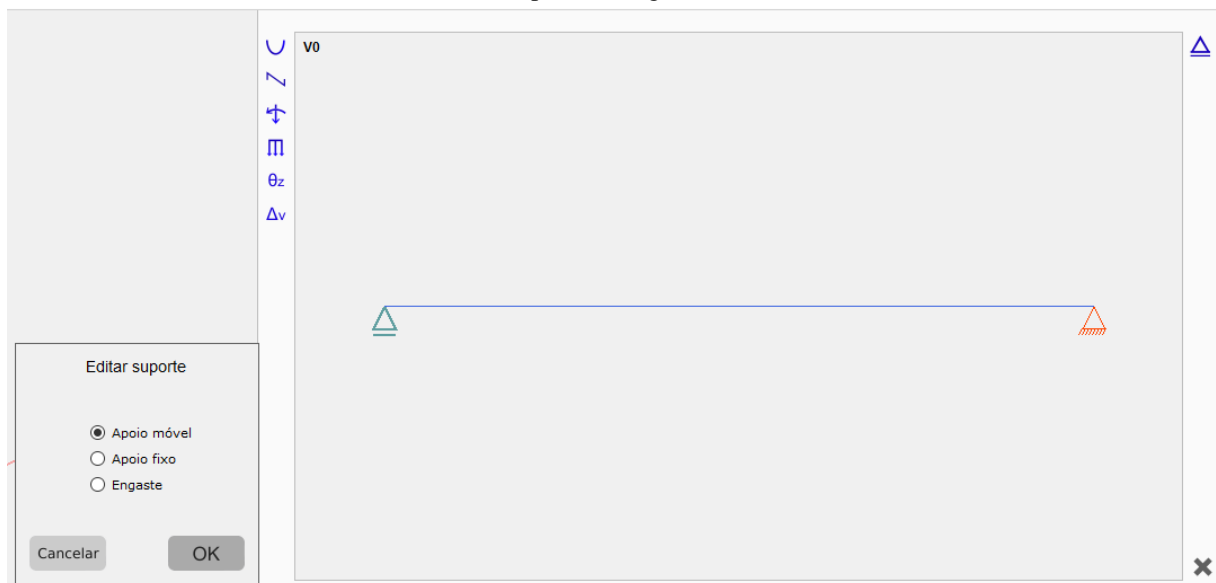
A seguir serão apresentados os resultados do estudo de caso 1, feito no capítulo 4.1, obtidos por meio do aplicativo desenvolvido.

FIGURA 38 – Modelo da viga isostática estudada



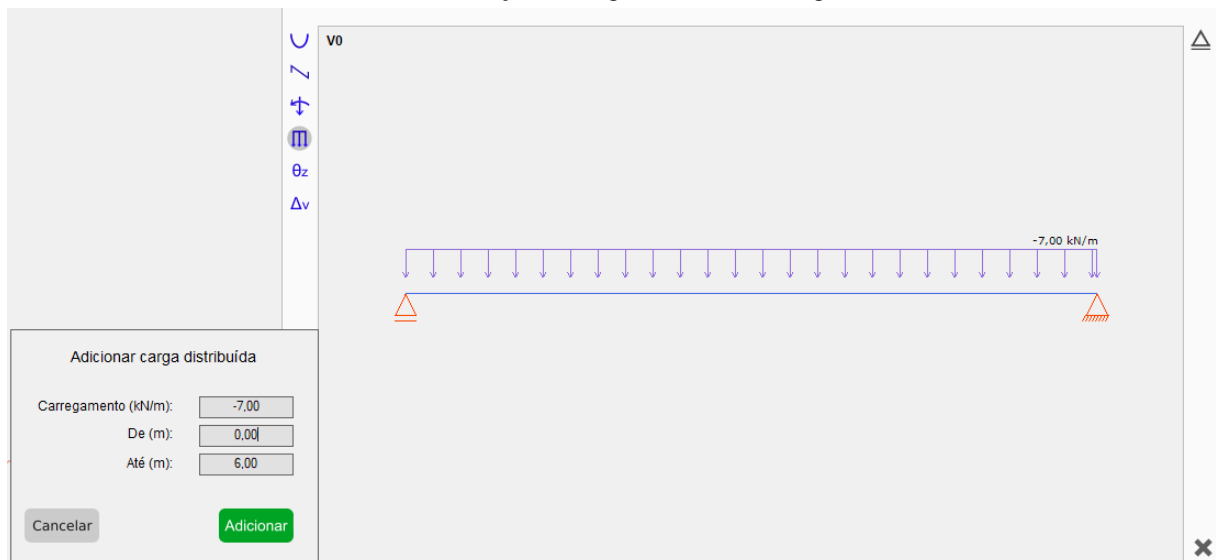
Fonte: Elaborado pelo autor, 2023

Primeiramente foi desenhada a viga em questão (em azul), desenhando também os pilares (em amarelo) que servirão de apoio para a viga, como mostra a figura acima.

FIGURA 39 – Apoios da viga isostática estudada

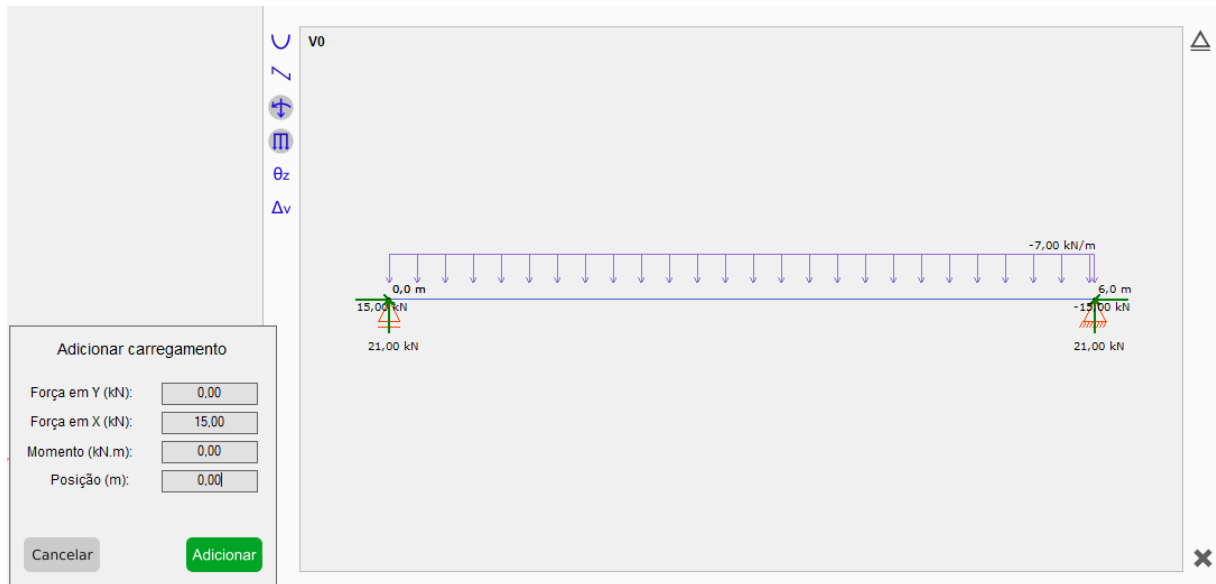
Fonte: Elaborado pelo autor, 2023

Por padrão o aplicativo detecta os apoios e os define como apoios fixos, portanto foi necessário alterar o tipo do apoio da direita como apoio móvel, como mostra a figura acima.

FIGURA 40 – Inserção da carga distribuída na viga isostática

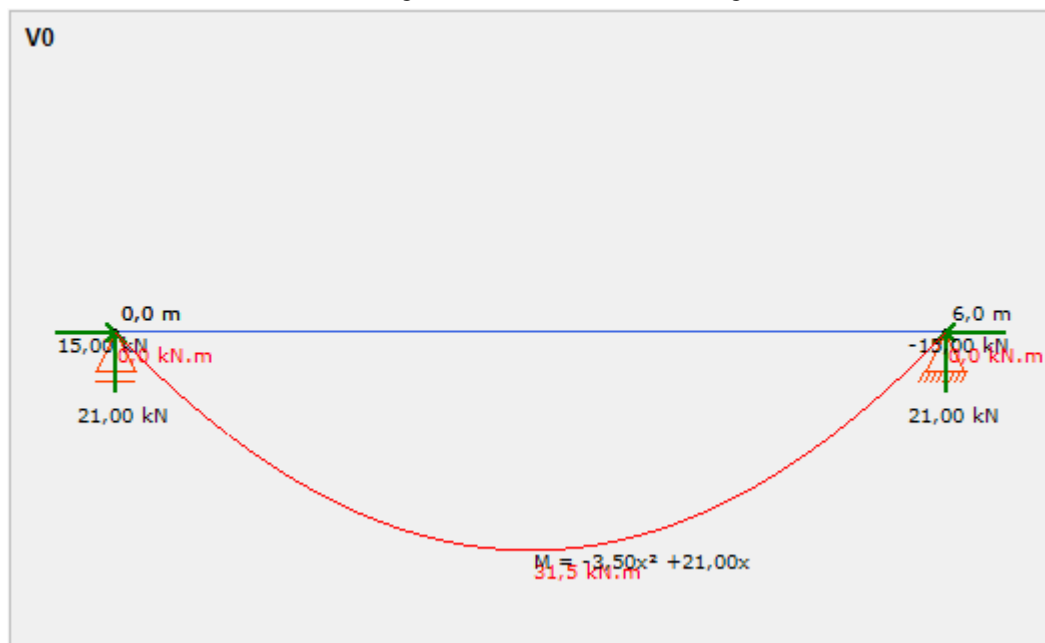
Fonte: Elaborado pelo autor, 2023

Depois disso foi adicionada a carga distribuída de 7 kN/m (figura 40).

FIGURA 41 – Inserção das forças pontuais na viga isostática

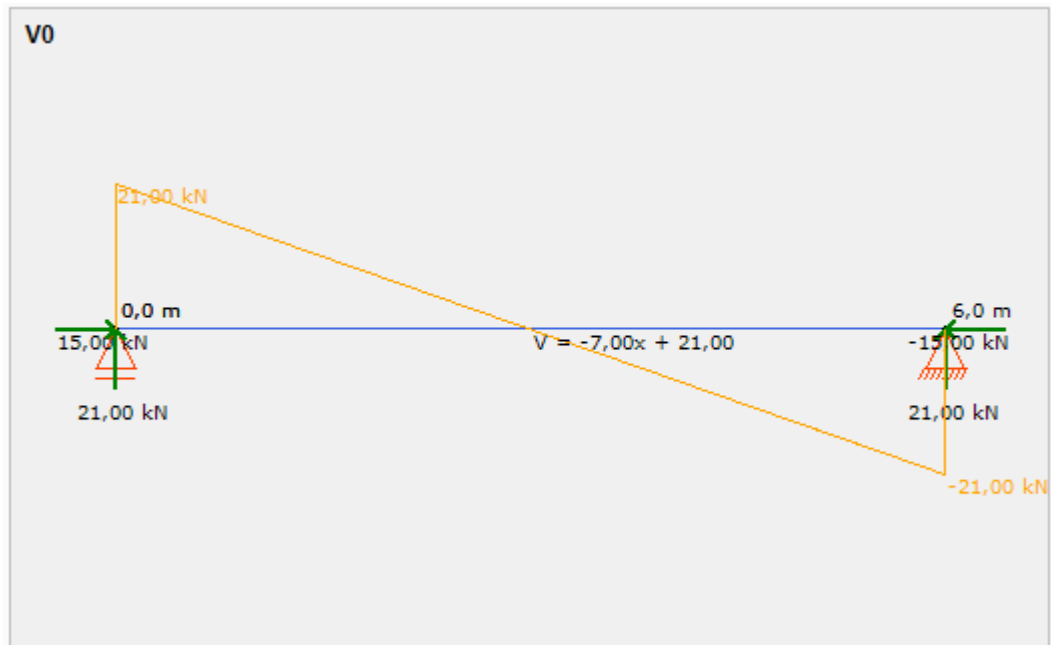
Fonte: Elaborado pelo autor, 2023

E em seguida foi adicionada a carga pontual de 15 kN. Como é possível notar na figura 41, as reações nos apoios já são calculadas instantaneamente no momento da inserção dos esforços.

FIGURA 42 – Diagrama de momento fletor da viga isostática

Fonte: Elaborado pelo autor, 2013

Como é possível ver na figura acima, tanto a equação de momento fletor, quanto as reações de apoio têm os mesmos valores obtidos no cálculo de estudo de caso.

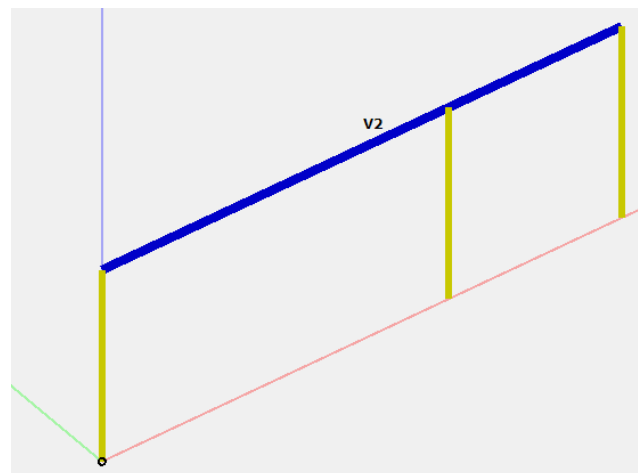
FIGURA 43 – Diagrama de força cortante da viga isostática

Fonte: Elaborado pelo autor, 2023

O mesmo ocorre para a equação de força cortante, que tem o mesmo valor que foi obtido no estudo de caso.

5.2 RESULTADOS DO ESTUDO DE CASO 2

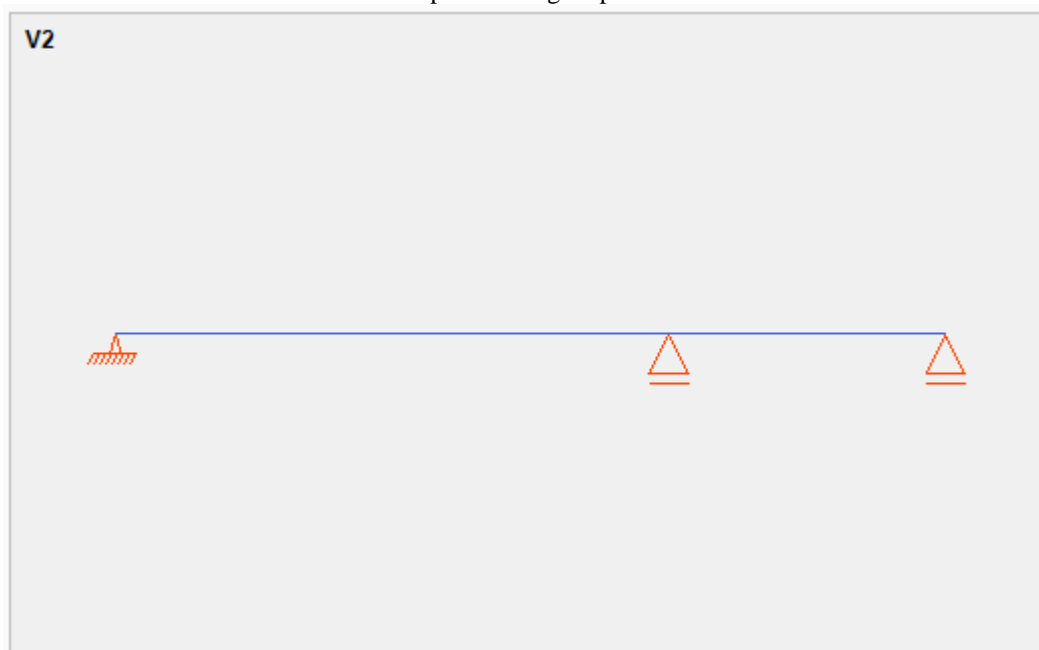
A seguir serão apresentados os resultados do estudo de caso 2, feito no capítulo 4.2, obtidos por meio do aplicativo desenvolvido.

FIGURA 44 – Modelo da viga hiperestática estudada

Fonte: Elaborado pelo autor, 2023

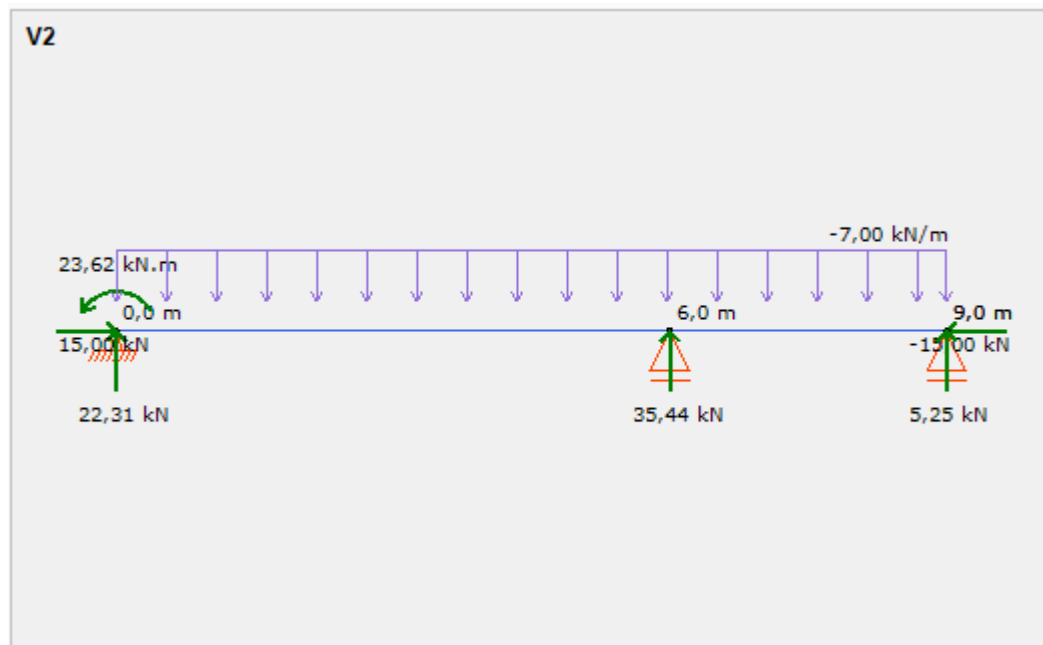
Primeiramente foi desenhada a viga em questão (em azul), desenhando também os pilares (em amarelo) que servirão de apoio para a viga, como mostra a figura 44.

FIGURA 45 – Apoios da viga hiperestática estudada



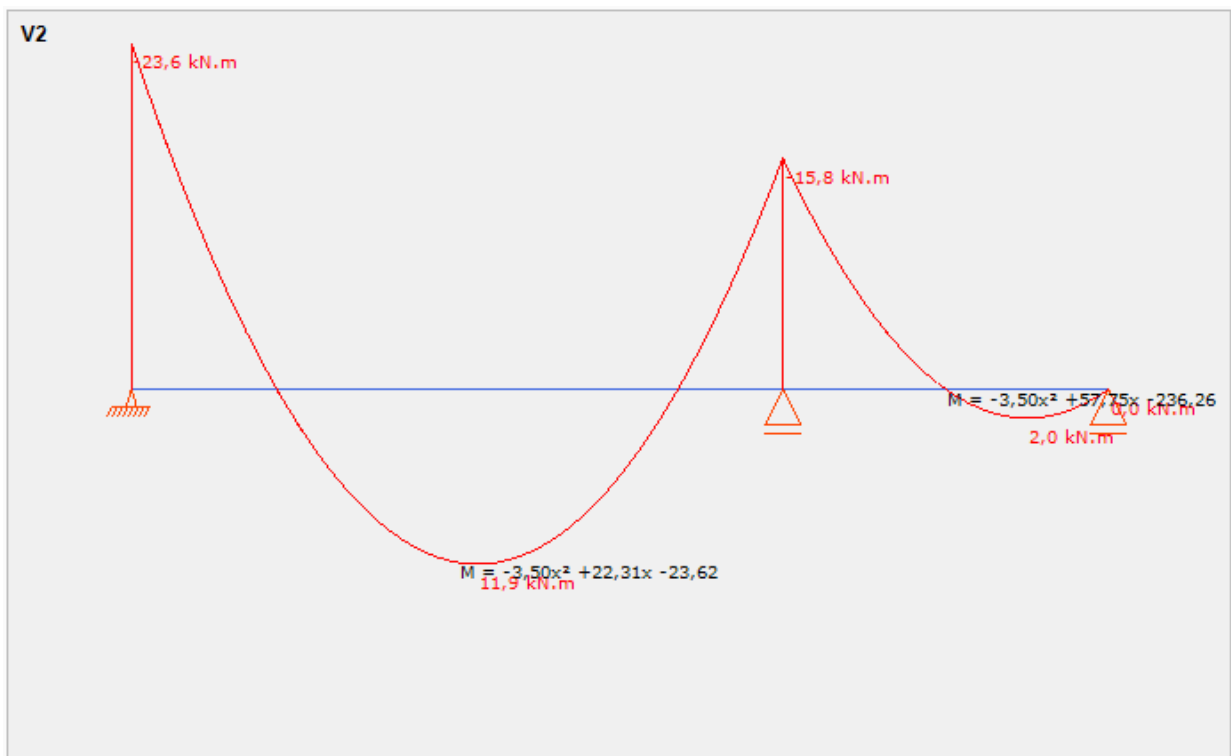
Fonte: Elaborado pelo autor, 2023

Como já mencionado, por padrão o *software* detecta os apoios como apoios fixos, portanto foram alterados os apoios para que o modelo fosse igual ao modelo estudado no estudo de caso, sendo um engaste na extremidade esquerda e os demais apoios móveis, como mostra a figura 45.

FIGURA 46 – Inserção dos carregamentos na viga hiperestática

Fonte: Elaborado pelo autor, 2023

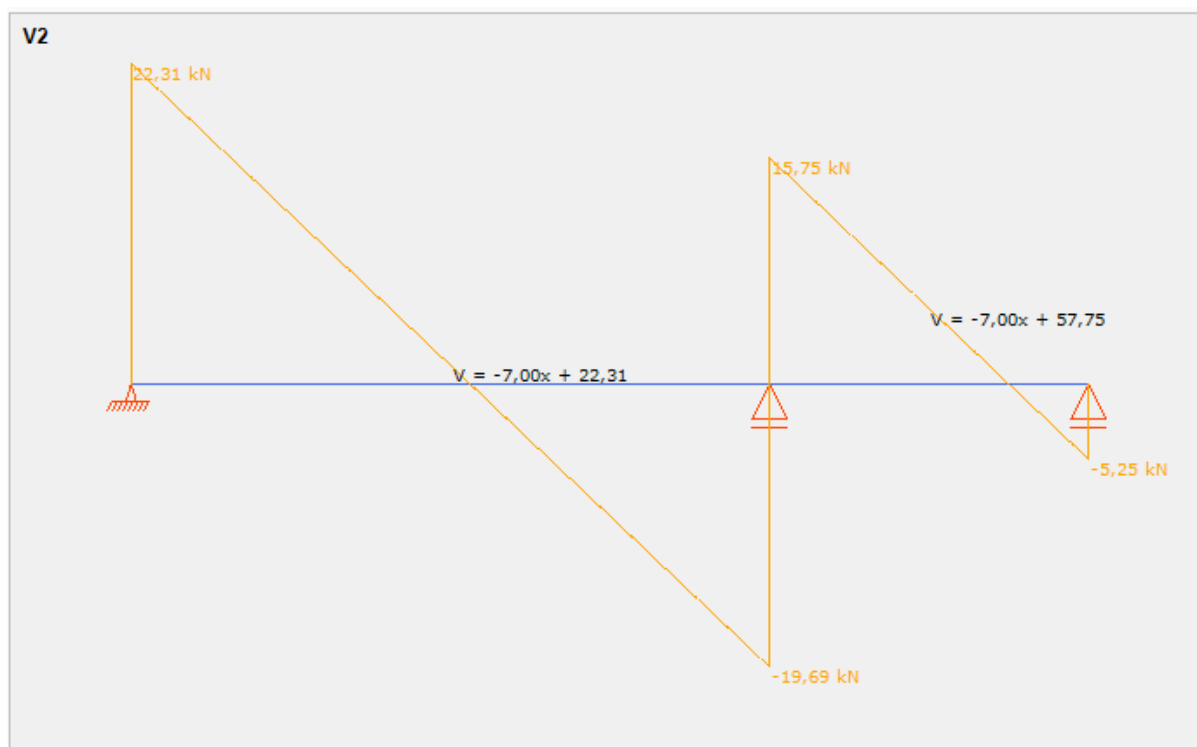
Depois disso foram inseridos os carregamentos, e como pode ser visto na figura 46 as reações dos apoios são calculadas instantaneamente no momento da inserção dos carregamentos, e têm os mesmos valores obtidos no estudo de caso.

FIGURA 47 – Equações de momento fletor da viga hiperestática

Fonte: Elaborado pelo autor, 2023

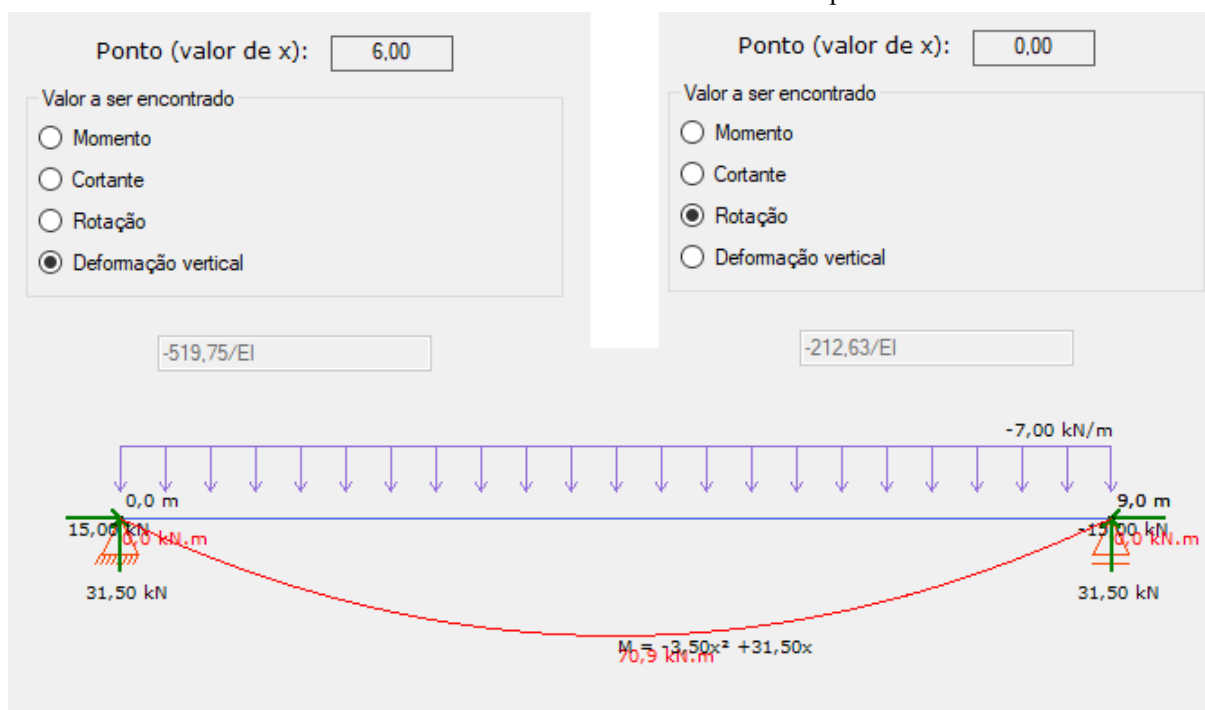
Além disso, como pode ser visto na figura 47, as equações de momento fletor também têm os mesmos valores encontrados no cálculo.

FIGURA 48 – Equações de força cortante da viga hiperestática



Fonte: Elaborado pelo autor, 2023

O mesmo ocorre com as equações de esforço cortante, que têm os mesmos valores obtidos no estudo de caso.

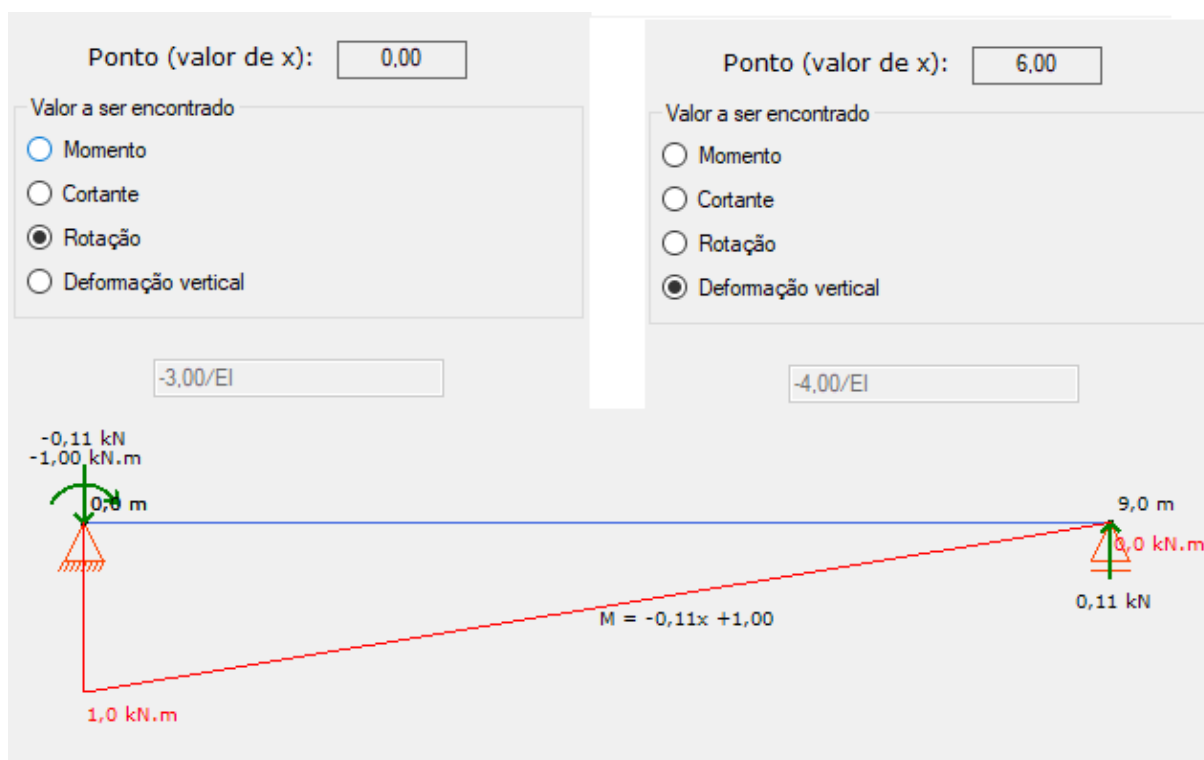
FIGURA 49 – Resultado do Sistema Principal

Fonte: Elaborado pelo autor, 2023

Além do resultado do sistema hiperestático, também é possível obter os valores calculados para os sistemas intermediários isostáticos.

Como é possível notar na figura 49, para o sistema principal, a equação de momento e valor de rotação no ponto A têm os mesmos valores encontrados no estudo de caso. Já o valor da deformação vertical no ponto B teve valor de $-519,75/EI$, enquanto no estudo de caso este valor foi de $-519,69/EI$. Esta diferença pode ter sido causada pelo fato de a quantidade de casas decimais utilizadas pelo aplicativo se diferir das aproximações feitas no cálculo manual.

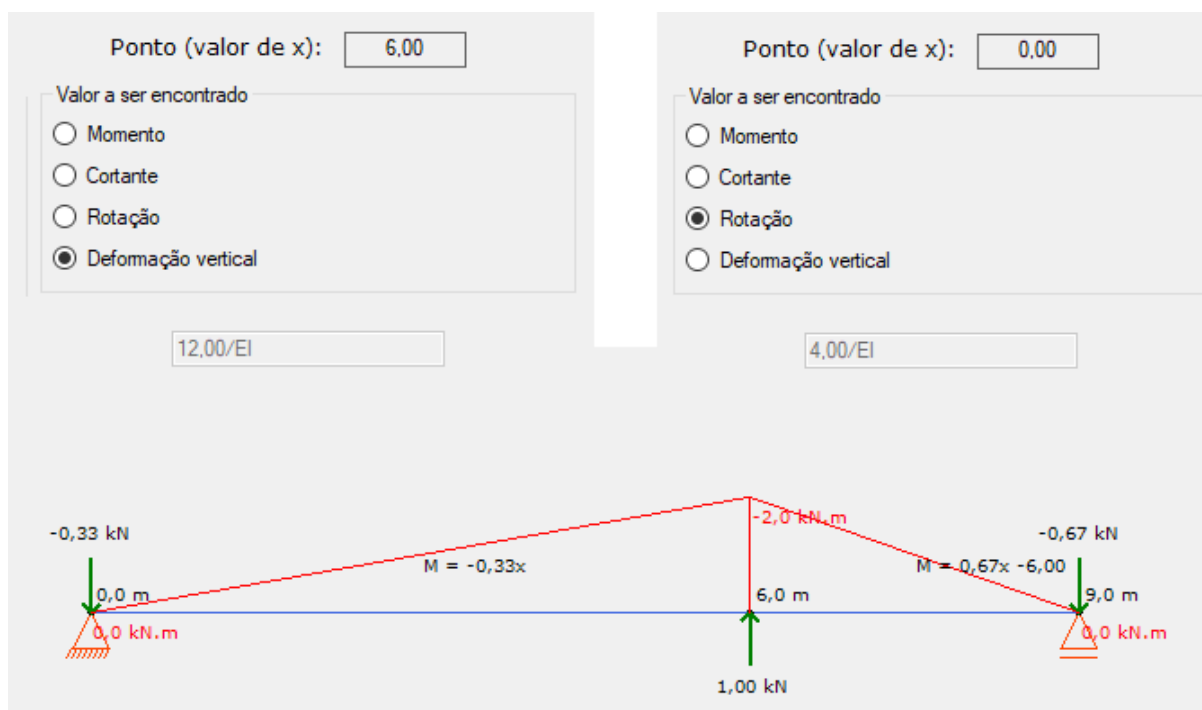
FIGURA 50 – Resultado do Sistema Unitário 1



Fonte: Elaborado pelo autor, 2023

Como é possível notar na figura 50, os resultados de equação de momento fletor e deformação no ponto B para o sistema unitário 2 são os mesmos obtidos no estudo de caso, com a exceção de que a rotação no ponto A possui sinal oposto, que se deve ao fato de que o momento unitário para sua obtenção no estudo de caso foi aplicado no sentido contrário ao da rotação que ocorre efetivamente no sistema.

FIGURA 51 – Resultado do Sistema Unitário 2



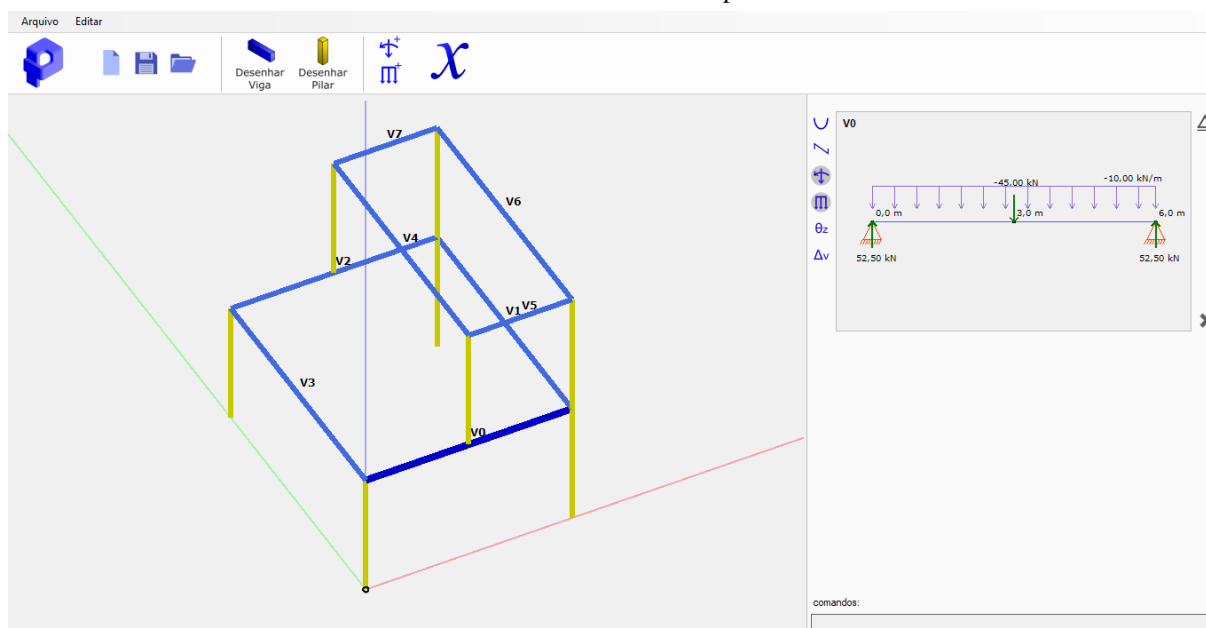
Fonte: Elaborado pelo autor, 2023

Como é possível notar na figura 51, os resultados de equação de momento fletor e deformação no ponto B para o sistema unitário 2 são os mesmos obtidos no estudo de caso. Da mesma forma que no resultado anterior, a exceção foi a rotação no ponto A, que possui sinal oposto, o que se deve ao fato de que o momento unitário para sua obtenção no estudo de caso foi aplicado no sentido contrário ao da rotação que ocorre efetivamente no sistema.

5.3 OUTROS RESULTADOS PERTINENTES

Neste capítulo, além da comparação entre os resultados do estudo de caso, serão explorados outros resultados pertinentes às funcionalidades do *software* que não foram abordados na apresentação dos estudos de caso. Esta seção oferecerá uma visão mais abrangente das capacidades do *software*, destacando aspectos adicionais que contribuem para sua utilidade e eficácia.

FIGURA 52 – Interface do aplicativo



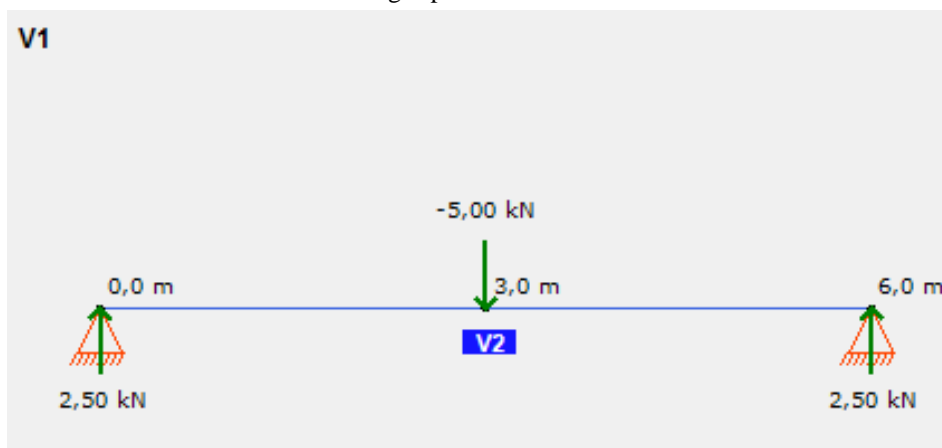
Fonte: Elaborado pelo autor, 2023

Com o propósito de demonstrar as funcionalidades da vista tridimensional, foi desenvolvido um modelo hipotético de uma estrutura composta por pilares (amarelo) e vigas (azul). A interface do aplicativo pode ser vista na figura 52.

Na parte superior esquerda estão presentes os botões de salvamento e carregamento de arquivos externos, em seguida os botões que possibilitam selecionar as estruturas a serem desenhadas, e por fim os botões que possibilitam a inserção de carregamentos e o botão que possibilita a obtenção de valores de esforços, rotações e deformações em um ponto específico da estrutura.

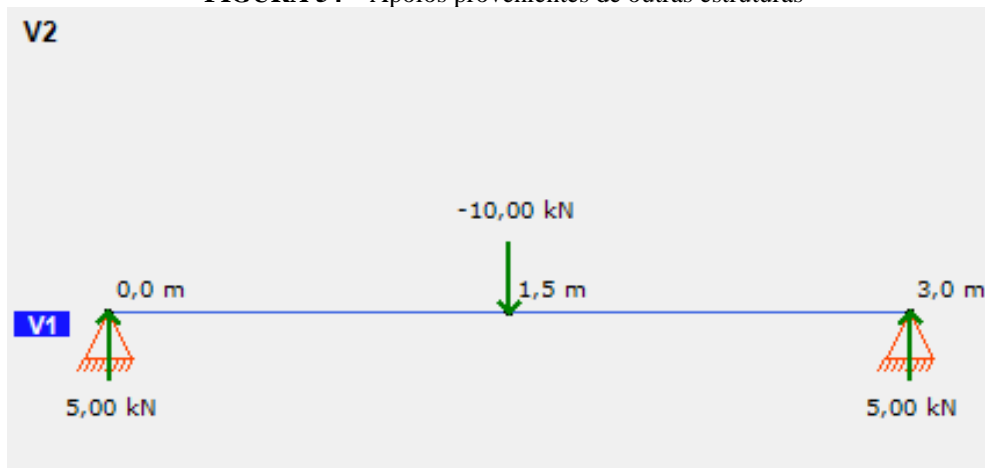
Na parte central está presente a vista tridimensional, utilizada para desenhar e manipular as estruturas. A vista pode ser transladada, rotacionada e aproximada com movimentos do *mouse*.

No canto superior esquerdo está localizado um campo onde é possível visualizar o diagrama de corpo livre da estrutura selecionada. No lado esquerdo do campo situam-se os botões que permitem a visualização dos diagramas de esforços internos, forças e cargas distribuídas, bem como os diagramas de rotação e deformação. Do lado direito está localizado o botão que permite alterar o tipo dos apoios e o botão que permite a exclusão de esforços do diagrama de corpo livre. Também é possível interagir com o campo e selecionar elementos, como carregamentos, apoios e relações com outras estruturas.

FIGURA 53 – Cargas provenientes de outras estruturas

Fonte: Elaborado pelo autor, 2023

No diagrama de corpo livre também são mostradas as relações da estrutura em questão com outras estruturas, como mostra a figura 53, que mostra uma viga genérica chamada de “V1”, onde aparece um retângulo azul com a legenda “V2”, indicando que a carga vertical de 5kN provém da viga “V2”. Esta relação pode ser alterada clicando sobre a legenda e mudando alterando a relação entre estrutura apoiada e estrutura de apoio.

FIGURA 54 – Apoios provenientes de outras estruturas

Fonte: Elaborado pelo autor, 2023

O mesmo ocorre quando a estrutura em questão é apoiada por outra do modelo, como mostra a figura 54, que retrata uma viga genérica chamada de “V2”, onde é mostrado um retângulo azul com a legenda “V1”, que indica que a viga em questão é apoiada pela viga “V2”. Da mesma forma que no exemplo anterior, essa relação pode ser alterada ao clicar sobre o retângulo e selecionar o tipo de relação entre as estruturas.

6 CONCLUSÃO

Este estudo representa um avanço significativo na área da engenharia civil, apresentando uma solução inovadora e acessível para o estudo de estruturas isostáticas e hiperestáticas. O *software* desenvolvido destaca-se como uma ferramenta crucial para estudantes e profissionais em formação, complementando lacunas educacionais na compreensão de conceitos complexos.

A revisão abrangente da literatura solidificou a base teórica, evidenciando uma abordagem detalhada aos conceitos de análise estrutural e resistência dos materiais. Essa fundamentação teórica é refletida na qualidade das implementações do *software*, tornando-o não apenas uma ferramenta prática, mas também uma contribuição valiosa para o conhecimento na área.

A interface do *software* foi projetada com foco na simplicidade e acessibilidade. Seu design permite uma interação descomplicada, facilitando o uso por parte dos usuários. Durante os testes, a eficácia dessa abordagem se confirmou, tornando evidente a facilidade com que o *software* pode ser utilizado de maneira proveitosa. A estrutura de interação foi pensada para proporcionar uma experiência intuitiva, visando otimizar a eficiência durante a utilização.

A arquitetura funcional do *software* revela uma abordagem criteriosa na concepção e implementação de cada funcionalidade. Cada etapa do processo de desenvolvimento foi meticulosamente planejada e executada, visando proporcionar uma experiência de usuário mais fluida e intuitiva. Esse feito destaca a capacidade do projeto de integrar conceitos teóricos de maneira coesa e prática, bem como abre portas para aplicações futuras no campo da análise estrutural.

Os testes, conduzidos com rigor e respaldados por estudos de caso, comprovaram a exatidão do *software*. Esses procedimentos não apenas evidenciaram sua eficácia na aplicação prática, mas também fortaleceram a confiança e credibilidade do sistema, destacando sua robustez e consistência em diversas situações de análise estrutural.

Durante este estudo, ficou evidente a intrincada complexidade associada aos cálculos manuais necessários para análises estruturais, especialmente em estruturas hiperestáticas. Essa complexidade não apenas desafia o processo de aprendizado, mas também coloca obstáculos à aplicação prática desses cálculos em projetos reais.

A introdução do *software* desenvolvido surge como uma resposta tangível a essas complexidades. A automação dos cálculos não apenas aprimora a eficiência, mas também reduz a propensão a erros inerente aos cálculos manuais. A facilidade de uso do *software*, em conjunto

com sua capacidade de lidar com estruturas isostáticas e hiperestáticas, oferece uma solução concreta para superar os desafios tradicionais no aprendizado e na aplicação prática.

Além de ser uma ferramenta de aprendizado, o *software* demonstra claramente sua utilidade em análises pontuais de projetos estruturais. A possibilidade de realizar análises detalhadas em estruturas individuais destaca-se como uma extensão valiosa de sua aplicabilidade. Essa capacidade de análise focalizada pode ser particularmente benéfica para profissionais e estudantes envolvidos em projetos específicos que demandam precisão na análise estrutural.

Em suma, este trabalho não só cumpre os objetivos estabelecidos, mas também oferece uma contribuição substancial ao ensino e à compreensão das complexidades envolvidas nas análises estruturais. O *software* não é apenas uma ferramenta educacional, é um marco no avanço do conhecimento na engenharia civil.

REFERÊNCIAS

AUTODESK. **Robot Structural Analysis Professional: BIM structural analysis software for engineers**. Disponível em: <<https://www.autodesk.com/products/robot-structural-analysis/overview>>. Acesso em: jun. 2023.

BOTELHO, Manoel H. C. **Resistência dos Materiais para entender e gostar: Um texto curricular**. São Paulo: Studio Nobel, 1998.

COMPUTERS & STRUCTURES, INC. **SAP2000: Structural Analysis and Design**. Disponível em: <<https://www.csiamerica.com/products/sap2000>>. Acesso em: jun. 2023.

DE TELECOMUNICAÇÕES, Curso de Engenharia; GRUPO, P. E. T. **Tutorial de Programação Orientada a Objeto**. 2009.

FEODOSIEV, Vsevolod Ivanovich. **Resistência dos Materiais**. Porto: Lopes da Silva, 1977.

FTOOL. **A Graphical-Interactive Program for Teaching Structural Behavior**. Disponível em: <<https://www.ftool.com.br/Ftool/>>. Acesso em: jun. 2023.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. 1ª edição. Porto Alegre: Editora da UFRGS, 2009. 120 p.

HIBBELER, Russell C. **Análise de Estruturas**. 8ª edição. São Paulo: Pearson Education do Brasil, 2013.

HIBBELER, Russell C. **Resistência dos Materiais**. 5ª edição. São Paulo: Pearson Prentice Hall, 2004.

LEITE, Thiago et al. **Orientação a objetos: aprenda seus conceitos e suas aplicabilidades de forma efetiva**. Editora Casa do Código, 2016.

LIMA, Elon Lages. **Geometria Analítica e Álgebra Linear**. Rio de Janeiro: IMPA, 2014.

MARTHA, Luiz Fernando. **Análise de Estruturas: Conceitos e Métodos Básicos**. Rio de Janeiro: Elsevier, 2010.

MARTHA, Luiz Fernando. **FTOOL—Um programa gráfico-interativo para ensino de comportamento de estruturas**. Rio de Janeiro: PUC-Rio, 2002.

MELCONIAN, Sarkis. **Mecânica técnica e resistência dos materiais**. 17ª edição. Érica, 1999.

MICROSOFT. **Um tour pela linguagem C#**. Disponível em: <<https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>>. Acesso em: jul. 2023.

MICROSOFT. **Visual Studio Community**. Disponível em: <<https://visualstudio.microsoft.com/pt-br/vs/community/>>. Acesso em: jul. 2023.

MIRANDA, Luiz Eduardo. **Resistência dos Materiais: Flexão Simples**. Disponível em: <<https://ebin.pub/resistencia-dos-materiais.html>>. Acesso em: mai. 2023.

MULTIPLUS. SAP2000: **Software para Análise Estrutural e Dinâmica, Linear e Não-Linear por Elementos Finitos**. Disponível em: <<https://multiplus.com/software/sap2000/index.html>>. Acesso em: jun. 2023.

RICARTE, Ivan Luiz Marques. **Programação orientada a objetos: uma abordagem com JAVA**. 2001. Departamento de Engenharia de Computação e Automação Industrial-Faculdade de Engenharia Elétrica e de Computação Universidade Estadual de Campinas. Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>>. Acesso em: jul. 2023.

SKEET, Jonathan. **C# in Depth**. Simon and Schuster, 2019.