



Ronald Junior Lemos Assunção

**Uma estrutura de classes orientada a objetos
para verificação de projeto de perfis metálicos
em MATLAB**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Civil do Departamento de Engenharia Civil e Ambiental da PUC-Rio.

Orientador: Luiz Fernando Martha

Rio de Janeiro
Agosto de 2021



Ronald Junior Lemos Assunção

**Uma estrutura de classes orientada a objetos
para verificação de projeto de perfis metálicos
em MATLAB**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Civil do Departamento de Engenharia Civil e Ambiental da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

Luiz Fernando Martha

Orientador

Departamento de Engenharia Civil e Ambiental – PUC-Rio

Daniel Carlos Taissum Cardoso

Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Luciano Rodrigues Ornelas de Lima

Universidade Estadual do Rio de Janeiro – UERJ

Rio de Janeiro, 20 de Agosto de 2021

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Ronald Junior Lemos Assunção

Graduou-se em Engenharia Civil pela Universidade Veiga de Almeida em 2018. Em 2018, deu início ao mestrado em estruturas no Departamento de Engenharia civil e ambiental da PUC-Rio.

Ficha Catalográfica

Assunção, Ronald Junior Lemos

Uma estrutura de classes orientada a objetos para verificação de projeto de perfis metálicos em MATLAB / Ronald Junior Lemos Assunção; orientador: Luiz Fernando Martha. – Rio de Janeiro: PUC-Rio, Departamento de Engenharia Civil e Ambiental , 2021.

v., 126 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Civil e Ambiental .

Inclui bibliografia

1. Engenharia civil – Teses. 2. Estruturas – Teses. 3. Programação orientada a objetos;. 4. Pórticos planos;. 5. MATLAB;. 6. Dimensionamento de estruturas de aço;. I. Martha, L. F.. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Civil e Ambiental . III. Título.

Agradecimentos

A Deus por iluminar o caminho.

Aos meus parentes, Elaine e Carlos, por acreditarem no meu sonho desde o início e me proporcionarem todo o suporte necessário para concluir minha jornada. Vocês são minha motivação para me aprimorar mais e mais.

A minha namorada Maria Fernanda, por me incentivar, pelo carinho e por todas as palavras de conforto que me acalmaram em momentos difíceis.

Ao meu amigo, Christian Leite Dias, por todo o suporte oferecido no decorrer desse trabalho, assim como pela parceria nesses anos de mestrado.

Ao meu orientador, Luiz Fernando, por acreditar no meu potencial e por me auxiliar de maneira muito atenciosa por todo o período. Agradeço, principalmente, por toda a compreensão, visto o momento conturbado.

A todos os professores que me auxiliaram nessa jornada, pelas correções, ensinamentos, conselhos e paciência ao longo do curso.

A PUC-Rio e todos os seus funcionários, essenciais em minha formação, por me proporcionarem por todos esses anos um ambiente propício de muito aprendizado e trocas.

Agradeço a todos os médicos que me amparam ao longo dos anos que proveram tratamentos para tornar minha vida mais produtiva e agradável.

Resumo

Assunção, Ronald Junior Lemos; Martha, L. F.. **Uma estrutura de classes orientada a objetos para verificação de projeto de perfis metálicos em MATLAB**. Rio de Janeiro, 2021. 126p. Dissertação de Mestrado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

Esta dissertação descreve a extensão de uma ferramenta educacional (LESM – Linear Elements Structure Model), desenvolvida na PUC-Rio e implementada no ambiente do MATLAB, para dimensionamento de pórticos metálicos bidimensionais seguindo os preceitos da norma brasileira NBR8800:2008. Todas as etapas de desenvolvimento são descritas detalhadamente, trazendo também referências à norma utilizada, visto que a funcionalidade deve atender rigorosamente a seus preceitos. As etapas adotadas são: levantamento de requisitos da funcionalidade, levantamento de requisitos do 'software', análise, projeto do programa, codificação, teste e operação. Vale ressaltar que todos os desdobramentos das etapas são baseados na linguagem orientada a objetos MATLAB, na qual o *software* LESM foi desenvolvido. As etapas de desenvolvimento são elucidadas através de esquemas UML (Linguagem de Modelagem Unificada) para que haja maior entendimento dos relacionamentos organizacionais e processuais utilizados. Dessa maneira podem ser traçados comparativos das novas estruturas de dados implementadas com as anteriormente disponíveis na documentação do LESM. Além disso, apesar da linguagem predominante utilizada no desenvolvimento ser a linguagem do MATLAB, há menções relacionais à JavaScript, utilizado para elaborar alguns recursos da nova funcionalidade. Por fim, após esclarecer todas as nuances do processo de desenvolvimento, são modelados exemplos práticos utilizando o LESM com o intuito de demonstrar a nova funcionalidade adicionada e a acurácia de seus resultados.

Palavras-chave

Programação orientada a objetos; Pórticos planos; MATLAB; Dimensionamento de estruturas de aço;

Abstract

Assunção, Ronald Junior Lemos; Martha, L. F. (Advisor). **An object-oriented class structure for design verification of metallic profiles in MATLAB**. Rio de Janeiro, 2021. 126p. Dissertação de mestrado – Department of Civil and Environmental Engineering, Pontifícia Universidade Católica do Rio de Janeiro.

This document describes the extension of an educational tool (LESM – Linear Elements Structure Model), developed on PUC-Rio and implemented in the MATLAB environment for metallic dimensioning of 2D frames, following the precepts of the Brazilian dimensioning standard, NBR8800:2008. The entire development steps are described in details, along with references of the standard used, since the new feature must strictly attend its precepts. The order of the adopted steps is: feature requirements gathering, software requirements gathering, analysis, program project, coding, test and operation. It is worth mentioning that all of the steps are based on the objects oriented language MATLAB, which the software LESM was developed. The previously mentioned phases are elucidated through UMLs schemes for a better understanding of the organizational relationships and processes used. Thus, comparatives of the implemented new data structure with the previously available in LESM documentation, disposed in UML diagrams, can be traced. Furthermore, despite the predominant language used in the development is MATLAB, will be also mentions to JavaScript, used to elaborate some resources of the new feature. Finally, after clarifying all aspects of the development process, practical examples using LESM are modeled to demonstrate the new feature along with the results accuracy.

Keywords

Object oriented programming; Planar frame structures; MATLAB; Design of steel structures;

Sumário

1	Introdução	15
1.1	Motivação	16
1.2	Objetivos	17
1.3	Escopo	18
1.4	Organização do trabalho	18
2	Referencial teórico	20
2.1	Programação orientada a objetos	20
2.1.1	Surgimento	20
2.1.2	Conceitos	21
2.1.2.1	Objetos	21
2.1.2.2	Classe	22
2.1.2.3	Herança	22
2.1.2.4	Polimorfismo	23
2.2	UML (Linguagem de Modelagem Unificada)	24
2.2.1	Propósito da modelagem de <i>softwares</i>	24
2.2.1.1	Diagramas de casos de uso	25
2.2.1.2	Diagramas de classes	26
2.2.1.3	Diagrama de objetos	26
2.2.1.4	Diagrama de sequência	27
2.3	MATLAB	28
2.4	LESM (<i>Linear Elements Structure Model</i>)	28
2.4.1	Definição	28
2.4.2	Implementação do método da rigidez direta	29
2.4.3	Arquitetura orientada a objeto	32
2.4.3.1	Divisão de Classes	32
2.5	Diretrizes de dimensionamento metálico segundo a NBR8800:2008	38
2.5.1	Critérios de garantia de segurança	39
2.5.1.1	Método dos estados limites	39
2.5.2	Combinação de ações últimas	40
2.5.3	Dimensionamento de perfil I	43
2.5.3.1	Tração	43
2.5.3.2	Flexão	44
2.5.3.3	Compressão	47
2.5.3.4	Cisalhamento	51
2.5.3.5	Flexo Compressão e Flexo Tração	52
3	Metodologia	53
3.1	Objetivos de Implementação	54
3.2	Levantamento de requisitos do software	55
3.3	Análise	57
3.3.1	Modificação na estrutura de dados	57
3.3.1.1	Drv	57
3.3.1.2	Materiais	58

3.3.1.3	Seções	59
3.3.1.4	Elementos de barra	60
3.3.1.5	Rotinas de cálculo para NBR8800:2008	62
3.3.1.6	Memorial de cálculo	62
3.3.2	Elaboração das GUI	64
3.3.2.1	GUI para materiais	64
3.3.2.2	GUI para Tipos de seções	64
3.3.2.3	GUI para combinação de ações	66
3.3.2.4	GUI para criação de grupos	67
3.3.3	Diagramas resultantes da Análise	69
3.4	Projeto do programa	70
3.4.1	Rotinas computacionais de dimensionamento	71
3.4.1.1	Fluxograma da rotina de cálculo para força resistente ao cisalhamento	71
3.4.1.2	Flexão	72
3.4.1.3	Fluxograma da rotina de cálculo para força resistente de compressão	75
3.4.2	Implementação da nova estrutura de dados	77
3.4.2.1	Implementação da nova Section	77
3.4.2.2	Implementação da nova Materials	79
3.4.2.3	Implementação da nova definição dos casos de carga	81
3.4.2.4	Implementação da nova estrutura para o Drv	81
3.4.2.5	Implementação da nova estrutura de grupos	82
3.4.2.6	Implementação da classe de dimensionamento	84
3.4.2.7	Implementação da classe de geração de memorial de cálculo	86
3.4.2.8	Nova arquitetura de classes e fluxo de ações	87
3.5	Codificação	90
4	Resultados	91
4.1	Modelagem pórtico 2D LESM	91
4.2	Modelagem pórtico 2D CypeCad	97
4.3	Análise numérica dos resultados obtidos pelos <i>softwares</i>	97
4.3.1	Análise dos valores de tração	97
4.3.2	Análise dos valores de flexão	99
4.3.3	Análise dos valores de compressão	105
4.3.4	Análise dos valores de cisalhamento	108
4.3.5	Análise dos valores de flexo-tração e flexo-compressão	111
4.4	Exibição de resultados para o usuário	114
4.4.1	Memorial de cálculo	114
4.4.2	GUI de taxas de utilização	119
5	Conclusão	122
5.1	Sugestões para trabalhos futuros	123
	Referências bibliográficas	125

Lista de figuras

Figura 2.1	Representação para um objeto. Fonte: Kamienski (1996)	21
Figura 2.2	Relacionamentos entre classes e objetos. Fonte: Kamienski (1996)	22
Figura 2.3	Esquemática do conceito de Herança. Fonte: Autor (2021)	23
Figura 2.4	Esquemática do conceito de Polimorfismo. Fonte: Autor (2021)	24
Figura 2.5	Exemplo de diagrama de caso de uso. Fonte: Guedes (2018)	25
Figura 2.6	Exemplo de diagrama de classes. Fonte: Guedes (2018)	26
Figura 2.7	Exemplo de diagrama de objetos. Fonte: Guedes (2018)	27
Figura 2.8	Exemplo de diagrama de sequência. Fonte: Guedes (2018)	27
Figura 2.9	Diagrama de atividades da versão não gráfica do programa LESM para solução do problema global discreto pelo método da rigidez direta. Fonte: Martha, (2018)	30
Figura 2.10	Métodos e propriedades da classe Drv. Fonte: Martha, (2018)	33
Figura 2.11	Métodos e propriedades da classe Material. Fonte: Martha, (2018)	34
Figura 2.12	Métodos e propriedades da classe Section. Fonte: Martha, (2018)	34
Figura 2.13	Métodos e propriedades da classe Node. Fonte: Martha, (2018)	35
Figura 2.14	Métodos e propriedades da classe Elem. Fonte: Martha, (2018)	36
Figura 2.15	Diagrama de classes do programa LESM (versão não gráfica) para implementar uma análise linear-elástica de estruturas reticuladas. Fonte: Martha, (2018)	38
Figura 2.16	Fatores condicionais de contorno. Fonte: ABNT, (2008)	48
Figura 2.17	Esbeltez limite para os elementos AL. Fonte: ABNT, (2008)	49
Figura 2.18	Esbeltez limite para os elementos AA. Fonte: ABNT, (2008)	50
Figura 3.1	Interação da metodologia do modelo de cascata (imagem adaptada). Fonte: Wazlawick (2013)	53
Figura 3.2	GUI de propriedades geométricas da seção. Fonte: Autor (2021)	56
Figura 3.3	GUI de propriedades dos materiais. Fonte: Autor (2021)	56
Figura 3.4	Elemento delimitado por dois nós. Fonte: Autor (2021)	60
Figura 3.5	Elemento redividido após inserção de um nó. Fonte: Autor (2021)	61
Figura 3.6	Demonstração do funcionamento da nova classe para grupos que caracterizem vigas e colunas. Fonte: Autor (2021)	61
Figura 3.7	Exportações dos valores dos esforços para o usuário fornecido pelo LESM. Fonte: Autor (2021)	63

Figura 3.8	GUI com a lista de materiais. Fonte: Autor (2021)	64
Figura 3.9	GUI para tipos de seções. Fonte: Autor (2021)	65
Figura 3.10	GUI de Seções com modo de consulta ativado. Fonte: Autor (2021)	65
Figura 3.11	GUI para combinação de ações. Fonte: Autor (2021)	66
Figura 3.12	GUI para combinação de ações com o modo de ajuda ativado para cargas permanentes. Fonte: Autor (2021)	67
Figura 3.13	GUI para combinação de ações com o modo de ajuda ativado para cargas variáveis. Fonte: Autor (2021)	67
Figura 3.14	GUI de criação de grupos de elementos de barras. Fonte: Autor (2021)	68
Figura 3.15	Digrama de casos de uso para concepção da nova funcionalidade. Fonte: Autor (2021)	69
Figura 3.16	[Digrama de sequência para concepção da nova funcionalidade. Fonte: Autor (2021)	70
Figura 3.17	[Fluxograma para a rotina de cálculo para força resistente ao cisalhamento. Fonte: Autor (2021)	72
Figura 3.18	[Fluxograma da rotina de cálculo do momento resistente para FLT. Fonte: Autor (2021)	73
Figura 3.19	[Fluxograma da rotina de cálculo do momento resistente para FLM. Fonte: Autor (2021)	74
Figura 3.20	[Fluxograma da rotina de cálculo do momento resistente para FLA. Fonte: Autor (2021)	75
Figura 3.21	[Fluxograma da rotina de cálculo para força resistente de compressão. Fonte: Autor (2021)	76
Figura 3.22	[Diagrama de objeto W_Beam. Fonte: Autor (2021)	79
Figura 3.23	[Diagrama de objeto Steel. Fonte: Autor (2021)	80
Figura 3.24	[Fluxo de funcionamento da classe NBR8800_Report. Fonte: Autor (2021)	86
Figura 3.25	[Construção do memorial de cálculo através dos métodos da classe NBR8800_Report. Fonte: Autor (2021)	87
Figura 3.26	[Diagrama de classes da nova estrutura do LESM. Fonte: Autor (2021)	88
Figura 3.27	[Fluxograma de sequência na nova estrutura do LESM. Fonte: Autor (2021)	89
Figura 4.1	Modelagem das barras do pórtico exemplo 2D. Fonte: Autor (2021)	92
Figura 4.2	Caso de carga PP modelado no LESM. Fonte: Autor (2021)	94
Figura 4.3	Caso de carga IMP modelado no LESM. Fonte: Autor (2021)	94
Figura 4.4	Caso de carga ARC modelado no LESM. Fonte: Autor (2021)	95
Figura 4.5	Caso de carga SOS modelado no LESM. Fonte: Autor (2021)	96
Figura 4.6	Caso de carga V1 modelado no LESM. Fonte: Autor (2021)	96
Figura 4.7	Trecho do memorial de cálculo do esforço resistente de tração para o elemento 16. Fonte: Autor (2021)	115

Figura 4.8	Trecho do memorial de cálculo do esforço resistente de flexão (FLM) para o elemento 16. Fonte: Autor (2021)	115
Figura 4.9	Trecho do memorial de cálculo do esforço resistente de flexão (FLA) para o elemento 16. Fonte: Autor (2021)	116
Figura 4.10	Trecho do memorial de cálculo do esforço resistente de flexão (FLT) para o elemento 16. Fonte: Autor (2021)	116
Figura 4.11	Trecho do memorial de cálculo do valor de flambagem elástica para o elemento 16. Fonte: Autor (2021)	116
Figura 4.12	Trecho do memorial de cálculo do fator Q_a para o elemento 16. Fonte: Autor (2021)	117
Figura 4.13	Trecho do memorial de cálculo do fator Q_s para o elemento 16. Fonte: Autor (2021)	117
Figura 4.14	Trecho do memorial de cálculo do fator χ para o elemento 16. Fonte: Autor (2021)	118
Figura 4.15	Trecho do memorial de cálculo do valor resistente à compressão para o elemento 16. Fonte: Autor (2021)	118
Figura 4.16	Trecho do memorial de cálculo do valor resistente à cisalhamento para o elemento 16. Fonte: Autor (2021)	119
Figura 4.17	GUI da taxa de utilização do esforço de tração para o elemento 16. Fonte: Autor (2021)	120
Figura 4.18	GUI da taxa de utilização do esforço de flexão para o estado limite FLM do elemento 16. Fonte: Autor (2021)	120
Figura 4.19	GUI da taxa de utilização do esforço de cisalhamento para o elemento 16. Fonte: Autor (2021)	121
Figura 5.1	Tabela de Pefis Guerdau	124

Lista de tabelas

Tabela 2.1	Coefficientes de ponderação para ações permanentes e variáveis	41
Tabela 2.2	Coefficientes de ponderação estatístico de acordo com a ação variável	42
Tabela 3.1	Propriedades físicas dos aços disponíveis na nova versão do LESM	59
Tabela 3.2	Propriedades físicas dos aços disponíveis na nova versão do LESM, Fonte: Autor (2021)	78
Tabela 3.3	Diagrama da classe W_Beam, Fonte: Autor (2021)	79
Tabela 3.4	Diagrama da classe Material, Fonte: Autor (2021)	80
Tabela 3.5	Diagrama da classe Steel, Fonte: Autor (2021)	80
Tabela 3.6	Diagrama da classe Elems_Structural_Group, Fonte: Autor (2021)	83
Tabela 3.7	Diagrama da classe NBR8800, Fonte: Autor (2021)	85
Tabela 3.8	Diagrama da classe NBR8800_Report, Fonte: Autor (2021)	86
Tabela 4.1	Casos de carga pórtico 2D, Fonte: Autor (2021)	93
Tabela 4.2	Comparação dos resultados dos esforços de tração entre os programas CypeCad e LESM, Fonte: Autor (2021)	98
Tabela 4.3	Comparação do momento solicitante entre os programas CypeCad e LESM, Fonte: Autor (2021)	100
Tabela 4.4	Comparação do momento resistente para o estado limite FLT entre os programas CypeCad e LESM, Fonte: Autor (2021)	102
Tabela 4.5	Comparação do momento resistente para o estado limite FLM entre os programas CypeCad e LESM, Fonte: Autor (2021)	103
Tabela 4.6	Comparação do momento resistente para o estado limite FLA entre os programas CypeCad e LESM, Fonte: Autor (2021)	104
Tabela 4.7	Comparação dos resultados dos esforços de compressão entre os programas CypeCad e LESM, Fonte: Autor (2021)	105
Tabela 4.8	Comparação dos resultados dos esforços de cisalhamento entre os programas CypeCad e LESM, Fonte: Autor (2021)	108
Tabela 4.9	Comparação dos resultados dos esforços de flexo-tração e flexo-compressão entre os programas CypeCad e LESM, Fonte: Autor (2021)	111

*Educação é uma descoberta progressiva de
nossa própria ignorância.*

Voltaire, .

Lista de Abreviaturas

2D – Duas dimensões

3 – Três Dimensões

AA – elemento apoiado-apoiado

AL – elemento apoiado-livre

FLA – Flambagem local da alma

FLM – Flambagem local da mesa

FLT – Flambagem lateral com torção

MATLAB® – MATrix LABoratory (the language of technical computing)

UML – Linguagem de Modelagem Unificada (do inglês, Unified Modeling Language)

1

Introdução

O avanço tecnológico progressivo da última década vem transformando grande parte das atividades e processos que anteriormente eram realizadas de forma manual. Tornando, assim, incontestável a mudança nos paradigmas organizacionais e no desenvolvimento adotado para essas atividades.

Para elucidar o impacto que o avanço tecnológico causa em determinadas áreas, pode-se trazer à tona o fato de que nos últimos anos a engenharia civil vem passando por modernizações em toda sua cadeia de processos. Segundo Kimura [1, p. 18] “desde o lançamento dos dados, passando pela análise estrutural, dimensionamento e detalhamento dos elementos, até a impressão de desenhos, de alguma forma, são influenciadas pela rapidez e precisão que a informática proporciona”.

Tendo em vista esse cenário, pode-se afirmar que um engenheiro provavelmente irá passar por diversos momentos em que *softwares* irão automatizar processos, como os citados acima. Dessa forma, é de suma importância que esse profissional tenha algum conhecimento prévio na manipulação dessas ferramentas. Contudo, vale ressaltar que muitos desses *softwares* possuem licenças pagas, tornando seu acesso quase inviável durante a fase de formação acadêmica. [2].

Por esse motivo, alguns *softwares* disponibilizam licenças de estudantes. Porém, a duração costuma ser curta, apenas para teste, e, raramente, distribuem informações sobre o funcionamento interno do programa, contribuindo ainda mais para que a interação desses *softwares* com jovens engenheiros seja pequena. Outro ponto de grande desvantagem, segundo Saraiva e Lima [2], é que *softwares* proprietários não podem ser usados, copiados, estudados, modificados e redistribuídos sem restrição, impossibilitando que, por parte dos usuários, haja qualquer ação de cooperação para o seu aperfeiçoamento.

Dessa forma, a melhor maneira de garantir um uso mais consistente de ferramentas computacionais durante a formação é através de *softwares* educacionais. Visando esse objetivo, surge o desenvolvimento do *software* LESM (Linear Elements Structure Model), desenvolvido pelo instituto Tecgraf da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Na versão atual, trata-se de um programa de computador para análise linear elástica

de estruturas reticuladas bidimensionais e tridimensionais, o qual “utiliza a linguagem de programação (script language) do ambiente MATLAB®, adotando uma disciplina de programação orientada a objetos”[3, p. 2].

O programa em questão é gratuito e de código aberto, trazendo grandes benefícios à comunidade educacional tanto, na questão ligada a utilização de *softwares* na formação do engenheiro, quanto no compartilhamento de seu funcionamento, através de seu código amplamente documentado. Assim, permite-se o entendimento completo do funcionamento da ferramenta, contribuindo para a comunidade acadêmica.

Como citado anteriormente, o LESM utiliza o paradigma da linguagem orientada a objetos. Esse é um ponto chave sobre a versalidade do *software*, sendo responsável por permitir o recebimento de atualizações e adições de funcionalidades. Essa questão é explorada ao longo da dissertação, utilizando os artifícios da linguagem visual, UML (Unified Modeling Language). A UML é utilizada para modelar *softwares* orientados a objetos de maneira mais ilustrada, com a finalidade de elucidar as características do sistema.[4, p. 19].

Sabendo que o *software* LESM não incluía nenhum tipo de dimensionamento de estruturas, foi proposto, por meio dessa dissertação, prover uma nova funcionalidade para verificação de perfis metálicos com o propósito de aumentar ainda mais a abrangência de seu uso. Para tal funcionalidade, utilizou-se como base o código normativo brasileiro, NBR8800:2008. Além disso, também foi gerado a documentação dessa implementação, elucidando todos os processos. Durante a descrição desses processos, elucidou-se como foi realizada toda a abstração das seções utilizadas da norma, seja no âmbito de rotinas de cálculos ou na implementação estrutural orientada a objetos, mais detalhada nas seções seguintes.

1.1 Motivação

Esse estudo tem como principal motivação disseminar o conhecimento a respeito do desenvolvimento de *softwares* voltados a engenharia civil, com enfoque no *software* LESM. O programa em questão tem cunho totalmente educacional, além de ser de disseminação livre, permitindo que qualquer aluno de universidade ou pesquisadores acompanhem o processo de construção, aperfeiçoamento e utilização.

O processo descrito neste documento visa agregar ao acervo em torno dessa temática com o intuito de fomentar mais contribuições a respeito desse segmento. Além disso, traz-se como resultado a nova funcionalidade de verificação de perfis metálico, contribuição de grande valor didático para o

corpo docente e discente das universidades que utilizam o LESM.

1.2

Objetivos

Os objetivos gerais desta dissertação são:

- Elaborar uma metodologia detalhada sobre o processo de implementação da nova funcionalidade de verificação de perfis ao *software* LESM, fazendo referência ao uso da linguagem orientada a objetos na implementação da estrutura de dados adotada;
- Criar um processo iterativo que ofereça suporte ao usuário, possibilitando a modelagem de estruturas metálicas com a finalidade de verificar a resistência dos perfis adotados nesse processo;
- Geração de relatórios de cálculos para cada verificação realizada para os perfis que compõem a modelagem estrutural.

Esta dissertação tem os seguintes objetivos específicos:

- Demonstrar todo o processo de implementação da nova funcionalidade de verificação de perfis metálico de pórticos 2D no *software* educacional LESM. Tem-se como a principal premissa a correlação do processo de estruturação das rotinas de cálculo, baseadas na linguagem orientada a objetos, com as diretrizes dadas pela NBR8800:2008. O intuito dessa análise é trazer esclarecimentos de como é realizada a abstração do código normativo para que o mesmo consiga ser inserido no *software*;
- Elaborar um método automatizado de verificação que siga, assertivamente, todas as especificidades da NBR8800:2008, exibindo de maneira clara para o usuário quais foram os critérios da norma utilizada para realizar a verificação de determinado perfil;
- Gerar gráficos que exibam a taxa de utilização ao longo da barra para cada esforço verificado;
- Gerar um memorial de cálculo, no qual o usuário seja capaz de interagir e analisar o dimensionamento referente ao esforço especificado.

1.3

Escopo

O documento limita-se a:

- Implementar a verificação de perfis metálicos de pórticos 2D, considerando uma análise de comportamento linear elástica;
- Adotar a NBR8800:2008 para verificação de esforços de tração, compressão, flexão, flexo-compressão e flexo-tração;
- Apresentar o esforço de tração com certa simplificação, visto que a versão atual do *software* ainda não detêm informações sobre ligações metálicas;
- Exibir apenas as verificações de estados limites últimos;
- Atender somente os perfis I e H laminados;
- Não abranger o anexo H da NBR8800:2008 para almas esbeltas;
- Gerar combinação de ações;
- Gerar a documentação UML.

Um ponto importante a ser considerado é que mesmo a verificação de estado limite de serviço não estando disponível como resultado do *software*, já se possui uma estrutura definida para futuras implementações.

1.4

Organização do trabalho

O presente trabalho divide-se em 5 capítulos, contendo uma introdução, seguida do referencial teórico, da metodologia, dos resultados e da conclusão.

Na introdução, apresenta-se fatos gerais relevantes ao assunto tratado nessa dissertação, além da motivação, objetivo e estrutura do trabalho, com o intuito de inicializar e instigar o leitor no tema.

Já no referencial teórico, constrói-se uma base teórica, fundamentada através de referências da literatura, a respeito das definições, dos diagramas, dos métodos, das linguagens de programação, dos *softwares* e de outros termos abordados.

Na metodologia, evidencia-se a organização a ser seguida na presente dissertação, visando atingir o objetivo, descrito na introdução. Dessa forma, faz-se um levantamento de todas as etapas de elaboração da nova funcionalidade que se pretende inserir na extensão do *software* LESM.

Nos resultados, apresenta-se os dados obtidos a partir das novas funcionalidades inseridas, comparando-os com outro *software* de cálculo estrutural, já consolidado no mercado.

Por fim, na conclusão são enumerados as vantagens e desvantagens da abordagem seguida na dissertação, assim como sugestões de aperfeiçoamento para trabalhos futuros.

2

Referencial teórico

2.1

Programação orientada a objetos

2.1.1

Surgimento

Inicialmente, a programação das máquinas era feita através de chaves binárias em programas com poucas instruções. Quando a complexidade dos *softwares* começou a aumentar, surgiu a necessidade de rever o processo, nascendo assim o Assembly [12]. Essa linguagem de programação, apesar de ser de baixo nível, era de rápido processamento e permitia ao programador gerenciar melhor as complexidades. Contudo, sua sintaxe era muito extensa e distante da linguagem humana, dificultando a construção de programas complexos. Tendo em vista esse cenário, surgem as linguagens de alto nível, como: FORTRAN, C e Pascal. As suas maiores vantagens são uma sintaxe mais semântica e curta, com um conceito de programação estruturada. Dessa forma, aumentou-se a velocidade de criação de programas, possibilitando a resolução de problemas um pouco maiores. [11]

No entanto, essas linguagens estruturadas possuem limitações em relação à escalabilidade e manutenção, visto que, conforme a aplicação atinge um tamanho considerável, a relação entre suas funções deixa de ser clara, sendo extramente complexo compreender seu funcionamento. Para resolver essa questão, nasce a programação orientada a objetos, aproveitando parte da linguagem estruturada, mas introduzindo novos conceitos, como:

- Objetos;
- Classes;
- Herança;
- Polimorfismo.

Em 1966 foi criada a primeira linguagem orientada a objetos, denominada Simula. Desenvolvida na Noruega, foi a primeira a definir o conceito de classes e objetos . A partir desse momento, novas linguagens, também orientadas a

objetos, porém mais aprimoradas, foram surgindo, como C++, C#, Java, Python e MATLAB. Dadas sua robustez e curva de aprendizado acelerado, ainda são amplamente utilizadas. Além disso, por mais que sigam os mesmos preceitos, cada uma possui aplicações específicas nas quais desempenham melhor. [11]

2.1.2 Conceitos

2.1.2.1 Objetos

Objetos podem ser considerados pequenas entidades autônomas que possuem características individuais, podendo ser acessadas e manipuladas. Basicamente o objeto é composto por: [5]

- Propriedades (atributos): Informações que definem características internas do objeto, as quais só podem ser acessadas por ele;
- Comportamento: São ações, denominadas métodos, que o objeto pode realizar. Com esses métodos podem-se executar instruções lógicas que mudam os estados das propriedades com determinada finalidade;
- Identidade: Cada objeto é único e independente.

Para elucidar a entidade de um objeto, tem-se o esquema mostrado na Figura 2.1:

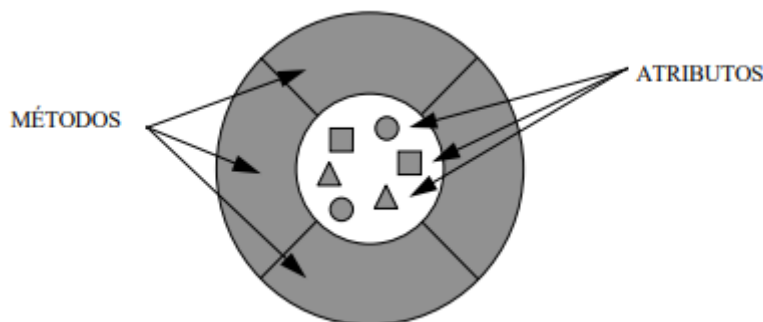


Figura 2.1: Representação para um objeto. Fonte: Kamienski (1996)

Salienta-se que, mesmo o objeto sendo uma entidade única, dentro de seus atributos podem ser inseridos referências a outros objetos. Essa é uma característica muito relevante, pois, esse tipo de prática facilita a conectividade das classes, tratadas em seguida, particionando entidades de objetos “maiores” em várias “menores”.

2.1.2.2 Classe

Para criação de objetos é necessário especificar previamente suas propriedades e comportamentos. Para isso, define-se uma estrutura denominada classe. Com as classes, torna-se possível extrair todas as informações que o objeto de um determinado segmento irá conter. Após essa etapa, pode-se iniciar o objeto e utilizá-lo de maneira semelhante a uma variável, porém, com métodos e propriedades acessíveis, definidas nas classes. Esse processo de criação de um objeto a partir de uma classe se chama instanciar um objeto.

Vale ressaltar que podemos ter múltiplas instâncias independentes, inclusive em objetos pertencentes a mesma classe, como elucidado na Figura 2.2. Assim, o comportamento desses objetos provenientes da mesma classe é o mesmo, porém, sua manipulação e utilização é independente. Caso o valor da propriedade de uma das instâncias seja alterada, apenas a que foi modificada terá seu valor atualizado.

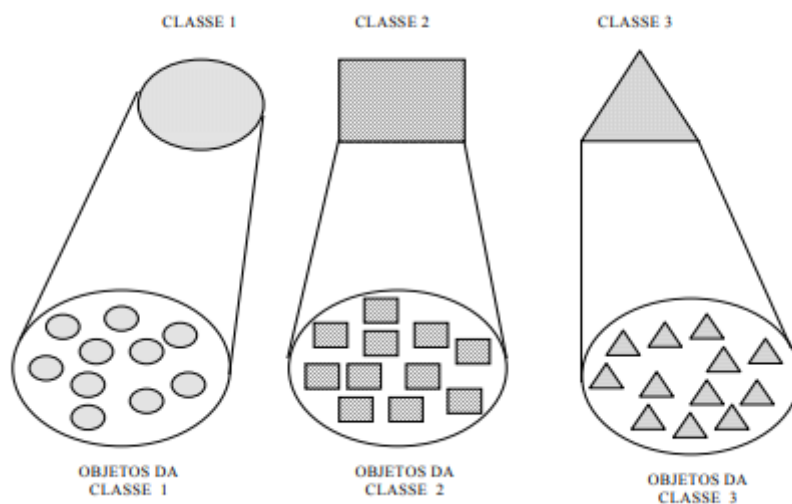


Figura 2.2: Relacionamentos entre classes e objetos. Fonte: Kamienski (1996)

2.1.2.3 Herança

O conceito de herança parte do princípio da passagem de características de um grupo mais abrangente para seus subgrupos. De acordo com Coad e Yourdon [11, p. 15] herança é: “Mecanismo para expressar a similaridade entre classes, simplificando a definição de classes similares a outras que já foram definidas. Ela representa generalização e especialização, tornando atributos e serviços comuns em uma hierarquia de classe.”

Para caracterizar melhor esse conceito, pode-se ter em mente o cenário em que se tem uma classe denominada animal e todas as suas subclasses herdam os atributos e métodos que a pertencem. Visto que todos os animais possuem gênero, idade e se alimentam é mais eficiente extrair essas características comuns a essas subclasses, aglomerá-las a uma classe mãe e repassá-las através de herança, tal como ilustra a Figura 2.3.

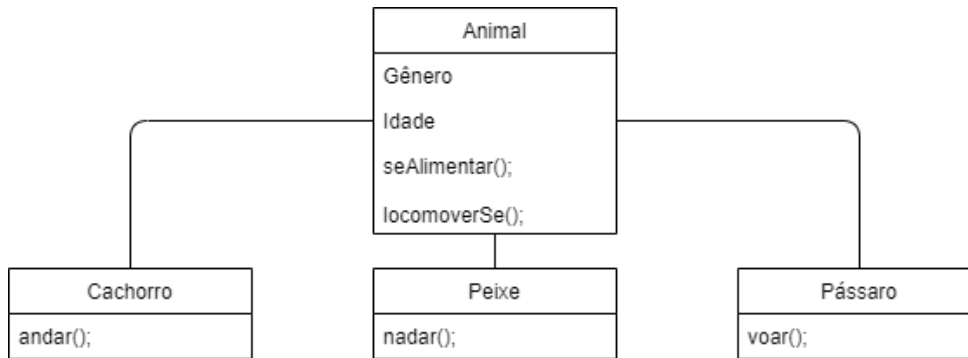


Figura 2.3: Esquemática do conceito de Herança. Fonte: Autor (2021)

2.1.2.4 Polimorfismo

O polimorfismo depende diretamente da herança, trazendo, porém, um novo conceito. Esse novo conceito oferece a possibilidade de redefinir algum método, por exemplo, de acordo com alguma especificidade da subclasse. Voltando ao exemplo anterior dos animais, pode-se utilizar do polimorfismo para extrair ainda mais o método ligado a locomoção de cada animal, definindo na classe mãe o método **locomover**, tal como ilustra a Figura 2.4. Ainda é possível, por meio da herança, passá-lo para as respectivas subclasses onde esse método será definido de acordo com sua particularidade.

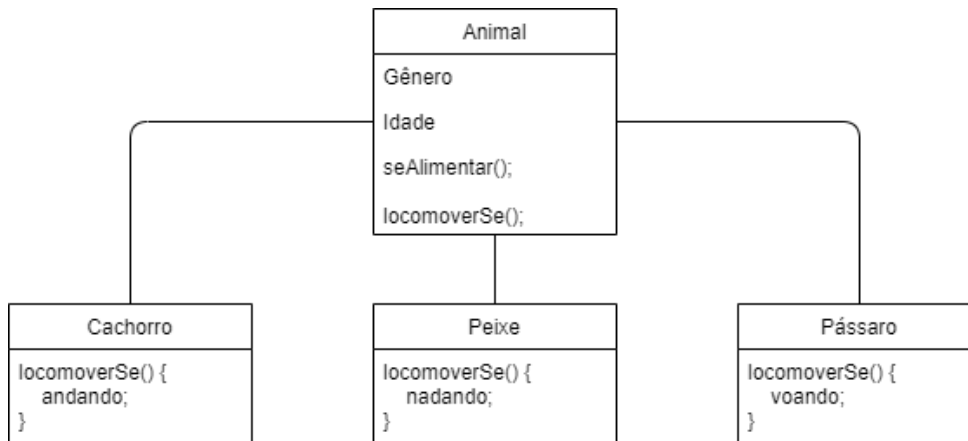


Figura 2.4: Esquematização conceito de Polimorfismo. Fonte: Autor (2021)

2.2

UML (Linguagem de Modelagem Unificada)

A linguagem UML tem como principal propósito modelar *softwares* baseados na utilização do paradigma da orientação a objetos. Trate-se de uma linguagem apenas de modelagem, a qual, segundo Pereira [6], “foi concebida com o intuito de estabelecer um padrão único a ser usado para a especificação das características dos sistemas computacionais projetados para atender às necessidades dos usuários desses sistemas.” Essa seção tem como intuito elucidar as possíveis utilizações dos diagramas UML, visando facilitar o entendimento dos capítulos futuros.

2.2.1

Propósito da modelagem de *softwares*

Durante a elaboração de um *software* é extremamente relevante realizar previamente uma modelagem de sua estrutura, pois deve-se abordar o programa como uma entidade “viva” com elevado potencial de crescimento. Dessa forma, cada vez que novas funcionalidades são inseridas ao projeto sua complexidade pode aumentar exponencialmente, sendo uma boa modelagem prévia e contínua fundamental para dar suporte ordenado a esse crescimento. Além disso, através desse processo também pode-se enumerar alguns benefícios que serão importantes ao projeto como: [4]

- Análise de Requisitos;
- Prototipação;
- Prazos e Custos;

- Manutenção.

Existem muitos outros modelos de diagramas UML, porém, será dado enfoque apenas aos citados a cima.

2.2.1.1

Diagramas de casos de uso

O diagrama de casos de uso, de forma geral, é utilizado durante a fase em que é feito o levantamento e análise de requisitos do sistema. Através dele, pode-se definir também fluxos de funcionamento de baixo detalhamento. Com isso, utiliza-se esse diagrama para embasar os próximos diagramas que irão constituir a documentação do *software*.

Para caracterizar esse diagrama, pode-se observar um exemplo, mostrado na Figura 2.5, no qual ilustra-se um diagrama de casos de uso de um sistema de banco.

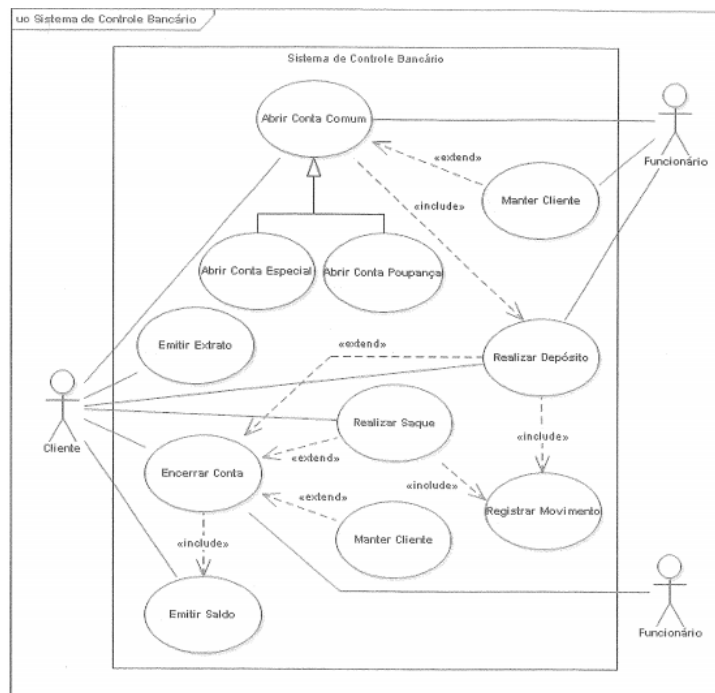


Figura 2.5: Exemplo de diagrama de caso de uso. Fonte: Guedes (2018)

Como se pode observar, o diagrama acima apenas ilustra, de maneira geral, as atividades que o *software* irá desempenhar e como irá se relacionar com o usuário nesse processo. O exemplo demonstra exatamente uma cadeia de ações que o funcionário e o cliente podem desempenhar utilizando o *software* e a correlação entre os indivíduos. Apesar do diagrama ser superficial, traz um ponto de partida muito conciso para o início do desenvolvimento do programa.

2.2.1.2

Diagramas de classes

O diagrama de classes é um dos mais importantes, visto que ilustra o relacionamento entre as classes que existem no *software*. Além disso, também elucida os métodos e propriedades dessas classes. Para o paradigma da orientação a objetos, a modelagem desse diagrama é fundamental para um entendimento global da aplicação.

Continuando o modelo de exemplificação anterior, o diagrama de classes para o sistema bancário ficaria da forma mostrada na Figura 2.6:

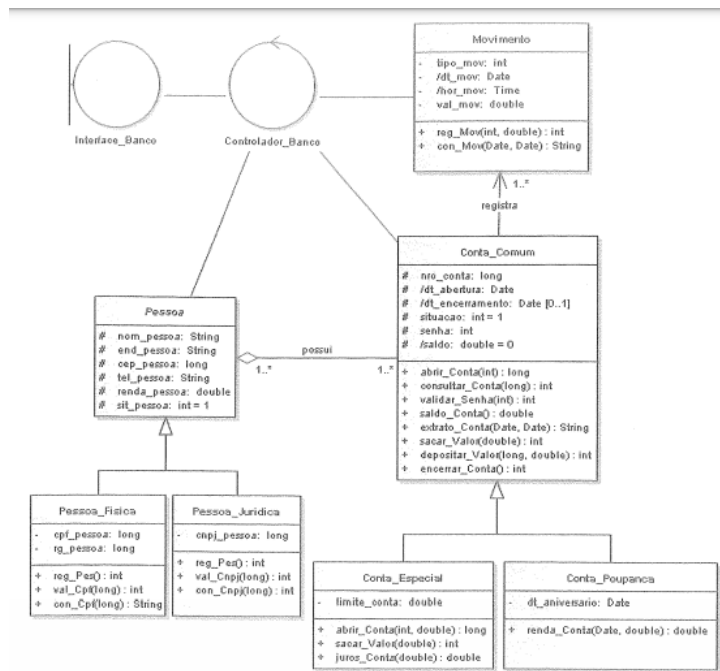


Figura 2.6: Exemplo de diagrama de classes. Fonte: Guedes (2018)

2.2.1.3

Diagrama de objetos

O diagrama de objetos é complementar ao de classe, porém com uma especificidade a mais. Os valores armazenados nos objetos estanciados, em determinado momento da execução do *software*, são mostrados (ver Figura 2.7). Esse recurso é extremamente importante, pois torna possível gerenciar o estado dos objetos precisamente em diferentes momentos da aplicação.

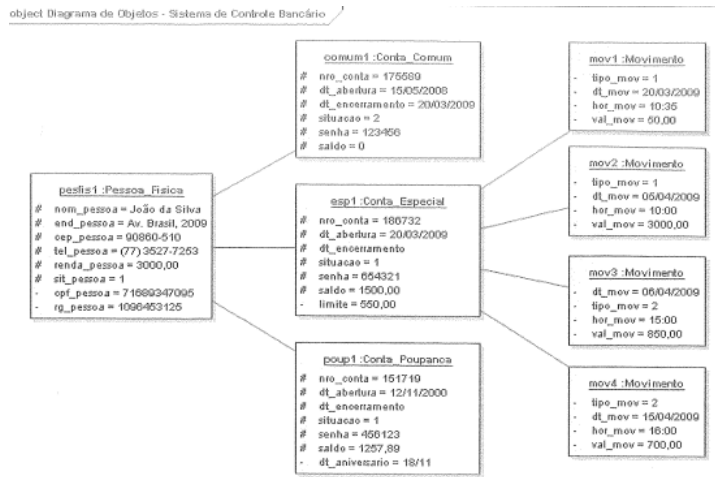


Figura 2.7: Exemplo de diagrama de objetos. Fonte: Guedes (2018)

Nota-se que, para esse sistema exemplificado acima, cada caixa do diagrama representa uma instância das classes, colocando em evidência os valores armazenados em suas propriedades.

2.2.1.4 Diagrama de sequência

O diagrama de sequência é responsável por ilustrar um aspecto temporal do conjunto de ações. Logo, geralmente, tem o intuito de representar ações encadeadas de maneira sequencial, conforme ilustrado na Figura 2.8. Em especial, pode-se observar que, além da ordem de execução, a relação de dependência de uma ação em relação a outra.

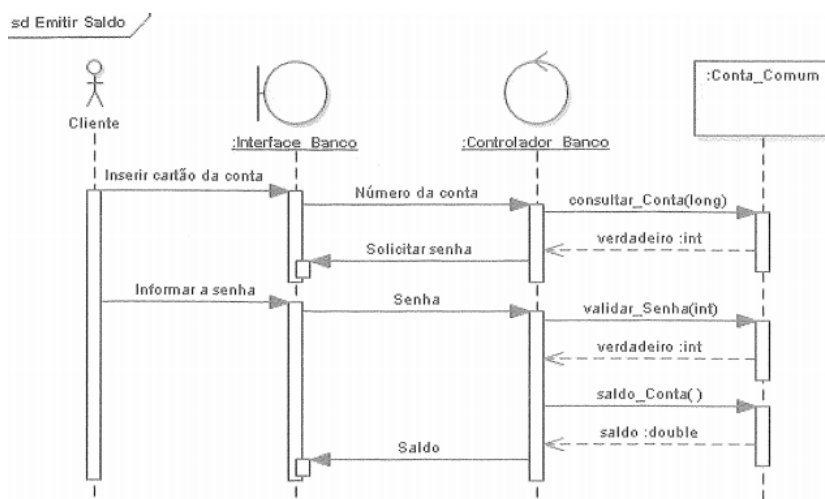


Figura 2.8: Exemplo de diagrama de sequência. Fonte: Guedes (2018)

2.3

MATLAB

Segundo a Mathworks (<https://www.mathworks.com/>) (empresa criadora do MATLAB):

MATLAB® é uma plataforma de programação projetada especificamente para engenheiros e cientistas para analisar e projetar sistemas e produtos que transformam nosso mundo. O coração do MATLAB é a linguagem MATLAB, uma linguagem baseada em matriz que permite a expressão mais natural da matemática computacional.

Além de facilitar o uso de expressões matemáticas na computação, outro benefício extraordinário dessa plataforma é a possibilidade da utilização do paradigma da orientação a objetos. Dessa forma, permite-se a criação de aplicações robustas e escaláveis. O MATLAB ainda possui diversas ferramentas que facilitam a criação de *interfaces* interativas dentro das aplicações desenvolvidas.

Outro ponto importante é que o MATLAB é uma linguagem com uma curva de aprendizado relativamente rápida, com uma vasta documentação e uma comunidade fortemente ativa em diversos fóruns.

2.4

LESM (*Linear Elements Structure Model*)

2.4.1

Definição

O LESM [16] é desenvolvido em MATLAB no Instituto Tecgraf de Desenvolvimento de Software Técnico-Científico da PUC-Rio (Tecgraf/PUC-Rio). Este programa é totalmente *open source* e foi concebido inicialmente como uma ferramenta não gráfica para análise linear-elástica de modelos de elementos unidimensionais, como treliças e pórticos 2D / 3D e grelhas. O foco da primeira versão do LESM foi fornecer um código-fonte didático e bem documentado para apresentar aos alunos a implementação computacional de métodos de análise estrutural. Foi tema de um livro sobre Análise Matricial de Estruturas [15] e atualmente está sendo utilizado em alguns cursos na PUC-Rio e em outras Universidades. Posteriormente foi desenvolvida uma interface gráfica para o programa [17].

Assim como citado acima, o programa foi desenvolvido utilizando a linguagem MATLAB, de acordo com os paradigmas da orientação a

objetos. Para as rotinas de cálculo, utilizou-se o método da rigidez direta, o qual foi totalmente abstraído através de classes que comportam as rotinas implementadas. Contudo, manteve-se toda a base de cálculo matricial, visto que o MATLAB comporta muito bem esse tipo de notação.

2.4.2

Implementação do método da rigidez direta

Apesar do método da rigidez direta já estar inserido, previamente a esse trabalho, no LESM, será descrito nessa seção como esse processo ocorreu, assim como sua motivação.

Para obter as soluções global e local pelo método da rigidez direta, deve-se decompor a estrutura analisada em dois estágios. No estágio I deve-se isolar o efeito local da solicitação, no qual os elementos de barra carregados estão em uma situação de engastamento perfeito. Fato que justifica a utilização desse modelo, visto que gera como resultado deslocamentos e rotações nodais nulos para esse primeiro estágio, garantindo que os resultados do modelo discretizado não se modifiquem caso diferentes níveis de discretização forem utilizados. Já no estágio II, considera-se apenas o efeito global da solicitação. Nesse estágio, as forças e momentos nodais são equivalentes às reações de engastamento do estágio I invertidas. Assim, a exceção do trecho carregado (visto que o carregamento nesse trecho provoca efeitos locais de engastamento que devem ser adicionados), o comportamento final da estrutura é correspondente ao comportamento global do estágio II.

Vale ressaltar também que, nesse segundo estágio, a solicitação acontece apenas nos nós do modelo e, em termos de deslocamentos e rotações nodais, os resultados obtidos são os mesmos para a estrutura do carregamento original. Dessa forma, evidencia-se que o estágio II reflete, na verdade, o comportamento global discretizado da estrutura.

Para utilização do método da rigidez direta para modelos reticulados são necessários alguns conceitos importantes:

- Cargas nodais: forças e momentos que atuam diretamente nos nós da discretização, no carregamento original da estrutura;
- Cargas equivalentes nodais: cargas nodais atuantes no estágio II (geradas a partir das reações de engastamento perfeito invertidas dos elementos de barra carregados no estágio I);
- Cargas nodais combinadas: solicitações do estágio II, retratando o efeito discretizado das solicitações externas atuando nos nós.

Através desse método é possível determinar completamente o vetor das forças nodais generalizadas globais. Isso ocorre por meio das componentes de reação de apoio e dos valores dos graus de liberdade desconhecidos (componentes de deslocamentos e rotações nodais livres), também encontrados com o método da rigidez direta.

Além disso, os esforços internos em todos os elementos de barra do modelo estrutural também podem ser encontrados através da análise do estágio II. Dessa forma, ao superpor as elásticas e esforços internos da situação de engastamento perfeito do estágio I, apenas para os elementos de barra carregados, obtêm-se os resultados completos.

Para elucidar a metodologia de análise por esse método, a Figura 2.9 mostra um diagrama de atividades da versão não gráfica do programa LESM, de acordo com o padrão UML.

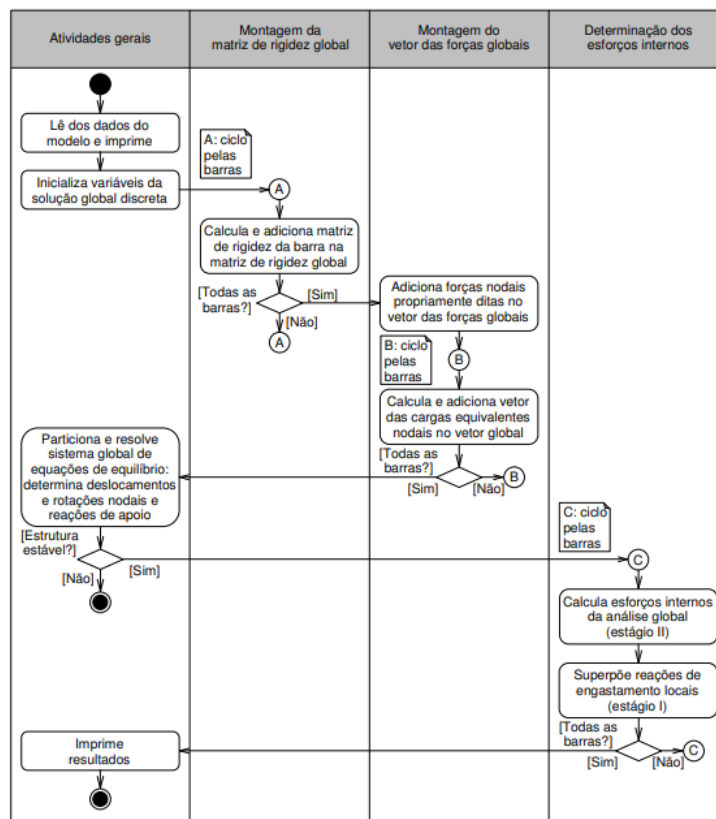


Figura 2.9: Diagrama de atividades da versão não gráfica do programa LESM para solução do problema global discreto pelo método da rigidez direta. Fonte: Martha, (2018)

Salienta-se que as principais variáveis globais do programa LESM são:

- Vetor dos deslocamentos e rotações nodais: graus de liberdade;

- Matriz de rigidez global: matriz dos coeficientes do sistema de equações proveniente da formulação do problema global discreto;
- Matriz ID: matriz que armazena a numeração global dos graus de liberdade. Inicializada seguida da leitura dos dados e viabilizadora da construção da matriz de rigidez global (procedimento mostrado na segunda coluna da acima).

Tanto o vetor dos deslocamentos e rotações nodais quanto a matriz de rigidez global são armazenados como propriedades de um objeto de classe *Drv* (*Analysis Driver Class*).

A matriz de rigidez global começa com valores nulos e, conforme se realiza um ciclo por todas as barras, é acrescentada a contribuição de cada matriz de rigidez local. Sendo assim, essa matriz global nada mais é do que a superposição das matrizes de rigidez locais (matrizes de rigidez dos elementos de barra do modelo). Para realização desse procedimento de superposição, necessita-se dos vetores de espalhamento, determinados através da matriz ID, para relacionar as posições em que os coeficientes das matrizes locais se ligam na matriz global.

O vetor das forças globais inicia com as forças nodais propriamente ditas e, assim como a matriz de rigidez global, conforme se realiza um ciclo por todas as barras, adicionam-se a esse vetor as cargas equivalentes nodais de cada barra. Para a construção desse vetor, discutida na terceira coluna da imagem acima, faz-se necessário o uso dos vetores de espalhamento.

Com a solução global do sistema, ficam estabelecidos os valores das reações de apoio assim como os valores dos graus de liberdade do problema. Para isso, basta solucionar o sistema de equações globais, por meio da partição da matriz de rigidez global e do vetor das forças globais (processo indicado na primeira coluna da Figura 2.9).

Em suma, a partir da superposição dos esforços internos resultantes da análise do estágio II (análise global discreta) com os esforços de engastamento (reações de apoio locais) das barras isoladas no estágio I, obtêm-se os esforços internos finais nas barras, processo mostrado na quarta coluna da imagem acima. Sendo essa atividade considerada primordial antes da impressão dos resultados.

Vale ressaltar que, para a montagem do vetor das forças globais, fazem-se necessárias as cargas equivalentes nodais de cada barra determinadas a partir do cálculo das reações de apoio locais. Ademais, as reações de apoio locais permanecem contidas nos elementos de barra para, posteriormente, serem utilizadas no cálculo dos esforços internos finais. Por fim, mediante a superposição dos deslocamentos originados da análise global discreta com os deslocamentos provenientes das solicitações externas atuando no interior das

barras no caso de engastamento perfeito, determinam-se os deslocamentos no interior das barras.

2.4.3

Arquitetura orientada a objeto

Com intuito de transformar todo o processo de concepção do método da rigidez direta, descrito na seção anterior, em uma rotina computacional orientada a objetos, foi realizada uma abstração total da metodologia, separando-a em classes com responsabilidades específicas na aplicação. Cada classe possui suas propriedades e métodos associados, nas quais as instâncias criadas a partir delas poderão desempenhar papéis associativos, agregativos, dependentes e de composição entre si. Para expor com mais detalhes a divisão de classes e suas correlações, a seguir será dado um breve explicativo sobre a organização adotada para tal, ilustrada também na Figura 2.9.

2.4.3.1

Divisão de Classes

A organização de classes orientadas a objeto da versão original do LESM é mostrada na Figura 2.15. Em seguida, são descritas as principais classes do programa.

- *Drv (Analysis Driver Class)*: Classe responsável pela análise estrutural, na qual são definidos métodos e propriedades globais, representando o modelo estrutural analisado. Essa classe contém uma lista de referências para as instâncias do modelo, sendo elas, segundo Martha [3]: “materiais (objetos da classe Material), seções transversais (objetos da classe Section), nós (objetos da classe Node), elementos de barra (objetos da classe Elem) e elementos de carregamento (objetos da classe Lelem)”. Nessa mesma classe, tem-se uma referência a um objeto instanciado relacionado a classe Anm, responsável por gerenciar o tipo de análise. Essa análise pode alternar entre treliça 2D e 3D, pórtico 2D e 3D ou grelha. Na classe Drv, apresentada na Figura 2.10, também se tem outras propriedades de grande importância para o processo de cálculo dos esforços, como as matrizes globais do modelo discreto, o vetor dos graus de liberdade, o vetor das forças combinadas e a matriz de rigidez global.

Vale ressaltar que, nas Figuras 2.10, a 2.14, o símbolo “~” indica as propriedades com visibilidade para todos os métodos das classes de análise e o “+” os métodos das respectivas classes.

Drv - analysis driver class
<p>~anm: Anm Referência para um objeto da classe <i>Anm</i></p> <p>~materials: Material[] Vetor de referência para objetos da classe <i>Material</i></p> <p>~sections: Section[] Vetor de referência para objetos da classe <i>Section</i></p> <p>~nodes: Node[] Vetor de referência para objetos da classe <i>Node</i></p> <p>~elems: Elem[] Vetor de referência para objetos da classe <i>Elem</i></p> <p>~K: double[][] Matriz global de rigidez</p> <p>~F: double[] Vetor global de carregamento</p> <p>~D: double[] Vetor global de deslocamento</p> <p>~ID: int[][] Matriz de numeração global dos graus de liberdade</p> <p>~nmat: int Número de materiais do modelo</p> <p>~nsec: int Número de seções transversais do modelo</p> <p>~nnp: int Número de nós do modelo</p> <p>~nel: int Número de elementos do modelo</p> <p>~neq: int Número total de equações de equilíbrio</p> <p>~neqfree: int Número de equações de equilíbrio de graus de liberdade livres</p> <p>~neqfixed: int Número de equações de equilíbrio de graus de liberdade fixos</p>
<p>+Drv() Método construtor</p> <p>+fictRotConstraint(int fict): void Insere ou remove apoios fictícios de rotação</p> <p>+dimKFD(): void Inicializa a matriz e vetores globais</p> <p>+assembleDOFNum(): void Monta a matriz de numeração global dos graus de liberdade</p> <p>+assembleGle(): void Monta os vetores de espalhamento dos elementos</p> <p>+gblMtx(): void Monta a matriz global de rigidez</p> <p>+assembleElemMtx(double[][] keg, int e): void Insere a matriz de rigidez de um elemento na matriz global de rigidez</p> <p>+elemLoads(): void Adiciona cargas equivalentes nodais ao vetor global de carregamento</p> <p>+assembleENL(double[] feg, int e): void Insere o vetor de cargas equivalentes nodais de um elemento no vetor global</p> <p>+solveEqnSystem(): int Resolve o sistema de equações de equilíbrio</p> <p>+elemIntForce(): void Calcula os esforços internos nas extremidades dos elementos</p> <p>+elemIntDispl(): void Calcula os deslocamentos internos dos elementos</p> <p>+process(): int Processa os dados do modelo pelo método da rigidez direta</p> <p>+clean(): void Força liberação de memória utilizada pelas propriedades</p>

Figura 2.10: Métodos e propriedades da classe *Drv*. Fonte: Martha, (2018)

- Material (Figura 2.11): Essa classe define as propriedades do material associado ao elemento de barra;

Material - material class
<p>~id: int Número identificador</p> <p>~elasticity: double Módulo de elasticidade</p> <p>~poisson: double Coeficiente de Poisson</p> <p>~shear: double Módulo de cisalhamento</p> <p>~thermExp: double Coeficiente de expansão térmica</p>
<p>+Material() Método construtor</p> <p>+clean(): void Força liberação de memória utilizada pelas propriedades</p>

Figura 2.11: Métodos e propriedades da classe Material. Fonte: Martha, (2018)

- Section (Figura 2.12): Classe responsável por conter métodos e propriedades referentes à seção associada ao elemento de barra;

Section - cross-section class
<p>~id: int Número identificador</p> <p>~area_x: double Área total</p> <p>~area_y: double Área de cisalhamento na direção local y</p> <p>~area_z: double Área de cisalhamento na direção local z</p> <p>~inertia_x: double Momento de inércia à torção</p> <p>~inertia_y: double Momento de inércia à flexão na direção local y</p> <p>~inertia_z: double Momento de inércia à flexão na direção local z</p> <p>~height_y: double Altura na direção local y</p> <p>~height_z: double Altura na direção local z</p>
<p>+Section() Método construtor</p> <p>+clean(): void Força liberação de memória utilizada pelas propriedades</p>

Figura 2.12: Métodos e propriedades da classe Section. Fonte: Martha, (2018)

- Node (Figura 2.13): Classe que gerencia as informações das estâncias dos nós para o modelo reticulado. No Node tem-se desde coordenadas até as cargas nodais, por exemplo;

Node - node class
<p>~id: int Número identificador</p> <p>~coord: double[] Vetor de coordenadas</p> <p>~ebc: int[] Vetor de identificadores de condição de suporte</p> <p>~nodalLoad: double[] Vetor de cargas aplicadas</p> <p>~prescDispl: double[] Vetor de deslocamentos prescritos</p>
<p>+Node() Método construtor</p> <p>+elemsIncidence(Drv drv): int, int Retorna o número total de elementos conectados ao nó e o número de elementos articulados (com rótula) no nó</p> <p>+clean(): void Força liberação de memória utilizada pelas propriedades</p>

Figura 2.13: Métodos e propriedades da classe Node. Fonte: Martha, (2018)

- Elem (Figura 2.14): Classe de extrema importância, definida para que objetos instanciados a partir dela contêm propriedades que armazenem desde a matriz de rotação do elemento e os esforços que o elemento está submetido. Além disso, a classe também possui diversos métodos acoplados para calcular essas grandezas;

```

Elem - element class
~nen: int
Número de nós do elemento (sempre 2)
~type: int
Identificador do tipo de elemento (Euler-Bernoulli ou Timoshenko)
~anm: Anm
Referência para um objeto da classe Anm
~load: Lelem
Referência para um objeto da classe Lelem
~material: Material
Referência para um objeto da classe Material
~section: Section
Referência para um objeto da classe Section
~nodes: Node[]
Vetor de referência para objetos da classe Node
~hingei: int
Identificador para a existência de rótula no nó inicial
~hingef: int
Identificador para a existência de rótula no nó final
~length: double
Comprimento
~cosine_X: double
Cosseno do ângulo diretor com o eixo global X
~cosine_Y: double
Cosseno do ângulo diretor com o eixo global Y
~cosine_Z: double
Cosseno do ângulo diretor com o eixo global Z
~vz: double[]
Vetor auxiliar no plano local xz
~gle: int[]
Vetor de espalhamento
~T: double[][]
Matriz nodal de transformação por rotação
~rot: double[][]
Matrix de transformação por rotação do elemento
~kel: double[][]
Matriz de rigidez no sistema local
~fel_distribLoad: double[]
Vetor de reações de engastamento no sistema local para cargas distribuídas
~fel_thermalLoad: double[]
Vetor de reações de engastamento no sistema local para cargas térmicas
~axial_force: double[]
Vetor de esforços axiais nas extremidades
~shear_force_Y: double[]
Vetor de esforços cortantes na direção local y nas extremidades
~shear_force_Z: double[]
Vetor de esforços cortantes na direção local z nas extremidades
~bending_moment_Y: double[]
Vetor de momentos fletores em torno do eixo local y nas extremidades
~bending_moment_Z: double[]
Vetor de momentos fletores em torno do eixo local z nas extremidades
~torsion_moment: double[]
Vetor de momentos torsores nas extremidades
~intDispl: double[][]
Vetor de deslocamentos internos

+Elem()
Método construtor
+flexuralStiffCoeff_XY(): double[][]
Monta a matriz de coeficientes de rigidez a flexão no plano local xy
+flexuralStiffCoeff_XZ(): double[][]
Monta a matriz de coeficientes de rigidez a flexão no plano local xz
+flexuralDisplShapeFcnVector_XY(double x): double[]
Avalia as funções de forma para efeitos de flexão no plano local xy
+flexuralDisplShapeFcnVector_XZ(double x): double[]
Avalia as funções de forma para efeitos de flexão no plano local xz

+rotTransMtx(): double[][]
Calcula a matriz nodal de transformação por rotação
+gblStiffMtx(): double[][]
Calcula a matriz de rigidez do elemento no sistema global
+axialStiffCoeff(): double[][]
Monta a matriz de coeficientes de rigidez axiais
+torsionStiffCoeff(): double[][]
Monta a matriz de coeficientes de rigidez a torção
+gblAnlIntForce(Drv drv): double[]
Calcula os esforços internos da análise global
+axialDisplShapeFcnVector(double x): double[]
Avalia as funções de forma para efeitos axiais
+gblAnlIntDispl(Drv drv, double x): double[]
Calcula os deslocamentos internos da análise global
+clean(): void
Força liberação de memória utilizada pelas propriedades
    
```

Figura 2.14: Métodos e propriedades da classe Elem. Fonte: Martha, (2018)

- Lelem: Nessa classe existem objetos e propriedades associadas que refletem os carregamentos na barra. Dentre essas cargas, têm-se forças unitárias ou forças distribuídas e variações de temperatura que geram o carregamento;
- Anm: Classe abstrata para generalizar propriedades e comportamentos para análise de treliças 2D e 3D, pórticos 2D e 3D, ou grelhas. Ademais, suas propriedades e métodos para cada caso específico é definido nas respectivas classes que herdam a classe Anm;
- Print: Essa classe é responsável pela impressão textual de resultados. Também está relacionada com diversas subclasses designadas para conter métodos específicos, havendo, assim, a impressão coerente com a análise designada.

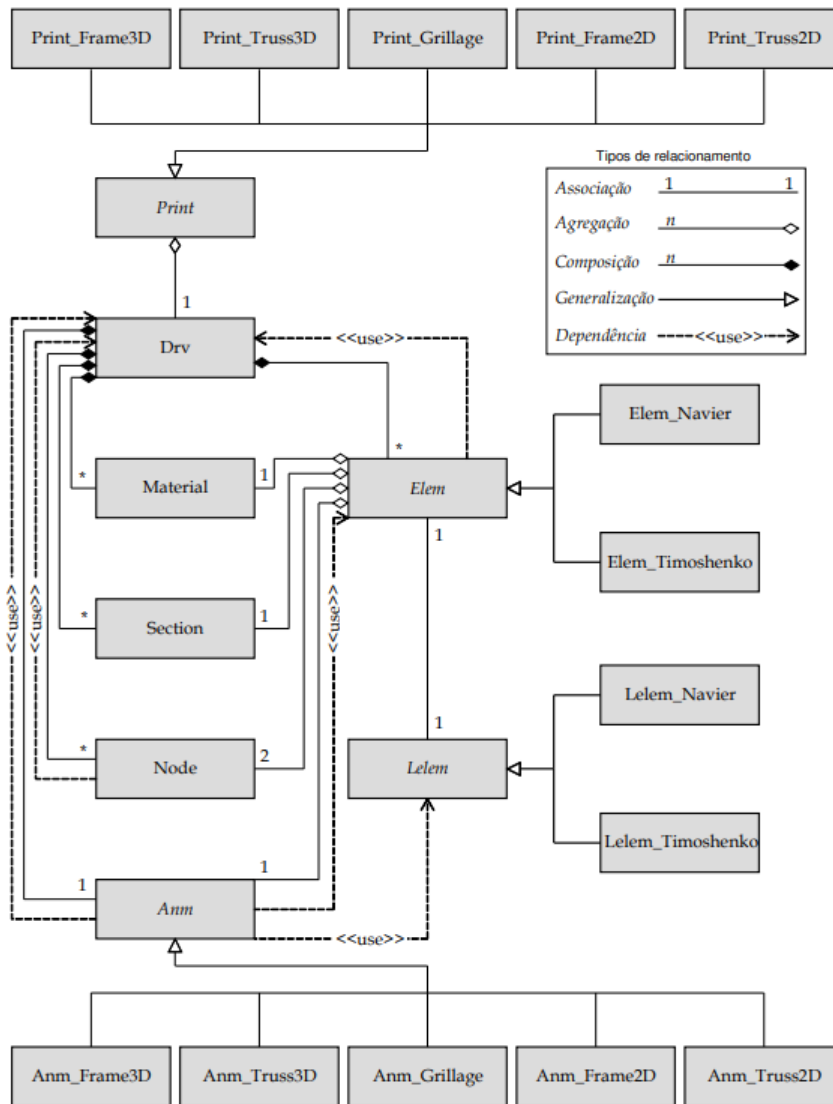


Figura 2.15: Diagrama de classes do programa LESM (versão não gráfica) para implementar uma análise linear-elástica de estruturas reticuladas. Fonte: Martha, (2018)

2.5

Diretrizes de dimensionamento metálico segundo a NBR8800:2008

Para que haja um entendimento prévio das etapas de verificação que serão utilizadas nesse documento, as seções deste capítulo irão sintetizar as diretrizes adotadas para o dimensionamento de perfis de aço tipo I para o estado do limite último utilizando como referência a ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS [12]

2.5.1

Critérios de garantia de segurança

Uma das premissas fundamentais para o desenvolvimento de projetos estruturais é garantir a segurança e integridade da estrutura. Referente a garantia de segurança de projetos metálicos, as normas disponíveis para tal utilizam duas metodologias: o método das tensões admissíveis, usado pela norma americana AISC, ou o método dos estados limites, utilizado pela NBR8800:2008.

Como a norma delimitada pelo escopo do projeto é a NBR8800:2008, a próxima seção será dedicada ao dimensionamento do método dos estados limites.

2.5.1.1

Método dos estados limites

Este método define que, para garantir uma segurança satisfatória da estrutura, a seguinte equação deve ser atendida:

$$Sd = S(\sum \gamma_i F_i) < R_d = R\left(\frac{f_k}{\gamma_m}\right) \quad (2-1)$$

Onde Sd é a solicitação de projeto, devendo ser menor que R_d , o qual refere-se à resistência de projeto. O valor da solicitação de projeto é obtida através da combinação

das ações F_i majorada pelo seu respectivo coeficiente γ_{fi} , no qual seu valor é diretamente ligado ao tipo do carregamento aplicado. Em relação a parcela da resistência do projeto, f_k é a função de resistência característica do material, o qual é minorado pelo coeficiente γ_m . Vale ressaltar que γ_m depende da propriedade mecânica do material.[14]

A formulação desse método é baseado em probabilidades geradas pelas ações atuantes. Os valores das ações a serem utilizadas nos cálculos podem se obtidos pelos seguintes processos:

- Critério estatístico, adotando-se valores característicos. Dessa forma, os valores das ações equivalem a uma probabilidade de serem ultrapassados.
- Critério determinístico, nesse critério os valores são dados por uma envoltória das solicitações criadas pelo carregamento atuante.

Como a aplicação do método estatístico não é trivial, os códigos normativos muitas vezes utilizam valores tabelados para a obtenção dos valores das ações. Assim, na obtenção dos ações, os valores probabilísticos tabelados são multiplicados pelo valor nominal do carregamento.

Outro ponto importante que a norma do estudo de caso, a NBR8800:2008, define são duas categorias de estado limite:

Estado limite último(ELU): definido para as combinações de ações no caso de a estrutura perder o equilíbrio e a estabilidade. Caso a estrutura ultrapasse esse limite, a estrutura pode sofrer rupturas, flambagens, esmagamentos, entre outras avarias.

Estado limite de serviço(ELS): representa o limite que a estrutura não deve atingir para que seu desempenho estrutural não seja comprometido pelos carregamentos nominais. Um exemplo de falha desse limite são grandes deformações geradas.

Para cada um desses estados é utilizado um tipo de combinação de ações. Porém, como o escopo é limitado apenas ao estado limite último, a próxima seção trará mais detalhes apenas deste estado-limite.

2.5.2

Combinação de ações últimas

Como dito anteriormente, para encontrar a solicitação de projeto, deve-se realizar combinações estatísticas de atuação de cargas na estrutura a fim de determinar o caso mais crítico e compará-lo ao valor resistente encontrado. Para isso, deve-se realizar comutações entre as cargas nominais atuantes, multiplicando seus respectivos fatores de ponderação estipulados pela norma. Vale ressaltar que esses fatores são utilizados, pois nem todas as solicitações acontecem necessariamente de forma concomitante, sendo assim, possuem uma probabilidade de ação combinada.

A norma estipula coeficientes de ponderação para dois tipo de cargas, as permanentes e as variáveis. Esses fatores são estipulados de acordo com a definição da carga e o tipo de combinação, apresentados pelas Tabelas 2.1 e 2.2.

Tabela 2.1: Coeficientes de ponderação para ações permanentes e variáveis

Combinações	Ações permanentes (γ_g)					
	Diretas					Indiretas
	Peso próprio de estruturas metálicas	Peso próprio de estruturas pré-moldadas	Peso próprio de estruturas moldadas no local e de elementos construtivos industrializados e empuxos permanente	Peso próprio de elementos construtivos industrializados com adições in loco	Peso próprio de elementos construtivos em geral e equipamentos	
Normais	1,25 (1,00)	1,30 (1,00)	1,35 (1,00)	1,40 (1,00)	1,50 (1,00)	1,20 (0)
Especiais ou de construção	1,15 (1,00)	1,20 (1,00)	1,25 (1,00)	1,30 (1,00)	1,40 (1,00)	1,20 (0)
Excepcionais	1,10 (1,00)	1,15 (1,00)	1,25 (1,00)	1,20 (1,00)	1,30 (1,00)	0 (0)
	Ações variáveis (γ_q)					
	Efeito de temperatura	Ação do vento	Ações truncadas	Demais ações variáveis, incluindo as decorrentes do uso e ocupação		
Normais	1,20	1,40	1,20	1,50		
Especiais ou de construção	1,00	1,20	1,10	1,30		
Excepcionais	1,00	1,00	1,00	1,00		

Tabela 2.2: Coeficientes de ponderação estatístico de acordo com a ação variável

Ações		γ_{12}		
		ψ_0	ψ_1	ψ_2
Ações variáveis causadas pelo uso e ocupação	Locais em que não há predominância de pesos e de equipamentos que permanecem fixos por longos períodos de tempo, nem de elevadas concentrações de pessoas	0,5	0,4	0,3
	Locais em que há predominância de pesos e de equipamentos que permanecem fixos por longos períodos de tempo, ou de elevadas concentrações de pessoas	0,7	0,6	0,4
	Bibliotecas, arquivos, depósitos, oficinas e garagens e sobrecargas em coberturas	0,8	0,7	0,6
Vento	Pressão dinâmica do vento nas estruturas em geral	0,6	0,3	0
Temperatura	Variações uniformes de temperatura em relação à média anual local	0,6	0,5	0,3
Cargas móveis e seus efeitos dinâmicos	Passarelas de pedestres	0,6	0,4	0,3
	Vigas de rolamento de pontes rolantes	1,0	0,8	0,5
	Pilares e outros elementos ou subestruturas que suportam vigas de rolamento de pontes rolantes	0,7	0,6	0,4

As combinações últimas possuem três classificações: normal, especial e excepcional. Para esse escopo, relata-se apenas o caso das normais, cuja formulação é dada por:

$$Sd = \sum_{i=1}^m (\gamma_{gi} F_{Gi,k}) + \gamma_{q1} F_{Q1,k} + \sum_{j=2}^n (\gamma_{qj} \Psi_{0j} F_{Qj,k}) \quad (2-2)$$

sendo:

$F_{Gi,k}$ o valor característicos das cargas permanentes;

$F_{Q1,k}$ o valor característico da carga variável principal;

$F_{Qj,k}$ o valor característico da carga variável secundária;

γ o Fator de ponderação principal;

Ψ o Fator de ponderação secundário.

A equação descrita acima deve ser formulada com todas as combinações possíveis ou necessárias para garantir que o caso solicitante mais crítico seja encontrado. Dessa forma, pode-se utilizá-lo como referência comparativa à

resistência encontrada para cada tipo de esforço que um perfil metálico pode ser submetido.

2.5.3 Dimensionamento de perfil I

Após obter o valor solicitante de projeto, deve-se verificar se os perfis utilizados no projeto conseguem suportar esta solicitação. Para isso, calcula-se a resistência desses perfis para cada tipo de esforço, sendo eles: tração, flexão, compressão, cisalhamento, flexo compressão e flexo tração.

Para encontrar esse valor resistente, a norma NBR8800:2008 apresenta diretrizes sugeridas a serem seguidas para cada tipo de esforço citado com o intuito de facilitar e garantir boas práticas na obtenção desses dados. Sendo assim, a seguir será apresentado um compilado dessas diretrizes, delimitado pelo escopo do documento.

2.5.3.1 Tração

A NBR8800:2008 estipula que, para perfis tracionados, com exceção das barras arredondadas rosqueadas, o $N_{t,Rd}$ utilizado deve ser o menor valor obtido entre o limite último de escoamento da seção bruta e a ruptura da seção líquida.

Para obter o valor resistente do escoamento da seção bruta, a norma estabelece a seguinte equação:

$$N_{t,Rd} = \frac{A_g * f_y}{\gamma_{a1}} \quad (2-3)$$

onde:

A_g representa a área bruta transversal da barra;

f_y representa Resistência ao escoamento do aço

Para o valor resistente da seção líquida temos:

$$N_{t,Rd} = \frac{A_e * f_u}{\gamma_{a2}} \quad (2-4)$$

A_e representa área líquida transversal da barra;

f_u representa a resistência a ruptura do aço.

Porém para efetuar o cálculo da acima devemos obter o valor da área efetiva A_e dado por:

$$A_e = C_t * A_n$$

sendo:

A_n o valor da área líquida da barra;

C_t o valor do coeficiente de redução da área líquida.

Como os valores de A_n e C_t são diretamente ligados ao detalhamento do tipo da ligação metálica, essa verificação não será contemplada no *software*, como dito anteriormente no escopo. Assim, não haverá maior detalhamento sobre essa verificação.

Após a obtenção do menor valor entre o limite último de escoamento da seção bruta e a ruptura da seção líquida, a seguinte diretriz deve ser atendida:

$$N_t, R_d \geq N_t, s_d \quad (2-5)$$

2.5.3.2

Flexão

Para realizar a verificação dos perfis à flexão, utiliza-se o anexo G da NBR8800:2008. Como o escopo da dissertação limita-se a perfis do tipo I e H, todas as equações demonstradas nessa seção serão referentes a eixos de dupla simetria, dada pelo anexo G.

- Para flexão em relação ao eixo de maior inércia, faz-se necessário analisar 3 estados limites:
 - 1) FLT: Flambagem lateral com torção.
 - 2) FLM: Flambagem Local da Mesa.
 - 3) FLA: Flambagem Local da Alma.

Flexão em relação ao eixo de maior inércia para FLT

Para o estado limite FLT, são necessários cinco parâmetros, dados por:

$$\lambda = \frac{L_b}{r_y} \quad (2-6)$$

$$\lambda_p = 1,76 * \sqrt{\frac{E}{f_y}} \quad (2-7)$$

$$\lambda_r = \frac{1,38\sqrt{I_y * J}}{r_y * J * \beta_1} \sqrt{1 + \sqrt{1 + \frac{27 * C_w * \beta_1^2}{I_y}}} \quad (2-8)$$

$$M_r = (f_y - \sigma_r)W \quad (2-9)$$

$$M_{cr} = \frac{C_b \pi^2 E I_y}{L_b^2} \sqrt{\frac{C_w}{I_y} \left(1 + 0,039 \frac{J L_b^2}{C_w}\right)} \quad (2-10)$$

Sendo:

L_b referente ao comprimento destravado;

r_y referente ao raio de giração do eixo;

F_y referente a resistência ao escoamento do aço.

O coeficiente β_1 é elucidado na nota 1 da tabela G1, dado por:

$$\beta_1 = (fy - \sigma_r) \quad (2-11)$$

onde:

σ_r é referente a tensão residual do aço, 30% de F_y de acordo com a NBR8800:2008

O coeficiente C_b é calculado conforme a equação a seguir:

$$C_b = \frac{12,5M_{max}}{2,5M_{max} + 3M_a + 4M_b + 3M_c} * R_m \leq 3 \quad (2-12)$$

Sendo:

Max o Momento máximo de cálculo em modulo;

Ma o Momento solicitante de cálculo situado a um quarto do vão, a partir da esquerda;

Mb o Momento solicitante de cálculo no meio do vão;

Mc o Momento fletor solicitante a três quartos do vão, a partir da esquerda;

Rm o Fator de monossimetria. Para esse caso, adotou-se o valor 1, visto que apenas a viga I, duplamente simétrica, está em estudo.

Após obter os cinco fatores listados anteriormente, deve-se consultar a seção G.2 da NBR8800:2008, a qual contém formulações para definição do valor do momento resistente de cálculo para FLT. Ademais, a escolha do uso dessas equações é dada por condicionais relacionadas aos valores de esbeltez do elemento estrutural a em questão.

Se $\lambda \leq \lambda_p$, M_{rd} é dado por:

$$M_{rd} = \frac{M_{pl}}{\gamma_{a1}} \quad (2-13)$$

Se $\lambda_p < \lambda \leq \lambda_r$,

$$M_{rd} = \frac{C_b}{\gamma_{a1}} [M_{pl} - (M_{pl} - M_r) \left(\frac{\lambda - \lambda_p}{\lambda_r - \lambda_p} \right)] \leq \frac{M_{pl}}{\gamma_{a1}} \quad (2-14)$$

Se $\lambda > \lambda_r$,

$$M_{rd} = \frac{M_{cr}}{\gamma_{a1}} \leq \frac{M_{pl}}{\gamma_{a1}} \quad (2-15)$$

Flexão em relação ao eixo de maior inércia para FLM

Para o estado limite FLM, os fatores usados nesse estado limite são dados por:

$$\lambda = \frac{b}{t} \quad (2-16)$$

$$\lambda_p = 0.38 * \sqrt{\frac{E}{fy}} \quad (2-17)$$

$$\lambda_r = 0.83 * \sqrt{\frac{E}{(fy - \sigma_r)}} \quad (2-18)$$

$$Mr = (fy - \sigma_r)W \quad (2-19)$$

$$Mcr = \frac{0,69E}{\lambda^2} W_c \quad (2-20)$$

Para o cálculo do momento resistente para FLM, tem-se as seguintes formulações:

Se $\lambda \leq \lambda_p$, Mrd é dado por:

$$Mrd = \frac{Mpl}{\gamma a1} \quad (2-21)$$

Se $\lambda_p < \lambda \leq \lambda_r$,

$$Mrd = \frac{1}{\gamma a1} [Mpl - (Mpl - Mr) \left(\frac{\lambda - \lambda_p}{\lambda_r - \lambda_p} \right)] \leq \frac{Mpl}{\gamma a1} \quad (2-22)$$

Se $\lambda > \lambda_r$,

$$Mrd = \frac{Mcr}{\gamma a1} \quad (2-23)$$

Flexão em relação ao eixo de maior inércia para FLA

Para o estado limite FLA, tem-se as seguintes expressões de coeficientes:

$$\lambda = \frac{h}{tw} \quad (2-24)$$

$$\lambda_p = 3,76 * \sqrt{\frac{E}{fy}} \quad (2-25)$$

$$\lambda_r = 5,76 * \sqrt{\frac{E}{fy}} \quad (2-26)$$

$$Mr = fyW$$

Para o cálculo do momento resistente, tem-se:

Todas as equações referentes ao momento resistente utilizado em FLM podem ser adotadas, exceto no caso em que $\lambda < \lambda_r$, visto que refere-se a vigas de alma esbelta. Como esse caso não pertence ao escopo, o usuário será notificado caso o perfil se enquadre e será indicado o uso do anexo H da norma NBR8800:2008.

Flexão em relação ao eixo de menor inércia

Ademais, para o caso do eixo de menor inércia, tem-se ainda:

Para o eixo de menor inércia em seções I e H com dois eixos de simetria, a norma estipula apenas a verificação de FLM, na qual os coeficientes são dados por:

$$\lambda = \frac{b}{t} \quad (2-27)$$

$$\lambda_p = 0.38 * \sqrt{\frac{E}{fy}} \quad (2-28)$$

$$\lambda_r = 0.83 * \sqrt{\frac{E}{(fy - \sigma_r)}} \quad (2-29)$$

Se $\lambda \leq \lambda_p$, M_{rd} é dado por:

$$M_{rd} = \frac{M_{pl}}{\gamma_{a1}} \quad (2-30)$$

Se $\lambda_p < \lambda \leq \lambda_r$,

$$M_r = \frac{1}{\gamma_{a1}} [M_{pl} - (M_{pl} - M_r) \frac{\lambda - \lambda_p}{\lambda_r - \lambda_p}] \leq \frac{M_{pl}}{\gamma_{a1}} \quad (2-31)$$

Se $\lambda > \lambda_r$,

$$M_{rd} = \frac{M_{cr}}{\gamma_{a1}} \quad (2-32)$$

2.5.3.3

Compressão

Segundo essa norma, a fórmula para carga crítica axial é dada por:

$$N_{c,rd} = \frac{\chi Q A g f_y}{\gamma_{a1}} \quad (2-33)$$

Como definido pela norma, deve-se ainda calcular os fatores de redução.

O fator χ , associado a resistência a compressão, é dado por:

- Para:

$$\lambda_0 \leq 1,5 : \chi = 0,658^{\lambda_0^2} \quad (2-34)$$

- Para:

$$\lambda_0 > 1,5 : \chi = \frac{0,877}{\lambda_0^2} \quad (2-35)$$

Visto que o fator χ é dependente do valor da esbeltez reduzida λ_0 , faz-se necessário, previamente, calculá-lo por meio da fórmula:

$$\lambda_0 = \sqrt{\frac{Q A g f_y}{N_e}} \quad (2-36)$$

Para tal cálculo, deve-se encontrar dois fatores, N_e e Q , denominados, respectivamente, força axial de flambagem elástica e fator de redução total associado a flambagem local.

No anexo E da norma, para perfis duplamente simétricos, são especificados algumas formulações:

- Flambagem por flexão em relação ao eixo central x;

$$N_{ex} = \frac{\pi^2 EI_x}{(K_x L_x)^2} \tag{2-37}$$

- Flambagem por flexão em relação ao eixo central y;

$$N_{ey} = \frac{\pi^2 EI_y}{(K_y L_y)^2} \tag{2-38}$$

- Flambagem por torção ao eixo longitudinal z.

$$N_{ez} = \frac{1}{r_0^2} \left[\frac{\pi^2 EC_w}{(K_z L_z)^2} + GJ \right] \tag{2-39}$$

Sendo:

I_x, I_y os valores de inércia para os eixos x e y respectivamente;

C_w a constante de empenamento;

r_0 o raio de giração da seção bruta, definido como:

$$r_0 = \sqrt{(r_x^2 * r_y^2 * x_0^2 * y_0^2)} \tag{2-40}$$

Onde:

r_x e r_y são os raios de giração do perfil em relação ao seus eixos centrais respectivamente;

x_0 e y_0 são as coordenadas do centro de cisalhamento na direção dos eixos centrais x e y.

$K_x L_x$ e $K_y L_y$ referem-se ao comprimento de flambagem do eixo x e y, respectivamente, multiplicado pelo fator condicional do contorno, dado pela tabela E.1 da norma, reproduzida na Figura 2.16.





<p>A linha tracejada indica a linha elástica de flambagem</p>	(a)	(b)	(c)	(d)	(e)	(f)	
	Valores teóricos de K_x ou K_y	0,5	0,7	1,0	1,0	2,0	2,0
	Valores recomendados	0,65	0,80	1,2	1,0	2,1	2,0
Código para condição de apoio	   	Rotação e translação impedidas Rotação livre, translação impedida Rotação impedida, translação livre Rotação e translação livres					

Figura 2.16: Fatores condicionais de contorno. Fonte: ABNT, (2008)

$K_z L_z$ refere-se ao comprimento de flambagem por torção, no qual o fator k_z é definido pela seção E.2.2, que estipula o valor 1 ou 2, dependendo das condições de contorno da viga. Sendo, respectivamente:

- a) Quando ambas as extremidades da barra possuírem rotação em torno do eixo longitudinal impedida e empenamento livre.
- b) Quando uma das extremidades da barra possuir rotação em torno do eixo longitudinal e empenamento livres e, a outra extremidade, rotação e empenamento impedidos [10]

Após o cálculo de N_{ex} , N_{ey} e N_{ez} , deve-se escolher o menor valor entre eles para prosseguir com a obtenção de λ_0 .

Sequencialmente, deve-se obter o fator Q (fator de redução total associado a flambagem local). Esse fator é composto por dois fatores, Q_s e Q_a . Sendo sua fórmula dada por:

$$Q = Q_a * Q_s \tag{2-41}$$

Obtenção do fator Q_s :

Primeiramente, deve-se analisar a razão b/t do elemento AL (Apoiado e Livre) e compará-lo a esbeltez limite para esse perfil. No caso em estudo, são perfis I e H, contemplados no grupo 4 da tabela F.1 da NBR8800:2008, retratada na Figura 2.17.

4	<ul style="list-style-type: none"> — Mesas de seções I, H, T ou U laminadas — Abas de cantoneiras ligadas continuamente ou projetadas de seções I, H, T ou U laminadas ou soldadas — Chapas projetadas de seções I, H, T ou U laminadas ou soldadas 		$0,56 \sqrt{\frac{E}{f_y}}$
---	--	--	-----------------------------

Figura 2.17: Esbeltez limite para os elementos AL. Fonte: ABNT, (2008)

É importante frisar que a esbeltez limite é dada por $0,56 \sqrt{\frac{E}{f_y}}$. Caso a esbeltez do elemento analisado for menor que a limite, adota-se o valor 1 para Q_s . Caso contrário, deve-se adotar a seguinte abordagem para o grupo 4:

Para $0,56 \sqrt{\frac{E}{f_y}} < \frac{b}{t} \leq 1,03 \sqrt{\frac{E}{f_y}}$ adota-se:

$$Q_s = 1,415 - 0,74 \frac{b}{t} \sqrt{\frac{f_y}{E}} \tag{2-42}$$

Para $\frac{b}{t} > 1,03\sqrt{\frac{E}{f_y}}$

$$Q_s = \frac{0,69E}{f_y(\frac{b}{t})^2} \quad (2-43)$$

:

Obtenção do fator Q_a :

Primeiro, deve-se verificar a esbelteza limite para os elementos AA (Apoiado e Apoiado) de acordo com a Figura 2.18.

2	<ul style="list-style-type: none"> — Almas de seções I, H ou U — Mesas ou almas de seção-caixão — Todos os demais elementos que não integram o Grupo 1 		$1,49\sqrt{\frac{E}{f_y}}$
---	---	--	----------------------------

Figura 2.18: Esbelteza limite para os elementos AA. Fonte: ABNT, (2008)

Se a razão entre b/t for inferior a $1,49\sqrt{\frac{E}{f_y}}$, adota-se o valor 1 para Q_a . Caso contrário, deve-se seguir a seguinte recomendação da norma:

$$Q_a = \frac{A_{ef}}{A_g} \quad (2-44)$$

Onde:

A_{ef} é a área efetiva da seção, dada por:

$$A_{ef} = A_g - \sum (b - b_{ef})t \quad (2-45)$$

Sendo:

b refere-se a largura do elemento comprimido;

t é a espessura do elemento comprimido;

b_{ef} refere-se a largura efetiva do elemento comprimido, dada pela seguinte formulação:

$$b_{ef} = 1,92t\sqrt{\frac{E}{\sigma}}\left[1 - \frac{C_a}{\frac{b}{t}}\sqrt{\frac{E}{\sigma}}\right] \leq b \quad (2-46)$$

Para o coeficiente C_a adota-se 0,34 para todos os perfis não tubulares. Em relação ao termo σ , pode-se ter duas abordagens:

- Calcular utilizando χ e adotando $Q=1$ para seu cálculo.

$$\sigma = \chi f_y \quad (2-47)$$

- Cálculo mais conservador, o qual será adotado nas rotinas do programa, sendo:

$$\sigma = fy \quad (2-48)$$

Após a coleta de todo esses fatores, retoma-se a equação (2.25) para obter o valor de força axial resistente do perfil.

2.5.3.4 Cisalhamento

- Perfil fletido em relação ao eixo perpendicular à alma

Para definir a força resistente de cálculo é necessário definir previamente alguns fatores que irão compor a formulação, sendo eles:

$$\lambda = \frac{h}{tw} \quad (2-49)$$

$$\lambda_p = 1,10 \sqrt{\frac{kvE}{fy}} \quad (2-50)$$

$$\lambda_r = 1,37 \sqrt{\frac{kvE}{fy}} \quad (2-51)$$

Para kv: Nesse caso, assume-se cinco, visto que o escopo não contempla enrijecedores.

Após a obtenção desses fatores, encontra-se a formulação adequada, utilizando as seguintes condicionais:

- Para $\lambda \leq \lambda_p : Vrd = \frac{Vpl}{\gamma a1}$ (2-52)

- Para $\lambda_p < \lambda \leq \lambda_r : Vrd = \frac{\lambda_p Vpl}{\lambda \gamma a1}$ (2-53)

- Para $\lambda > \lambda_r : Vrd = 1,24 \left(\frac{\lambda_p}{\lambda}\right)^2 \frac{Vpl}{\lambda a1}$ (2-54)

Para encontrar o fator Vpl, referente a força cortante de plastificação da alma, utiliza-se a seguinte formulação:

$$Vpl = 0,60 A_w F_y \quad (2-55)$$

Onde:

Aw representa a área efetiva de cisalhamento, equivalente a:

$$A_w = dtw \quad (2-56)$$

d representa a altura da seção transversal;

tw representa a espessura da alma.

- Perfil fletido em relação ao eixo perpendicular as mesas

Para este caso específico utilizam-se as mesmas diretrizes adotadas nas equações 2-49, 2-50 e 2-51. Contudo, nas equações dependentes de kv, utiliza-se 1,2 para este coeficiente.

Após a obtenção dos lambdas, o procedimento segue para determinar a força cortante resistente de cisalhamento através das equações 2-52, 2-53 e 2-54. Para encontrar o valor de V_{pl} , utiliza-se a mesma fórmula 2-55, porém a área efetiva de cisalhamento é dada por:

$$A_w = 2b_f * t_f \quad (2-57)$$

2.5.3.5

Flexo Compressão e Flexo Tração

Esses fenômenos fazem alusão a combinação de esforços, sendo cargas axiais combinadas com flexão no eixo x e y. Contudo, as cargas axiais podem ser de tração ou compressão. Assim, deve-se verificar essas duas variações para as seguintes formulações:

se $\frac{N_{sd}}{N_{rd}} \geq 0, 2$:

$$\frac{N_{sd}}{N_{rd}} + \frac{8}{9} \left(\frac{M_{x, sd}}{M_{x, rd}} + \frac{M_{y, sd}}{M_{y, rd}} \right) \leq 1, 0 \quad (2-58)$$

se $\frac{N_{sd}}{N_{rd}} < 0, 2$:

$$\frac{N_{sd}}{2N_{rd}} + \left(\frac{M_{x, sd}}{M_{x, rd}} + \frac{M_{y, sd}}{M_{y, rd}} \right) \leq 1, 0 \quad (2-59)$$

Sendo:

N_{sd} refere-se a força solicitante de compressão ou de tração;

N_{rd} é a força resistente de cálculo de compressão ou tração;

$M_{x, Sd}$ e $M_{y, Sd}$ são os momentos fletores de cálculo para o eixo x e y, respectivamente;

$M_{x, Rd}$ e $M_{y, Rd}$ são os momentos fletores resistentes para o eixo x e y, respectivamente.

3 Metodologia

Neste capítulo será descrito como foi realizado todo o processo de implementação da nova funcionalidade no programa LESM. Para tal, foi necessário explorar modelos de processos de construção de *softwares*, visando uma maior ordenação e disciplina em seu desenvolvimento. Dessa forma, após algumas pesquisas sobre modelos que se encaixassem melhor ao projeto, determinou-se o uso do modelo cascata. Esse modelo consiste em definir pontos de controle, os quais devem ser executados e seguidos de maneira rigorosa para que o processo seja confiável e preciso, independente do tamanho da aplicação. Por ser um processo muito estrito, cada subprocesso dentro dos pontos de controle devem ser muito bem executados para que hajam poucos retrocessos entre os pontos, mesmo que ainda seja permitida essa prática.[7].

Visto isso, foram definidos pontos de controle para a construção do *software* dado pela Figura 3.1:

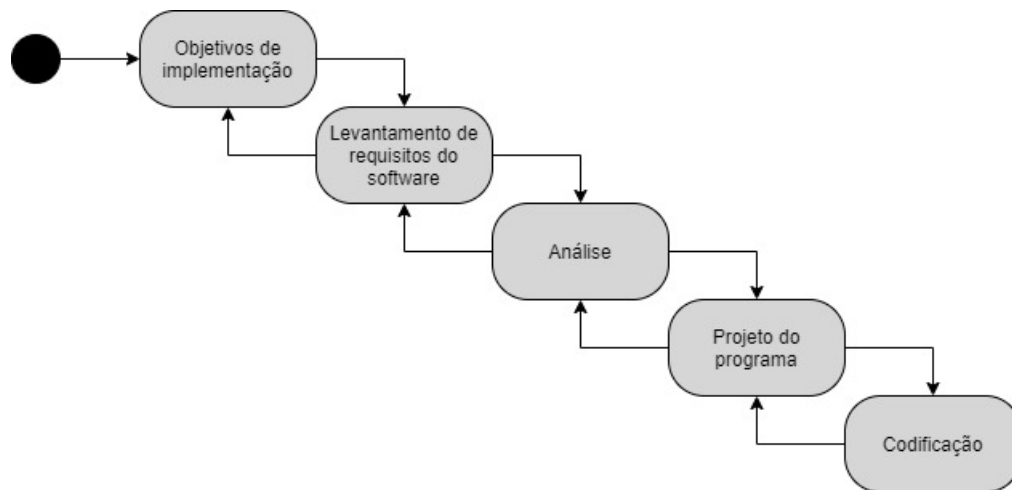


Figura 3.1: Interação da metodologia do modelo de cascata (imagem adaptada). Fonte: Wazlawick (2013)

Com o intuito de representar de forma clara todo o processo realizado em cada ponto de controle previamente definido, essa metodologia é descrita ao longo das próximas seções, nomeadas de acordo com o nome de cada ponto de controle utilizado.

3.1

Objetivos de Implementação

Esta fase foi responsável por criar um levantamento sobre todas as funcionalidades que se pretendia inserir na nova extensão do *software*. Essa etapa foi de suma importância, visto que todos os pontos definidos nesse momento serviram de embasamento para se obter uma métrica mais específica sobre qual é a real extensão do projeto em número de funcionalidades. Contudo, antes de enumerá-las, levantou-se quais objetivos gerais esperava-se atingir com a nova extensão e, a partir dos objetivos, definiu-se as novas funcionalidades necessárias para atingí-los.

Com isso, foram definidos dois objetivos principais: a verificação de perfis metálicos, conforme a norma NBR8800:2008, e a forma como os resultados obtidos são demonstrados, visando uma representação mais lúdica aos usuários, com um enfoque na didática para os alunos de graduação.

Ademais, para uma maior assertividade no levantamento das funcionalidades, foram elaboradas ainda duas perguntas, utilizando os próprios objetivos como base, a serem satisfeitas pela nova extensão. Sendo assim as perguntas e suas funcionalidades ficaram dispostas da seguinte maneira:

- Quais requisitos técnicos devem ser inseridos para dimensionar perfis I?
 - 1) Disponibilizar bibliotecas de perfis I e H da Gerdau para o usuário na modelagem da estrutura.
 - 2) Disponibilizar a possibilidade de rotação dos perfis.
 - 3) Disponibilizar a biblioteca com os tipos de aço a serem usados na modelagem.
 - 4) Disponibilizar a possibilidade de tipificação para caso de carga.
 - 5) Disponibilizar a associação de coeficientes aos casos de carga para suas combinações.
 - 6) Disponibilizar a combinação automática de casos de carga seguindo os preceitos da NBR8800.
 - 7) Disponibilizar a criação de estruturas (grupos) de vigas ou de colunas.
 - 8) Disponibilizar a verificação de perfis automática para estado limite último para (tração, flexão, compressão, cisalhamento, flexo compressão e flexo tração) em estruturas caracterizadas como pórtico 2D, seguindo as diretrizes da NBR8800.

- 9) Disponibilizar a visualização sobre quais perfis passam ou falham nas verificações de esforços realizadas.
- 10) Disponibilizar a visualização sobre o fator de segurança obtido para cada verificação realizada ao longo das seções da barra.
 - Quais requisitos didáticos devem ser inseridos?
 - 1) Disponibilizar um memorial de cálculo bem estruturado e interativo que demonstre todo o procedimento realizado na verificação de perfis, referenciando a seção da norma para cada processo.
 - 2) Disponibilizar opções de ajuda no momento de definição de carga, segundo a NBR8800:2008, de modo que, após a escolha das descrições do tipo, o coeficiente da carga seja atribuído de maneira automática.
 - 3) Disponibilizar uma visualização clara sobre a seção que falhou no perfil com contraste de cores.

3.2

Levantamento de requisitos do software

Para realizar esse ponto de controle, foram enumeradas alterações que devem ser feitas na versão atual do *software* LESM, visto que, quando realiza-se alguma adição de funcionalidade, pode ser que o *software* ainda não esteja preparado para recebê-la.

Contudo, antes de enumerar as alterações necessárias, houve uma análise do que poderia ser reutilizado da versão atual e o que iria necessitar de mudanças. Para tal, utilizou-se como referência todas as novas funcionalidades a serem inseridas no programa, enumeradas na Seção 3.1.

Primeiramente, foi observado se na *interface* de navegação da versão atual do LESM era disponibilizado algum tipo de classificação de parâmetros que pode ser associado na modelagem dos elementos de barra, como material e seção. Notou-se que, para os materiais e seções, não existia nenhum tipo de GUI (*Graphical user interface* - Interface gráfica do utilizador) disponibilizando a lista de alguns tipos utilizados no mercado, possuindo apenas a possibilidade de o usuário introduzir manualmente as características da barra modelada, como pode ser observados nas Figuras 3.2 e 3.3.

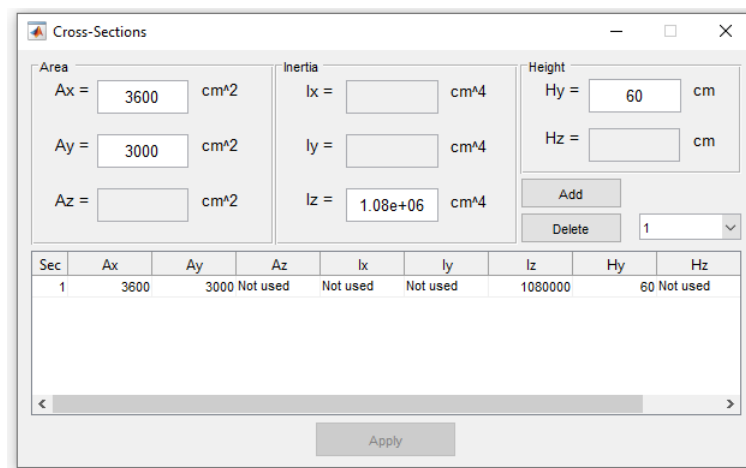


Figura 3.2: GUI de propriedades geométricas da seção. Fonte: Autor (2021)

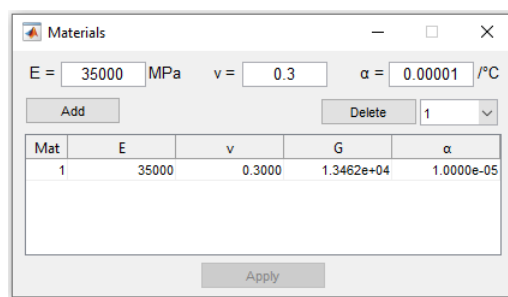


Figura 3.3: GUI de propriedades dos materiais. Fonte: Autor (2021)

Outro ponto observado na usabilidade do *software*, foi a inexistência de GUIs que permitam ao usuário a caracterização dos casos de cargas (variáveis e permanentes) e a criação de grupos de vigas e colunas. Além disso, o programa também não disponibiliza nenhuma possibilidade de verificação dos perfis, assim como a visualização de seus resultados. Dessa forma, faz-se necessário criar novas GUIs, possibilitando a interação do usuário com a nova funcionalidade e a coleta das informações registradas, para que o *software* as utilize na verificação dos perfis metálico. Devido a essas inserções, fica implícito que a estrutura de dados interna do código também sofrerá alterações.

Dessa forma, foram definidos pontos a serem modificados ou criados para atribuir a nova extensão ao programa:

- 1) Criação da GUI para seleção de perfil;
- 2) Criação da GUI para seleção do material;
- 3) Criação da GUI para caracterização do tipo de carga;

- 4) Criação da GUI para criação de grupos de vigas e colunas;
- 5) Criação da GUI para visualização do coeficiente de segurança ao longo da viga;
- 6) Modificação da estrutura de dados para introdução de casos de carga;
- 7) Modificação na estrutura de dados para criação de barras;
- 8) Modificação na estrutura de dados para introdução de propriedades geométrica das barras;
- 9) Modificação na estrutura de dados para introdução de materiais na barra;
- 10) Modificação na estrutura de dados para suporte a rotina de cálculos de verificação.

Cada ponto levantado será descrito com maior detalhamento nas seções seguintes, com intuito de demonstrar como foram implementadas tais criações ou modificações ao longo da construção do programa.

3.3 Análise

Após enumerar os requisitos do sistema e do *software*, passe-se para etapa de análise. Nessa parte, apresentam-se soluções de implementação aos tópicos citados, trazendo à tona detalhes das tomadas de decisões realizadas para cada um deles. Dessa forma, visa-se alinhar de forma clara todos os processos, métodos e até mesmo tecnologias utilizadas. Esse processo é de suma importância, pois, após a conclusão desta etapa, pode-se dar início ao desenvolvimento da nova estrutura do *software*.

3.3.1 Modificação na estrutura de dados

3.3.1.1 Drv

Antes de realizar as alterações necessárias para a nova funcionalidade, observou-se a existência de um estudo anterior do mesmo *software* com a seguinte titulação **Extensão de programa gráfico para análise de vibrações de modelos estruturais reticulados**[9]. Na pesquisa em questão é descrito como foi introduzida a análise de vibrações em estruturas reticuladas. Analisando este artigo, nota-se que houveram algumas alterações em relação

ao diagrama de classes, mostrada na Figura 2.15. Assim, para evitar uma incoerência entre as versões, foi realizada uma análise dessas alterações.

A priori, as modificações mais significativas são em relação a estrutura da classe *Drv*. As funções principais da classe foram divididas, alocando informações referentes a estrutura modelada, como materiais, seções, nós, elementos de barra e etc em instâncias de uma classe denominada *Model*. Em relação ao método da rigidez direta, a *Drv* tornou-se responsável apenas por essa tarefa, sendo abstraída e dividida em subclasses de acordo com o tipo de análise. Assim, criou-se o *Drv_LES*, responsável somente pela análise linear elástica e as classes *Solver* que possuem funções auxiliares utilizados em sua respectiva *Drv*.

Após a análise, observou-se ainda a falta de um estrutura que armazene o resultado dos esforços para diversos casos de carga, visto que, na versão atual do LESM, têm-se apenas os valores dos esforços para o caso de carga corrente. Assim, criou-se uma nova classe, denominada *Drv_LES_Design*. Nessa classe, a fatoração da matriz, em que adquirem-se os deslocamentos, passou a ser realizada de maneira simultânea para todos casos de carga, não apenas para o corrente, apresentado pela seguinte expressão:

$$K.d = F \quad (3-1)$$

$$K.d_{[lc]} = F_{[lc]} \quad (3-2)$$

sendo:

K a matriz de rigidez;

d refere-se ao vetor de deslocamentos;

F refere-se ao vetor das forças;

$Lc = load\ case$ (caso de carga)

Pode-se observar que, agora, armazenam-se os valores de deslocamento para cada vetor de forças gerados pelos casos de carga.

3.3.1.2

Materiais

Foi observado que, na versão atual do LESM, os materiais são utilizados de maneira genérica, como pode-se observar na Figura 2.11. As propriedades disponíveis apenas definem as características físicas do material, sem que haja um controle do tipo desse material. Assim, o programa não evidencia se é um aço, uma madeira ou algum outro tipo de material.

A não determinação do tipo do material cria uma problemática para realização das verificações, pois, na engenharia, a natureza do material escolhido é muito importante, sendo um dos fatores utilizados para definir

a norma utilizada. Com isso, decidiu-se criar uma maneira de introduzir uma nova estrutura de dados escalável que permita disponibilizar uma lista de tipos específicos de materiais.

Para consolidar tal decisão, foi decidido extrair a classe Material com a finalidade de manter nessa classe apenas as propriedades comuns a todos os tipos de materiais, sendo elas: densidade, módulo de elasticidade, coeficiente de Poisson e tensão de escoamento. Em seguida, definiu-se que, para cada novo material a ser introduzido, é criada uma classe exclusiva com suas propriedades únicas (Tabela 3.1) e as demais propriedades são herdadas da classe Material. Tendo em vista que o escopo do projeto é sobre o perfil de aços, apenas a classe Steel é introduzida, seguindo a estrutura do novo conceito apresentado. Outro ponto, definido por ora, é que a lista de aços que será disponibilizada ao usuário serão os seguintes:

Tabela 3.1: Propriedades físicas dos aços disponíveis na nova versão do LESM

Denominação	Módulo de elasticidade (GPa)	Módulo de elasticidade transversal do aço (GPa)	Limite de escoamento (MPa)	Resistência a tração (MPa)	Poisson	Coefficiente de dilatação térmica	Densidade (g/cm ³)
ASTM A36	200	75	250	400	0.3	1.2*10 ⁻⁵	7.85
ASTM A572	205	75	345	450	0.3	1.2*10 ⁻⁵	7.80

3.3.1.3 Seções

Na versão atual, às seções associadas às barras possuem apenas propriedades necessárias para realizar o método da rigidez direta, apresentadas na Figura 2.12. Isso ocorre, pois, nesta versão, apenas são realizados o cálculo dos esforços nas barras. Com isso, dado que até então não existia diferença entre tipos de perfis, notou-se a necessidade de criar uma maneira de extrair o conceito de seção para as barras, visto que a norma NBR8800:2008 difere as diretrizes procedurais para a realização dos cálculos de esforços resistentes dependendo do tipo de perfil adotado.

Analogamente à seção anterior, extraiu-se a classe Section com o intuito de manter as propriedades comuns a todos tipos de perfis, independente de sua geometria. Para isso, as propriedades que a classe possuía foram mantidas. Porém, notou-se a necessidade de adicionar mais propriedades, comuns aos perfis, relevantes ao dimensionamento, como: área bruta, raio de giração, constante de torção, coeficiente de empenamento, módulo de resistência elástico, entre outras.

Após definir essa nova forma mais abrangente da classe *Section*, pensou-se em como introduzir as famílias de perfis I para o *software*. Com isso, definiu-se que, para introduzir um novo tipo de perfil à estrutura do programa, deve-se criar uma nova classe que contenha todas as características particulares da geometria do perfil em questão. Essa fase de introdução de um novo perfil será mais detalhada na seção de projeto do programa. Contudo, vale ressaltar que essa nova classe deverá obrigatoriamente herdar a classe *Section* a fim de reaproveitar suas propriedades e métodos.

Destaca-se que esse novo formato da estrutura de perfis, incluídas ao *software* LESM, foi baseado em um trabalho de conclusão de curso de Almeida [8].

Depois de definir o novo comportamento da classe *Section* e sua subclasse, foi decidido qual seria a lista de perfis disponíveis para utilização do usuário, com o intuito de deixá-los condizentes com os usados no mercado brasileiro. Dessa forma, disponibilizou-se a família do perfil I, dado pelo catálogo comercial da GUERDAU, na 5.1, ilustrada no Apêndice A.

3.3.1.4 Elementos de barra

No LESM, ao modelar um elemento de barra, automaticamente instanciam-se um objeto que possui todas os métodos e propriedades da classe *Elem*, apresentada na Figura 2.14. Ademais, vale ressaltar que, por trás desse processo, existe uma diretriz previamente imposta em relação aos limites do elemento modelado, dada pela seguinte definição: na versão atual do LESM um elemento é **sempre** delimitado por dois nós. Para exemplificar tal afirmação, a Figura 3.4 apresenta uma viga bi apoiada modelada no *software*.

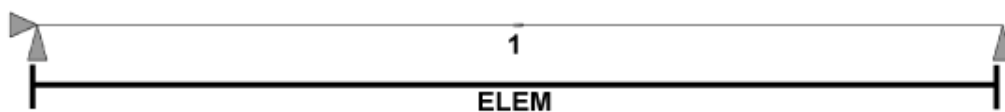


Figura 3.4: Elemento delimitado por dois nós. Fonte: Autor (2021)

Observando a Figura 3.4, nota-se que a instância *Elem* está delimitada pelos nós referentes aos apoios. Contudo, caso o usuário deseje inserir uma carga pontual no centro do elemento, o LESM define que, para tal ação, o usuário deve, obrigatoriamente, inserir previamente um novo nó no ponto onde deseja inserir a carga. Ao inserir o novo nó, o elemento se dividirá em duas

instâncias Elem. Dado a definição de delimitação das estâncias do tipo Elem, o modelo assumirá o aspecto mostrado na Figura 3.5.

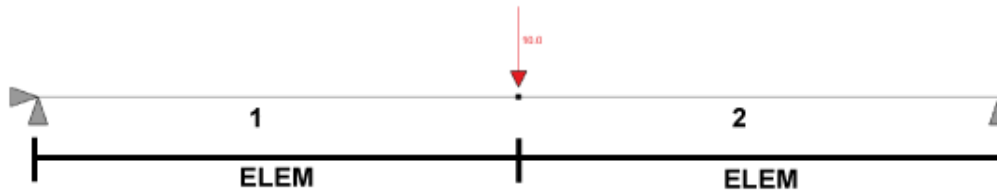


Figura 3.5: Elemento redividido após inserção de um nó. Fonte: Autor (2021)

Para o método da rigidez direta, esse tipo de organização é bem atendido para calcular os esforços no *software*. Contudo, ao dimensionar a estrutura, faz-se necessário indicar ao programa que esses dois elementos são, na realidade, uma única viga. Pensando nessa problemática, observou-se a necessidade de se acrescentar uma nova classe, análoga à Elem, denominada `Elems_Structural_Group`. Essa classe ficará responsável por receber os índices das instâncias Elem como uma única entidade, a ser dimensionada. Essa nova dinâmica de modelagem assumirá a forma mostrada na Figura 3.6.

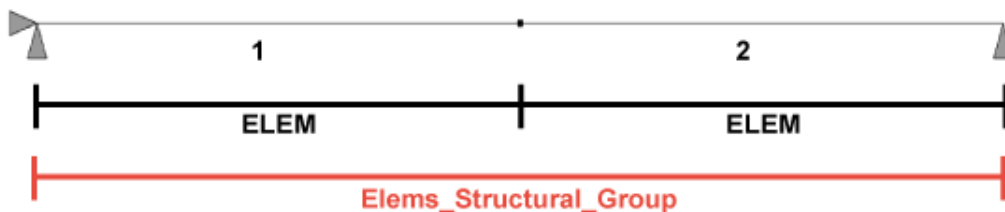


Figura 3.6: Demonstração do funcionamento da nova classe para grupos que caracterizem vigas e colunas. Fonte: Autor (2021)

Além da classe `Elems_Structural_Group` trazer esse novo conceito de modelagem, também traz a inserção de métodos que percorram as instâncias Elem, tratando e incorporando os esforços ao novo grupo criado. Ademais, decidiu-se inserir propriedades que caracterizem a viga ou a coluna modelada, como: fator de modificação para diagrama de momento fletor, comprimento de flambagem na direção x e y, coeficiente de flambagem na direção x e y, distância entre duas seções contidas a flambagem lateral com torção, entre outras.

3.3.1.5

Rotinas de cálculo para NBR8800:2008

Como dito anteriormente na Seção 2.4.2, atualmente o LESM calcula apenas os esforços na barra, onde a Classe Drv, juntamente com as suas classes adjacentes (classes Solver e Drv_LES), faz o suporte necessário ao método da rigidez direta. Contudo, ainda não existe uma classe que utiliza os valores de esforços obtidos para o dimensionamento das barras. Pensando nesse aspecto, ainda que o escopo seja limitado apenas à NBR8800:2008, é importante criar uma estrutura abstrata capaz de ser utilizada futuramente, caso sejam inseridos mais códigos normativos. Assim, a ideia é criar uma classe que contenha propriedades e métodos comuns a todas as normas. Em seguida, criaria-se outra classe com as diretrizes da norma inserida com suas especificidades, que herdará as informações da classe abstrata anteriormente criada.

Seguindo esse conceito, decidiu-se inserir uma classe chamada NBR8800 que contenha todas as rotinas de cálculo, elucidados na Seção 2.5. Assim, essas rotinas são transcritas para dentro de métodos, contidos na classe, responsáveis por analisar se o perfil está apto ou não para o esforço solicitante.

Outro ponto relevante é a maneira na qual a nova classe irá receber os elementos a serem dimensionados. Esse fato mostra-se importante, visto que, para instanciar a NBR8800, decidiu-se que, a atribuição de um parâmetro com, pelo menos, uma instância do tipo `Elems_Structural_Group`.

3.3.1.6

Memorial de cálculo

Nesta etapa da análise, observou-se a maneira como o LESM exporta os valores dos esforços para o usuário na versão atual. A partir de um teste, obteve-se o esquema visual na saída mostrado na Figura 3.7.

```

_____ ANALYSIS RESULTS _____
-----
NODAL DISPLACEMENTS / ROTATIONS
-----
NODE    DISPL X [mm]    DISPL Y [mm]    ROTAT Z [rad]
  1      0.00e+00      0.00e+00      0.00000e+00
  2      0.00e+00     -1.47e-07      0.00000e+00
  3      0.00e+00      0.00e+00      0.00000e+00
-----

SUPPORT REACTIONS
-----
NODE    FORCE X [kN]    FORCE Y [kN]    MOMENT Z [kNm]
  1      0.000          20.000         13.333
  3      0.000          20.000        -13.333
-----

INTERNAL FORCES AT ELEMENT NODES
-----
ELEMENT  AXIAL FORCE [kN]    SHEAR FORCE [kN]    BENDING MOMENT [kNm]
          Nodei  Nodef  Nodei  Nodef  Nodei  Nodef
  1      0.000    0.000    20.000  0.000    13.333  6.667
  2      0.000    0.000     0.000  20.000    -6.667 -13.333
-----

ELEMENTS INTERNAL DISPLACEMENTS IN LOCAL SYSTEM
-----
Axial and transversal displacements in 10 cross-sections from x = 0 to x = L

ELEM 1
X [m]    0.000    0.222    0.444    0.667    0.889    1.111    1.333    1.556    1.778    2.000
du [mm]  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000
dv [mm]  +0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000

ELEM 2
X [m]    0.000    0.222    0.444    0.667    0.889    1.111    1.333    1.556    1.778    2.000
du [mm]  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000  +0.000
dv [mm]  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000

```

Figura 3.7: Exportações dos valores dos esforços para o usuário fornecido pelo LESM. Fonte: Autor (2021)

Inicialmente, pensou-se em reaproveitar a estrutura da Figura 3.7 para exportar os resultados das verificações dos perfis. Contudo, como a intenção do memorial de cálculo é criar uma representação dinâmica e didática, decidiu-se reformular o visual das saídas de dados, visto que os arquivos do tipo .txt são simples e não permitem muitas estilizações. Para melhorar esse aspecto, definiu-se que, após a verificação, o LESM exporte um arquivo HTML, contendo todo o seu passo a passo. Optou-se por esse tipo de arquivo, pois as possibilidades de estilização e manipulação são significativamente maiores que no .txt. Vale salientar que, em conjunto a esse arquivo, duas bibliotecas auxiliares de código aberto, disponibilizadas pelo programa, serão utilizadas. Sendo elas:

- Bootstrap: Biblioteca utilizada para facilitar a implementação de design responsivos em aplicações que utilizam JavaScript, CSS e HTML.

- MathJax: Biblioteca JavaScript para introduzir latex em arquivos HTML.

3.3.2

Elaboração das GUI

3.3.2.1

GUI para materiais

Elaborou-se uma interface simples, mostrada na Figura 3.8, na qual o usuário consegue selecionar o tipo de aço desejado. Após a seleção, basta selecionar o “OK”, na extremidade inferior esquerda, para escolher o aço nos materiais disponíveis para as barras.

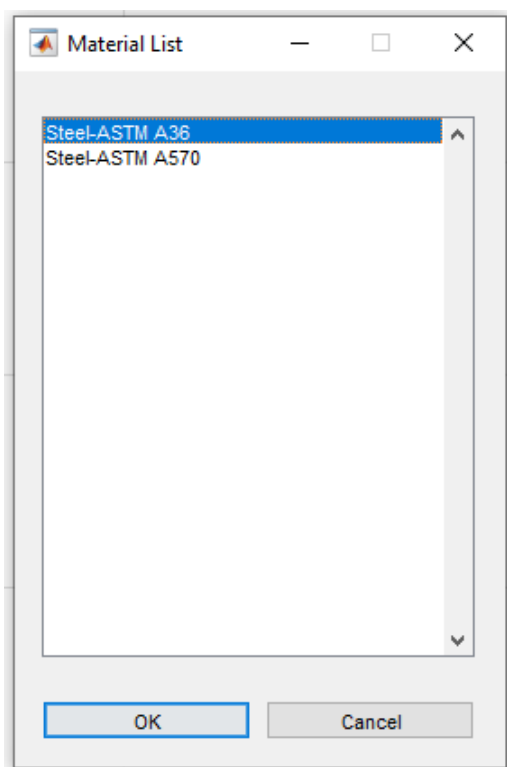


Figura 3.8: GUI com a lista de materiais. Fonte: Autor (2021)

3.3.2.2

GUI para Tipos de seções

A nova GUI de seções foi elaborada com o intuito de esclarecer ao usuário o tipo de perfil criado, além de suas respectivas propriedades geométricas, como mostrado na Figura 3.9. Ademais, na lista de perfis disponibilizada ao usuário, foi reservada uma área na GUI para demonstrar o fabricante do perfil, a lista

de famílias e suas respectivas listas de nomes de seções disponíveis, tal como na Figura 3.10.

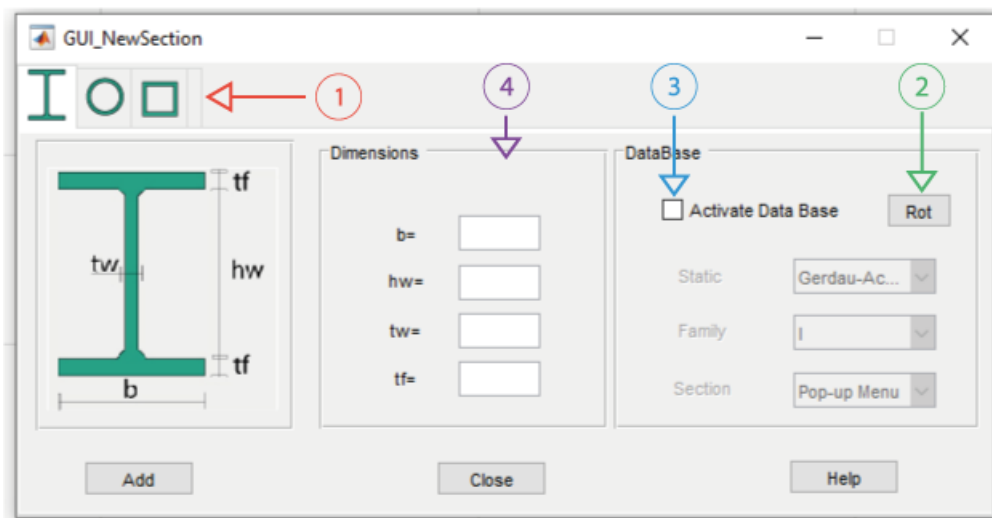


Figura 3.9: GUI para tipos de seções. Fonte: Autor (2021)

De acordo com a imagem acima, tem-se que:

- 1) Área responsável por selecionar a geometria de perfil desejada;
- 2) Botão disponível para rotacionar o perfil;
- 3) *Check Box* utilizado para ativar o modo de consulta de perfis listados;
- 4) Área para inserção ou verificação de propriedades geométricas da seção.

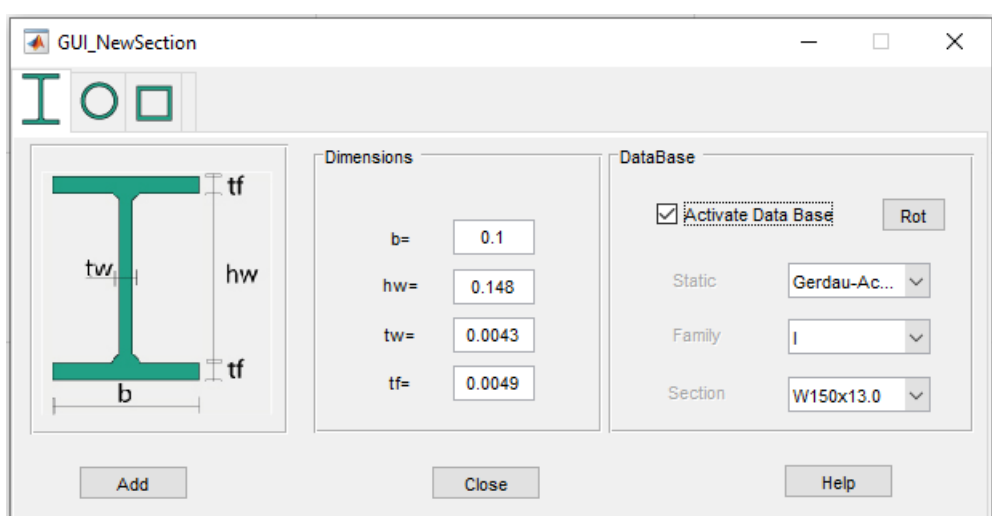


Figura 3.10: GUI de Seções com modo de consulta ativado. Fonte: Autor (2021)

3.3.2.3

GUI para combinação de ações

Essa GUI é responsável por criar os casos de carga para o sistema. Nessa tela, o usuário define o nome da carga, o tipo (variável ou permanente) e seus fatores de ponderação, assim como elucidado na Figura 3.11. Além disso, esses componentes podem ser inseridos manualmente ou, ao ativar um modo de ajuda (Figuras 3.12 e 3.13), uma lista de classificações da norma, dadas pelas Tabelas 2.1 e 2.2, é exibida. Assim, através das classificações selecionadas pelo usuário, o próprio *software* insere os fatores.

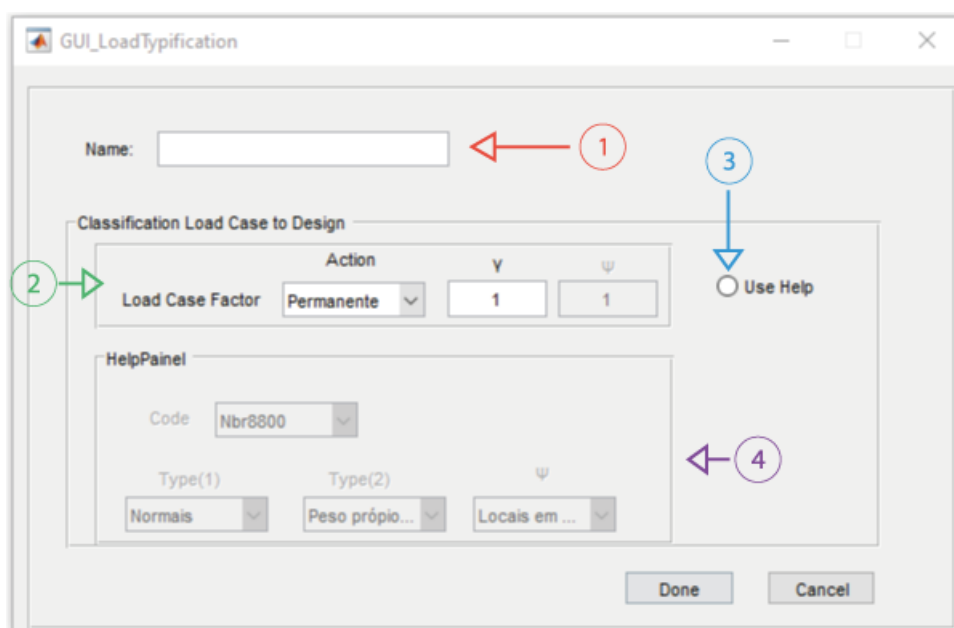


Figura 3.11: GUI para combinação de ações. Fonte: Autor (2021)

De acordo com a imagem acima, tem-se que:

- 1) Campo para inserir o nome do caso de carga;
- 2) Área de escolha do tipo de carga e inserção dos fatores ou visualizações;
- 3) 'Radio Button' para ativar modo ajuda;
- 4) Área de caracterização do tipo, de acordo com a norma escolhida.

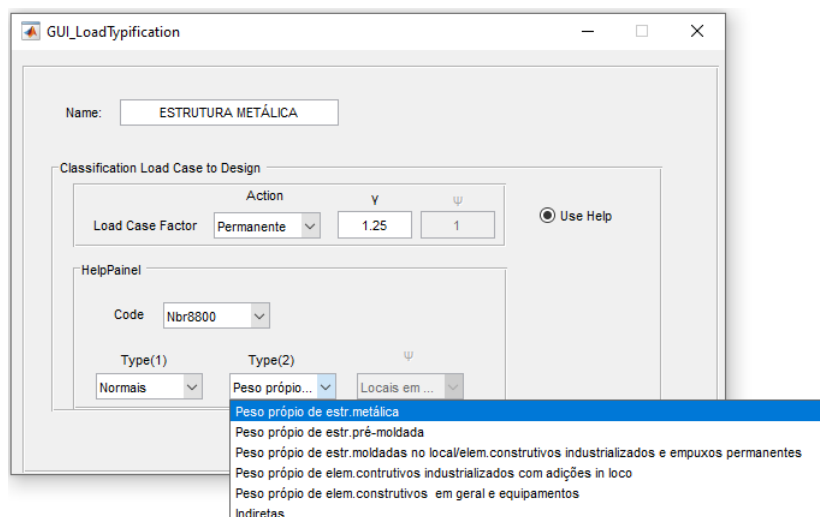


Figura 3.12: GUI para combinação de ações com o modo de ajuda ativado para cargas permanentes. Fonte: Autor (2021)

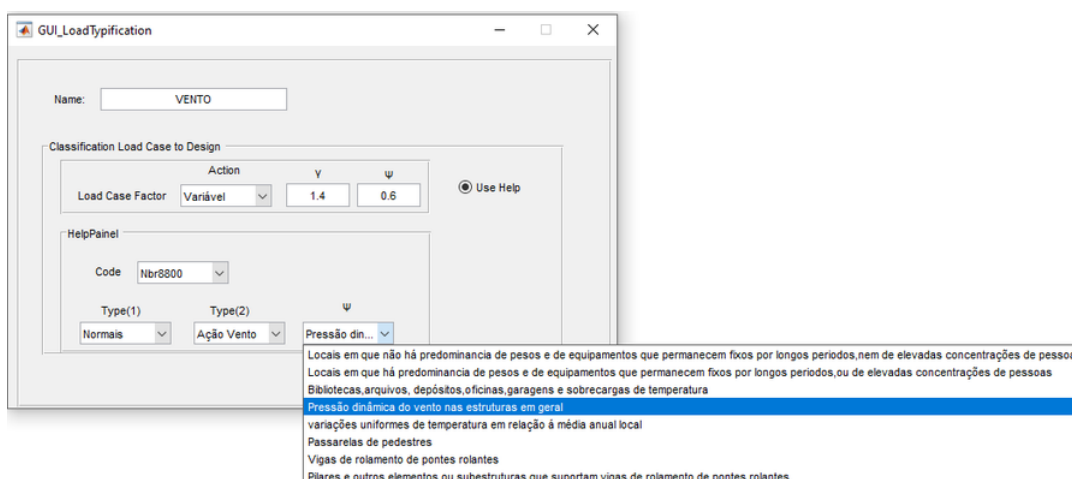


Figura 3.13: GUI para combinação de ações com o modo de ajuda ativado para cargas variáveis. Fonte: Autor (2021)

3.3.2.4 GUI para criação de grupos

Para criação de grupos de barras foi elaborada uma GUI que permite ao usuário introduzir manualmente alguns parâmetros relevantes ao dimensionamento, conforme mostrado na Figura 3.14. Embora alguns desses parâmetros possam ser obtidos automaticamente, optou-se por essa abordagem mais flexível, visando ampliar a autonomia dos usuários sobre os coeficientes

utilizados no dimensionamento. Outro objetivo dessa GUI, foi evidenciar ao usuário as barras que compõem o grupo e suas propriedades geométricas, através de áreas de visualização para esses aspectos.

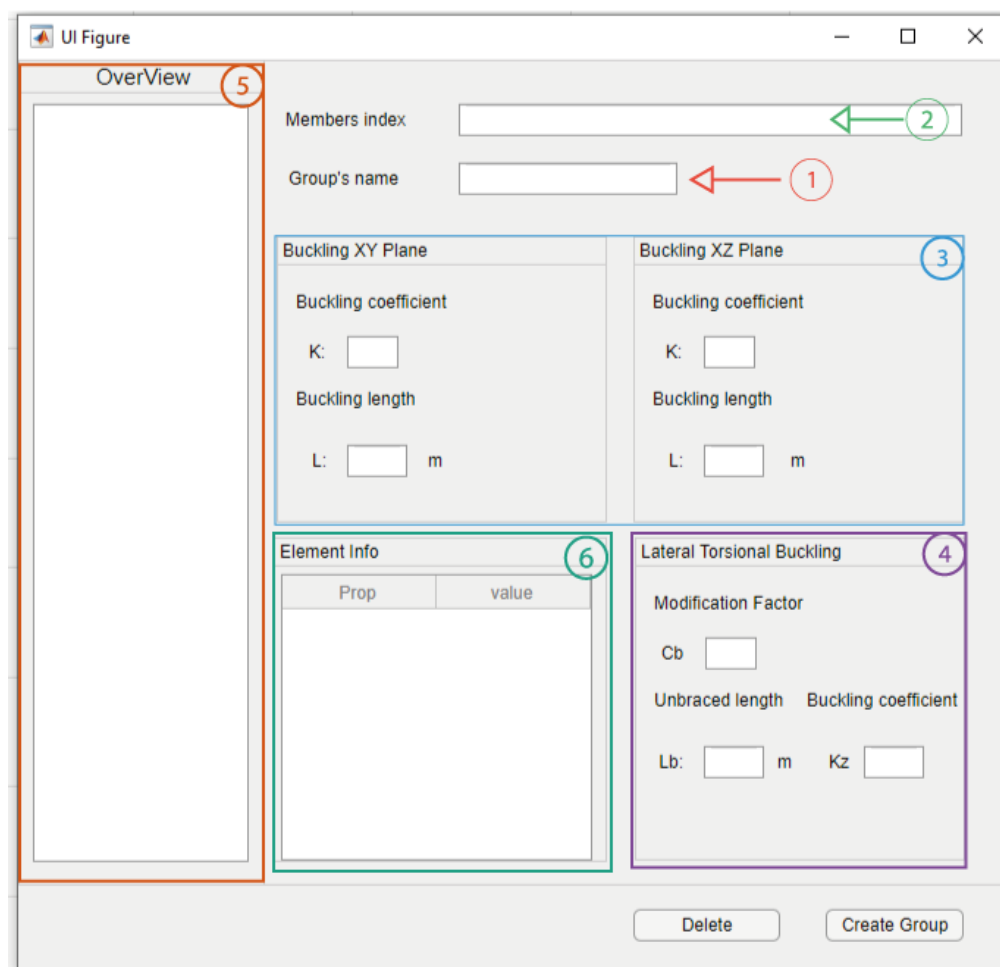


Figura 3.14: GUI de criação de grupos de elementos de barras. Fonte: Autor (2021)

- 1) Área para inserção do nome do agrupamento de elementos de barra;
- 2) Área para inserção dos números de identificação de cada elemento de barra;
- 3) Área para inserção do coeficiente de flambagem por flexão e seus respectivos comprimentos de flambagem;
- 4) Área para inserção do fator de modificação C_b , distância entre travamentos L_b e o coeficiente de flambagem por torção K_z ;
- 5) Área para visualização geral dos grupos criados;

- 6) Área para visualização de propriedades geométricas dos grupos selecionados.

3.3.3 Diagramas resultantes da Análise

Como a análise ainda é uma fase introdutória do projeto, foi necessário sintetizar todas as premissas e alterações relatadas na Seção 3.3. Para isso, foram escolhidos os diagramas UML, os quais serão utilizados como ponto de partida para uma elaboração mais detalhada de classes e relacionamentos na fase de projeto do programa. Como apresentado anteriormente na Seção 2.2, já se sabe que os diagramas introdutórios para modelagem são os de casos de uso e de sequência. Assim, elaborou-se com base nas análises feitas, diagramas UML dos processos citados acima, tal como nas Figuras 3.15 e 3.16.

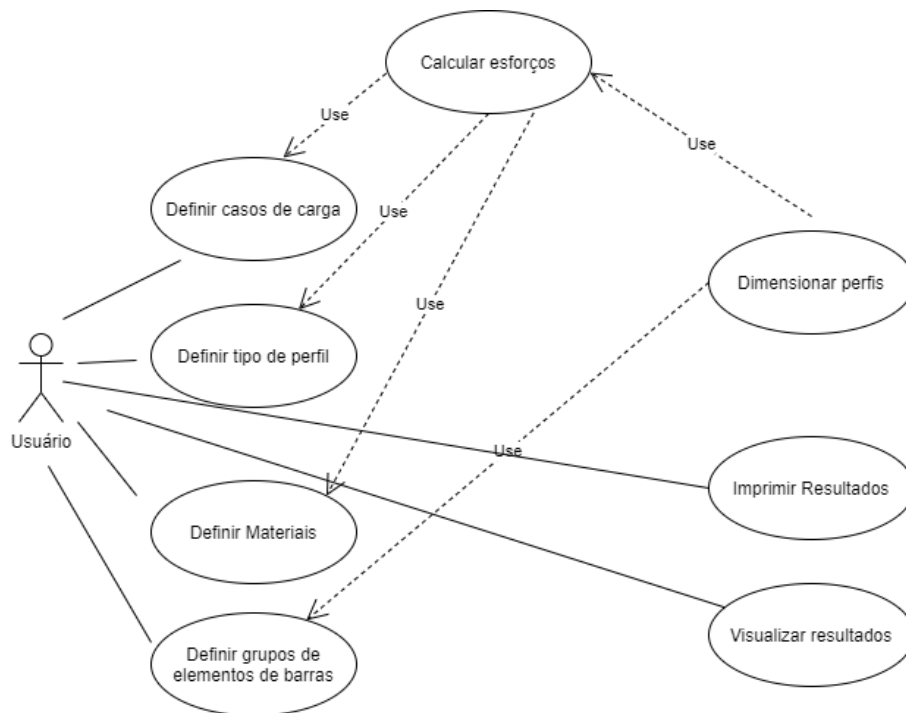


Figura 3.15: Diagrama de casos de uso para concepção da nova funcionalidade. Fonte: Autor (2021)

A Figura 3.15 elucida as atividades que um usuário pode realizar ao utilizar a nova funcionalidade. Dentre essas atividades estão: como definir casos de carga, definição de materiais, de grupos de elementos e outros. Também é esboçado a relação de dependência entre as atividades através da palavra *use*.

Após a confecção do diagrama de casos de uso, foi necessário criar o de sequência (Figura 3.16). Esse diagrama tem como intuito melhorar o

entendimento da ordem em que as ações enumeradas podem ser executadas no diagrama de casos de uso.

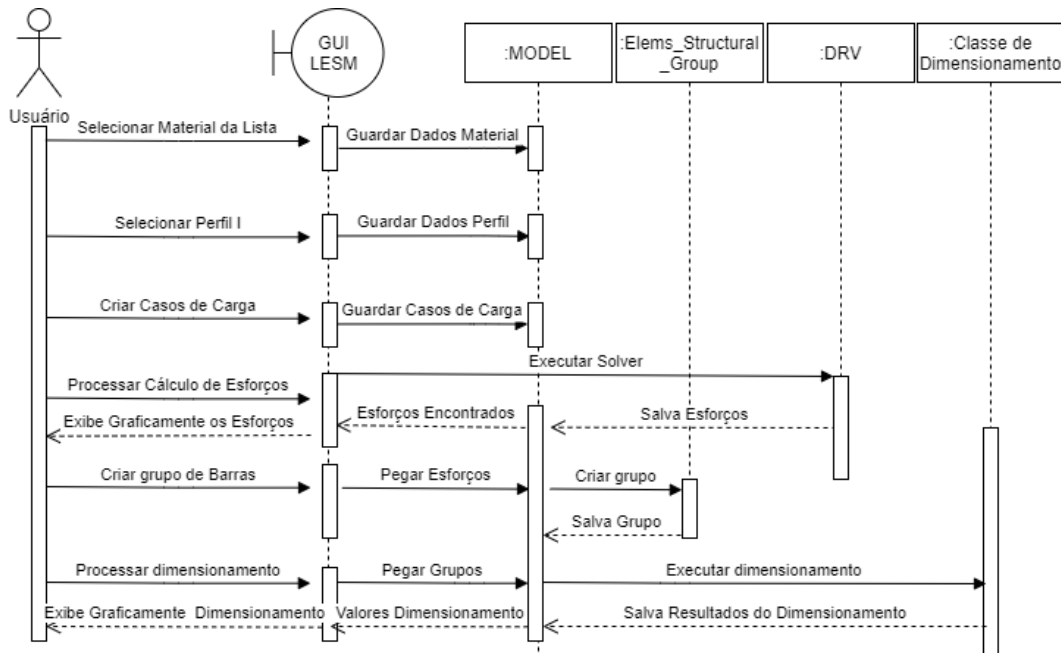


Figura 3.16: [Diagrama de sequência para concepção da nova funcionalidade. Fonte: Autor (2021)

Nesse diagrama, observam-se as ações que o usuário pode realizar no contexto de dependência e independência, diretamente ligada a ordem de execução. Vale salientar que as ações “Processar Cálculo de Esforços” e “Processar Dimensionamento” são, respectivamente, independente e dependente. Enquanto o processamento do cálculo de esforços não depende de nenhuma etapa prévia, para o dimensionamento metálico faz-se necessário que o programa já tenha realizado o cálculo dos esforços e criado previamente o grupo de barras.

3.4 Projeto do programa

Cessada a análise, inicia-se a fase de projeto do programa. Nessa etapa, reúnem-se todos os planejamentos e as definições anteriormente descritas, formalizando-as para possibilitar a criação de um processo de implementação, seguido posteriormente na codificação. Portanto, para que o projeto seja claro e objetivo, devem-se criar diagramas de classes com suas relações e propriedades, elucidando o fluxo das rotinas de cálculo. Além disso, esse último ponto mostra-se essencial, visto que se utiliza a linguagem orientada a objetos no programa.

3.4.1

Rotinas computacionais de dimensionamento

Nesta etapa foi necessário formalizar as rotinas de cálculo, exemplificadas na Seção 2.5. Dessa forma, foram elaborados fluxogramas, apresentando os cálculos de cisalhamento, flexão e compressão, com o intuito de esclarecer a implementação procedural das etapas de dimensionamento no meio computacional.

Ademais, os fluxogramas para as rotinas de tração, flexo-compressão e flexo-tração não foram construídos. Isso se deve ao fato de seus métodos de cálculo serem simples e com poucas etapas. Vale ressaltar ainda que, para a tração, apenas utilizou-se a fórmula 2-3, dado o escopo reduzido dessa funcionalidade. Para flexo-compressão e flexo-tração, por mais que suas rotinas sejam dependentes do resultado das rotinas de cargas axiais e de flexão, seus métodos de cálculo possuem somente duas variações, dadas pelas fórmulas 2-58 e 2-59.

3.4.1.1

Fluxograma da rotina de cálculo para o cortante resistente

Na Figura 3.17, mostra-se o fluxograma com o passo a passo para o cálculo da rotina de cisalhamento. Além disso, destaca-se que as etapas demonstradas serão introduzidas posteriormente na classe de dimensionamento (classe NBR8800) no *software* LESM.

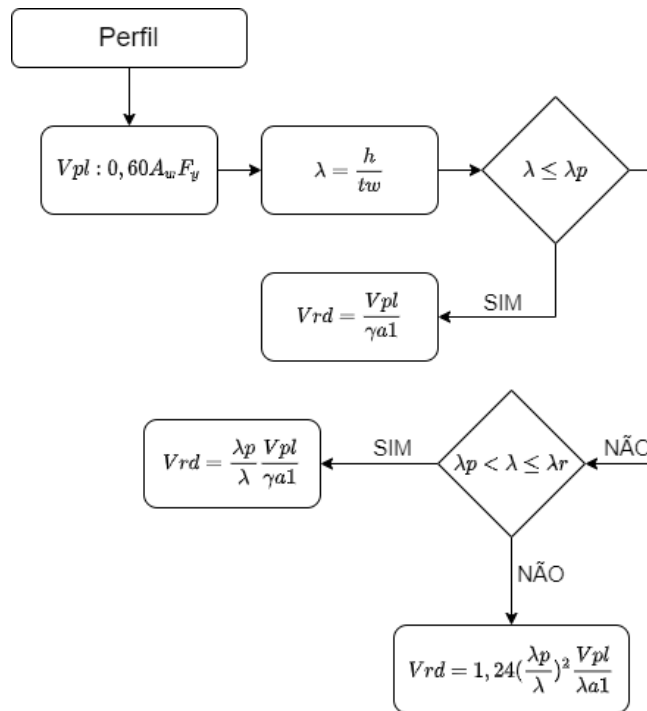


Figura 3.17: [Fluxograma para a rotina de cálculo para força resistente ao cisalhamento. Fonte: Autor (2021)]

Os valores de λ , λ_p e λ_r podem ser encontrados na seção 2.5.3.4.

3.4.1.2 Flexão

Fluxograma da rotina de cálculo do momento resistente para FLT

Assim como para o cisalhamento, também tem-se o fluxograma com as etapas para a execução da rotina de cálculo para o momento resistente de FLT, elucidado na Figura 3.18. Do mesmo modo, destaca-se que as etapas demonstradas serão introduzidas posteriormente na classe de dimensionamento (classe NBR8800) no *software* LESM.

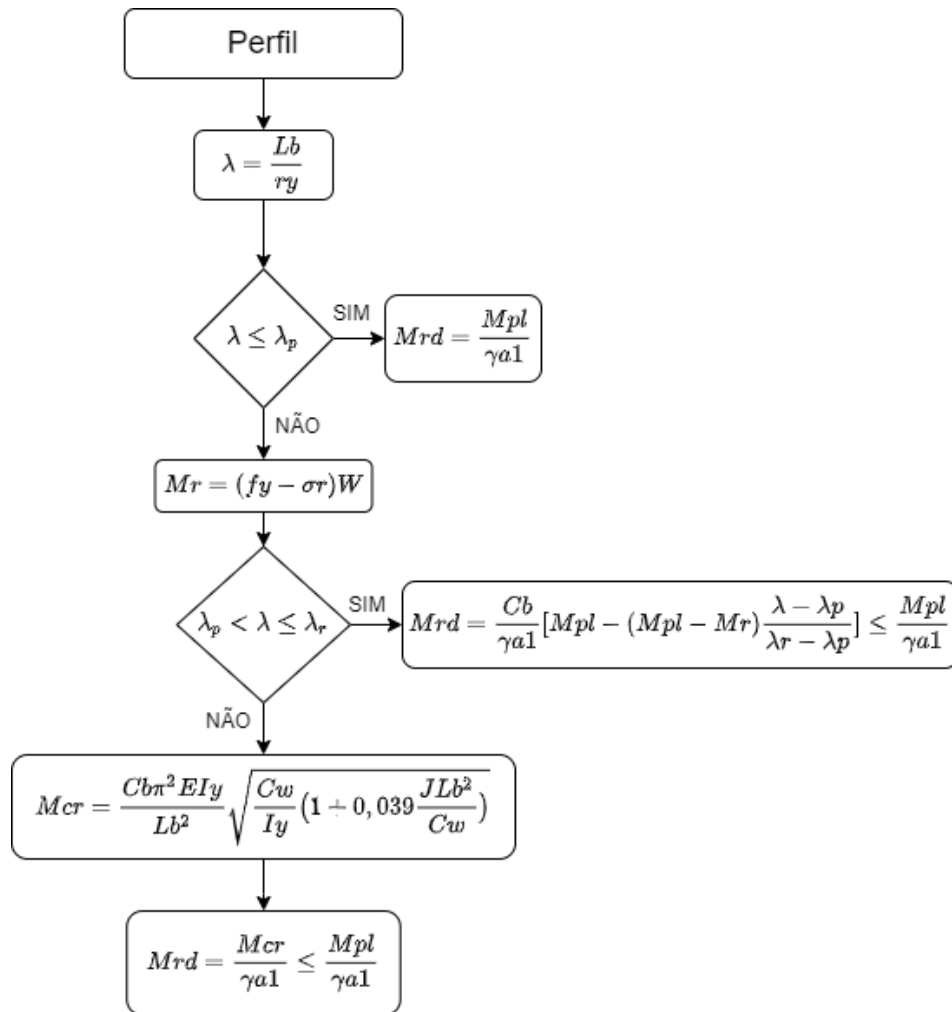


Figura 3.18: [Fluxograma da rotina de cálculo do momento resistente para FLT. Fonte: Autor (2021)]

Os valores de λ , λ_p e λ_r podem ser encontrados na seção 2.5.3.2.

Fluxograma da rotina de cálculo do momento resistente para FLM

Para o momento resistente de FLM, o fluxograma, ilustrado na Figura 3.19, destaca os passos para a execução de sua rotina de cálculo. Também elucidada-se que as etapas demonstradas serão introduzidas posteriormente na classe de dimensionamento (classe NBR8800) no *software* LESM.

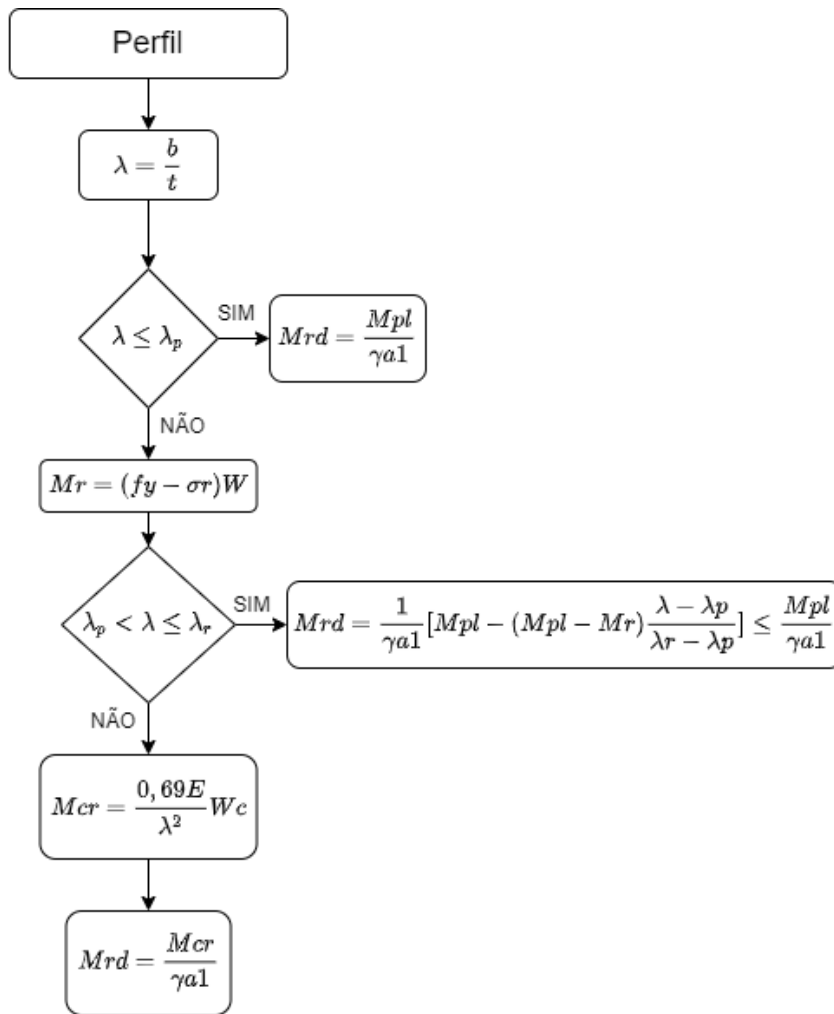


Figura 3.19: [Fluxograma da rotina de cálculo do momento resistente para FLM. Fonte: Autor (2021)]

Os valores de λ , λ_p e λ_r podem ser encontrados na seção 2.5.3.2.

Fluxograma da rotina de cálculo do momento resistente para FLA

O passo a passo para a execução da rotina de cálculo do momento resistente de FLA é demonstrado no fluxograma, elucidado na Figura 3.20. Como citado anteriormente, as etapas demonstradas serão introduzidas posteriormente na classe de dimensionamento (classe NBR8800) no *software* LESM.

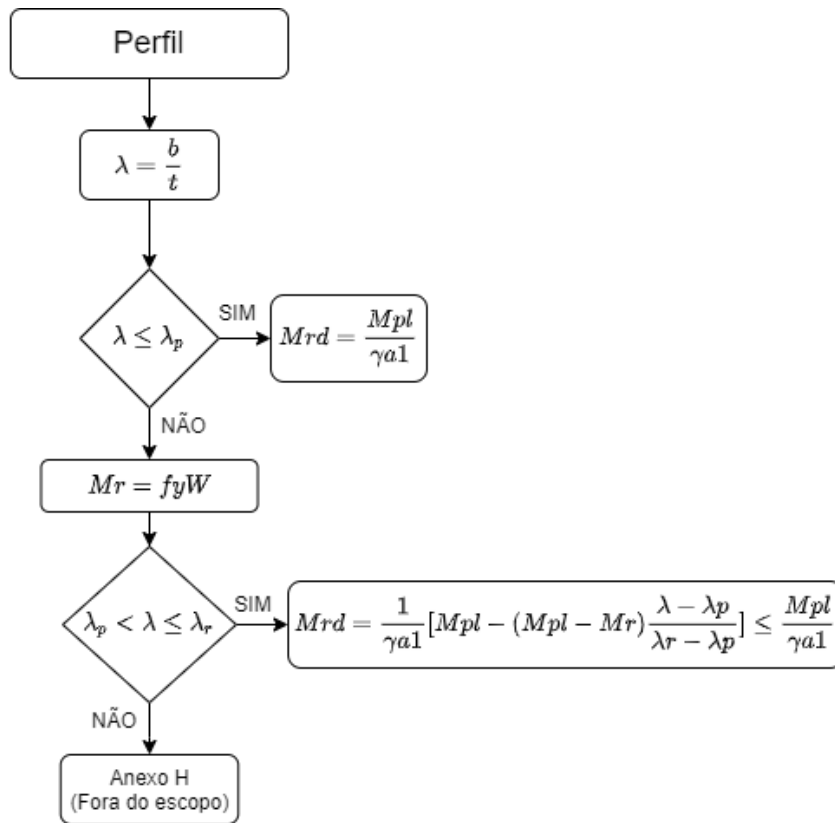


Figura 3.20: [Fluxograma da rotina de cálculo do momento resistente para FLA. Fonte: Autor (2021)]

Os valores de λ , λ_p e λ_r podem ser encontrados na seção 2.5.3.2.

3.4.1.3

Fluxograma da rotina de cálculo para força resistente de compressão

Por fim, tem-se, no fluxograma mostrado na Figura 3.21, as etapas para o cálculo da rotina de compressão. Assim como nos demais casos, os passos demonstrados serão introduzidos posteriormente na classe de dimensionamento (classe NBR8800) no *software* LESM.

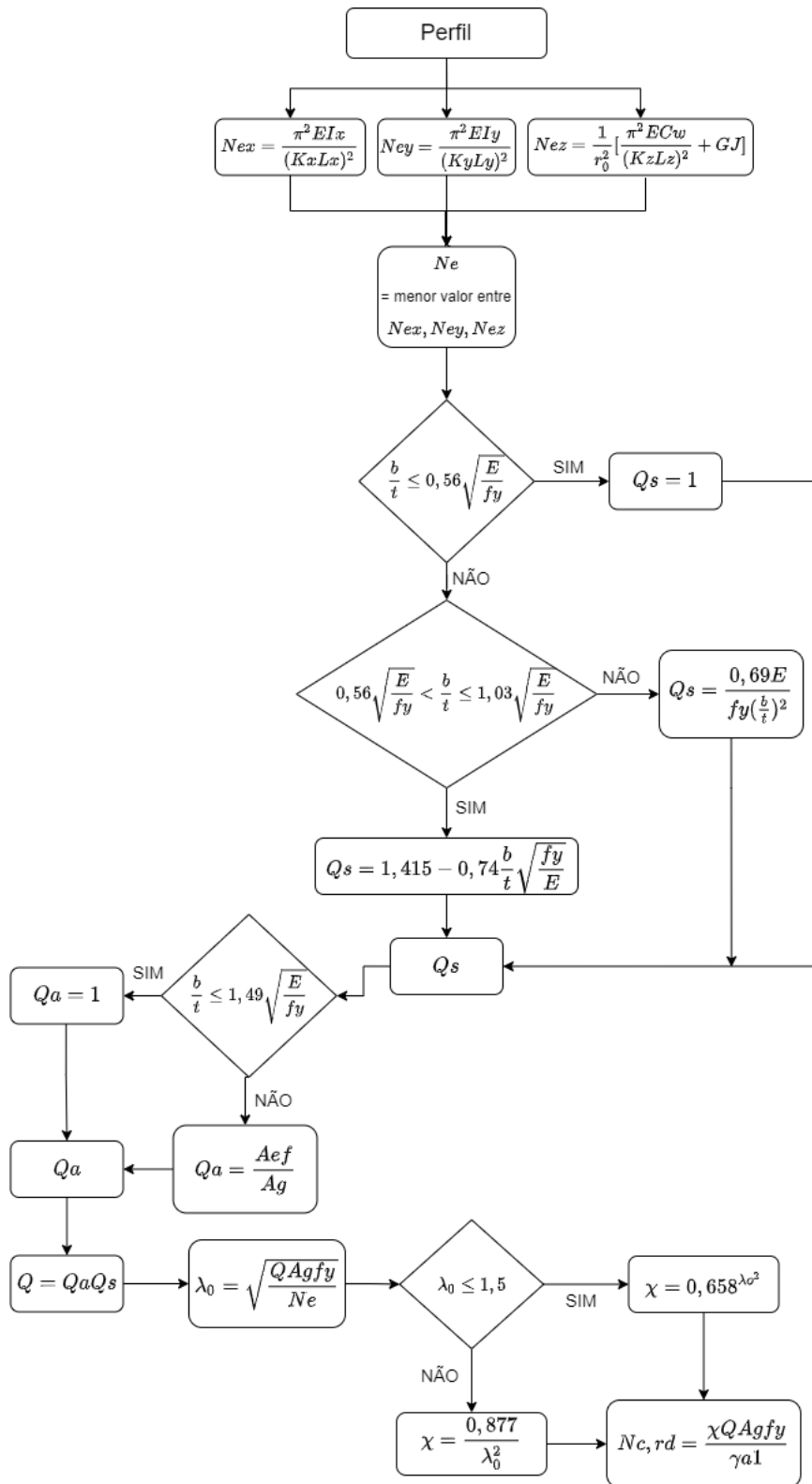


Figura 3.21: [Fluxograma da rotina de cálculo para força resistente de compressão. Fonte: Autor (2021)]

3.4.2

Implementação da nova estrutura de dados

Nessa etapa, inseriu-se todas as modificações apontadas na fase de análise. Também foi realizada a arquitetura de funcionamento conjunto das classes inseridas ou modificadas, além da implementação de métodos e propriedades. Vale ressaltar que todo esse processo será elucidado a seguir através de diagramas de classes.

Para facilitar o entendimento dessa seção, os tópicos apresentados seguirão a mesma ordem do diagrama da Figura 3.16, o qual reflete a ordem de funcionamento entre as classes.

3.4.2.1

Implementação da nova Section

Como dito na Seção 3.3.1.3, houve a necessidade de adicionar propriedades comuns a todos os tipos de perfis. Como se pode observar na nova estrutura da classe Section, dada pela Tabela 3.2, foram adicionadas novas propriedades, abrangendo do valor da área bruta até o módulo de resistência plástico. Também foram adicionados métodos abstratos (métodos não implementados), pois, caso o usuário não ative o modo de consulta de perfis, dado pela Figura 3.9, o valor dessas novas propriedades inseridas serão obtidas através desses métodos abstratos, passados por herança.

Tabela 3.2: Propriedades físicas dos aços disponíveis na nova versão do LESM,
 Fonte: Autor (2021)

Section	Section Class
~Ag:	double =>Área da seção trasnversal
~Ix:	double =>Momento de inércia referente ao eixo forte
~Wx:	double =>Módulo de resistência elástico ao eixo x
~Wy:	double =>Módulo de resistência elástico ao eixo y
~rx:	double =>Raio de giração referente ao eixo x
~Iy:	double =>Momento de inércia referente ao eixo fraco
~ry:	double =>Raio de giração referente ao eixo y
~cw:	double =>Constante de empenamento da seção trasnversal
~r0:	double =>Raio de giração polar da seção bruta em relação ao centro de cisalhamento
~J:	double =>Constante de torção da seção trasnversal
~Zx:	double =>Módulo de resistência plástico referente ao eixo x
~Zy:	double =>Módulo de resistência plástico referente ao eixo y
+ AgSolver(W_Beam):	double =>Método abstrato para cálculo de área bruta trasnversal
+ IxSolver(W_Beam):	double =>Método abstrato para cálculo do momento de inercia em relação ao eixo x
+ IySolver(W_Beam):	double =>Método abstrato para cálculo do momento de inercia em relação ao eixo y
+ rxSolver(W_Beam):	double =>Método abstrato para cálculo do raio de giração ao eixo x
+ rySolver(W_Beam):	double =>Método abstrato para cálculo do raio de giração ao eixo y
+ r0Solver(W_Beam):	double =>Método abstrato para cálculo do raio de giração polar da seção
+ WxSolver(W_Beam):	double =>Método abstrato para cálculo do módulo de resistência elástico do eixo x
+ cwSolver(W_Beam):	double =>Método abstrato para cálculo do módulo de resistência elástico do eixo y
+ JSolver(W_Beam):	double =>Método abstrato para cálculo da constante de torção da seção
+ clean():	void =>Força liberação de memória utilizada pelas propriedades

Nesse caso, em que o perfil não é selecionado, os métodos abstratos são herdados e implementados, através de polimorfismo, de acordo com a particularidade da família do perfil por uma nova classe adicionada, a `W_beam`, dada pela Tabela 3.3. Isso ocorre pois os coeficientes obtidos por esses métodos variam dependendo do tipo de seção.

Vale ressaltar ainda que, além de conter todas as características da classe `Section`, a `W_beam` possui propriedades exclusivas da família de perfil `l`, como: `bf`, `tf`, `d`, `h` e `tw`.

Tabela 3.3: Diagrama da classe W_Beam, Fonte: Autor (2021)

W_Beam - W_Beam Class
~bf: double =>Largura da mesa
~tf: double =>Espessura de mesa
~d: double =>Altura total da seção transversal
~h: double =>Altura total da alma do perfil
~tw: double =>Espesura da alma
+W_Beam () =>Construtor
+AgSolver (W_Beam): double =>Método abstrato para cálculo da Área bruta trasnversal
+IxSolver (W_Beam): double =>Método abstrato para cálculo do momento de inércia do eixo x
+IySolver (W_Beam): double =>Método abstrato para cálculo do momento de inércia do eixo y
+r0Solver (W_Beam): double =>Método abstrato para cálculo do raio de giração polar da seção
+rxSolver (W_Beam): double =>Método abstrato para cálculo do raio de giração do eixo x
+rySolver (W_Beam): double =>Método abstrato para cálculo do raio de giração do eixo y
+WxSolver (W_Beam): double =>Método abstrato para cálculo do módulo de resistência elástico do eixo x
+cwSolver (W_Beam): double =>Método abstrato para cálculo do módulo de resistência elástico do eixo y
+JSolver (W_Beam): double =>Método abstrato para cálculo da constante de torção da seção
+ clean(): void =>Força liberação de memória utilizada pelas propriedades

Com essa nova definição, caso o usuário selecione uma viga W150x13, por exemplo, o objeto salvo na classe Model teria o formato demonstrado na Figura 3.22.

W150x13: W Beam
Ag: 0.0017
Ix: 6.3500e-06
Wx: 8.5811e-05
Wy: 1.6400e-05
rx: 0.0618
ly: 8.2000e-07
ry: 0.0222
cw: 4.1979e-09
r0: 0.0657
J: 1.7200e-08
Zx: 9.6400e-05
Zy: 2.5500e-05

Figura 3.22: [Diagrama de objeto W_Beam. Fonte: Autor (2021)]

3.4.2.2 Implementação da nova Materials

Como planejado na análise, a classe Material possui apenas as propriedades comuns a todo tipo de material, como pode-se observar na nova estrutura de dados, dada pela Tabela 3.4. Sendo assim, foi criada também a classe Steel

para instanciar um dos tipos de aços disponíveis no programa, dado pela Tabela 3.5.

Tabela 3.4: Diagrama da classe Material, Fonte: Autor (2021)

Material - Material Class
~id: int =>Número indicador
~elasticity: double =>Módulo de elasticidade
~poisson: double =>Coeficiente de Poison
~shear: double =>Coeficiente cisalhamento
~themExp: double =>Coeficiente de expansão térmica
~density: double =>Densidade mássica
~leakage: double =>Tensão de escoamento
~yield: double =>Tensão de ruptura
+clean(): void =>Força liberação de memória utilizada pelas propriedades

Tabela 3.5: Diagrama da classe Steel, Fonte: Autor (2021)

Steel - Steel Class
+Steel() =>Construtor

Nessa nova definição, quando o usuário escolhe um tipo de aço na GUI, dado pela Figura 3.8, um novo objeto do tipo Steel é salvo na Model. Supondo que o aço escolhido seja o ASTM-A36, a configuração mostrada na Figura 3.23 é salva.

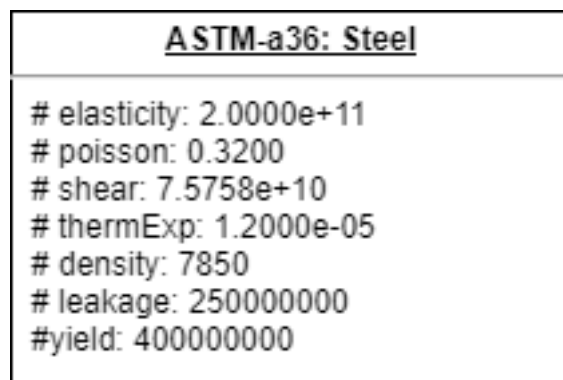


Figura 3.23: [Diagrama de objeto Steel. Fonte: Autor (2021)

3.4.2.3

Implementação da nova definição dos casos de carga

Para os casos de carga, não foi criada uma nova classe, apenas foi adicionada uma nova propriedade para a classe *Model*, denominada *loadsInformations*. Nessa propriedade será salva uma matriz de informações, referente a cada caso de carga, inserida pelo usuário na GUI, dada pela Figura 3.11. As informações contidas nessa matriz são: o tipo da carga, o coeficiente λ_g para as cargas permanentes, λ_q e Ψ_0 para as cargas variáveis. A posteriori, essas informações serão consultadas e combinadas através da fórmula 2-2. Visto isso, a notação indicial para a matriz *loadsInformations* será dada por:

$$[loadInformations]_{(i,j)} \quad (3-3)$$

Sendo:

i referente as propriedades citadas acima;

j refere-se ao número de casos de carga.

Observação: Na linha que contém o fator Ψ_0 , caso a carga seja permanente, o valor adicionado é 0.

Complementando o processo de combinação de ações, criou-se uma função auxiliar, denominada *CombinationLoads*. Essa nova função será chamada, posteriormente, e usada para realizar as combinações dos fatores salvos na variável *loadInformations*. Após a combinação desses fatores, os mesmo serão passados por um processo iterativo, o qual acessará a matriz de esforços, salva nos grupos, e retornará uma nova matriz de esforços combinados, salva nas propriedades com sufixo *cases* na classe *Elemes_Structural_Group*. Vale dizer que essa função calcula todas as combinações possíveis.

3.4.2.4

Implementação da nova estrutura para o Drv

Como dito na Seção 3.3.1.1, o *Drv* foi abstraído com o intuito de introduzir outros tipos de análise no futuro. Dessa forma, foi criada uma nova classe, denominada *Drv_Les_Design*. Em suma, a nova classe realiza a fatoração das matrizes para a resolução do método da rigidez direta e obtenção dos esforços nas barras modeladas. Ademais, foi criada a classe *Model*, responsável por armazenar as informações gerais, como os vetores de objetos de elementos, os materiais, as seções, os números de nós, entre outras.

Dessa forma, as informações associadas ao modelo dentro do *Drv*, dado pela Figura 2.10, foram movidas para a *Model*. Assim, o *Drv*, atualmente, apenas consulta as informações de modelo, contidas na *Model*, para criar as

matrizes de rigidez dos elementos e as condições de contorno. Além disso, também dá prosseguimento ao método da rigidez direta.

Portanto, ao selecionar o material, o tipo de seção, definir os casos de carga e modelar a estrutura, todos esses dados serão salvos na Model. Ademais, ao clicar em “Processar dados”, o Drv consulta as informações salvas na Model, executa o método da rigidez direta e popula o vetor de objetos do tipo Elems com os esforços contidos para cada barra. Devido ao Drv_Les_design, as barras armazenam os esforços para cada caso de carga em suas propriedades, dada na Figura 2.14. Dessa forma, após processar os dados, as propriedades, antes dadas por vetores de esforços, serão representadas agora por matrizes, sendo:

$$[matrizDoEsforçoCorrente]_{(i,j)} \quad (3-4)$$

i refere-se ao número de casos de carga;

j refere-se ao número de divisões do elemento de barra.

Observação: As divisões do elemento de barra (Elem), elucidados pelo índice j da matriz dos esforços, representam pontos ao longo da seção, na qual esses esforços são obtidos pelo *software*. No LESM, cada elemento é dividido em 50 seções.

3.4.2.5

Implementação da nova estrutura de grupos

Para implementar esse novo conceito de grupos de barra, criou-se uma nova classe, denominada Elems_Structural_Group, como dito na fase de Análise. Sua estrutura é dada pela Tabela 3.6. Essa nova classe contém propriedades, como Kx, Ky, Lx, Ly, Kz, Lb e Cb, que serão salvas quando o usuário criar o grupo (instanciar um elemento da classe) através da GUI, dada pela Figura 3.14. Também vale ressaltar que as propriedades, como material, section e elems, são implementadas assim que o grupo é criado. Assim, as identificações das barras, dadas pelos “ids”, informados pelo usuário, são consultadas na Model, repassando o material dos elementos e a seção que os compõem para o grupo.

Tabela 3.6: Diagrama da classe Elems_Structural_Group, Fonte: Autor (2021)

Elems_Structural_Group - Elems_Structural_Group Class	
~name: string =>	Nome do agrupamento de elementos de barra
~elems: Elem[] =>	Vetor de referência para objetos da classe Elem
~ids: int =>	Identificação do agrupamento de elementos de barra
~length: double =>	Comprimento total do agrupamento de barras
~Kx: double =>	Coefficiente de flambagem por flexão em x
~Ky: double =>	Coefficiente de flambagem por flexão em y
~Lx: double =>	Comprimento de flambagem em x
~Ly: double =>	Comprimento de flambagem em y
~Cb: double =>	Fator de modificação para diagrama de momento fletor
~Lb: double =>	Distância entre duas seções contidas à flambagem lateral com torção
~Kz: double =>	Coefficiente de flambagem por torção
~axialForce:double[] =>	Matriz dos carregamentos axiais não combinados
~axialForceCase:double[] =>	Matriz dos carregamentos axiais combinados
~tractionCase:double[] =>	Matriz dos carregamentos axiais combinados referentes a tração
~compressionCase:double[] =>	Matriz dos carregamentos axiais combinados referentes a compressão
~maxCompressionCase:double =>	Valor máximo da força axial de compressão entre as combinações
~maxCompressionCase:string =>	Nome da combinação mais crítica para a compressão
~compressionVerify:double =>	Valor da força axial resistente de compressão
~maxTraction:double =>	Valor máximo da força axial de tração entre as combinações
~maxTraction:string =>	Nome da combinação mais crítica para a tração
~tractionVerify:double =>	Valor da força axial resistente de tração
~bending_XY:double[] =>	matriz dos momentos no plano XY não combinados
~bendingCases_XY:double[] =>	matriz dos momentos no plano XY combinados
~bendingXYVerify:double[] =>	vetor com o valor do momento de resistente FLT, FLA e FLM
~maxBendMoment_XY:double =>	Valor de momento máximo de tração entre combinações
~maxBendMoment_XY_CombinationId:string =>	Nome da combinação mais crítica para a flexão
~shear_XY:double[] =>	Matriz dos valores de esforços cortante não combinados
~shearCases_XY:double[] =>	Matriz dos valores de esforços cortante combinados
~shearXYVerify:double =>	Valor da força cortante resistente
~maxShearForce_XY(): double =>	Valor máximo da força cortante entre combinações
~maxShearForce_XY_combinationId: string =>	Nome da combinação mais crítica para o cortante
+Elems_Structural_Group() =>	Construtor
+separeTractionCases(): void =>	Método para aglutinar somente as cargas axiais de tração
+getMaxTraction(): void =>	Método para obter o maior valor de tração combinada
+separeCompressCases(): void =>	Método para gluttinar somente as cargas axiais de compressão
+getMaxCompress(): void =>	Método para obter o maior valor de compressão combinada
+getMaxBending(): void =>	Método para obter o maior valor de flexão combinada
+getMaxShear(): void =>	Método para obter o maior valor de cortante combinado
+trasformElems(): void =>	Método para calcular o tamanho do grupo dos elementos de barras

As propriedades com o prefixo “max” são necessárias pois, armazenam o valor máximo do esforço (S_d) dentre todos os casos de carga possíveis, que posteriormente será utilizado pela classe de dimensionamento para comparar esse valor com o resistente obtido, com o intuito de registrar se o perfil está apto ou não para ser utilizado. Já as propriedades com o sufixo “cases” são necessária pois, armazenam os valores dos esforços para cada caso de carga ao

longo da seção com o intuito do mesmo ser consultado após o dimensionamento, caso o usuário queira visualizar a taxa de utilização de cada esforço ao longo da viga selecionada.

Dessa forma, o gráfico da taxa de utilização é montado da seguinte maneira: após a classe de dimensionamento (classe NBR8800) realizar o cálculo do esforço resistente, o resultado será registrado nas propriedades com o sufixo “*verify*“. Em seguida, juntamente com as propriedades com o sufixo “*cases*“, pode-se ter o valor do fator de utilização, dado pela seguinte razão:

$$\eta_{[lc,i]} = \left| \frac{Cases_{[lc,i]}}{Verify} \right| \quad (3-5)$$

sendo:

n o fator de utilização;

lc o ‘*load case*’ o caso de carga utilizado;

i é referente a seção da viga analisada.

3.4.2.6

Implementação da classe de dimensionamento

Para a execução da verificação do perfil, criou-se a classe NBR8800. Essa classe tem como principal função receber as instâncias de *Elms_Structural_Group*, criadas pelo usuário, percorrê-las e verificá-las, através de seus métodos. Ademais, todos os cálculos executados serão registrados no memorial de cálculo para a livre consulta do usuário. Pensando nas premissas levantadas acima, foi desenvolvida a seguinte disposição de propriedades e métodos para essa classe, demonstrados na Tabela 3.7.

Tabela 3.7: Diagrama da classe NBR8800, Fonte: Autor (2021)

NBR8800 - NBR8800 Class

~ActionList:string[] =>Lista de tipificação do tipo de combinação
~Type1List:string[] =>Lista de tipos de combinação
~Type2Permanent:string[] =>Lista de tipificação de carga do tipo direta e indireta
~Type2Variable:string[] =>Lista de tipificação de carga do tipo variáveis
~Type2PsiList:string[] =>Lista de tipificação de carga do tipo variáveis para o fator psi
~permanentValues: double[] =>Lista dos coeficientes de ponderação de cargas permanentes
~keyPermanentValues: string[] =>Chave de identificação da lista de cargas permanentes
~variableValues:double[] =>Lista de valores de coeficientes de ponderação de cargas variáveis
~keyVariableValues: string[] =>Chave de identificação da lista de cargas variáveis
~psiValues:double[] =>Lista de valores de coeficientes de ponderação de psi
~keyPsiValues: string[] =>Chave de identificação da lista de psi
~designReportFile:NBR8800_report =>objeto de referência a classe NBR8800_Report
+NBR8800() =>Construtor
+LoadCombination():double =>Método para combinar as chaves de fatores
+solve():void =>Realizar dimensionamento
+classifyFlm(): any[] =>método para classificar seção e obter fatores para FLM
+classifyFla(): any[] =>método para classificar seção e obter fatores para FLA
+classifyFlt(): any[] =>método para classificar seção e obter fatores para FLT
+MrdFlmFlaProcess(): void =>Método para calcular o momento resistente para FLA e FLM
+MrdFltProcess(): void =>Método para calcular o momento resistente para FLT
+NrdSolve(): void =>Método para calcular a força axial resistente de tração
+getNeValue(): double =>Método retorna a menor força axial de flambagem elástica
+getLocalBuckingFactorQa: double =>Método retorna o fator de redução Qa
+getLocalBuckingFactorQs: double =>Método retorna o fator de redução Qs
+getQsiFactor: double =>Método retorna o fator de redução Qsi
+CompressSolve(): void =>Método para calcular a força axial de compressão resistente
+shearResistance(): void =>Método para calcular o cisalhamento resistente a seção
+axialBendingCombination(): void =>Método para calcular o flexo compressão e flexo tração

Observando o diagrama acima, pode-se identificar diversas propriedades e métodos. A maioria das propriedades presentes são responsáveis por armazenar constantes estipuladas pela norma, como as lista de classificação dos casos de carga, os fatores associados a cada tipo de carga, entre outros. Para os métodos, tem-se uma estrutura mais robusta. As rotinas definidas na Seção 3.4.1 são transformadas em dois tipos de métodos: um utilizado para obter os coeficientes necessários para a verificação do perfil e o outro que utiliza desses coeficientes para encontrar o valor do esforço resistente da barra.

Em relação ao registro do memorial dos cálculos, considerou-se mais semântico e conciso criar uma nova classe para auxiliar nesse processo. Com isso, foi criada a classe NBR8800_Report, possuindo os métodos necessários para realização desses registros. Sendo assim, quando a classe NBR8800 é instanciada, além das barras (Elems_Structural_groupo), deve ser passada

também uma instância da nova classe criada. Dessa forma, os métodos dessa nova instância serão sempre chamados dentro de cada método da classe NBR8800, registrando todas as atividades. O fluxo de funcionamento dessa classe, NBR8800_Report, é expresso na Figura 3.24.

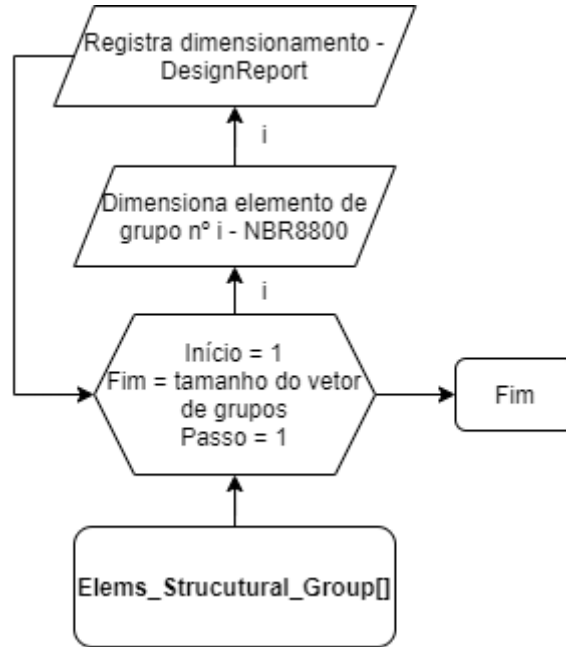


Figura 3.24: [Fluxo de funcionamento da classe NBR8800_Report. Fonte: Autor (2021)]

3.4.2.7

Implementação da classe de geração de memorial de cálculo

Como dito na seção anterior, a classe NBR8800_Report tem o intuito de registrar todo o cálculo realizado nas etapas de verificação. Para isso, a classe dispõe de métodos que garantem a construção desses registros associados a cada esforço verificado. Para tal propósito, essa classe foi elaborada de acordo com o demonstrado na Tabela 3.8.

Tabela 3.8: Diagrama da classe NBR8800_Report, Fonte: Autor (2021)

NBR8800_Report - NBR8800_Report
+NBR8800_Report() =>Construtor
+createHtml(): void =>Cria a estrutura HTML
+createTractionSection(): void =>Cria a seção de tração do memorial de cálculo
+createFlexSection(): void =>Cria a seção de flexão do memorial de cálculo
+createShearSection(): void =>Cria a seção de cisalhamento do memorial de cálculo
+createCompressionSection(): void =>Cria a seção de compressão do memorial de cálculo

Os métodos apresentados acima trazem todo o suporte para a elaboração do documento HTML com os registros dos cálculos executados. Para essa construção ser possível, o uso da classe fica restrito ao momento em que os métodos da classe NBR8800 são chamados, visto que nem todos os valores obtidos nas rotinas são salvos no programa. Sendo assim, o valor encontrado deve ser registrado pelo programa no documento HTML no momento em que é calculado, já que quando o próximo método for chamado, esse valor não existirá mais.

A fim de sintetizar o processo descrito acima, elaborou-se um esquema que ilustra as chamadas dos métodos a cada rotina de cálculo, dado pela Figura 3.25

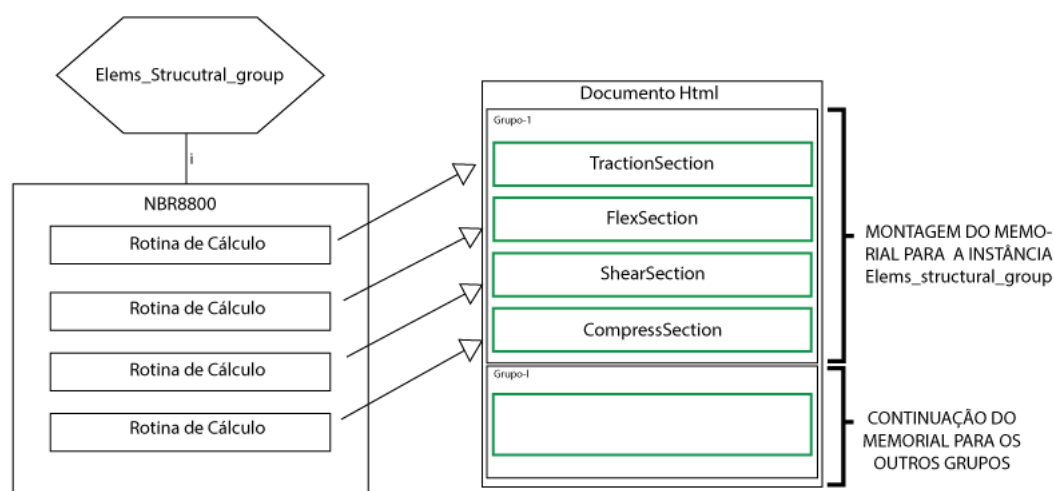


Figura 3.25: [Construção do memorial de cálculo através dos métodos da classe NBR8800_Report. Fonte: Autor (2021)]

3.4.2.8

Nova arquitetura de classes e fluxo de ações

Após definir as novas implementações de classes e rotinas, criou-se um diagrama UML de classes, dado na Figura 3.26, e um fluxograma de sequência de ações, mostrado na Figura 3.27, para elucidar de maneira mais técnica todo o novo fluxo de dados do *software* LESM. Além disso, essa etapa também tem como intuito criar uma documentação que sintetize todas as alterações realizadas, facilitando a fase de codificação e gerando um material consultivo rico e detalhado.

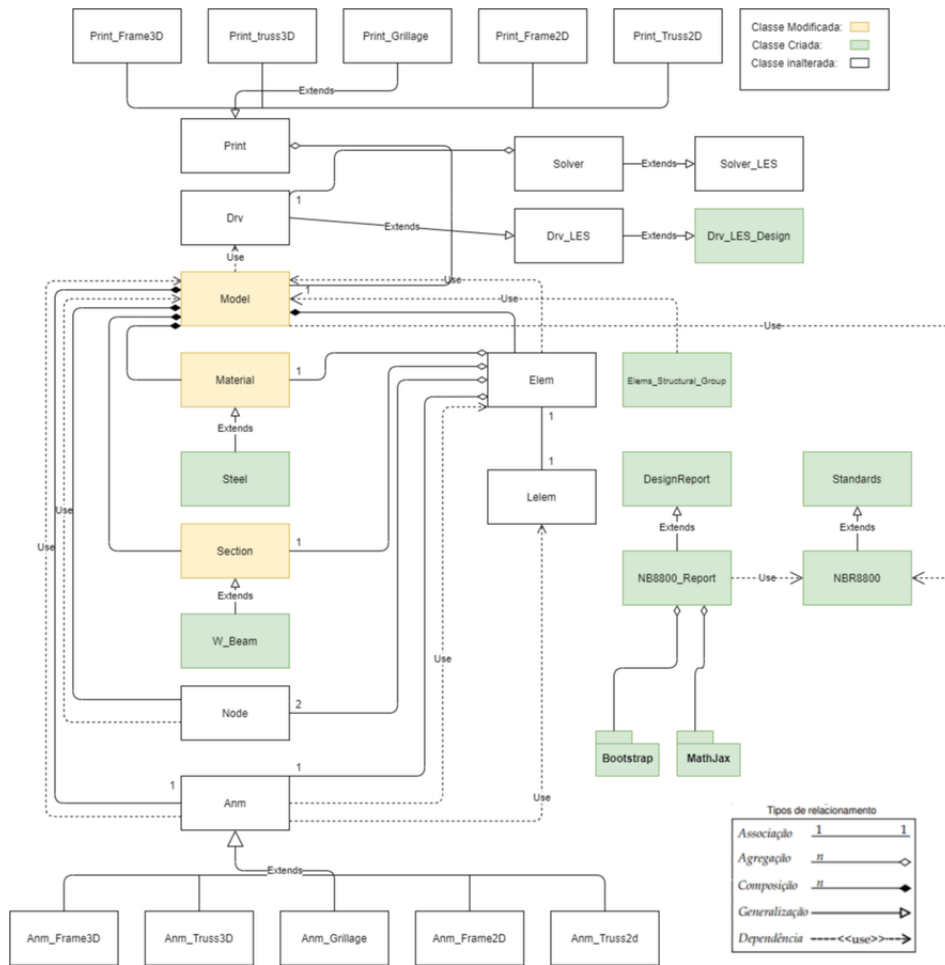


Figura 3.26: [Diagrama de classes da nova estrutura do LESM. Fonte: Autor (2021)]

O diagrama de classes acima oferece uma visão geral dos resultados gerados a partir do desenvolvimento dos processos implementados, descritos nas seções anteriores. Buscou-se apresentar os relacionamentos entre as classes, que enriquecem a documentação do *software* e facilitam o entendimento geral do programa, de uma maneira mais formal e documentada.

Além disso, as setas que conectam as “caixas” do diagrama, representando os tipos de relacionamentos entre as classes, possuem definições claras, especificadas pela UML. Dessa forma, gera-se uma clarificação ainda maior e completa dos fluxos representados pela figura acima. Ademais, para os tipos de relacionamento, mostrados no quadro de legenda da Figura 3.26, tem-se as seguintes definições[13]:

- Associação: Relacionamento estrutural que demonstra que um objeto está conectado a outro. Dessa forma, o usuário é capaz de navegar do objeto de uma classe até o outro e vice-versa.

- Agregação: Associação que demonstra o número de “partes” de outras instâncias, formadoras de uma das extremidades da relação. Geralmente indicada pelo adorno de um número no diagrama.
- Composição: Tipo de relação que demonstra que a instância de uma classe está presente em outra. Caso a classe, na qual a instância foi introduzida, seja deletada ambas são destruídas.
- Generalização: Relacionamento especificado para demonstrar algum tipo de herança entre classes mães e filhas.
- Dependência: Relacionamento que indica que uma alteração em uma classe poderá afetar um outro item que a utiliza.

Vale enfatizar que as duas classes, denominadas DesignReport e Standards, mostradas no diagrama acima, foram criadas apenas para garantir uma melhor concordância em relação ao conceito de abstração. Assim, caso futuramente deseje-se inserir mais normas que compartilhem métodos já existentes, será possível reaproveitá-los para realizar verificações e gerar memoriais de cálculo.

Além do diagrama de relacionamento entre as classes, decidiu-se gerar um fluxograma mais sucinto, expressando através de “caixas” um resumo descrevendo a atividade realizada por cada etapa inserida no programa, mostrado na Figura 3.27.

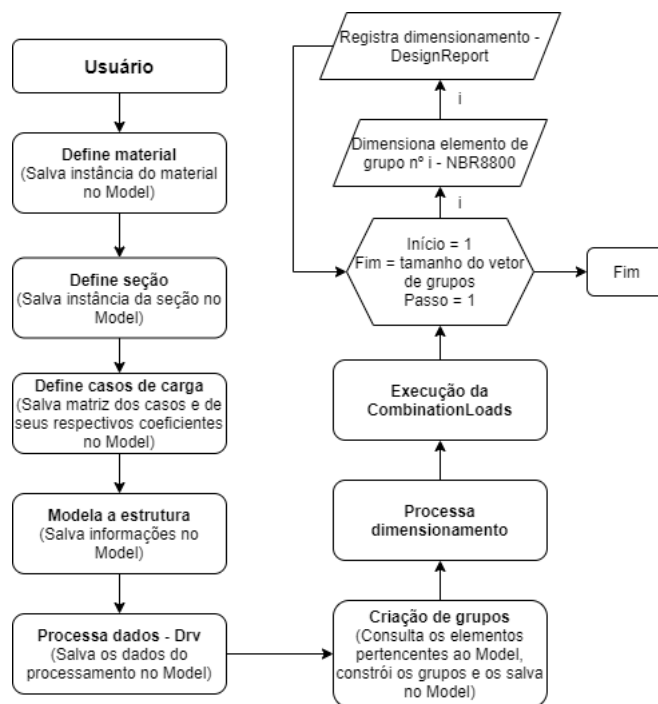


Figura 3.27: [Fluxograma de sequência na nova estrutura do LESM. Fonte: Autor (2021)

3.5 Codificação

A codificação foi a fase em que, após definido todo o relacionamento entre a arquitetura de classes desenvolvida e gerada toda a documentação necessária para guiar o desenvolvimento, começa-se a criar as linhas de código que irão construir de forma utilizável a nova funcionalidade para o usuário. Vale comentar que, mesmo na fase de codificação, existiram momentos em que algumas etapas anteriores precisaram ser revisitadas e modificadas, o que está de acordo com o modelo cascata escolhido.

Após a finalização da fase de codificação, a nova funcionalidade será testada com um exemplo prático, a fim de verificar o atendimento dos objetivos. Essa averiguação será tratada na seção de resultados.

4

Resultados

Este capítulo demonstra a acurácia da nova funcionalidade de dimensionamento, desenvolvida conforme descrito anteriormente. Também será exibido o memorial de cálculo, exportado em HTML.

A fim de validar os valores obtidos pela nova funcionalidade, decidiu-se realizar um método comparativo com outra ferramenta computacional, o CypeCad 2022, obtido em: <http://versoes.cype.pt/avaliacao.htm>. Esse *software* possui funcionalidades para projeto de cálculo estrutural em concreto armado, pré-moldado e em estruturas de aço. Desenvolvido pela Cype, tal ferramenta foi escolhida, pois contém em seu acervo normativo a NBR8800:2008. Além disso, por mais que seja um programa pago, oferece uma versão de avaliação de 10 dias, não necessariamente corridos, a qual foi utilizada para a comparação em questão.

Devido à restrições do próprio CypeCad, definidas no contrato da versão de avaliação, não serão apresentados nessa dissertação nenhuma figura, diagrama ou qualquer outra forma visual que divulgue a documentação, *interface*, elementos e outras partes integrantes do programa.

Para tal comparação entre os resultados obtidos pelos *softwares*, será modelado um pórtico de 3 pavimentos com diferentes casos de carga. Assim, serão analisados todos os esforços mencionados nessa dissertação.

4.1

Modelagem pórtico 2D LESM

O modelo de análise será um pórtico de 3 pavimentos, modelado com 27 instâncias Elems_Structural_Group. O aço utilizado para tal foi o ASTM572. Em relação aos perfis utilizados, as barras enumeradas do 1 ao 15 correspondem ao perfil HP250X62.2, as barras do 16 ao 23 correspondem ao perfil W530X82 e as barras do 24 ao 27 equivalem ao perfil W460x60.

Em relação aos coeficientes utilizados na criação de grupos, dado pela Figura 3.14, foi realizada a seguinte atribuição para as barras do 1 ao 15: Lb será equivalente ao comprimento da barra e o mesmo vale para o comprimento de flambagem. Em relação as barras 16, 19, 20, 23, 24 e 27, o valor de Lb

utilizado foi de 2m. Por fim, para as barras 17, 18, 21, 22, 25 e 26 foi utilizado para Lb o valor de 2,25m. Para os outros fatores, com o intuito de simplificar os cálculos, adotou-se, para todas as barras, os seguintes valores: Cb=1, Lz=1, K_X=1, K_y=1.

Após escolher os materiais, os perfis e montar os grupos de barras, a modelagem resultante no *software* é demonstrada na Figura 4.1.

$$\eta_{[lc,i]} = \left| \frac{Cases_{[lc,i]}}{Verify} \right| \quad (4-1)$$

sendo:

n o fator de utilização;

lc o 'load case' o caso de carga utilizado;

i é referente a seção da viga analisada.

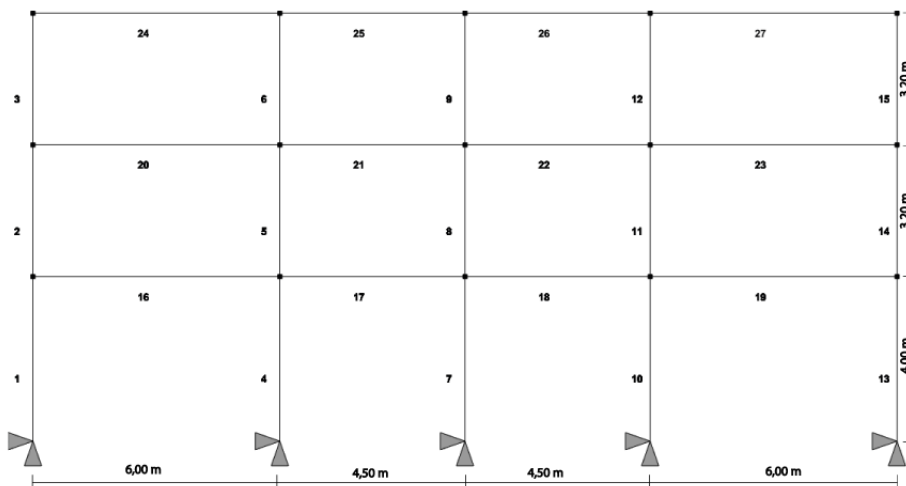


Figura 4.1: Modelagem das barras do pórtico exemplo 2D. Fonte: Autor (2021)

Após a modelagem das barras utilizadas, serão introduzidos os casos de cargas, demonstrados na Tabela 4.1.

Tabela 4.1: Casos de carga pórtico 2D, Fonte: Autor (2021)

Nome	Tipo	Composição	Carregamento	γ	ψ
PP	Permanente	Peso próprio Viga(HP250x62)x4m	-2,5 (kN)	1,25(1,00)	XXX
PP	Permanente	Peso próprio Viga(HP250x62)x3.2m	-2,0 (kN)	1,25(1,00)	XXX
PP	Permanente	Peso próprio Viga(W530x82)	-0,8 (kN)	1,25(1,00)	XXX
PP	Permanente	Peso próprio Viga(W530x60)	-0,6 (kN)	1,25(1,00)	XXX
IMP	Permanente	Impermeabilização	10,5 (kN)	1,4(1,00)	XXX
ARC	Permanente	Alvenaria+revestimento +concreto	32,7 (kN)	1,3(1,00)	XXX
SOS	Variável	Serviços + sobrecarga de cobertura	6 (kN)	1,5	0,8
SOS	Variável	Serviços + sobrecarga de ocupação	33 (kN)	1,5	0,8
V1	Variável	Vento	13,4 (kN)	1,4	0,6
V1	Variável	Vento	11,9 (kN)	1,4	0,6
V1	Variável	Vento	6 (kN)	1,4	0,6

Os carregamentos acima foram modelados no *software* utilizando a nova GUI, dada pela Figura 3.11. Nesse diálogo, cada carga inserida foi previamente caracterizada, de acordo com a norma NBR8800:2008, e os coeficientes de ponderação foram implementados automaticamente. Vale salientar que as cargas que possuem o mesmo nome são pertencentes ao mesmo caso de carga.

Após a criação, os casos de carga ficam disponíveis para aplicação das cargas associadas. Sendo assim, foram adicionadas as cargas à estrutura, conforme mostrado nas Figuras 4.2, 4.3, 4.4, 4.5 e 4.6.

- **Caso de carga PP**

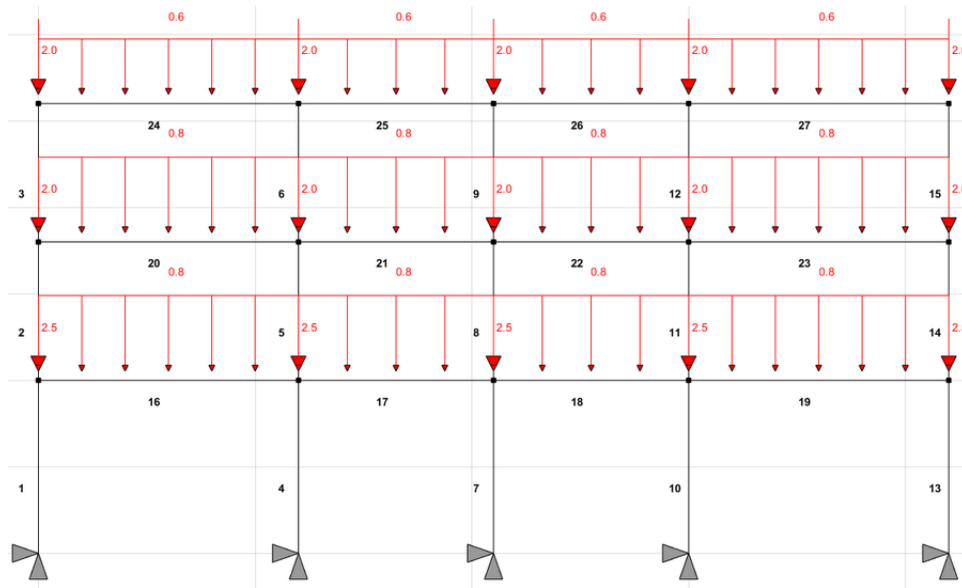


Figura 4.2: Caso de carga PP modelado no LESM. Fonte: Autor (2021)

Para o caso de carga PP, foram adicionadas cargas equivalentes ao peso próprio das vigas modeladas, uma vez que o LESM ainda não contabiliza automaticamente esse carregamento. Para as colunas, inseriu-se carregamentos nodais. Já para as vigas, foram adicionados carregamentos distribuídos ao longo da mesma.

- Caso de carga IMP

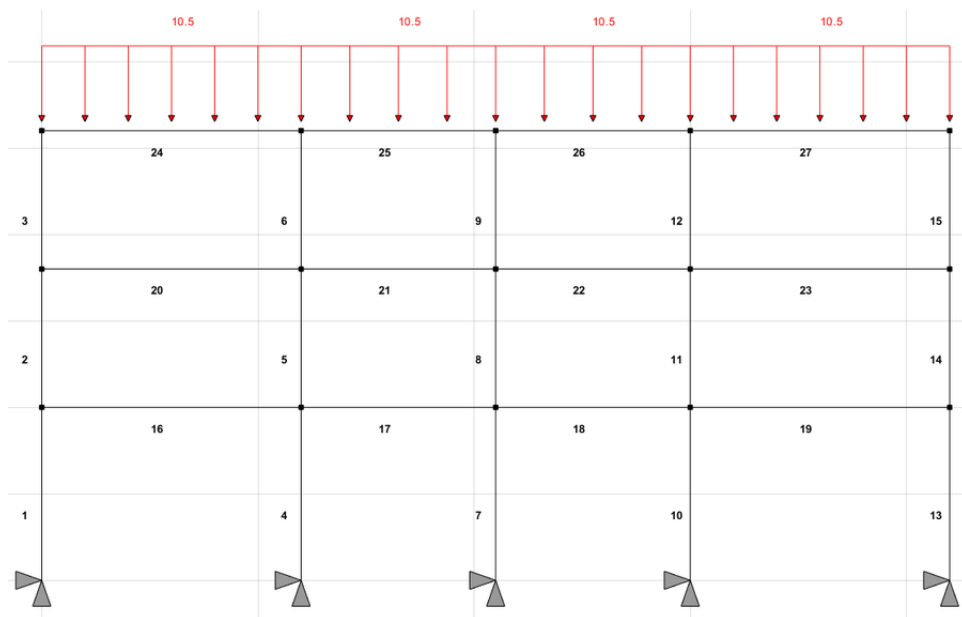


Figura 4.3: Caso de carga IMP modelado no LESM. Fonte: Autor (2021)

Para a carga referente à impermeabilização, considerou-se um carregamento de $1,75\text{kN/m}^2$, multiplicado por uma área de influência de 6m , resultando no carregamento modelado de $10,5\text{ kN/m}$. Esse carregamento foi apenas utilizado no pavimento de cobertura.

- **Caso de carga ARC**

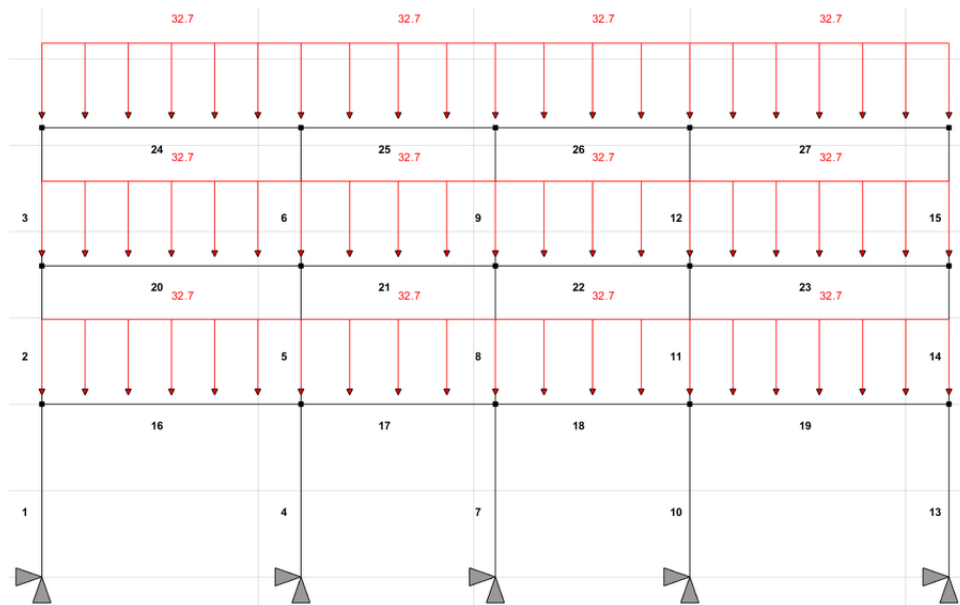


Figura 4.4: Caso de carga ARC modelado no LESM. Fonte: Autor (2021)

O caso de carga, nomeado como ARC, é dado por: alvenaria, possuindo $1,3\text{kN/m}^2$ de carga, revestimento, com carga equivalente a $1,4\text{ kN/m}^2$ e peso próprio do concreto da laje com o carregamento de $2,75\text{ kN/m}^2$. Esses carregamentos foram somados e multiplicados por uma área de influência de 6m , resultando em um carregamento de $32,7\text{ kN/m}$, replicados nas vigas de todos os pavimentos.

- **Caso de carga SOS**

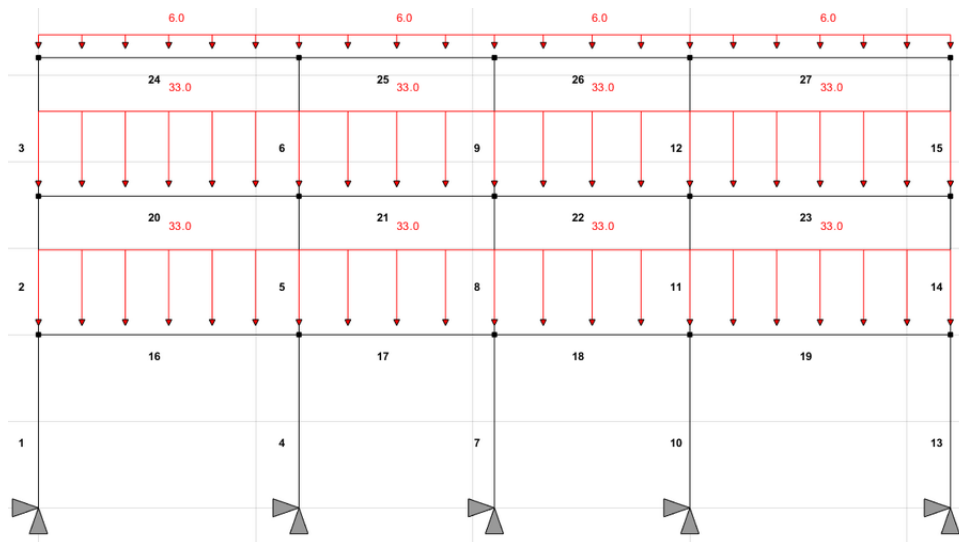


Figura 4.5: Caso de carga SOS modelado no LESM. Fonte: Autor (2021)

Esse caso de carga é composto por sobrecarga de ocupação, com carregamento equivalente a $5,0 \text{ kN/m}^2$ e serviço equivalente a $0,5 \text{ kN/m}^2$. Esses dados foram então somados e multiplicados pela área de influência de 6m , resultando em 33kN/m , aplicados no primeiro e segundo pavimento.

A carga aplicada na cobertura é composta pela carga de serviço somada a sobrecarga de ocupação da cobertura, com carga de $0,5 \text{ kN/m}^2$. Multiplicando pela mesma área de influência, resulta em 6 kN/m .

• Caso de carga V1

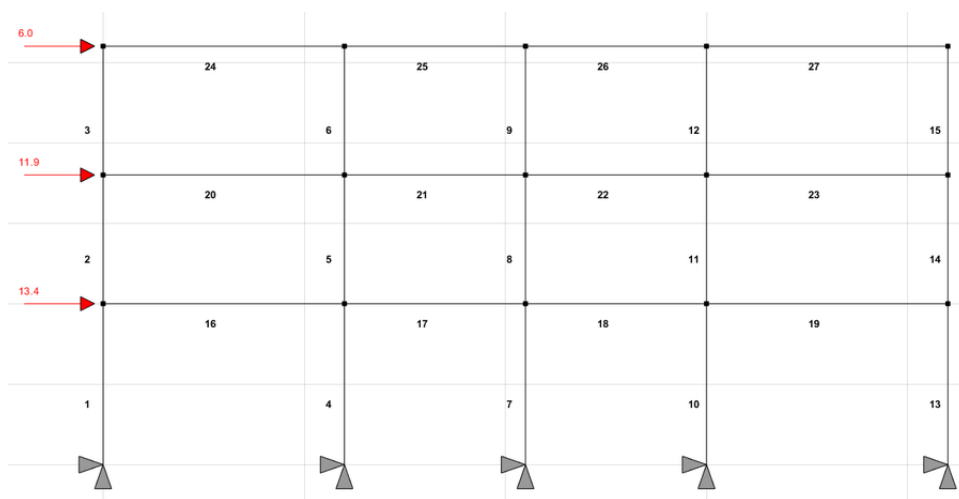


Figura 4.6: Caso de carga V1 modelado no LESM. Fonte: Autor (2021)

O carregamento V1 é responsável pelo vento aplicado na estrutura, esse vento foi “simplificado” para ser introduzido como carga nodal na lateral esquerda do pórtico, sendo os seguintes valores: 13,4 kN, 11 kN e 6kN.

4.2

Modelagem pórtico 2D CypeCad

O modelo exemplo, desenvolvido na Seção 4.1, foi reproduzido fielmente utilizando a interface do CypeCad. Toda a estrutura foi replicada utilizando as mesmas dimensões, materiais, perfis e apoios.

Em relação ao carregamento aplicado, todos os casos de carga também foram reproduzidos de forma idêntica, com exceção do carregamento de peso próprio. Esse fato ocorreu, pois o CypeCad monta automaticamente o caso de carga referente a peso próprio dos perfis utilizados.

Após a modelagem, através do CypeCad, utilizou-se a função “calcular” do *software*, selecionando a opção de não dimensionar perfis. Nessa opção, todos os esforços são calculados, além de exibir todas as diretrizes da norma utilizada para realizar a verificação das barras. Ao final, ainda é exibido se o perfil resiste ou não ao esforço solicitante. Vale ressaltar que os valores coletados para a análise de resultados foram os referentes ao E.L.U.

4.3

Análise numérica dos resultados obtidos pelos *softwares*

Para realizar um controle da acurácia e confiabilidade dos resultados atingidos na nova extensão do *software* LESM, coletou-se os dados obtidos na verificação das barras por ambos os *softwares*. Para essa análise, decidiu-se comparar o caso de carga crítico, selecionado pelos programas, o valor solicitante para esse caso de carga e o valor resistente do perfil. Esse método comparativo será feito separadamente para cada tipo de esforço com o intuito de analisar possíveis divergências para cada caso.

Vale salientar que a métrica de acurácia utilizada para comparar as grandezas encontradas em ambos os *softwares* é o erro relativo entre elas.

4.3.1

Análise dos valores de tração

Para os esforços de tração nos elementos do exemplo, utilizou-se para a comparação dos resultados, obtidos pelo LESM e pelo CypeCad, os valores de força axial solicitante e força axial resistente de projeto, assim como a composição da combinação do caso de carga crítico, demonstrados na Tabela 4.2.

Tabela 4.2: Comparação dos resultados dos esforços de tração entre os programas CypeCad e LESM, Fonte: Autor (2021)

Tração								
Elementos	Combinação LESM	Combinação CypeCad	Nc, Sd-LES (kN)	Nc, Sd-CypeCad (kN)	E%	Nc, Rd-LES (kN)	Nc, Rd-CypeCad (kN)	E%
Elemento 1	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 2	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 3	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 4	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 5	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 6	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 7	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 8	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 9	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 10	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 11	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 12	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 13	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 14	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 15	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 16	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	47,499	44,590	6,520	3277,500	3277,500	0,000
Elemento 17	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	30,014	28,350	5,870	3277,500	3277,500	0,000
Elemento 18	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	30,014	28,350	5,870	3277,500	3277,500	0,000
Elemento 19	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	47,499	44,590	6,520	3277,500	3277,500	0,000

Tração								
Elementos	Combinação LESM	Combinação CypeCad	Nc, Sd- LESM (kN)	Nc, Sd- CypeCad (kN)	E%	Nc, Rd- LESM (kN)	Nc, Rd- CypeCad (kN)	E%
Elemento 20	1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,3*ARC	19,350	18,430	4,990	3277,500	3277,500	0,000
Elemento 21	1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,3*ARC	11,056	10,670	3,620	3277,500	3277,500	0,000
Elemento 22	1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,3*ARC	11,056	10,670	3,620	3277,500	3277,500	0,000
Elemento 23	1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,3*ARC	19,350	18,430	4,990	3277,500	3277,500	0,000
Elemento 24	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 25	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 26	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX
Elemento 27	perfil não tracionado	perfil não tracionado	XXX	XXX	XXX	XXX	XXX	XXX

Ao analisar as grandezas referentes à tração, nota-se uma total compatibilidade entre os elementos do 16 ao 23, que possuem tração. Além disso, percebe-se também uma compatibilidade quase unânime entre os casos de carga crítico escolhido por ambos os *softwares*, exceto pelo elemento 19. Nesse caso, a parcela de impermeabilização no LESM é multiplicada por 1,4, enquanto que no CypeCad isso não ocorre. Possivelmente esse fato se deve a uma discrepância numérica entre as forças axiais solicitantes encontradas para as combinações desse elemento. Criando, assim, uma diferenciação na escolha do caso mais crítico pelos programas.

Em relação à força axial resistente, tem-se um erro reativo de 0%, expressando um total acordo entre a metodologia de obtenção desse valor pelos *softwares*.

4.3.2 Análise dos valores de flexão

Para os esforços de flexão utilizou-se, para a comparação dos resultados entre os programas, a composição da combinação do caso de carga crítico, além dos valores de momento solicitante de projeto e momento resistente de projeto, mostrados na Tabela 4.3. Esses dados foram separados dos valores obtidos para

cada tipo de estado limite (FLT, FLM e FLA), evidenciados, respectivamente nas Tabelas 4.4, 4.5 e 4.6 para facilitar a visualização dos resultados.

Ademais, a divisão das análises dos esforços para cada tipo de estado limite (FLT, FLM e FLA,) deu-se visando discernir de forma clara os possíveis motivos para as discrepâncias entre os valores obtidos pelos programas.

Tabela 4.3: Comparação do momento solicitante entre os programas CypeCad e LESM, Fonte: Autor (2021)

Momento Solicitante					
Elementos	Combinação LESM	Combinação CypeCad	MSd-LESM (kN)	MSd-CypeCad (kN)	E%
Elemento 1	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,5*SOS+1,3ARC	42,878	44,310	3,230
Elemento 2	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,5*SOS+1,3ARC	96,695	92,170	4,910
Elemento 3	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,5*SOS+1,3ARC	112,364	108,130	3,920
Elemento 4	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,4*V1+ 1,2*SOS+1,3ARC	50,169	50,540	0,730
Elemento 5	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+1,4*V1+ 1,2*SOS+1,3ARC	41,402	39,370	5,160
Elemento 6	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,4*V1+ 1,2*SOS+1,3ARC	43,707	41,620	5,010
Elemento 7	1,4*V1 1,25*PP 1,3*ARC 1,4*IMP	PP+IMP+1,4*V1+ARC	36,342	36,342	0,000
Elemento 8	1,4*V1 1,25*PP 1,3*ARC 1,4*IMP	PP+IMP+1,4*V1+ARC	10,810	10,730	0,750
Elemento 9	1,4*V1 1,25*PP 1,3*ARC 1,4*IMP	PP+IMP+1,4*V1+ARC	3,468	3,450	0,510
Elemento 10	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	PP+IMP+1,4*V1+ARC	31,180	31,070	0,350
Elemento 11	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+1,5*SOS+1,3ARC	35,456	33,470	5,940
Elemento 12	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,5*SOS+1,3ARC	70,874	72,150	1,770
Elemento 13	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,4*V1+ 1,2*SOS+1,3ARC	98,229	93,800	4,720
Elemento 14	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	113,671	109,480	3,830
Elemento 15	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	294,665	290,490	1,440

Momento Solicitante					
Elementos	Combinação LESM	Combinação CypeCad	MSd- LESM (kN)	MSd- CypeCad (kN)	E%
Elemento 16	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	229,253	226,710	1,120
Elemento 17	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+1,5*SOS+1,3ARC	242,229	239,560	1,110
Elemento 18	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	279,655	275,520	1,500
Elemento 19	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+1,5*SOS+1,3ARC	266,698	266,140	0,210
Elemento 20	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	266,698	266,140	0,210
Elemento 21	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+1,5*SOS+1,3ARC	194,318	198,380	2,050
Elemento 22	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	198,701	202,640	1,940
Elemento 23	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+1,5*SOS+1,3ARC	262,727	262,090	0,240
Elemento 24	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	196,902	195,310	0,820
Elemento 25	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,5*SOS+1,3ARC	154,245	154,730	0,310
Elemento 26	0,6*V1 1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+IMP+0,84*V1+ 1,5*SOS+1,3ARC	155,307	155,790	0,310
Elemento 27	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP+1,4*IMP+1,5*SOS+1,3ARC	195,971	194,350	0,830

Analisando a combinação de ação crítica dos elementos, escolhidas pelos *softwares*, observa-se desconformidades mais relevantes nos elementos de 7 a 10, visto que a composição do caso de carga se difere totalmente. Contudo, ao verificar mais a fundo, nota-se que esse fato não é um erro na escolha, mas sim na representação gráfica de saída do LESM. Nesse caso, os valores de momento dos casos PP, ARC e IMP são 0. Assim, independente do valor que os multiplica, o resultado será sempre nulo, sendo apenas a parcela V1 responsável por influenciar o resultado final encontrado. Portanto, o dado da combinação escolhida obtido é compatível em ambos os programas, havendo somente a falta de um tratamento na saída gráfica para esse tipo de situação no LESM.

Para os demais casos, as discrepâncias encontradas na escolha de

combinações críticas se deve, possivelmente, a erros semelhantes ao descrito acima ou pequenos desajustes de escolha de coeficientes na parcela de vento (V1).

Para o valor do momento solicitante, a maior porcentagem de erro relativo é de 5,16%. Provavelmente, esses erros apresentados se devem a pequenas diferenciações na metodologia de obtenção de esforços entre os *softwares* e/ou alguns erros numéricos na consideração de casas decimais.

Tabela 4.4: Comparação do momento resistente para o estado limite FLT entre os programas CypeCad e LESM, Fonte: Autor (2021)

Momento Resistente (FLT)			
Elementos	Mrd-FLT-LESM (kN)	Mrd-FLT-CypeCad (kN)	E%
Elemento 1	242,280	226,490	6,970
Elemento 2	245,580	238,730	2,870
Elemento 3	245,580	238,730	2,870
Elemento 4	242,280	226,490	6,970
Elemento 5	245,580	238,730	2,870
Elemento 6	245,580	238,730	2,870
Elemento 7	242,280	226,490	6,970
Elemento 8	245,580	238,730	2,870
Elemento 9	245,580	238,730	2,870
Elemento 10	242,280	226,490	6,970
Elemento 11	245,580	238,730	2,870
Elemento 12	245,580	238,730	2,870
Elemento 13	242,280	226,490	6,970
Elemento 14	245,580	238,730	2,870
Elemento 15	245,580	238,730	2,870
Elemento 16	643,140	635,960	1,130
Elemento 17	637,400	317,840	3,170
Elemento 18	637,400	317,840	3,170
Elemento 19	643,140	635,960	1,130
Elemento 20	643,140	635,960	1,130
Elemento 21	637,400	317,840	3,170
Elemento 22	637,400	317,840	3,170
Elemento 23	643,140	635,960	1,130
Elemento 24	393,350	367,270	7,100
Elemento 25	388,510	352,210	10,310

Momento Resistente (FLT)			
Elementos	Mrd-FLT-LESM (kN)	Mrd-FLT-CypeCad (kN)	E%
Elemento 26	388,510	352,210	10,310
Elemento 27	393,350	367,270	7,100

Em relação ao momento resistente encontrado para estado limite FLT, para ambos os *softwares*, nota-se uma discrepância considerável, com erro relativo máximo de 10,31%, referentes aos elementos 25 e 26. Como a obtenção desses valores é dependente de propriedades geométricas de cada tipo de perfil, especula-se que exista um possível desalinhamento dos fatores utilizados para se obter λ_r , dado pela equação 2-8, ou por M_{cr} , dado pela equação 2-10, podendo ocasionar nessa divergência.

Tabela 4.5: Comparação do momento resistente para o estado limite FLM entre os programas CypeCad e LESM, Fonte: Autor (2021)

Momento Resistente (FLM)			
Elementos	Mrd-FLM-LESM (kN)	Mrd-FLM-CypeCad (kN)	E%
Elemento 1	231,260	230,340	0,400
Elemento 2	231,260	230,340	0,400
Elemento 3	231,260	230,340	0,400
Elemento 4	231,260	230,340	0,400
Elemento 5	231,260	230,340	0,400
Elemento 6	231,260	230,340	0,400
Elemento 7	231,260	230,340	0,400
Elemento 8	231,260	230,340	0,400
Elemento 9	231,260	230,340	0,400
Elemento 10	231,260	230,340	0,400
Elemento 11	231,260	230,340	0,400
Elemento 12	231,260	230,340	0,400
Elemento 13	231,260	230,340	0,400
Elemento 14	231,260	230,340	0,400
Elemento 15	231,260	230,340	0,400
Elemento 16	645,620	645,620	0,000
Elemento 17	645,620	645,620	0,000
Elemento 18	645,620	645,620	0,000
Elemento 19	645,620	645,620	0,000
Elemento 20	645,620	645,620	0,000

Momento Resistente (FLM)			
Elementos	Mrd-FLM-LESM (kN)	Mrd-FLM-CypeCad (kN)	E%
Elemento 21	645,620	645,620	0,000
Elemento 22	645,620	645,620	0,000
Elemento 23	645,620	645,620	0,000
Elemento 24	645,620	645,620	0,000
Elemento 25	645,620	645,620	0,000
Elemento 26	645,620	645,620	0,000
Elemento 27	645,620	645,620	0,000

Analisando o valor de Mrd para o estado limite de FLM, percebe-se que, do elemento 1 ao 15, houve um pequeno erro, de 0,40% entre os resultados obtidos pelos *softwares*. Já do elemento 16 ao 27 observa-se um erro relativo de 0%. Ao analisar esses dados, deduz-se que o tipo de perfil que compõe os elementos detentores do erro, HP250X62.0, deve possuir uma pequena divergência geométrica na lista de propriedades do perfil disponibilizada por cada programa, o que não ocorre para as barras após a 15.

Tabela 4.6: Comparação do momento resistente para o estado limite FLA entre os programas CypeCad e LESM, Fonte: Autor (2021)

Momento Resistente (FLA)			
Elementos	Mrd-FLA-LESM (kN)	Mrd-FLA-CypeCad (kN)	E%
Elemento 1	247,930	247,930	0,000
Elemento 2	247,930	247,930	0,000
Elemento 3	247,930	247,930	0,000
Elemento 4	247,930	247,930	0,000
Elemento 5	247,930	247,930	0,000
Elemento 6	247,930	247,930	0,000
Elemento 7	247,930	247,930	0,000
Elemento 8	247,930	247,930	0,000
Elemento 9	247,930	247,930	0,000
Elemento 10	247,930	247,930	0,000
Elemento 11	247,930	247,930	0,000
Elemento 12	247,930	247,930	0,000
Elemento 13	247,930	247,930	0,000
Elemento 14	247,930	247,930	0,000
Elemento 15	247,930	247,930	0,000

Momento Resistente (FLA)			
Elementos	Mrd-FLA-LESM (kN)	Mrd-FLA-CypeCad (kN)	E%
Elemento 16	645,620	645,620	0,00
Elemento 17	645,620	645,620	0,00
Elemento 18	645,620	645,620	0,00
Elemento 19	645,620	645,620	0,00
Elemento 20	645,620	645,620	0,00
Elemento 21	645,620	645,620	0,00
Elemento 22	645,620	645,620	0,00
Elemento 23	645,620	645,620	0,00
Elemento 24	645,620	645,620	0,00
Elemento 25	645,620	645,620	0,00
Elemento 26	645,620	645,620	0,00
Elemento 27	645,620	645,620	0,00

Sobre o valor de Mrd encontrado para ambos os *softwares*, constatou-se uma compatibilidade de 100% entre os valores obtidos, confirmando que, para a rotina de obtenção desse dado, os *softwares* estão em total acordo, para o caso estudado.

4.3.3 Análise dos valores de compressão

Como pode-se observar na Tabela 4.7, assim como para tração, para os esforços de compressão, utilizou-se os valores de força axial solicitante e força axial resistente de projeto para a comparação dos resultados, obtidos pelo LESM e pelo CypeCad. Também analisou-se a composição da combinação do caso de carga crítico.

Tabela 4.7: Comparação dos resultados dos esforços de compressão entre os programas CypeCad e LESM, Fonte: Autor (2021)

Compressão								
Elementos	Combinação LEMS	Combinação CypeCad	Nc,Sd-LESM (kN)	Nc,Sd-CypeCad (kN)	E%	Nc,Rd-LESM (kN)	Nc,Rd-CypeCad (kN)	E%
Elemento 1	1,5*SOS	1,25*PP	717,056	715,420	0,230	1843,000	1829,110	0,760
	1,25*PP	1,4*IMP						
	1,3*ARC	1,5*SOS						
	1,4*IMP	1,3*ARC						
Elemento 2	1,5*SOS	1,25*PP	458,198	456,430	0,390	2055,800	2045,860	0,490
	1,25*PP	1,4*IMP						
	1,3*ARC	1,5*SOS						
	1,4*IMP	1,3*ARC						

Compressão								
Elementos	Combinação LEMS	Combinação CypeCad	Nc,Sd-LESM (kN)	Nc,Sd-CypeCad (kN)	E%	Nc,Rd-LESM (kN)	Nc,Rd-CypeCad (kN)	E%
Elemento 3	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	189,426	188,910	0,270	2055,800	2045,860	0,490
Elemento 4	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	1427,300	1426,910	0,030	1843,000	1829,110	0,760
Elemento 5	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	891,166	892,890	0,190	2055,800	2045,860	0,490
Elemento 6	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	379,140	379,280	0,040	2055,800	2045,860	0,490
Elemento 7	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	1065,300	1067,660	0,220	1843,000	1829,110	0,760
Elemento 8	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	685,968	685,040	0,140	2055,800	2045,860	0,490
Elemento 9	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	281,430	281,680	0,090	2055,800	2045,860	0,490
Elemento 10	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	1427,700	1427,190	0,040	1843,000	1829,110	0,760
Elemento 11	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	891,663	893,290	0,180	2055,800	2045,860	0,490
Elemento 12	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	379,227	379,360	0,040	2055,800	2045,860	0,490
Elemento 13	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	724,928	723,360	0,220	1843,000	1829,110	0,760
Elemento 14	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	460,075	458,360	0,370	2055,800	2045,860	0,490

Compressão								
Elementos	Combinação LEMS	Combinação CypeCad	Nc,Sd- LESM (kN)	Nc,Sd- CypeCad (kN)	E%	Nc,Rd- LESM (kN)	Nc,Rd- CypeCad (kN)	E%
Elemento 15	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	189,806	189,300	0,270	2055,800	2045,860	0,490
Elemento 16	1,25*PP 1,3*ARC 1,4*IMP	Não calculado	0,000	0,000	0,000	908,710	886,550	2,500
Elemento 17	1,25*PP 1,3*ARC 1,4*IMP	1,0*PP 1,0*IMP 1,4*V1 1,0*ARC	0,000	0,520	0,000	1511,200	1474,150	2,510
Elemento 18	1,25*PP 1,3*ARC 1,4*IMP	Não calculado	0,000	0,000	0,000	1511,200	1474,150	2,510
Elemento 19	1,25*PP 1,3*ARC 1,4*IMP	Não calculado	0,000	0,000	0,000	908,710	886,550	2,500
Elemento 20	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,0*PP 1,0*IMP 1,4*V1 1,2*SOS 1,0*ARC	10,469	10,320	1,440	908,710	886,550	2,500
Elemento 21	1,4*V1 0,8*SOS 1,0*PP 1,0*ARC 1,0*IMP	1,0*PP 1,0*IMP 1,4*V1 1,2*SOS 1,0*ARC	7,890	7,810	1,020	1511,200	1474,150	2,510
Elemento 22	1,4*V1 0,8*SOS 1,0*PP 1,0*ARC 1,0*IMP	1,0*PP 1,0*IMP 1,4*V1 1,2*SOS 1,0*ARC	3,674	3,670	0,110	1511,200	1474,150	2,510
Elemento 23	1,25*PP 1,3*ARC 1,4*IMP	Não calculado	0,000	0,000	0,000	908,710	886,550	2,500
Elemento 24	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	69,525	66,800	4,080	356,670	347,970	2,500
Elemento 25	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	44,080	42,690	3,260	634,090	618,620	2,500
Elemento 26	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	42,836	41,470	3,290	634,090	618,620	2,500
Elemento 27	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	65,907	63,190	4,300	356,670	347,970	2,500

Em relação a carga solicitante de compressão, houve uma discrepância, com erro relativo máximo de 4,08%. Acredita-se que essa divergência possa ter sido gerada por uma falta de alinhamento entre os *softwares* na metodologia de obtenção do esforços.

Em relação ao N_c, R_d houve uma pequena diferença, com erro relativo máximo de 2,50%. Nesse caso, presume-se que possa haver discordância no cálculo dos coeficientes, utilizados na equação 2-33, visto que a metodologia de obtenção do valor resistente de compressão, dada pela norma, possui uma grande quantidade de passos. Assim, essa etapa mostra-se mais suscetível a possíveis erros, propagados ao decorrer da metodologia.

Para as combinações críticas, observa-se poucas divergências. Contudo, para as que ocorreram, acredita-se que sua causa provável seja devido ao erro de saída gráfica no LESM, elucidado na Seção 4.3.2.

4.3.4 Análise dos valores de cisalhamento

Na Tabela 4.8, tem-se que, para os esforços de cisalhamento, além da composição da combinação do caso de carga crítico, utilizaram-se também os valores da carga de cisalhamento solicitante e carga de cisalhamento resistente de projeto para a comparação dos resultados, obtidos pelo LESM e pelo CypeCad.

Tabela 4.8: Comparação dos resultados dos esforços de cisalhamento entre os programas CypeCad e LESM, Fonte: Autor (2021)

Cisalhamento								
Elementos	Combinação LEMS	Combinação CypeCad	Vc,Sd- LESM (kN)	Vc,Sd- CypeCad (kN)	E%	Vc,Rd- LESM (kN)	Vc,Rd- CypeCad (kN)	E%
Elemento 1	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	10,729	11,080	3,250	486,070	486,070	0,000
Elemento 2	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	58,219	55,650	4,620	486,070	486,070	0,000
Elemento 3	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	65,191	62,470	4,360	486,070	486,070	0,000
Elemento 4	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,4*V1 1,5*SOS 1,3*ARC	12,542	12,630	0,700	486,070	486,070	0,000

Cisalhamento								
Elementos	Combinação LEMS	Combinação CypeCad	Vc,Sd- LESM (kN)	Vc,Sd- CypeCad (kN)	E%	Vc,Rd- LESM (kN)	Vc,Rd- CypeCad (kN)	E%
Elemento 5	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,4*V1 1,5*SOS 1,3*ARC	25,329	24,230	4,530	486,070	486,070	0,000
Elemento 6	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	25,445	24,110	5,540	486,070	486,070	0,000
Elemento 7	1,4*V1 1,25*PP 1,3*ARC 1,4*IMP	1,0*PP 1,0*IMP 1,4*V1 1,0*ARC	9,085	9,090	0,050	486,070	486,070	0,000
Elemento 8	1,4*V1 1,25*PP 1,3*ARC 1,4*IMP	1,0*PP 1,0*IMP 1,4*V1 1,0*ARC	6,291	6,170	1,970	486,070	486,070	0,000
Elemento 9	1,4*V1 1,25*PP 1,3*ARC 1,4*IMP	1,0*PP 1,0*IMP 1,4*V1 1,0*ARC	2,073	2,040	1,620	486,070	486,070	0,000
Elemento 10	1,0*PP 1,0*IMP 1,4*V1 1,0*ARC	1,0*PP 1,0*IMP 1,4*V1 1,0*ARC	7,795	7,770	0,320	486,070	486,070	0,000
Elemento 11	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	21,221	20,080	5,680	486,070	486,070	0,000
Elemento 12	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	24,276	22,930	5,870	486,070	486,070	0,000
Elemento 13	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,4*V1 1,5*SOS 1,3*ARC	17,719	18,040	1,780	486,070	486,070	0,000
Elemento 14	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	59,935	57,470	4,290	486,070	486,070	0,000
Elemento 15	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	65,907	63,190	4,300	486,070	486,070	0,000
Elemento 16	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	308,479	308,260	0,070	943,920	943,920	0,000

Cisalhamento								
Elementos	Combinação LEMS	Combinação CypeCad	Vc,Sd- LESM (kN)	Vc,Sd- CypeCad (kN)	E%	Vc,Rd- LESM (kN)	Vc,Rd- CypeCad (kN)	E%
Elemento 17	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	230,565	228,910	0,720	943,920	943,920	0,000
Elemento 18	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	236,493	234,790	0,730	943,920	943,920	0,000
Elemento 19	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	302,452	302,240	0,070	943,920	943,920	0,000
Elemento 20	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	293,383	293,960	0,200	943,920	943,920	0,000
Elemento 21	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	217,629	218,640	0,460	943,920	943,920	0,000
Elemento 22	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	219,535	220,510	0,440	943,920	943,920	0,000
Elemento 23	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	291,918	293,920	0,680	943,920	943,920	0,000
Elemento 24	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	215,174	215,570	0,180	684,980	684,980	0,000
Elemento 25	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	161,845	161,630	0,130	684,980	684,980	0,000
Elemento 26	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	162,100	162,313	0,130	684,980	684,980	0,000
Elemento 27	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	214,814	215,200	0,180	684,980	684,980	0,000

Analisando o valor de Vc,Sd, nota-se uma discrepância máxima de 5,68%, podendo também estar relacionada a algum desalinhamento na obtenção desse esforço pelo *software* a ou pequenos erros numéricos computacionais. Para Vc,Rd, percebe-se um erro relativo de 0%, configurando um alinhamento perfeito entre os *softwares* sobre a rotina de cálculo desse valor.

4.3.5

Análise dos valores de flexo-tração e flexo-compressão

Para os esforços de flexo-tração e flexo-compressão, demonstrados na Tabela 4.9, utilizou-se das composições das combinações do caso de carga crítico, assim como dos valores da força cortante solicitante, força cortante resistente de projeto e taxa de utilização para a comparação dos resultados, obtidos pelos programas.

Tabela 4.9: Comparação dos resultados dos esforços de flexo-tração e flexo-compressão entre os programas CypeCad e LESM, Fonte: Autor (2021)

Flexo-Tração e Flexo-Compressão					
Elementos	Combinação LEMS	Combinação CypeCad	n-LESM (kN)	n-CypeCad (kN)	E%
Elemento 1	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	0,554	0,563	1,620
Elemento 2	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	0,595	0,579	2,690
Elemento 3	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	0,532	0,515	3,300
Elemento 4	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,917	0,925	0,920
Elemento 5	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,592	0,587	0,900
Elemento 6	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,281	0,273	3,00

Flexo-Tração e Flexo-Compressão					
Elementos	Combinação LEMS	Combinação CypeCad	n-LESM (kN)	n-CypeCad (kN)	E%
Elemento 7	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,4*V1 1,5*SOS 1,3*ARC	0,669	0,675	0,900
Elemento 8	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,359	0,358	0,170
Elemento 9	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,4*V1 1,5*SOS 1,3*ARC	0,081	0,081	0,000
Elemento 10	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	0,832	0,839	0,850
Elemento 11	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	0,570	0,565	0,850
Elemento 12	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,5*SOS 1,3*ARC	0,273	0,264	3,260
Elemento 13	1,4*V1 0,8*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,4*IMP 1,4*V1 1,5*SOS 1,3*ARC	0,636	0,647	1,730
Elemento 14	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,601	0,586	2,630
Elemento 15	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,538	0,521	3,210

Flexo-Tração e Flexo-Compressão					
Elementos	Combinação LEMS	Combinação CypeCad	n-LESM (kN)	n-CypeCad (kN)	E%
Elemento 16	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,458	0,462	0,820
Elemento 17	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	0,360	0,371	3,050
Elemento 18	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,384	0,391	1,820
Elemento 19	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	0,442	0,440	0,480
Elemento 20	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,416	0,421	1,240
Elemento 21	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	0,312	0,321	2,870
Elemento 22	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,312	0,328	4,970
Elemento 23	1,5*SOS 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 1,5*SOS 1,3*ARC	0,409	0,413	1,090
Elemento 24	1,5*SOS 0,6*V1 1,25*PP 1,3*ARC 1,4*IMP	1,25*PP 1,0*IMP 0,84*V1 1,5*SOS 1,3*ARC	0,598	0,628	4,780

Flexo-Tração e Flexo-Compressão					
Elementos	Combinação LEMS	Combinação CypeCad	n-LESM (kN)	n-CypeCad (kN)	E%
Elemento 25	1,5*SOS	1,25*PP	0,429	0,471	8,850
	1,25*PP	1,4*IMP			
	1,3*ARC	1,5*SOS			
	1,4*IMP	1,3*ARC			
Elemento 26	1,5*SOS	1,25*PP	0,433	0,476	8,930
	0,6*V1	1,0*IMP			
	1,25*PP	0,84*V1			
	1,3*ARC	1,5*SOS			
Elemento 27	1,5*SOS	1,25*PP	0,590	0,619	4,750
	1,25*PP	1,4*IMP			
	1,3*ARC	1,5*SOS			
	1,4*IMP	1,3*ARC			

Em relação à comparação da taxa de utilização, registrou-se o erro relativo máximo de 8,85%. Visto que a análise dos esforços de flexo-tração e flexo-compressão utiliza-se dos valores solicitantes axiais e de flexão, possivelmente, esses e os demais erros encontrados, são causados devido aos motivos citados anteriormente nas seções 4.3.1, 4.3.3 e 4.3.2. Dessa forma, a junção das divergências desses valores solicitantes, ocasionaram, provavelmente, os erros relativos apontados na Tabela 4.9 acima.

4.4

Exibição de resultados para o usuário

Este capítulo mostra as diferentes maneiras em que o usuário poderá consultar os resultados obtidos pelo programa, após a verificação dos perfis para cada tipo de esforço abordado na dissertação, com exceção dos de flexo-compressão e flexo-tração. Visto que, até o presente momento, apesar de o programa calcular e gerar os valores dessa verificação, o memorial de cálculo ainda não contempla visualmente esses resultados. Dessa forma, apenas podem ser acessados através do terminal do MATLAB.

4.4.1

Memorial de cálculo

Como dito nas seções anteriores, assim que as verificações são executadas é gerado um memorial de cálculo em HTML com todo o procedimento utilizado. Esse documento é gerado na própria pasta em que o LESM se

encontra. Dessa forma, o usuário pode acessá-lo e abri-lo com o navegador de sua preferência.

Para demonstrar a *layout*, realizou-se um recorte do memorial de cálculo do elemento 16, contido no exemplo modelado anteriormente. As Figuras 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 e 4.16 elucidam esse recorte.

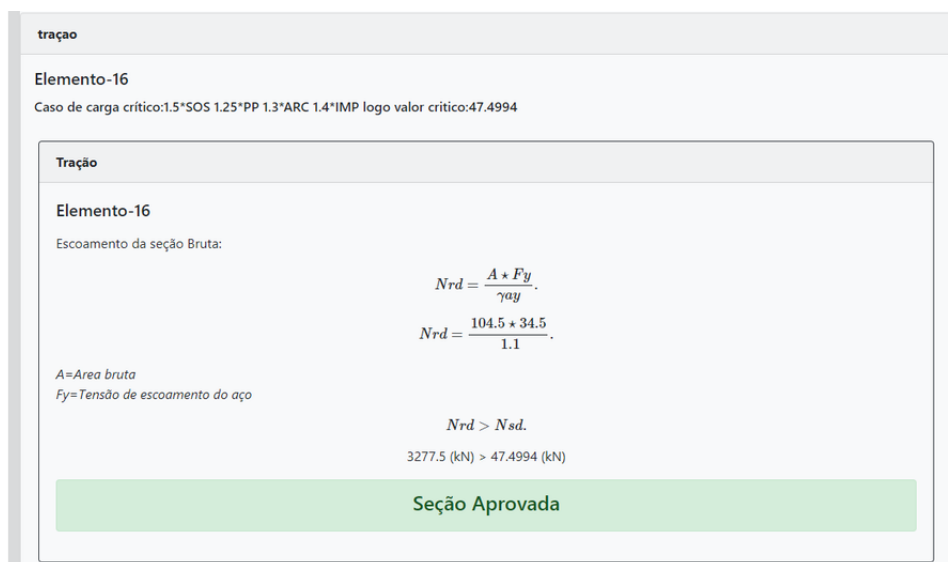


Figura 4.7: Trecho do memorial de cálculo do esforço resistente de tração para o elemento 16. Fonte: Autor (2021)

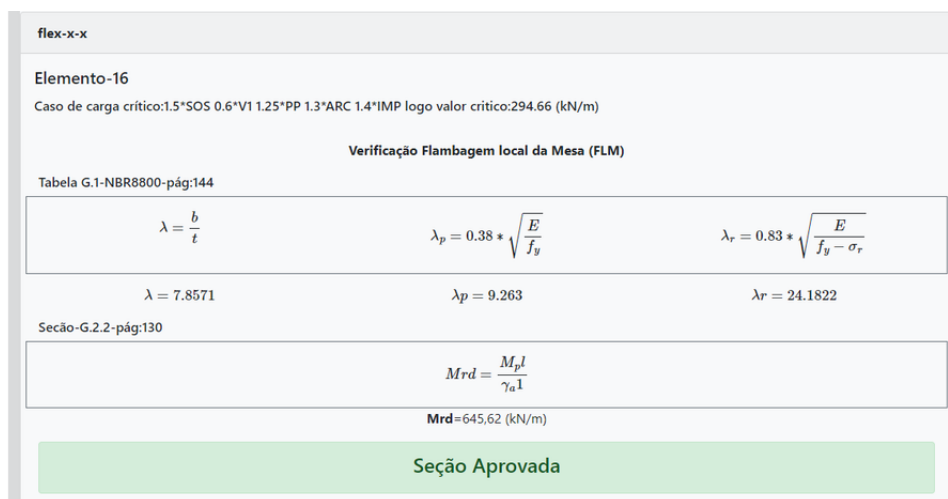


Figura 4.8: Trecho do memorial de cálculo do esforço resistente de flexão (FLM) para o elemento 16. Fonte: Autor (2021)

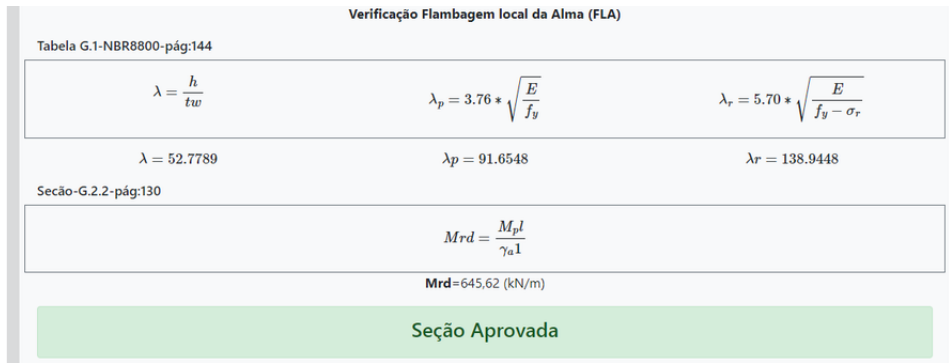


Figura 4.9: Trecho do memorial de cálculo do esforço resistente de flexão (FLA) para o elemento 16. Fonte: Autor (2021)

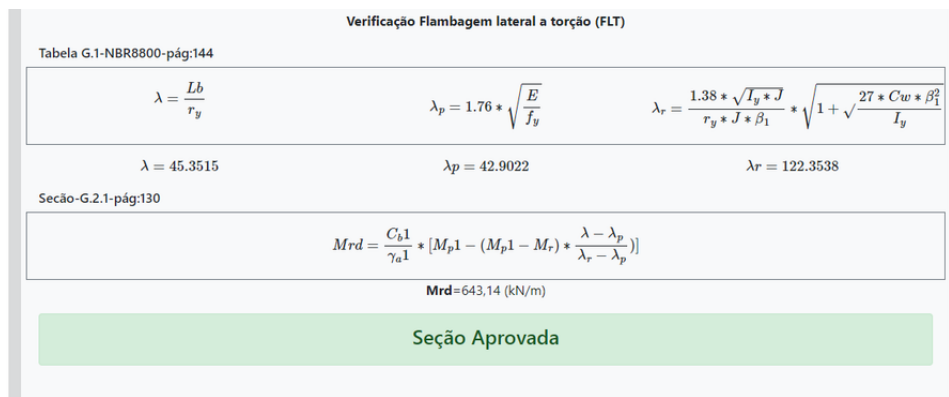


Figura 4.10: Trecho do memorial de cálculo do esforço resistente de flexão (FLT) para o elemento 16. Fonte: Autor (2021)

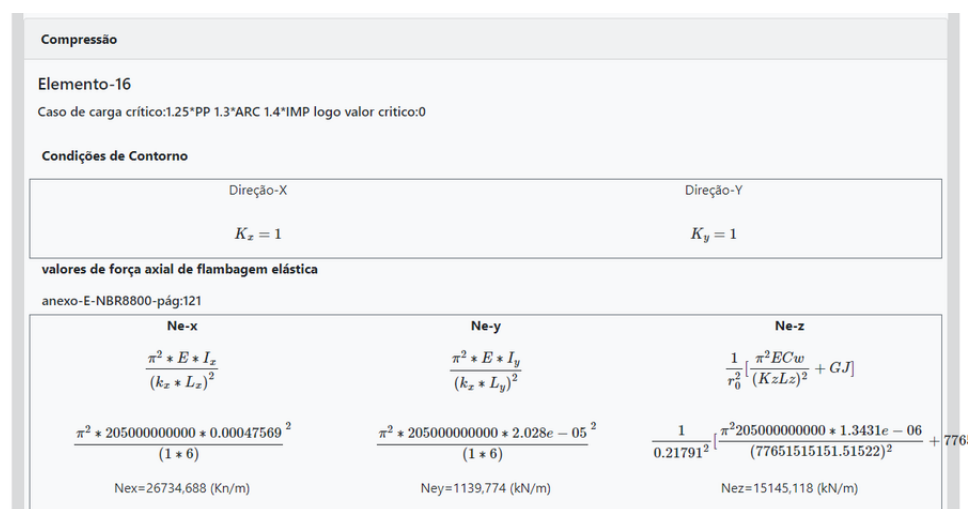


Figura 4.11: Trecho do memorial de cálculo do valor de flambagem elástica para o elemento 16. Fonte: Autor (2021)

Fator Qa
Seção-F3-NBR8800-pág:129

Esbeltez	Esbeltez Lim
$\frac{b_w}{t_w}$	$1.49 * \sqrt{\frac{E}{F_y}}$
$\frac{0.477}{0.0095}$	$1.49 * \sqrt{\frac{205000000000}{345000000}}$

Como:
50.2105 >= 36.3207

Logo:

$$B_{ef} = 1.92 * t_w * \sqrt{\frac{E}{F_y}} * [1 - \frac{C_a}{\frac{b}{t}} * \sqrt{\frac{E}{F_y}}]$$

$$B_{ef} = 1.92 * 0.0095 * \sqrt{\frac{205000000000}{345000000}} * [1 - \frac{0.38}{\frac{0.477}{0.0095}} * \sqrt{\frac{205000000000}{345000000}}]$$

$$B_{ef} = 0.37123$$

$$A_g = A_g - \sum((b_w - b_{ef}) * t_w)$$

$$A_{ef} = 0.0094452$$

$$Q_a = \frac{A_{ef}}{A_g}$$

$$Q_a = \frac{0.0094452}{0.01045}$$

$$Q_a = 0.90385$$

Fator Os

Figura 4.12: Trecho do memorial de cálculo do fator Qa para o elemento 16.
Fonte: Autor (2021)

PUC-Rio - Certificação Digital N° 1821325/CA

Fator Qs
Seção-F2-NBR8800-pág:126

Esbeltez	Esbeltez Lim
$\frac{b_f/2}{t_f}$	$0.56 * \sqrt{\frac{E}{F_y}}$
$\frac{0.209}{0.0133}$	

Como:
7.8571 <= 13.6507

Logo:
Qs = 1

Fator Q
Seção-F1-NBR8800-pág:126

Q Resultante

$$Q_a * Q_s$$

$$Q_s = 0.90385 * 1$$

Logo:
Qs = 0.90385

Figura 4.13: Trecho do memorial de cálculo do fator Qs para o elemento 16.
Fonte: Autor (2021)

Fator X
Seção-5.3.3.1-NBR8800-pág:44

Esbeltez Reduzida

$$\gamma_o = \sqrt{Q * Ag * \frac{f_y}{N_e}}$$

$$\gamma_o = \sqrt{0.90385 * 0.01045 * \frac{345000000}{1139774.8149}}$$

$\gamma_o = 1.6909$

Como:
 $\gamma_o > 1.5$

Logo:
 $\chi = \frac{0.877}{\gamma_o^2}$

$\chi = 0.30675$

Figura 4.14: Trecho do memorial de cálculo do fator χ para o elemento 16.
Fonte: Autor (2021)

Nrd
Seção-5.3.2-NBR8800-pág:44

Nrd resultante

$$Nrd = \frac{X * Q * Ag * f_y}{\gamma_a}$$

$$Nrd = \frac{0.30675 * 0.90385 * 0.01045 * 345000000}{1.1}$$

Nrd=908,711 (kN)

Seção Aprovada

Figura 4.15: Trecho do memorial de cálculo do valor resistente à compressão para o elemento 16. Fonte: Autor (2021)

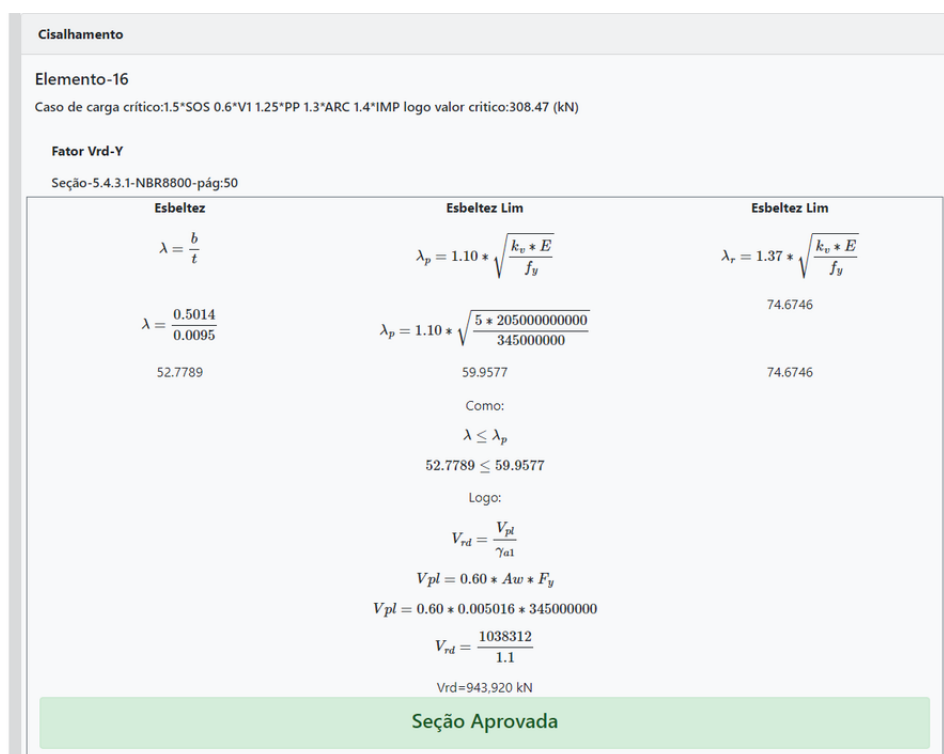


Figura 4.16: Trecho do memorial de cálculo do valor resistente à cisalhamento para o elemento 16. Fonte: Autor (2021)

4.4.2 GUI de taxas de utilização

Para a visualização da taxa de utilização dos perfis verificados, desenvolveu-se uma GUI em que o usuário poderá visualizar esses valores, ao longo do comprimento da viga para o esforço selecionado, e para o caso de carga desejado. A GUI conta com diversos menus, sendo um responsável por exibir a lista dos esforços disponíveis, denominado *Stress List*. Outro menu, no qual o usuário é capaz de selecionar qual o número do elemento que deseja utilizar, chamado de *Element Index*. Por fim, um menu, denominado *Category*, que exibe uma lista de estados limites verificados, visto que o esforço pode possuir mais de um, como é o caso da flexão, detentora de FLT, FLA E FLM. Ademais, a GUI também contém uma lista, denominada *LoadCombinationList*, que possui todas as combinações possíveis dos casos de carga, no qual o usuário consegue navegar.

Para demonstrar essa nova GUI com um caso conhecido, também será utilizado o elemento 16 do modelo utilizado para exemplificá-la, através das imagens 4.17, 4.18, 4.19.

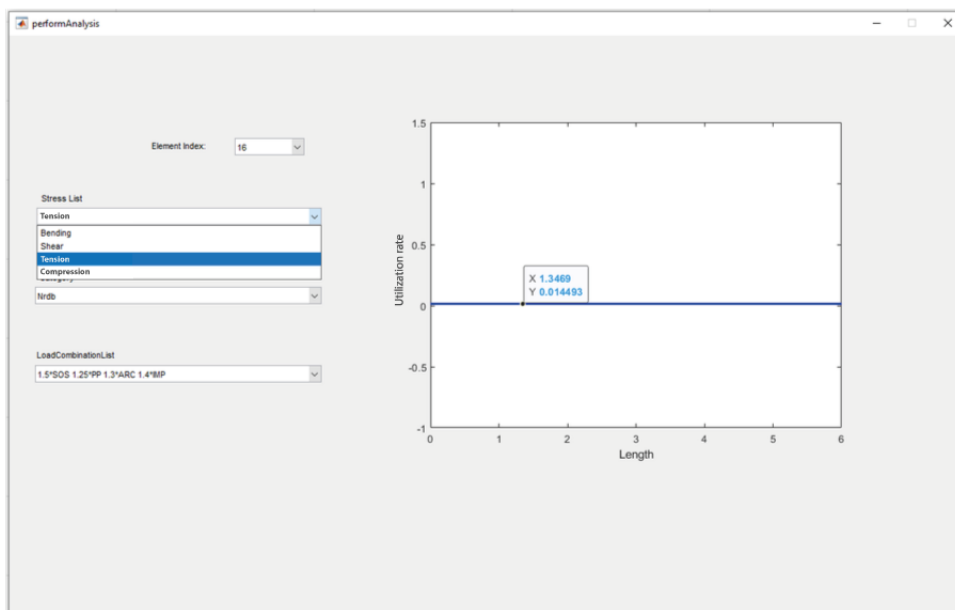


Figura 4.17: GUI da taxa de utilização do esforço de tração para o elemento 16. Fonte: Autor (2021)

PUC-Rio - Certificação Digital Nº 1821325/CA

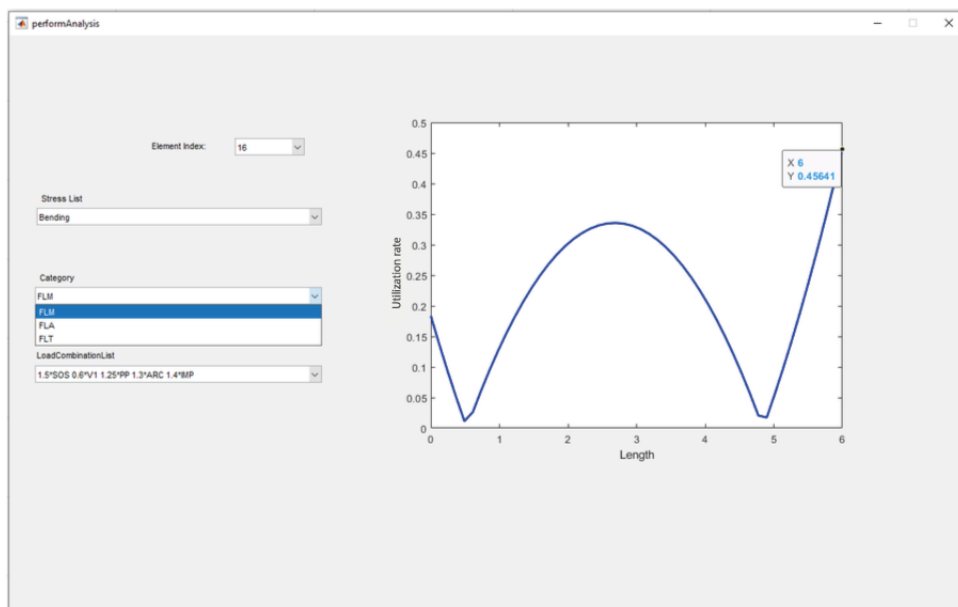


Figura 4.18: GUI da taxa de utilização do esforço de flexão para o estado limite FLM do elemento 16. Fonte: Autor (2021)

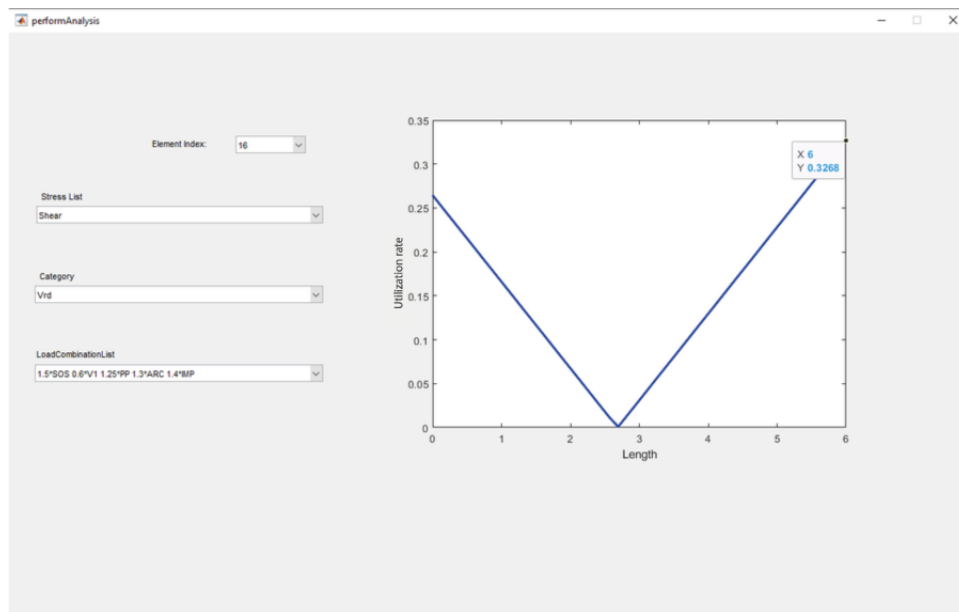


Figura 4.19: GUI da taxa de utilização do esforço de cisalhamento para o elemento 16. Fonte: Autor (2021)

5 Conclusão

Tendo em vista que os programas de dimensionamento estrutural são, em sua maioria, pagos, dificulta-se o contato de engenheiros em formação com essas ferramentas. Contudo, no cenário atual, em que a grande parte dos cálculos e análises são realizados por meios computacionais, o domínio desses *softwares* mostra-se essencial. Dessa forma, o presente trabalho propõe-se a elaborar uma alternativa gratuita e didática, visando mitigar essa falta de acesso.

Além disso, com o intuito de contribuir com o entendimento do funcionamento do programa em questão, assim como de seus cálculos realizados, o código do LESM é inteiramente aberto. Portanto, qualquer indivíduo, discente ou docente, está apto não apenas a verificação das metodologias de resolução dos dimensionamentos, mas também a contribuir com o *software*, modificando ou adicionando novas funcionalidades.

Com isso, para o desenvolvimento do presente trabalho, além de conhecimentos relacionados à engenharia civil, também utilizou-se de paradigmas computacionais fundamentados nos preceitos da orientação a objetos. Ademais, no decorrer da dissertação, comprovou-se que a utilização desse modelo facilitou a implementação da nova funcionalidade de dimensionamento. Devido ao seu alto grau de abstração e de seus conceitos bases de herança e polimorfismo, tornou-se simples a inserção de ramificações no LESM. Outrossim, a utilização de diagramas UML, em concordância com os princípios da orientação a objetos, permitiu esboçar todas as etapas do projeto, desde a elaboração conceitual até a geração da documentação final.

Baseado na esquematização do projeto deste trabalho, realizou-se toda a codificação, visando implementar a nova funcionalidade de verificação de perfis metálicos no LESM. Ao final, para comprovar seu funcionamento, elaborou-se um exemplo numérico, cujos resultados foram comparados com os obtidos por outro *software* de dimensionamento amplamente utilizado, o CypeCad.

Ao analisar os dados obtidos, percebe-se que evidenciam uma boa acurácia nas combinações de casos de carga e nos valores dos esforços solicitantes, resistentes dos perfis e nas taxas de utilização geradas por ambos os programas, visto que o maior erro relativo encontrado foi de 10,31%.

As discrepâncias detectadas, devem-se, provavelmente a algum desalinha-

mento entre a metodologia de cálculo dos esforços solicitantes dos *softwares* ou a possíveis erros em propriedades geométricas dos perfis utilizados. Assim, esses erros podem ocasionar pequenas divergências nos coeficientes utilizados no cálculo dos valores resistentes.

Outro ponto relevante, é a maneira na qual o usuário pode visualizar os dados gerados pelo LESM. A geração do memorial de cálculo, exibido passo a passo e contendo referências às páginas e seções das normas utilizadas, garante um melhor entendimento ao indivíduo utilizador do programa. Em relação ao gráfico das taxas de utilização, conclui-se que a sua exibição, ao longo das seções da barra sobre cada esforço, mostra-se extremamente didática, contribuindo para que o usuário seja capaz de identificar as seções críticas.

Dessa forma, a elaboração deste trabalho, auxilia na diminuição da barreira entre a comunidade acadêmica e os *softwares* de engenharia. Com isso, além de oferecer acesso a um programa de cálculo estrutural, também elucidam-se no memorial de cálculo todas as etapas para obtenção dos coeficientes e resultados, assim como a metodologia seguida, contribuindo para um entendimento integral dos cálculos de dimensionamento das estruturas. Por fim, disponibiliza-se abertamente o código do LESM, para que não só os usuários possam compreender ainda mais o funcionamento do programa, mas também sejam capazes de contribuir com o seu desenvolvimento.

5.1 Sugestões para trabalhos futuros

Com o intuito de aderir novas funcionalidades ao programa e complementar a implementação da verificação de perfis I, seguindo os preceitos da NBR8800:2008 realizada ao longo desta dissertação, propõem-se os seguintes temas para trabalhos futuros:

- 1) Inserção da verificação dos estados limites último e de serviço para perfis soldados, seguindo os preceitos da NBR8800:2008, em conjunto com o aumento da gama de perfis disponibilizados;
- 2) Implementação da funcionalidade de verificação de perfis metálicos também para pórticos 3D;
- 3) Implementação da análise de não linearidade geométrica para deslocamentos moderados;
- 4) Implementação de ligações semirrígidas com comportamento não linear;
- 5) Adaptação da análise de esforços internos para considerar a formação de rótulas plásticas.

Apêndice A

TABELA DE BITOLAS

Table with columns: BITOLA (mm x kg/m), Massa Linear (kg/m), d (mm), b (mm), t (mm), h (mm), d' (mm), Área (cm²), EIXO X-X (I, W, r, Z, I, W, r, Z), EIXO Y-Y (I, W, r, Z), ESBELTEZ (MESA, ALMA, C, u), and BITOLA (in x lb/ft).

* Bitolas produzidas sob encomenda.

Características Dimensionais e Propriedades Geométricas de acordo com as normas ABNT NBR 15980:2011 e ASTM A242/A4M.

As seções indicadas na tabela acima são produzidas em aço ASTM A572 Grau 50. Informações sobre produção em outros aços, consultar a equipe comercial de Gerdaul.



Figura 5.1: Tabela de Pefis Guerdau

Referências bibliográficas

- [1] KIMURA, A.. **Informática aplicada em estruturas de concreto armado.** 2018.
- [2] DE ANDRADE SARAIVA, A.; LIMA, D.. **Software livre aplicado academicamente e profissionalmente na Engenharia Mecânica.** Congresso Nacional Universidade ,EAD e Software Livre. Universidade Federal de Minas Gerais, Belo Horizonte., 2012.
- [3] MARTHA, L. F.. **ANÁLISE MATRICIAL DE ESTRUTURAS COM ORIENTAÇÃO A OBJETOS.** 11 2018.
- [4] GUEDES, G.. **UML 2: Uma Abordagem Prática.** 2018.
- [5] KAMIENSKI, C. A.. **Introdução ao paradigma de orientação a objetos.** Centro Federal de Educação Tecnológica da Paraíba, 1996.
- [6] DE MORAES PEREIRA, L. A.. **Análise e Modelagem de Sistemas com a UML.** 2011.
- [7] DE PÁDUA PAULA FILHO, W.. **Engenharia de Software: fundamentos, métodos e padrões.** março 2000.
- [8] ALMEIDA, T. G.. **Dimensionamento de Barras de Aço em MATLAB,** 2019.
- [9] LOPES, P. C.; RANGEL, R. L. ; MARTHA, L. F.. **Extensão de programa gráfico para análise de vibrações de modelos estruturais reticulados.** XL Ibero-Latin-American Congress on Computational Methods in Engineering, 2019.
- [10] ASSOCIACAO BRASILEIRA DE NORMAS TECNICAS. **NBR8800:Projeto de estruturas de aço e de estruturas mistas de aço e concreto de edifícios,** 2008.
- [11] COAD, P.; YOURDON, E.. **Object-Oriented Analysis.** Prentice-Hall, New Jersey, 2 edition, 1991.
- [12] BADALOTTI, G. M.. **Lógica e Organização de Computadores.** Uniasselvi, 2016.

- [13] BOOCH, G.; RUMBAUGH, J. ; JACOBSON, I.. **UML Gui do Usuário**. Elsevier Ltda, 2 edition, 2012.
- [14] DO COUTO PFEIL, W.; PFEIL, M. S.. **Estruturas de Aço - Dimensionamento Prático de Acordo com a NBR 8800:2008**. LTC - Livros Técnicos e Científicos Editora LTDA., Rio de Janeiro, 8ª edition, 2009.
- [15] MARTHA, L. F.. **Análise matricial de estruturas com orientação a objetos**. GEN LTC, 2018.
- [16] RANGEL, R. L.; MARTHA, L. F.. **LESM – An object-oriented MATLAB program for structural analysis of linear element models**. *Comput. Appl. Eng. Educ.*, 27(3):553 – 571, Janeiro 2019.
- [17] LOPES, P.; RANGEL, R. ; MARTHA, L.. **An interactive user interface for a structural analysis software using computer graphics techniques in MATLAB**. *Computer Applications in Engineering Education*, Março 2021.