

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
CURSO DE ENGENHARIA CIVIL**

João Antônio Brasil Severgnini

**METALPY: MÓDULO COMPUTACIONAL PARA
DETERMINAÇÃO DA CAPACIDADE
RESISTENTE DE PERFIS LAMINADOS E
SOLDADOS DE AÇO**

Porto Alegre
2022

JOÃO ANTÔNIO BRASIL SEVERGNINI

**METALPY: MÓDULO COMPUTACIONAL PARA
DETERMINAÇÃO DA CAPACIDADE RESISTENTE DE PERFIS
LAMINADOS E SOLDADOS DE AÇO**

Trabalho de Conclusão de Curso apresentado à Comissão de
Graduação do Curso de Engenharia Civil da Escola de Engenharia
da Universidade Federal do Rio Grande do Sul, como parte dos
requisitos para obtenção do título de Engenheiro Civil.

Orientador: Felipe Schaedler de Almeida

Porto Alegre
2022

JOÃO ANTÔNIO BRASIL SEVERGNINI

**METALPY: MÓDULO COMPUTACIONAL PARA
DETERMINAÇÃO DA CAPACIDADE RESISTENTE DE
PERFIS LAMINADOS E SOLDADOS DE AÇO**

Este Trabalho de Diplomação foi julgado adequado como pré-requisito para a obtenção do título de ENGENHEIRO CIVIL e aprovado em sua forma final pela Banca Examinadora, pelos Professores Orientadores e pela Comissão de Graduação do Curso de Engenharia Civil da Universidade Federal do Rio Grande do Sul.

Porto Alegre, setembro de 2022

BANCA EXAMINADORA

Prof. Enio Mesacasa Júnior (UFRGS)

Dr. pela Escola de Engenharia de São Carlos - Universidade de São Paulo

Prof. Felipe Schaedler de Almeida (UFRGS)

Dr. pela Universidade Federal do Rio Grande do Sul

Prof. Marcelo Maia Rocha (UFRGS)

Dr. pela Universitat Innsbruck, Áustria

AGRADECIMENTOS

Começo agradecendo a pessoa que fez todas as minhas conquistas se tornarem possíveis, que é meu exemplo de generosidade, respeito e resiliência, minha mãe Clélia. Obrigado por tudo que fez e faz por mim.

Agradeço minhas irmãs e meu pai por todo o apoio, companheirismo e vida compartilhada.

Agradeço a todos os professores que tive por me ajudarem a evoluir em cada uma das etapas da minha vida. Um agradecimento especial aos professores Felipe Schaedler de Almeida e João Ricardo Masuero por todo o conhecimento repassado e dedicação durante o período no qual fui aluno e orientando.

Agradeço a Vantec estruturas por me dar minha primeira oportunidade profissional e todo aprendizado que tive durante o tempo de estágio.

Agradeço ao grupo Simon engenharia e todos os seus colaboradores pelos conhecimentos passados durante o dia-a-dia, por contribuir com o meu crescimento profissional e pessoal, e pelo apoio e compreensão durante a elaboração deste trabalho.

Agradeço a todos os meus amigos que me apoiam e torcem por mim.

A menos que modifiquemos a nossa maneira de pensar,
não seremos capazes de resolver os problemas causados
pela forma como nos acostumamos a ver o mundo

Albert Einstein

RESUMO

Este trabalho apresenta a elaboração de um sistema computacional para a determinação da capacidade resistente de perfis estruturais laminados e soldados de aços. Este sistema foi concebido na forma de uma biblioteca, implementada na linguagem de programação Python utilizando o paradigma da programação orientada a objeto. A biblioteca é formada por dois componentes fundamentais, organizados em módulos separados. O primeiro módulo define classes destinadas à obtenção e representação das características básicas de cada tipo de perfil estrutural. O segundo módulo traz classes que abrigam métodos de cálculo da capacidade resistente para cada perfil segundo normas de projeto específicas. A implementação descrita nesse trabalho contempla perfis laminados tipo I, tubo retangular ou circular, além de perfis I e caixão soldados. Os métodos para a determinação de resistência à tração, compressão, flexão e corte destes perfis estão disponíveis segundo os critérios definidos nas normas ABNT/NBR8800:2008 e ANSI/AISC360-16. O código foi desenvolvido segundo uma arquitetura que visa favorecer a expansão da biblioteca, facilitando a inclusão de mais tipos de perfis ou de critérios de resistência definidos por outras normas. Também foram preocupações centrais no projeto da biblioteca a facilidade da reutilização e manutenção do código, bem como a sua integração com programas de terceiros. As primeiras aplicações apresentadas nesse trabalho visam ilustrar a utilização da biblioteca no fluxo usual de verificação de segurança de elementos estruturais e demonstrar a precisão dos resultados, através da comparação com exemplos disponíveis na bibliografia técnica. Em seguida, são apresentadas aplicações que demonstram a versatilidade da biblioteca em termos de integração com programas de terceiros.

Palavras-chave: estruturas de aço, Python, programação orientada a objetos.

ABSTRACT

This work shows the development of a computational system to determine the strength of rolled and built-up steel structural profiles. This system was designed as a library, implemented with Python programming language using object-oriented paradigm. This library contains two fundamentals components, organized in separated modules. The first module defines classes intended to obtain and represent basic characteristics of each type of structural steel profiles. The second module contains classes with methods to calculate strength according to specific design standards. The implementation described in this work cover profiles of the types rolled I shape , HSS, round HSS, built-up I shaped and box section. Functions for the determination of compression strenght, tension strenght, bending and shear strength are available according to the criteria giveb by ABNT/NBR8800:2008 and ANSI/AISC360-16. The code was developed according to an architecture that favors the expansion of the library, enabling the inclusion of more profiles or other regulation specification of project. Furthermore, ease code maintenance and integration with third-party software are main concerns in the library development. The first examples show the application of the library in the usual flow of safety check of structural elements and demonstrate its accuracy with respect to the results taken from technical literature. Additional examples demonstrate the versatility of the library in terms of integratios with other softwares.

Keywords: steel structures, python, object-oriented programming.

LISTA DE FIGURAS

Figura 2.1 – Perfis abordados	15
Figura 3.1 – Dimensões características de perfis I laminados	17
Figura 3.2 – Dimensões características de perfis I soldados	18
Figura 3.3 – Dimensões características de perfis tubo retangulares	18
Figura 3.4 – Dimensões características de perfis tubo circulares	19
Figura 3.5 – Dimensões características de seção caixão	19
Figura 3.6 – Elementos apoiados-livres	24
Figura 3.7 – Elementos apoiados-apoiados	24
Figura 3.8 – Área efetiva de cisalhamento	25
Figura 3.9 – Flambagem Lateral por torção	28
Figura 4.1 – Diagrama de classes do pacote Metalpy	35
Figura 4.2 – Módulo de perfis do pacote Metalpy	37
Figura 4.3 – Módulo de normas do pacote Metalpy	39
Figura 5.1 – Perfil CVS 450X116 submetido a flexo-compressão	41
Figura 5.2 – Repostas do exemplo 1	44
Figura 5.3 – Coluna de perfil W14X82	45
Figura 5.4 – Respostas do Exemplo 2	47
Figura 5.5 – Solicitações de cálculo do pilar	48
Figura 5.6 – Respostas do Exemplo 3	50
Figura 5.7 – Respostas do Exemplo 4	51
Figura 5.8 – Modelo estrutural da treliça analisada	52
Figura 5.9 – Cargas do modelo	53

Figura 5.10 – Resultados da verificação de segurança	56
Figura 5.11 – Aba Metalpy na interface do Revit	57
Figura 5.12 – Plugins adicionados ao Revit	58
Figura 5.13 – Utilização do plugin "Escolher Perfil"	61
Figura 5.14 – Apresentação dos resultados do plugin "Escolher Perfil"	61
Figura 5.15 – Utilização do plugin "Verificar Perfil"	64
Figura 5.16 – Apresentação dos resultados do plugin "Verificar Perfil"	65

LISTA DE TABELAS

Tabela 3.1 – Estados limites de perfis submetidos a flexão de acordo com a NBR8800:2008 e AISC360-16	27
--	----

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ANSI	American National Standards Institute
ELU	Estado limite último
AA	Elementos apoiados-apoiados
AL	Elementos apoiados-livres
FLT	Flambagem Lateral com torção
FLM	Flambagem local da mesa
FLA	Flambagem local da alma
FL	Flambagem local
POO	Programação orientada à objetos

SUMÁRIO

1 INTRODUÇÃO	13
2 DIRETRIZES	15
2.1 OBJETIVO	15
2.2 DELIMITAÇÃO	15
2.3 LIMITAÇÃO	16
3 CRITÉRIOS DE RESISTÊNCIA	17
3.1 PROPRIEDADES GEOMÉTRICAS DOS PERFIS DE AÇO	17
3.2 TRAÇÃO	22
3.3 COMPRESSÃO	22
3.4 ESFORÇO CORTANTE	25
3.5 MOMENTO FLETOR	27
3.5.1 Plastificação	27
3.5.2 Flambagem Lateral com torção	28
3.5.3 Flambagem local	29
3.6 ESFORÇOS COMBINADOS	30
4 ESTRUTURAÇÃO DO CÓDIGO	32
4.1 ORIENTAÇÃO A OBJETO	32
4.1.1 Classes	33
4.1.2 Herança	33
4.1.3 Polimorfismo	34
4.2 METALPY	34
4.2.1 Módulo perfis	36
4.2.2 Módulo normas	39
4.2.3 Relação entre o módulo perfis e módulo normas	40
5 EXEMPLO	41
5.1 EXEMPLOS DE VERIFICAÇÃO	41
5.1.1 Perfil I soldado submetido a flexo-compressão	41

5.1.2	Perfil I laminado submetido a flexo-tração	44
5.1.3	Perfil tubo retangular submetido a flexo-compressão	47
5.1.4	Força resistente ao corte de perfil tubo circular	50
5.2	APLICAÇÕES	52
5.2.1	Rotina de verificação integrada ao SAP2000	52
5.2.2	Plugins integrados ao Revit	56
5.2.2.1	Plugin de escolha de perfis	58
5.2.2.2	Plugin de verificação de perfis	61
6	CONSIDERAÇÕES FINAIS	66
	REFERÊNCIAS	67
	APÊNDICES	69

1 INTRODUÇÃO

Os computadores são empregados no auxílio a projetos estruturais desde o início de sua disseminação. Nas primeiras épocas, a automatização das rotinas de cálculo se dava por programas desenvolvidos pelos próprios projetistas. Com o avanço da capacidade de processamento e armazenamento dos computadores, muitos softwares comerciais de engenharias foram desenvolvidos. Hoje temos softwares extremamente poderosos para nos auxiliar nos cálculos e na geração de desenhos e modelos mais realistas da estrutura, integrando todo o processo de projeto.

Com a disponibilidade dos softwares comerciais, os projetistas passaram de autores a utilizadores de programas. Como resultado, durante um bom tempo a programação foi deixada de lado nos escritórios de projeto. Os softwares comerciais trouxeram inúmeros benefícios para os projetistas, poupando tempo, facilitando a geração de revisões e permitindo uma melhor integração entre as diversas disciplinas. Porém, a inacessibilidade ao código fonte dos programas comerciais fez com que os projetista tivessem que se adaptar ao software, e muitas vezes se limitando aos recursos que ele fornece.

Recentemente, os fornecedores de softwares vêm adotando medidas para aliviar estas limitações através das Interfaces de Programação de Aplicação (APIs, do inglês Application Programming Interface), que são mecanismos que permitem que dois componentes de softwares se comuniquem usando um conjunto de definições e protocolos (AWS, 2022). As APIs não só trouxeram a possibilidade da integração entre os diferentes softwares existentes no mercado, como também a possibilidade de os usuários criarem suas próprias funcionalidades utilizando as ferramentas fornecidas pelo programa. Desta forma, o projetista deixa de se adequar ao software e passa adequar o software as suas necessidades. Neste cenário a programação tem voltado a integrar a rotina dos escritórios de projeto.

Este trabalho propõe, implementa e avalia uma arquitetura para um sistema computacional destinado à determinação da capacidade resistente de perfis de aço, bem como para determinação de propriedades geométricas dos perfis e parâmetros de estabilidade de barras formadas por um determinado perfil. O desenvolvimento deste trabalho resultou na implementação da biblioteca Metalpy (SEVERGNINI, 2022), que tem por finalidade fornecer funções para que o projetista de estruturas de aço possa criar suas próprias rotinas e programas de acordo com suas necessidades. Outra finalidade da biblioteca é contribuir para a disseminação do uso de programação entre os engenheiros e principalmente entre os estudantes de engenharia. Nesse contexto, espera-se que a biblioteca implementada possa auxiliar tanto no ensino de estruturas de aço quanto no ensino de

programação para engenheiros civis.

Metalpy é um pacote desenvolvido na linguagem de programação Python, e foi implementada utilizando o paradigma de programação orientada a objetos. Este paradigma se baseia na construção de classes que modelam objetos, onde objetos são entidades que armazenam dados e apresentam funções diretamente relacionadas a esses dados. A biblioteca está estruturada em dois módulos bem definidos. O módulo perfis que apresenta classes que representam os perfis, armazenando todos os dados relacionados a determinado perfil e apresentando funções para determinação de parâmetros de estabilidade e suas propriedades geométricas. O módulo normas contém as classes das normas NBR8800 e AISC360, que apresentam funções para a determinação da capacidade resistente dos perfis de acordo com as normas NBR8800:2008 (ABNT, 2008) e AISC360-16 (ANSI, 2016), que serão referidas ao longo do texto como NBR8800 e AISC360, respectivamente. A biblioteca foi desenvolvida de forma que facilmente pode ser ampliada, possibilitando o acréscimo de novos tipos de perfis e outras normas que regulamentam os projetos de estruturas de aço. Metalpy foi desenvolvido em código aberto, permitindo livre acesso ao código fonte pelos usuários. Todos os arquivos de implementação do pacote, assim como, exemplos de utilização estão disponíveis no repositório Metalpy no GitHub.

Outros trabalhos abordam o desenvolvimento de ferramentas computacionais capazes de auxiliar no dimensionamento e verificações de perfis de aço. Teixeira (2015) e Simsen (2014) desenvolveram programas computacionais para verificação de perfis do tipo I laminados e soldados de acordo com os critérios da NBR8800. Guanabara (2010) implementou um programa na linguagem visual basic, que além de verificar a segurança de acordo com a NBR8800, apresenta uma ferramenta capaz de encontrar o melhor perfil a ser utilizado, a partir dos dados de solicitações e comprimento do perfil fornecidos pelo usuário. Neste trabalho Guanabara (2010) abordou perfis do tipo I, C, L, T, tubos circulares e tubos retangulares. Em todos esses trabalhos, os programas implementados apresentavam uma interface gráfica e seu uso era restrito as funcionalidades implementadas pelos criadores. Diferente dos trabalhos apresentados, Metalpy tem a proposta de fornecer ferramentas para para que o usuário crie seus próprios programas de acordo com as suas necessidades. Isso dá ao usuário liberdade e flexibilidade quanto a possibilidade de aplicações, mas que requer dele um conhecimento básico de programação.

O segundo capítulo deste trabalho apresenta os objetivos, as delimitações e as limitações do trabalho. O terceiro capítulo aborda os métodos de determinação da capacidade resistente de perfis de aço de acordo com as normas NBR8800 e AISC360. O quarto capítulo apresenta uma revisão sobre os principais conceitos da programação orientada a objetos, a estruturação da biblioteca e explicação dos componentes da biblioteca Metalpy. No capítulo 5 são apresentados exemplos de utilização e aplicações do código. E no último capítulo são abordadas algumas considerações finais a respeito do desenvolvimento do trabalho.

2 DIRETRIZES

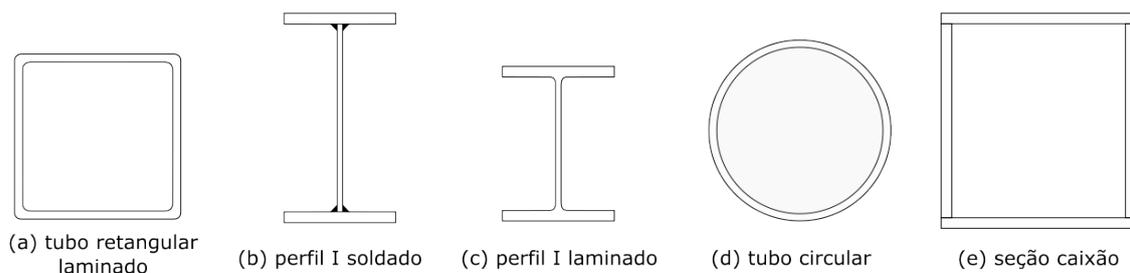
2.1 OBJETIVO

O objetivo principal deste trabalho é propor, implementar e avaliar uma arquitetura para um sistema computacional destinado à determinação da capacidade resistente de perfis de aço (laminados e soldados) de acordo com as normas NBR8800 e AISC360. A biblioteca também se propõem a servir como uma plataforma para o desenvolvimento de novos programas e rotinas criadas por outros usuários. O objetivo secundário deste trabalho é apresentar exemplos de aplicações de uso da biblioteca Metalpy, visando apresentar suas potencialidades e demonstrar formas de integração com outros softwares de engenharia.

2.2 DELIMITAÇÃO

Este trabalho aborda a determinação da capacidade resistente à tração, compressão, esforço cortante, momento fletor e a verificação de esforços combinados de perfis de aço, de acordo com o método dos estados limites últimos (ELU) das normas NBR8800:2008 (ABNT, 2008) e AISC360-16 (ANSI, 2016). São contemplados os perfis: perfil tubo retangular, perfis I, soldado e laminado (W, H e HP), perfil tubular circular e seção caixão retangular, apresentados na figura 2.1. Para perfis I soldados monossimétricos, as avaliações relativas ao momento fletor estão limitadas ao caso de flexão no plano da alma produzindo compressão na mesa superior. Também está contemplado o caso de perfis I com alma esbelta fletido no plano da alma.

Figura 2.1 – Perfis abordados



Fonte: Elaborado pelo autor

2.3 LIMITAÇÃO

Este trabalho não trata de nenhum aspecto referente às ligações. Portanto, a determinação da capacidade resistente à tração contempla apenas o estado limite de escoamento da seção bruta. Também não são abordados os efeitos de cargas localizadas nos perfis ou os estados limites de serviço. A biblioteca não apresenta uma interface gráfica.

3 CRITÉRIOS DE RESISTÊNCIA

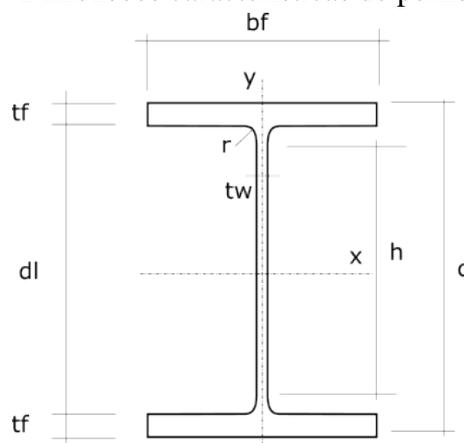
Este capítulo tem por objetivo apresentar uma breve revisão sobre os critérios de determinação da capacidade resistente para os tipos de perfis incluídos no escopo deste trabalho. São apresentados os critérios de segurança para cada ELU de acordo com ABNT (2008), destacando as diferenças entre as recomendações da NBR8800 e AISC360. As equações que definem a resistência dos perfis estão apresentadas em termos de valores característicos. Para determinação das resistências de cálculo dos perfis, a NBR8800 divide o valor de resistência característica pelo fator de ponderação γ_{a1} e a AISC360 multiplica pelo fator de ponderação Φ .

3.1 PROPRIEDADES GEOMÉTRICAS DOS PERFIS DE AÇO

Cada um dos perfis abordados neste trabalho são definidos por dimensões características, que influenciam diretamente no seu comportamento estrutural. Esta seção apresenta tais características usando a nomenclatura empregada na biblioteca Metalpy. Também são abordadas algumas propriedades geométricas que são essenciais ao dimensionamento estrutural e podem ser calculadas a partir das dimensões básicas do perfil tendo como referência o sistema de eixos apresentados nas figuras.

A figura 3.1 apresenta a seção de um perfil I laminado, onde estão indicadas as dimensões que o caracterizam. Estes perfis são caracterizados pela altura total (d), largura da mesa (bf), distância entre as faces internas das mesas (dl), altura da parte plana da alma (h), espessura da mesa (tf), espessura da alma (tw) e raio de concordância (r).

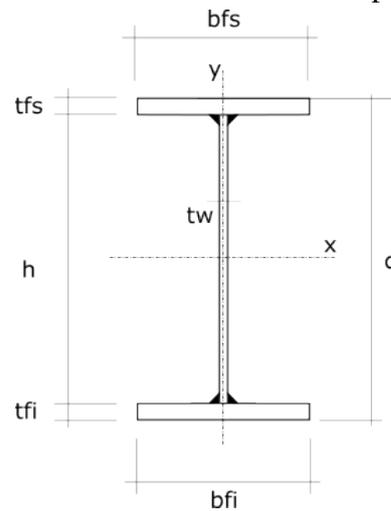
Figura 3.1 – Dimensões características de perfis I laminados



Fonte: Elaborado pelo autor

Na figura 3.2 está apresentado uma seção típica de um perfil I soldado e as dimensões que o caracterizam. Este perfis são caracterizados pela altura total (d), largura da mesa superior (b_{fs}), largura da mesa inferior (b_{fi}), altura da alma (h), espessura da mesa superior (t_{fs}) e espessura da mesa inferior (t_{fi}). Como estes perfis podem ser monossimétricos, as dimensões das mesas superior e inferior podem diferir.

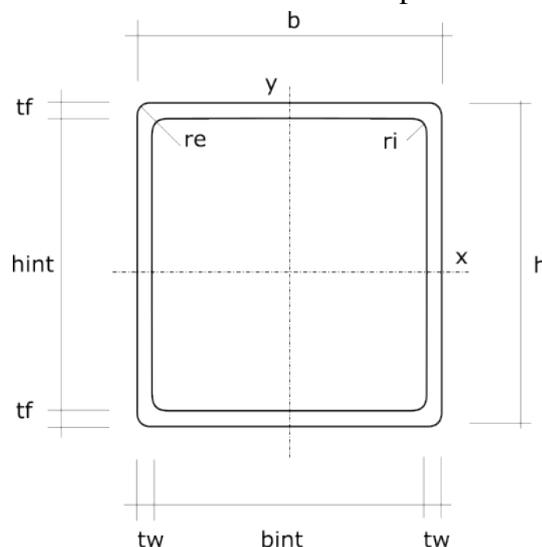
Figura 3.2 – Dimensões características de perfis I soldados



Fonte: Elaborado pelo autor

A figura 3.3 apresenta a seção de um perfil tubular retangular laminado e suas dimensões características. Estes perfis são caracterizados pela sua altura total (h), distância entre as faces internas das mesas (h_{int}), largura total (b), distância entre as faces internas das almas (b_{int}), espessura das mesas e das almas (t), raio de concordância externo (r_e) e raio de concordância interno (r_i).

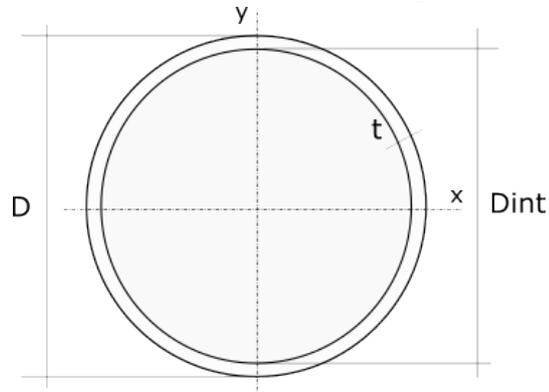
Figura 3.3 – Dimensões características de perfis tubo retangulares



Fonte: Elaborado pelo autor

A figura 3.4 apresenta a seção de um perfil tubular circular e as dimensões que o caracterizam. Estes perfis são caracterizados pelo diâmetro externo (D), diâmetro interno (D_{int}) e espessura da parede do tubo (t).

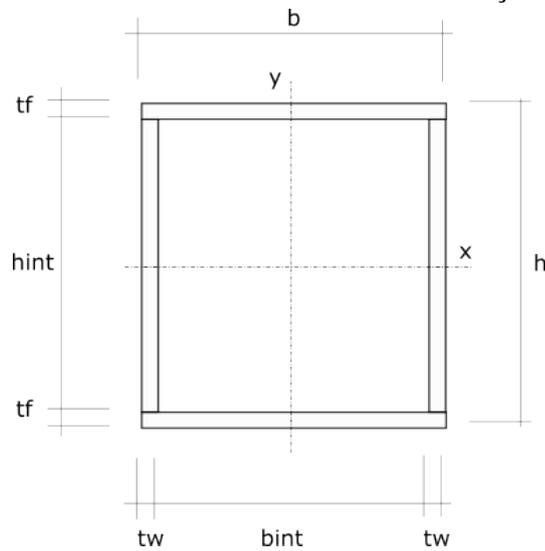
Figura 3.4 – Dimensões características de perfis tubo circulares



Fonte: Elaborado pelo autor

A figura 3.5 apresenta um perfil de seção caixão, indicando as dimensões que o caracterizam. Estes perfis são caracterizados pela altura total (h), distância entre as faces internas das mesas (h_{int}), largura total (b), distância entre as faces internas das almas (b_{int}), espessuras das mesas (t_f) e espessura das almas (t_w).

Figura 3.5 – Dimensões características de seção caixão



Fonte: Elaborado pelo autor

As dimensões apresentadas acima para cada tipo de perfil são utilizadas na determinação das seguintes propriedades geométricas:

- A_g - Área total;
- A_{wx} - Área efetiva de cisalhamento para cargas aplicadas paralelas ao eixo x;
- A_{wy} - Área efetiva de cisalhamento para cargas aplicadas paralelas ao eixo y;
- I_x - Momento de inércia em relação ao eixo x;
- I_y - Momento de inércia em relação ao eixo y;
- Z_x - Módulo de resistência plástico em relação ao eixo x;
- Z_y - Módulo de resistência plástico em relação ao eixo y;
- J - Constante de torção pura;
- C_w - Constante de empenamento.

Estas propriedades, juntamente com as propriedades do aço, definem grande parte dos comportamento estrutural dos perfis, influenciando diretamente na capacidade de resistir aos esforços. A partir destas, pode-se obter outras propriedades, como:

- $W_{xs} = I_x/y_{max}$ - Módulo de resistência elástico superior em relação ao eixo x;
- $W_{xi} = I_x/|y_{min}|$ - Módulo de resistência elástico inferior em relação ao eixo x;
- $W_{ys} = I_y/x_{max}$ - Módulo de resistência elástico superior em relação ao eixo y;
- $W_{yi} = I_y/|x_{min}|$ - Módulo de resistência elástico inferior em relação ao eixo y;
- $r_x = \sqrt{I_x/A}$ - Raio de giração em relação ao eixo x;
- $r_y = \sqrt{I_y/A}$ - Raio de giração em relação ao eixo y.

Seções simétricas em relação a algum eixo apresentam o mesmo valor para o módulo de resistência superior e inferior em relação ao eixo de simetria. Para perfis duplamente simétricos tem se $W_x = W_{xs} = W_{xi}$ e $W_y = W_{ys} = W_{yi}$.

Perfis laminados têm suas dimensões limitadas pelo seu processo de fabricação e apresentam suas dimensões e propriedades geométricas catalogadas pelos fabricantes. O módulo computacional

desenvolvido neste trabalho faz uso de um banco de dados da AISC (ANSI, 2017) para obtenção das propriedades geométricas de perfis I laminados. Para perfis tubulares, além da tabela da AISC, é utilizado o catalogo fornecido por Vallourec (s.d).

Os perfis soldados apresentados são formados pela união de chapas de aço ligadas por filetes de solda. Sendo assim, há grande flexibilidade em relação às dimensões da seção transversal no dimensionamento desses perfis. A principal limitação está na espessura padronizada das chapas usadas na sua fabricação. No módulo computacional desenvolvido, os perfis soldados têm as propriedades geométricas calculadas em função das suas dimensões características.

Como exemplo deste processo, as equações 3.1 a 3.4 determinam, respectivamente: a área total, os momentos de inércia em relação ao eixo X e Y e a constante de torção pura para perfis de seção caixão em função das suas dimensões. A constante de empenamento C_w para perfis de seção caixão é igual a zero.

$$A_g = 2b.t_f + 2h_{int}.t_w \quad (3.1)$$

$$I_x = \frac{b.t_f^3 + h_{int}^3.t_w}{6} + b.t_f.(h - t_f)^2 \quad (3.2)$$

$$I_y = \frac{b^3.t_f + h_{int}.t_w^3}{6} + h_{int}.t_w.(b - t_w)^2 \quad (3.3)$$

$$J = \frac{(t_w^3 + t_f^3)(b + h - t_w - t_f)}{3} + \frac{(b - t_w)^2(h - t_f)^2(t_f + t_w)}{b + h + t_w - t_f} \quad (3.4)$$

As equações 3.5 a 3.9 determinam, respectivamente: a área total, os momentos de inércia em relação ao X e Y, a constante de torção pura e a constante de empenamento para perfis I soldados com dupla simetria em função das duas dimensões.

$$A_g = 2b_f.t_f + h.t_w \quad (3.5)$$

$$I_x = \frac{h^3.t_w}{12} + \frac{b_f.t_f^3}{6} + b_f.t_f.(h - t_f)^2 \quad (3.6)$$

$$I_y = \frac{h.t_w^3}{12} + \frac{b_f^3.t_f}{6} \quad (3.7)$$

$$J = \frac{2b_f^3.t_f^3 + ht_w^3}{3} \quad (3.8)$$

$$C_w = \frac{t_f b_f^3 h^2}{24} \quad (3.9)$$

3.2 TRAÇÃO

Para determinar a força axial de tração resistente de um elemento de aço são avaliados para os ELUs de escoamento da seção bruta e ruptura da seção líquida, além do colapso por rasgamento, dependendo da forma de ligação. A força axial resistente é definida como o menor entre os valores obtidos em cada um desses estados limites (ABNT, 2008). Neste trabalho, não são consideradas os ELUs referentes as regiões de ligação das barras, sendo assim, a resistência a tração é definida pelo ELU de escoamento da seção bruta.

O ELU de escoamento da seção bruta ocorre quando a tensão normal na seção bruta da barra atinge a resistência ao escoamento do aço (f_y), fazendo com que o elemento sofra um alongamento excessivo e podendo levar a ruína da estrutura (FAKURY, 2016). A força axial de tração resistente característica para este estado limite é calculada pela equação 3.10:

$$N_{t,Rk} = A_g f_y \quad (3.10)$$

Para este estado limite os fatores de ponderação γ_{a1} e Φ_t são, respectivamente, equivalente à 1,1 e 0,90. As duas normas abordadas recomendam que o índice de esbeltez de barras tracionadas não supere o valor de 300.

3.3 COMPRESSÃO

Barras de aços comprimidas estão sujeitas a fenômenos de instabilidade. Estes fenômenos podem ser separados em naturezas distintas, fenômeno de instabilidade global e fenômeno de instabilidade local. Estes fenômenos de instabilidade ocorrem de forma simultânea, reduzindo a

força axial de compressão resistente da barra (PFEIL, 2022). A resistência à compressão é dada por:

$$N_{c,Rk} = \chi f_y A_e \quad (3.11)$$

Onde χf_y é a tensão média na falha do perfil por compressão e A_e é uma área efetiva da seção transversal que incorpora a redução de resistência em decorrência da flambagem local do perfil.

O coeficiente de redução χ , associado à flambagem global, é calculado por:

- para $\lambda_0 \leq 1,5$:

$$\chi = 0,658^{\lambda_0^2} \quad (3.12)$$

- para $\lambda_0 > 1,5$:

$$\chi = \frac{0,877}{\lambda_0^2} \quad (3.13)$$

Onde o índice de esbeltez reduzido da barra λ_0 é definido como:

$$\lambda_0 = \sqrt{\frac{A_e f_y}{N_e}} \quad (3.14)$$

Nesta expressão, N_e é a carga crítica de flambagem elástica da barra, tomada como a menor entre as forças de flambagem por flexão, torção ou flexotorção, sendo a última importante para perfis monossimétricos (Ver Anexo E da NBR8800:2008).

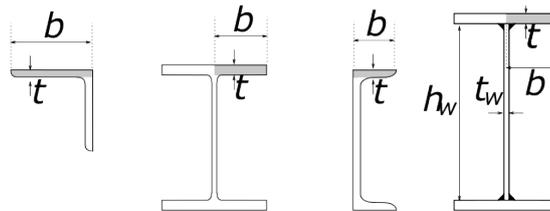
Os perfis tubulares têm o fator χ determinado conforme a NBR16239 (ABNT, 2013a), calculado por:

$$\chi = \frac{1}{(1 + \lambda_0^{4,48})^{1/2,24}} \quad (3.15)$$

Para AISC360-16 os efeitos de instabilidade global são considerados da mesma forma. A determinação da força axial resistente a compressão nominal é dada pela expressão $F_{cr} A_e$, onde F_{cr} é equivalente à $\chi \cdot f_y$. da equação 3.11 definida pela NBR8800:2008.

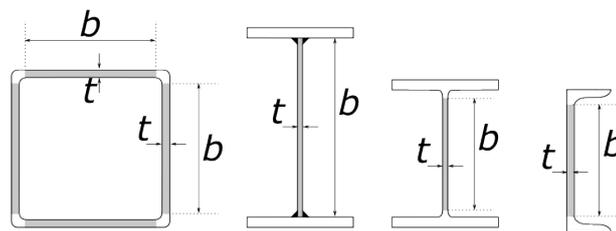
A área efetiva (A_e) é determinada em função dos fenômenos de flambagem local, que dependem das proporções e condições de vinculação dos elementos que formam a seção de paredes finas dos perfis laminados. Os elementos podem ser apoiados em apenas uma das bordas, denominado apoiado livre (AL), apresentados na figura 3.6, ou pode ser apoiado nas duas bordas, denominado apoiado-apoiado (AA), representado na figura 3.7.

Figura 3.6 – Elementos apoiados-livres



Fonte: Adaptado de Almeida (2020)

Figura 3.7 – Elementos apoiados-apoiados



Fonte: Adaptado de Almeida (2020)

Na ABNT (2008) o termo A_e da equação 3.11 é substituído por $Q.A_g$, onde o coeficiente associado à flambagem local $Q = Q_a.Q_s$ é determinado pelas parcelas relativas aos elementos AA (Q_a) e aos elementos AL (Q_s). O fator Q_a considera a resistência pós-flambagens dos elementos AA do perfil e é dado pela relação entre a área efetiva do perfil (determinada através do método da largura efetiva) e a área bruta do perfil, conforme é apresentado na seção F.3 do Anexo F da NBR8800:2008. O fator Q_s está associado à tensão de flambagem nos elementos AL (PFEIL, 2022). A determinação deste fator é feita de acordo com o tipo de perfil e seu cálculo está apresentado no item F.2 do Anexo F da NBR8800:2008.

Na AISC360 (ANSI, 2016), o processo para o cálculo da área efetiva do perfil é realizado através da determinação das larguras efetivas de cada elemento do perfil. Este procedimento é similar ao que é realizado para obtenção do fator Q_a , porém, este se aplica tanto para elementos AA quanto

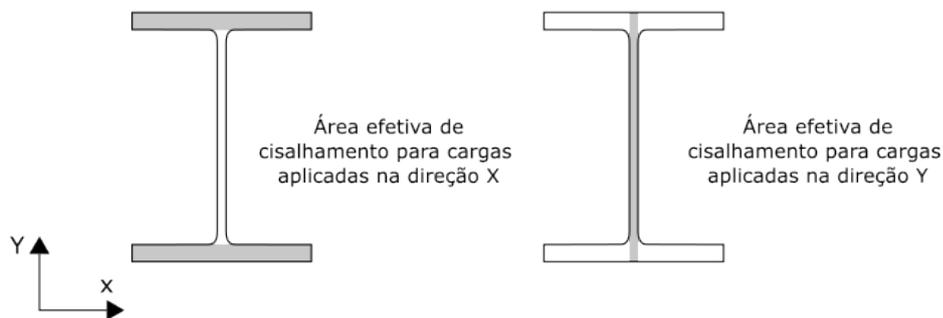
para elementos AL. O cálculo da área efetiva é apresentado na seção E.7 da AISC360 (ANSI, 2016).

Para determinação da força resistente de cálculo, γ_{a1} e Φ_t são, respectivamente, equivalente à 1,1 e 0,90. As duas normas abordadas limitam o índice de esbeltez de barras comprimidas em 200.

3.4 ESFORÇO CORTANTE

No geral, os elementos dos perfis capazes de resistir ao esforço cortante são os perpendiculares ao eixo de flexão (paralelo ao sentido de aplicação do carregamento). A figura 3.8 toma como exemplo a área efetiva de um perfil I, que, quando submetido a flexão em relação ao eixo de maior inércia (carregamento aplicados no sentido do eixo Y), tem o esforço cortante sendo resistido pela alma do perfil. Na situação alternativa, submetido a flexão no eixo de menor inércia (carregamento aplicado no sentido do eixo X), tem o esforço cortante sendo resistido pelas duas mesas. A mesma lógica se mantém para os demais perfis, exceto para perfis tubo circulares.

Figura 3.8 – Área efetiva de cisalhamento



Fonte: Elaborado pelo autor

Elementos estruturais de aço submetidos a esforço cortante também podem sofrer com os efeitos de instabilidade local. Os estados-limites últimos são dados pelo escoamento e por flambagem por cisalhamento (ABNT, 2008). A força cortante resistente característica do perfil é determinada pela NBR8800:2008 pelas equações 3.16, 3.17 e 3.18, de acordo com os parâmetros de esbeltez dos elementos que resistem ao esforço cortante.

- para $\lambda \leq \lambda_p$:

$$V_{Rk} = V_{pl} \quad (3.16)$$

- para $\lambda_p < \lambda \leq \lambda_r$:

$$V_{Rk} = \frac{\lambda_p}{\lambda} V_{pl} \quad (3.17)$$

- para $\lambda > \lambda_r$:

$$V_{Rk} = 1,24 \left(\frac{\lambda_p}{\lambda} \right)^2 V_{pl} \quad (3.18)$$

Os parâmetros de esbeltez limite λ_p e λ_r são dados pelas equações 3.19 e 3.20:

$$\lambda_p = 1,1 \sqrt{\frac{k_v E}{f_y}} \quad (3.19)$$

$$\lambda_r = 1,37 \sqrt{\frac{k_v E}{f_y}} \quad (3.20)$$

Nestas equações k_v é o parâmetro que leva em consideração a existência de enrijecedores transversas e a distância entre eles. Este parâmetro é definido na seção 5.4.3.1 da NBR8800.

A força cortante correspondente a plastificação (V_{pl}) é dada pela equação 3.21:

$$V_{pl} = 0,6 f_y A_w \quad (3.21)$$

Onde, A_w é área efetiva de cisalhamento e é definida na seção 5.4.3.1 para cada tipo de perfil transversal.

A AISC360-16 apresenta o mesmo procedimento para determinação da força cortante resistente que a NBR8800:2008, com algumas pequenas diferenças em relação a determinação da resistência para os perfis I. A primeira delas é o valor atribuído ao coeficiente k_v para perfis sem enrijecedores. A NBR8800:2008 apresenta um valor de k_v igual a 5, enquanto que a AISC360-16 atribui um valor de 5,34 para o mesmo coeficiente. Outra diferença é a apresentação de um roteiro de cálculo alternativo para perfis I com a relação entre a distância dos enrijecedores transversais e a altura da alma menor ou igual a 3. O roteiro de cálculo é apresentado na seção G.2.2 da AISC360-16, e ele propõe um acréscimo de resistência para os elementos que respeitam a relação entre distância dos enrijecedores e altura da alma comentada acima.

Para a obtenção da força cortante resistente de cálculo, os fatores de ponderação γ_{a1} e Φ_v são, respectivamente, iguais à 1,1 e 0,9.

3.5 MOMENTO FLETOR

O momento fletor resistente é tomado como sendo o menor valor de momento resistente calculado para cada estado limite. Na flexão o colapso do elemento estrutural pode se dar por meio da plastificação do perfil, flambagem lateral com torção (FLT) e flambagem local de elementos comprimidos, este podendo ser pela flambagem local da mesa (FLM) ou flambagem local da alma (FLA) (FAKURY, 2016). Os estados limites últimos na flexão não são os mesmos para todos os tipos de perfis. Alguns estados limites não são aplicáveis a determinados perfis. A tabela 3.1 apresenta os tipos de perfis e os estados limites empregados na determinação do momento resistente de cada um deles de acordo com a NBR8800 e AISC360.

Perfis	Estados Limites considerados	
	NBR8800	AISC360
Perfis I, H duplamente simétricos e U fletidos em relação ao eixo de maior inércia (eixo X)	FLM, FLT, FLA e Plastificação	FLM, FLT e Plastificação
Perfis I e H monossimétricos fletidos em relação ao eixo de maior inércia (eixo X)	FLT, FLM, FLA e Plastificação	FLT, FLM, escoamento da mesa comprimida e escoamento da mesa tracionada
Perfis I, H duplamente simétricos e U fletidos em relação ao eixo de menor inércia (eixo Y)	FLM, FLA e Plastificação	FLM e Plastificação
Tubos retangulares	FLT, FLM, FLA e Plastificação	FLT, FLM, FLA e plastificação
Tubos circulares	FL e Plastificação	FL e Plastificação

Tabela 3.1 – Estados limites de perfis submetidos a flexão de acordo com a NBR8800:2008 e AISC360-16

Da tabela acima, pode se observar que uma das diferenças entre as duas normas apresentadas é o fato da norma americana não apresentar a flambagem local da alma como um estado limite último para flexão de perfis I, H e U fletidos em relação ao eixo de maior inércia (eixo X). Apesar da AISC360-16 não considerar diretamente este ELU, perfis que apresentam almas não compactas e esbeltas são tratados com critérios diferentes de perfis que apresentam almas compactas.

3.5.1 Plastificação

O estado-limite último de plastificação ocorre quando o momento fletor solicitante em uma seção transversal de uma barra fletida atinge o seu momento de plastificação (M_{pl}), que é determinado através da equação 3.22. Quando o valor de momento de plastificação é atingido, ocorre o surgimento de uma rótula plástica, gerando deformações indefinidas (FAKURY, 2016).

$$M_{pl} = f_y Z \quad (3.22)$$

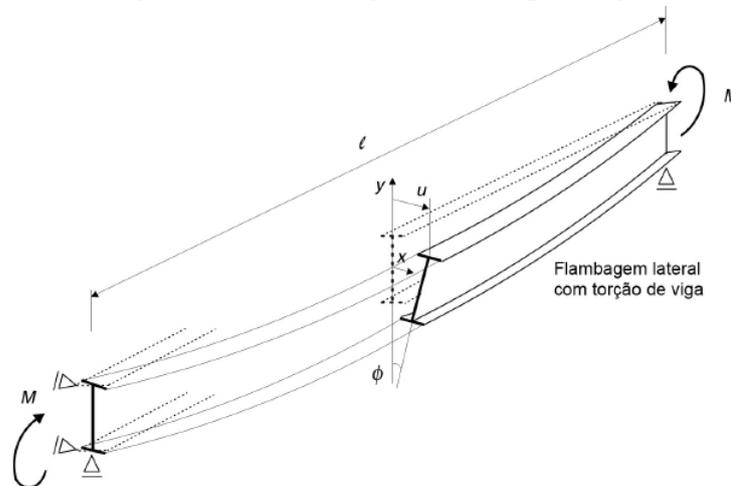
Segundo Fakury (2016), o surgimento de uma rótula plástica representa o colapso apenas de estruturas isostáticas, porém, em um elemento hiperestático acarreta em solicitações não previstas em outras partes da estrutura.

Este estado limite só ocorre quando o elemento em flexão apresenta esbelteza inferior ao mínimo estabelecida para ocorrência de flambagem lateral por torção, o que pode ocorrer em elementos curtos ou contidos lateralmente, e quando o perfil apresenta somente elementos compactos, não suscetíveis a flambagem local.

3.5.2 Flambagem Lateral com torção

A flambagem lateral com torção, ilustrada na figura 3.9, é o fenômeno no qual uma viga submetida a flexão sofre um deslocamento lateral, perpendicular ao sentido de aplicação da carga, juntamente com um giro em relação ao eixo longitudinal da viga (PFEIL, 2022).

Figura 3.9 – Flambagem Lateral por torção



Fonte: Adaptado de Pfeil (2022)

A FLT pode ocorrer em regime inelástico ou em regime elástico. No regime inelástico, a flambagem lateral acontece após a barra atingir o momento fletor de início de escoamento (M_r) e antes de atingir o momento de plastificação (M_{pl}). Este fenômeno se torna possível quando a esbelteza do elemento fletido (λ), em relação ao eixo perpendicular ao eixo de flexão, calculado pela equação 3.23, é menor que a esbelteza limite para flambagem elástica (λ_r) e maior que a esbelteza limite de plastificação (λ_p).

$$\lambda = \frac{L_b}{r} \quad (3.23)$$

Na equação 3.23, r é o raio de giração em relação ao eixo perpendicular ao eixo de flexão, e L_b é o comprimento destravado do perfil em relação ao deslocamento lateral. O valor do momento fletor resistente característico é obtido pela expressão 3.24:

$$M_{Rk} = C_b \left[M_{pl} - (M_{pl} - M_r) \frac{\lambda - \lambda_p}{\lambda_r - \lambda_p} \right] \leq M_{pl} \quad (3.24)$$

No regime elástico, ela ocorre antes da barra atingir o momento M_r , quando $\lambda \geq \lambda_r$. Para um perfil qualquer, o momento crítico para ocorrência da flambagem lateral por torção em regime elástico é dado pela equação 3.25:

$$M_{cr} = C_b \frac{\pi}{L_b} \sqrt{EI_y GJ \left(1 + \frac{EC_w}{GJ} \frac{\pi^2}{L_b^2} \right)} \quad (3.25)$$

E o momento fletor resistente característico é dado pela expressão 3.24:

$$M_{Rk} = M_{cr} \leq M_{pl} \quad (3.26)$$

Na NBR8800 e na AISC360 a equação 3.25 é adaptada para diferentes tipos de perfis.

O coeficiente C_b , presente nas equações 3.24 e 3.26, leva em consideração a distribuição do momento fletor ao longo do elemento fletido. A NBR8800 apresenta o procedimento para determinar o valor do coeficiente C_b no item 5.4.2.4.

Este estado limite só é aplicável quando a flexão ocorre em torno do eixo de maior inércia do elemento e este não apresenta contenção lateral em todo o seu comprimento.

3.5.3 Flambagem local

Os elementos comprimidos de uma viga submetida flexão também podem estar sujeitos ao fenômeno de flambagem local, reduzindo a capacidade resistente do perfil a flexão (PFEIL, 2022).

As seções são classificadas em 3 categorias, de acordo com a existência de elementos esbeltos: seções compactas, seções semi-compactas e seções esbeltas (PFEIL, 2022). Seções compactas

são aquelas que não apresentam elementos esbeltos, para estes perfis a flambagem local não é um estado limite último. Seções semi-compactas são aquelas que apresentam pelo menos um elemento com parâmetro de esbeltez (λ) entre o parâmetro de esbeltez limite de plastificação (λ_p) e o parâmetro de esbeltez limite de início de escoamento (λ_r). O momento resistente para os perfis contido na tabela G.1 do anexo G da NBR8800:2008 é dado pela equação 3.27:

$$M_{Rk} = \left[M_{pl} - (M_{pl} - M_r) \frac{\lambda - \lambda_p}{\lambda_r - \lambda_p} \right] \leq M_{pl} \quad (3.27)$$

Seções esbeltas são aquelas que apresentam pelo menos um elemento com o parâmetro de esbeltez maior que o (λ_r), indicando que ele sofre flambagem local ainda no regime elástico. Para estes tipos de perfis o momento fletor crítico (M_{cr}) é definido no anexo G da NBR8800:2008, de acordo com o tipo de perfil e se a flambagem ocorre na mesa ou na alma do perfil.

Em perfis tubo circulares, além do estado limite de plastificação, a flambagem local é o único estado limite aplicável. De forma análoga ao apresentado anteriormente, pode ocorrer em regime inelástico ($\lambda_p < \lambda \leq \lambda_r$) e regime elástico ($\lambda > \lambda_r$). A NBR8800 e AISC360 determinam o momento fletor resistente característico para tubo circulares pelas equações 3.28 e 3.29:

- para $\lambda_p < \lambda \leq \lambda_r$:

$$M_{Rk} = \left(\frac{0,021E}{D/t} + f_y \right) W \quad (3.28)$$

- para $\lambda > \lambda_r$:

$$M_{Rk} = \frac{0,33E}{D/t} W \quad (3.29)$$

3.6 ESFORÇOS COMBINADOS

Os critérios vistos até então se limitam a obtenção da capacidade resistente para elementos submetidos a um tipo de solicitação isolada. Pilares e terças de cobertura são exemplos de elementos que são solicitados por momentos fletores em duas direções ao mesmo tempo que podem estar sofrendo compressão ou tração (FAKURY, 2016).

Esforços axiais ocorrendo simultaneamente com esforços de flexão podem agravar ou atenuar os efeitos causados por cada um das solicitações isoladamente (FAKURY, 2016). Para isso as normas NBR8800 e AISC360 apresentam uma verificação de segurança de elementos submetidos a esforços combinados. O elemento submetido a esforços axiais e de flexão ao mesmo tempo deve respeitar a expressão 3.30 ou a expressão 3.31, de acordo com a relação entre o esforço axial solicitante e a força axial resistente.

- para $\frac{N_{Sd}}{N_{Rd}} \geq 0,2$:

$$\frac{N_{Sd}}{N_{Rd}} + \frac{8}{9} \left(\frac{M_{x,Sd}}{M_{x,Rd}} + \frac{M_{y,Sd}}{M_{y,Rd}} \right) \leq 1 \quad (3.30)$$

- para $\frac{N_{Sd}}{N_{Rd}} \leq 0,2$:

$$\frac{N_{Sd}}{2N_{Rd}} + \left(\frac{M_{x,Sd}}{M_{x,Rd}} + \frac{M_{y,Sd}}{M_{y,Rd}} \right) \leq 1 \quad (3.31)$$

Das expressões acima, N_{Rd} , $M_{x,Rd}$ e $M_{y,Rd}$ são a resistência axial de cálculo (podendo ser de tração e compressão) e os momentos fletores resistentes de cálculo obtidos dos valores característicos, como apresentado nas seções 3.2, 3.3 e 3.5. N_{Sd} , $M_{x,Sd}$ e $M_{y,Sd}$ são os esforços solicitantes obtidos através da análise da estrutura quando submetidos as cargas atuantes previstas.

4 ESTRUTURAÇÃO DO CÓDIGO

Este capítulo tem por objetivo apresentar a estruturação do pacote Metalpy, como ele está organizado e como se relaciona cada componente do código. Para que se tenha uma melhor compreensão do funcionamento do pacote, será apresentada uma breve introdução sobre os principais conceitos do paradigma de programação orientada a objetos, que serve de fundamento para toda estrutura do módulo desenvolvido.

4.1 ORIENTAÇÃO A OBJETO

Programação orientada a objetos (POO) é um paradigma de no qual um programa é estruturado em blocos, responsáveis por armazenar dados e manipula-los por meio de funções próprias. Na POO estes blocos são chamados de objetos, os dados armazenados são os atributos deste objeto, e as funções que ele pode realizar são seus métodos (MARTHA, 2019).

Para ilustrar o conceito de objeto, será tomado um exemplo do próprio pacote Metalpy. Dentro de um programa criado com o pacote Metalpy, um perfil de aço do tipo I laminado é representado como um objeto. Este objeto armazena os dados relacionados a este perfil, como: área da seção transversal, momentos de inércia, material, entre outras propriedades. Estes dados são os *atributos* do objeto perfil I laminado. Este objeto também apresenta uma função capaz de calcular os índices de esbeltez de diferentes barras, formadas por este perfil, de acordo com os seus comprimentos de flambagem. No contexto da programação orientada a objetos, esta função é um *método* do objeto. Neste método são utilizados os comprimentos de flambagem, que são parâmetros externos, fornecidos pelo usuário (*inputs*), e os raios de giração, que é um atributo interno do objeto, para obter os índices de esbeltez da barra.

Um programa criado com o paradigma de orientação a objetos é formado por diferentes objetos que se relacionam (algumas dessas relações serão vistas diretamente na explicação do funcionamento do pacote Metalpy) com a finalidade de executar uma determinado tarefa. Esta abordagem tem por objetivo gerar um aproveitamento do código, pois um mesmo objeto pode ser utilizado em diferentes programas, e também facilitar a manutenção do código, já que cada objeto pode ser substituído ou alterado de forma isolada (MARTHA, 2019).

A seguir serão apresentados alguns conceitos importantes da programação orientada a objetos utilizados na construção do pacote Metalpy.

4.1.1 Classes

Classes são os elementos básicos que definem quais métodos e atributos um objeto irá apresentar. Elas podem ser interpretadas como um molde para a construção de objetos que apresentam as mesmas características e os mesmos comportamentos (MARTHA, 2019).

Tomado como exemplo a classe `PerfilILam` do pacote `Metalpy`, que define perfis do tipo I laminado e será apresentada na seção 4.2. Nesta classe são definidos atributos que caracterizam os perfis I laminados, como: altura total, largura das mesas, espessura das mesas e espessura da alma. A partir desta classe, podem ser criados objetos de perfil I laminado, que apresentam valores diferentes para cada uma dessas propriedades. Por exemplo, um perfil `W530X85` e um perfil `W460X74` apresentam os mesmos atributos com valores diferentes. Na orientação a objetos, eles são tidos como objetos distintos, criados a partir da mesma classe.

Com a finalidade de um melhor aproveitamento do código e para evitar a reescrita de códigos com a mesma função, a POO traz os conceitos de herança e polimorfismo, que surgem a partir de uma abordagem de hierarquia de classes. Estes dois conceitos serão abordados a seguir.

4.1.2 Herança

Segundo Martha (2019, pg. 43), a definição de herança na programação orientada a objetos é dada da seguinte forma:

Herança é uma característica que permite que uma determinada classe herde as características de outra classe, chamada de superclasse. A classe descendente, chamada de subclasse, adquire todos os métodos e atributos da superclasse, podendo sobrescrevê-los.

A partir desta definição, pode se concluir que herança possibilita que classes mais genéricas (superclasses) deem origem a classes mais específicas (subclasses). A superclasse tem o objetivo de definir atributos e métodos que são comuns as subclasses.

No pacote `Metalpy`, este conceito é utilizado na criação das classes que descrevem os perfis metálicos. Todas as classes de perfis metálicos (`PerfilILam`, `PerfilI`, `TuboRet`, `TuboCir` e `Caixao`) são subclasses da classe `PerfilEstrutural`. A classe `PerfilEstrutural` é uma superclasse, que apresenta os atributos que são comuns a todos os tipos de perfis, como: área da seção transversal, momento de inércia e raio de giração. Ela também apresenta métodos comuns a qualquer tipo de perfil como: métodos que calculam as cargas críticas de flambagem e índice de esbeltez.

As subclasses herdam todos os atributos e métodos e ainda apresentam atributos e métodos

próprios, o que faz com que ela se diferencia das outras subclasses. Tomando como exemplo as classes de perfis I e perfis tubo circulares, as duas classes compartilham os atributos herdados da classe `PerfilEstrutural`, mas possuem atributos que definem características particulares, como largura das mesas, para o caso dos perfis I, e diâmetro nominal, para o caso dos perfis tubo circulares.

4.1.3 Polimorfismo

Polimorfismo é a propriedade que permite que uma subclasse sobrescreva métodos herdados da superclasse. Esta característica tem a finalidade de permitir que os métodos sejam readequados de forma a atender melhor as necessidades da subclasse.

Por exemplo, métodos que calculam as propriedades geométricas de perfis tubo circulares são diferentes dos métodos que calculam as propriedades geométricas de perfis tubo retangulares. Ambos os métodos são comuns às duas classes, porém apresentam uma implementação diferente. Polimorfismo permite que este método seja definido na superclasse, mas seja reimplementado em cada uma das subclasses de forma que sua função seja mais adequada dentro desta classe.

4.2 METALPY

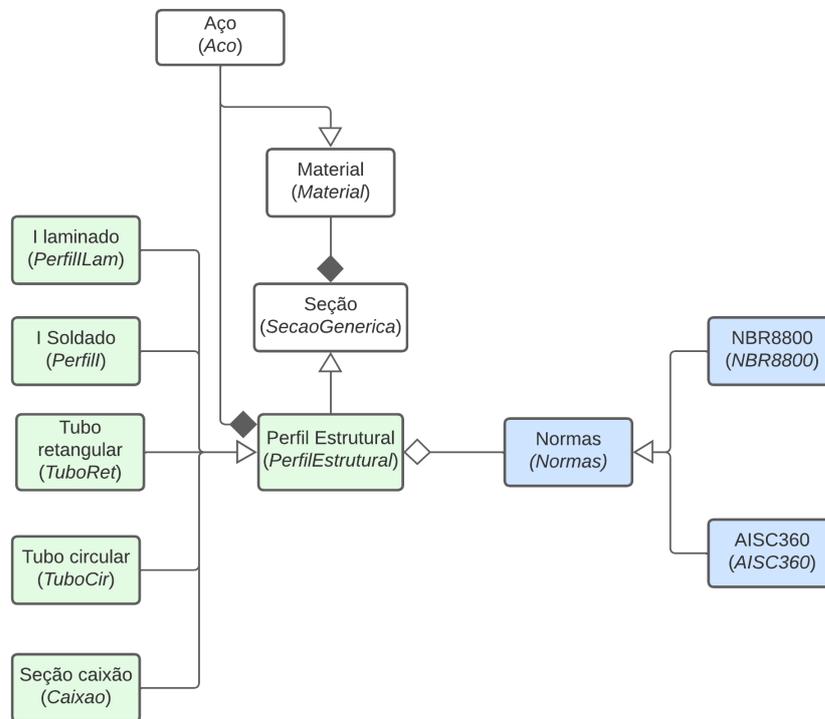
Metalpy é um pacote desenvolvido na linguagem de programação Python e tem a sua arquitetura projetada a partir do paradigma de programação orientada a objetos. Este pacote está implementado em código aberto, o que permite que os usuários acessem o seu código fonte. Essa característica possibilita que os usuários tenham acesso aos critérios utilizados nos cálculos de determinação da capacidade resistente dos perfis, as formulações utilizadas nos cálculos dos parâmetros geométricos e dos parâmetros de estabilidade. Além disso, permite que os usuários modifiquem e acrescentem funcionalidades ao código, de acordo com suas necessidades.

Este pacote foi criado com a finalidade principal de determinar a capacidade resistente de perfis de aço laminados e soldados, a partir dos critérios da norma brasileira *NBR8800* e da norma americana *AISC360*. Ao mesmo tempo, Metalpy foi pensada e estruturada de forma a servir como base para implementação de outros tipos de perfis, materiais e com a possibilidade de se adicionar critérios de outras normativas. Essa característica foi possibilitada pelas propriedades da programação orientada a objetos apresentados anteriormente, permitindo que o usuário possa utilizar as classes básicas já implementadas para criar suas próprias classes.

A figura 4.1 apresenta um diagrama de classes simplificado de toda a estrutura do pacote Metalpy. Neste diagrama são apresentadas todas as classes existente e a relação entre elas (definidas pelas

setas).

Figura 4.1 – Diagrama de classes do pacote Metalpy



Fonte: Elaborado pelo autor

A seta branca com formato triangular indica uma relação de generalização. A classe para a qual a seta está apontada é uma classe genérica (uma superclasse como visto no item 4.1.2) que é herdada por todas as classes que apontam para ela. No diagrama, essa relação existe entre as classes que representam cada um dos perfis e a classe `PerfilEstrutural`, entre a classe `Aco` e a classe `Material`, e nas classes que representam as normas `NBR8800` e `AISC360` e a classe `Normas`.

A seta preta com formato de um losango representam uma relação de composição, um objeto da classe para qual a seta está apontada é composto por um objeto da classe que aponta a seta. Essa relação pode ser vista entre a classe `SecaoGenerica`, que representa uma seção qualquer, e a classe `Material`, que representa um material que compõe a seção, e também, entre as classes `PerfilEstrutural` e a classe `Aco`. Apesar da classe `PerfilEstrutural` ser uma subclasse de `SecaoGenerica`, ela não pode ser composta diretamente por um objeto da classe `Material`, somente da classe `Aco`. Isso ocorre porque a classe `Material` não apresenta atributos que representam a tensão de escoamento e tensão de ruptura do material, necessários para a classe `PerfilEstrutural`.

A seta branca com formato de losango indica uma relação de agregação. A classe que aponta

agrega informação à classe que recebe a seta. No figura 4.1 a classe `Normas` acrescenta métodos nas classes de perfis que permitem que sejam determinadas a capacidade resistente dos perfis.

Como pode se observar na figura 4.1, a estrutura do pacote `Metalpy` está dividida em três partes destacadas na figura por três diferentes cores. Em branco estão destacadas as classes que servem como base do pacote, em verde são destacadas as classes que compõem o módulo de perfis e em azul as classes que compõem o módulo de normas.

A classe `SecaoGenerica` que compõe a base do pacote foi criada com a finalidade de compatibilizar o pacote `Metalpy` com outros pacotes que possam ser desenvolvidos, como um módulo de análise estrutural por exemplo. Essa classe apresenta os atributos básicos necessários de qualquer elemento estrutural que são normalmente utilizados por um programa de análise estrutural, e estão listados abaixo:

- mat:** Material definido pela classe `Material`;
- A:** Área da seção transversal;
- I_x:** Momento de inércia em relação ao eixo X;
- I_y:** Momento de inércia em relação ao eixo Y;
- J:** Constante de torção pura;
- EA:** Rigidez axial;
- EI_x:** Rigidez a flexão em torno do eixo X;
- EI_y:** Rigidez a flexão em torno do eixo Y;
- GJ:** Rigidez à torção.

Esta classe é herdada pela classe `PerfilEstrutural`, fazendo com que todos os perfis criados apresentem as propriedades listadas. Sendo assim, um código de análise estrutural que tenha a classe `SecoeGenerica` como classe base para definir as seções dos elementos estruturais também pode utilizar as classes de perfis criadas a partir da classe `PerfilEstrutural`.

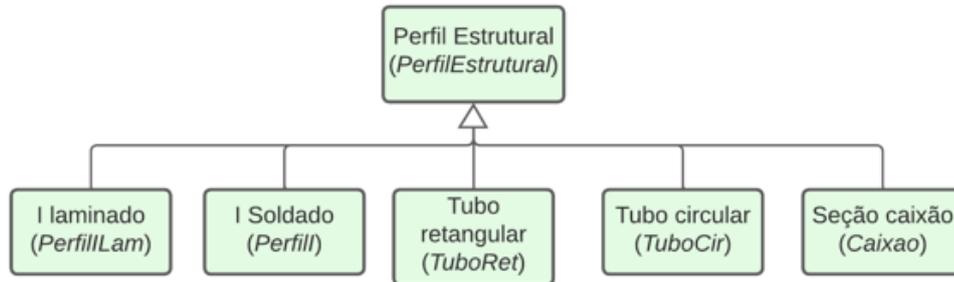
Nas próximas seções deste capítulo serão abordados os dois módulos principais que compõem o pacote `Metalpy`, o módulo **perfis** e o módulo **normas**, e como eles se relacionam de forma a obter a capacidade resistente dos perfis.

4.2.1 Módulo perfis

O módulo de perfis do pacote `Metalpy` dispõe de classes que representam os perfis estruturais. Os objetos criados a partir destas classes armazenam os atributos relacionados aos perfis, como propriedades geométricas e parâmetros de resistência dos materiais, e apresentam métodos que possibilitam os cálculos dos parâmetros de estabilidade de barras formadas pelo perfil que o objeto representa. O módulo conta com classes de perfis dos tipos: I laminado (`PerfilILam`), I soldados (`PerfilI`), tubos retangulares laminados (`TuboRet`), tubos circulares (`TuboCir`) e

seção caixão retangular (Caixao). A figura 4.2 traz um diagrama com a estrutura deste módulo.

Figura 4.2 – Módulo de perfis do pacote Metalpy



Fonte: Elaborado pelo autor

Como pode se observar no diagrama da figura 4.2, o módulo de perfis conta com a classe genérica `PerfilEstrutural` que serve como base para todas as outras classes do módulo. As classes específicas de cada perfil se relacionam por herança (ver 4.1.2) com a classe `PerfilEstrutural`, herdando todos os atributos e métodos desta classe.

Devido ao fato da classe `PerfilEstrutural` ser herdada por todas as outras classes, ela apresenta atributos e métodos que são comuns a todos os tipos de perfis. Além dos atributos herdados da classe `SecaoGenerica`, apresentados anteriormente, a classe `PerfilEstrutural` apresenta os seguintes atributos:

- Wxs:** Módulo de resistência elástico superior em relação ao eixo X (horizontal);
- Wxi:** Módulo de resistência elástico inferior em relação ao eixo X;
- Wys:** Módulo de resistência elástico superior em relação ao eixo Y (vertical), tomado como o lado direito em relação ao eixo Y;
- Wyi:** Módulo de resistência elástico inferior em relação ao eixo Y (vertical), tomado como o lado esquerdo em relação ao eixo Y;
- Zx:** Módulo de resistência plástico em relação ao eixo X;
- Zy:** Módulo de resistência plástico em relação ao eixo Y;
- Awx:** Área efetiva de cisalhamento para cargas aplicadas no sentido do eixo X;
- Awy:** Área efetiva de cisalhamento para cargas aplicadas no sentido do eixo Y;
- x0:** Coordenada X do centro de corte em relação ao centro geométrico do perfil.
- y0:** Coordenada Y do centro de corte em relação ao centro geométrico do perfil.
- Cw:** Constante de empenamento do perfil;

- Afy:** Resistência axial ao escoamento;
- Afu:** Resistência axial à ruptura;
- Mplx:** Momento de plastificação em relação ao eixo X;
- Mply:** Momento de plastificação em relação ao eixo Y;
- Mrx:** Momento de início de escoamento em relação ao eixo X, sem considerar as tensões residuais;
- Mry:** Momento de início de escoamento em relação ao eixo Y, sem considerar as tensões residuais;
- Vplx:** Força cortante de plastificação em relação ao eixo X;
- Vply:** Força cortante de plastificação em relação ao eixo Y;

E apresenta os seguintes métodos:

- indice_esbeltez(Lx, Ly):** Calcula o índice de esbeltez em relação ao eixo X e Y para uma barra com comprimentos de flambagem Lx e Ly;
- par_estabilidade(klx, kly, klz):** Calcula as cargas críticas de flambagem e o momento crítico de flambagem lateral com torção para uma barra com comprimentos de flambagem klx, kly e klz;
- definir_norma(norma):** Associa os métodos de uma das classes de normas ao objeto para determinar a capacidade resistente do perfil;
- prop():** Retorna os parâmetros geométricos do perfil;
- _prop_geo():** Calcula ou busca no banco de dados os parâmetros geométricos do perfil.

Em adição aos atributos listados para a classe genérica `PerfilEstrutural`, cada classe específica de perfis possuem atributos específicos do tipo de perfil. Por exemplo, a classe `PerfilILam` contém atributos que definem a altura total do perfil, altura da alma, largura das mesas e as espessuras da alma e das mesas, enquanto que a classe `TuboCir` contém atributos que definem o diâmetro externo, diâmetro interno e a espessura do perfil. O apêndice A apresenta os diagramas de todas as classes do módulo de perfis com os atributos e métodos de cada uma. As classes específicas de cada perfil podem ser separadas em dois grupos, de acordo com a forma que são definidos os parâmetros geométricos.

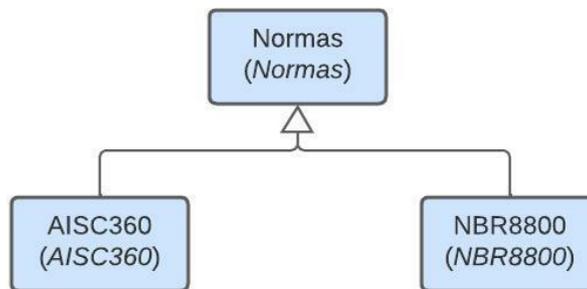
O primeiro grupo é constituído pelas classes `PerfilILam`, `TuboRet` e `TuboCir`. Os parâmetros geométricos destes perfis são obtidos através de bancos de dados, já que os perfis que essas classes representam são perfis normatizados e catalogados, sendo assim, não é necessário métodos para determinar as propriedades geométricas destes perfis. Os dados dos perfis são obtidos através do banco de dados fornecidos por ANSI (2017) e, para os perfis tubo retangulares e circulares, o banco de dados também conta com os perfis do catálogo da Vallourec (s.d). Futuramente, este banco de dados pode ser expandido, contemplando perfis de outros catálogos e perfis normatizados por outras organizações.

O segundo grupo é formado pelas classes `PerfilI` e `Caixao`. Estes perfis podem não apresentar dimensões padronizadas, todos os atributos relacionados a geometria do perfil são calculados através do método `_prop_geo()`. Este método calcula e retorna os atributos geométricos listados para a classe `PerfilEstrutural`.

4.2.2 Módulo normas

O módulo de normas contém as classes que apresentam os critérios de resistência das normas de projeto de estruturas de aço. O módulo conta com a classe `NBR8800`, que apresenta os métodos de acordo com os critérios da NBR8800:2008, e a classe `AISC360`, que apresenta os métodos de determinação da capacidade resistente de perfis de acordo com o AISC360-16. O módulo está estruturado como apresentado na figura 4.3:

Figura 4.3 – Módulo de normas do pacote Metalpy



Fonte: Elaborado pelo autor

Este módulo apresenta a classe base `Norma` e as subclasses `AISC360` e `NBR8800`, que herdam a superclasse `Norma`. Neste módulo a superclasse `Norma` serve como uma guia para a implementação de subclasses de normas específicas. Na POO, uma classe com essa característica é chamada de interface. Esta classe não apresenta nenhum atributo, apenas a indicação dos métodos que devem ser implementados pelas classes específicas, para que a nova norma implementada funcione corretamente dentro do pacote `Metalpy`.

Todos os métodos indicados pela classe `Normas` tem como parâmetros de entrada um objeto que representa um perfil, o qual se deseja calcular a resistência, e o parâmetro `data`. Este parâmetro indica para o método retornar valores intermediários utilizados no cálculo da capacidade resistente dos perfis. É um parâmetro do tipo booleano, podendo ser `True` ou `False`. Os métodos retornam os valores intermediários, caso o parâmetro `data` seja indicado como `True`. Todos os métodos contidos na classe `normas` estão listados abaixo:

`Ntrd_brt(perfil, data=False)`: Determina a força de tração resistente de cálculo para o ELU de escoamento da seção bruta;

`Ncrd(perfil, klx, kly, klz, data=False)`: Determina a compressão resistente de cálculo de um perfil com comprimentos de flambagem `klx`, `kly` e `klz`;

- Vrdx(perfil, Lv=None, data=False):** Determina a força cortante resistente de cálculo para cargas aplicadas na direção X (horizontal) do perfil. Caso o perfil seja um tubo circular, deve ser fornecido o parâmetro (Lv), que indica a distância entre as seções de esforço cortante máximo e nulo;
- Vrdy(perfil, a=None, Lv=None, data=False):** Determina a força cortante resistente de cálculo para cargas aplicadas na direção Y (vertical) do perfil. O parâmetro Lv é o mesmo descrito para o método Vrdx, e o parâmetro a é a distância entre enrijecedores, caso o perfil seja um I laminado ou soldado;
- Mrdx(perfil, Lb=None, Cb=1, data=False):** Determina o momento fletor resistente de cálculo de um perfil em relação ao eixo X. Para perfis sujeitos a flambagem lateral com torção, deve ser fornecido o comprimento destravado (Lb) e o fator Cb como parâmetros de entrada.
- Mrdy(perfil, Lb=None, Cb=1, data=False):** Determina o momento fletor resistente de cálculo de um perfil em relação ao eixo Y.
- verif_NM(perfil, ..., data=False):** Verifica perfis flexo-comprimidos ou flexo-tracionados.

As classes específicas de cada norma implementam os métodos listados acima de acordo com os seus critérios de segurança. Além destes métodos, elas também contam com métodos auxiliares para realização de cálculos intermediários. Por exemplo, para o cálculo do fator Q , que determina a redução capacidade resistente de um perfil comprimido devido ao fenômeno de flambagem local, a classe NBR8800 apresenta o método auxiliar `_Q()`. Já a classe AISC360 apresenta o método auxiliar `_Aef()`, responsável por calcular a área efetiva de um perfil comprimido. Todos os métodos das classes NBR8800 e AISC360 estão apresentados no apêndice B.

4.2.3 Relação entre o módulo perfis e módulo normas

Os módulo perfis e o módulo normas são módulos independentes e podem se relacionar de duas formas. Na primeira delas, os métodos implementados nas classes Normas tem como um dos seus argumentos um objeto do tipo Perfil Estrutural. Neste caso, o perfil é passado como um dado de entrada dos método de determinação da capacidade resistente. Os exemplos de aplicação que serão apresentados na seção 5.2 utilizam essa forma de relação.

A segunda forma é associando um objeto que representa uma das normas a um objeto que representa um perfil estrutural. Nesse caso, os métodos das classes de normas são adquiridos pelo objeto que representa o perfil, apresentando os mesmos argumentos originais, com exceção do próprio perfil. Esta associação é feita pelo método `definir_norma(norma)` da classe de perfis. Alternativamente, está associação pode ser feita na inicialização do objeto perfil, através do parâmetro `norma`. Esta relação é utilizada nos exemplos da seção 5.1.

5 EXEMPLO

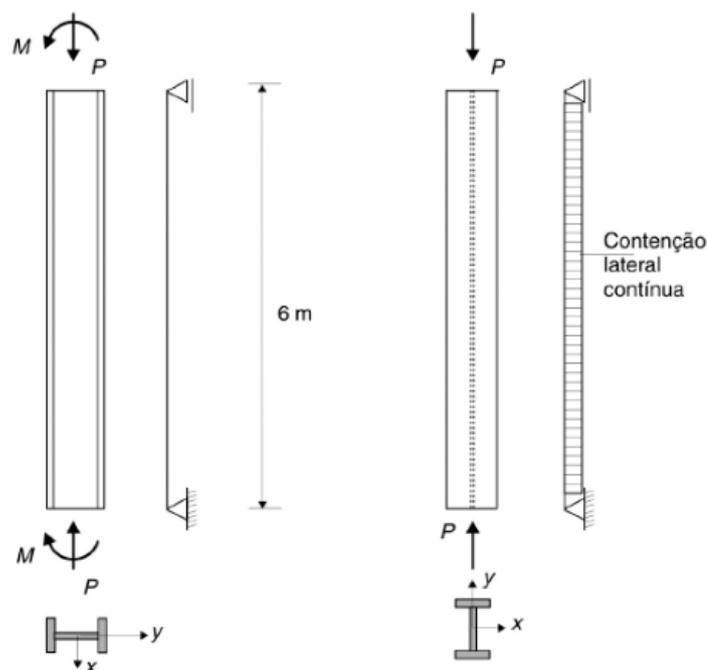
5.1 EXEMPLOS DE VERIFICAÇÃO

Nesta seção serão apresentadas exemplos de utilização do pacote Metalpy. Estes exemplos tem a finalidade de demonstrar a forma de uso do pacote e expor algumas funcionalidades. Os exemplos desta seção são exemplos resolvidos extraídos de livros e manuais de projetos de estruturas de aço com o objetivo de validar os resultados calculados com o pacote Metalpy, os comparando com os resultados apresentados pelas referências. Estes e outros exemplos podem ser acessados em Severgnini (2022).

5.1.1 Perfil I soldado submetido a flexo-compressão

Este exemplo foi apresentado por Pfeil (2022) e consiste na verificação de segurança de uma coluna submetida à flexo-compressão. A coluna, apresentada na figura 5.1, é formada por um perfil I soldado padronizado CVS 450X116 (ver ABNT (2013b)) e aço MR250. A coluna está sujeita a uma força de compressão $P=800$ kN e um momento fletor $M=50$ kN.m em relação ao eixo de maior inércia do perfil. A coluna apresenta um comprimento de 6m e está continuamente contida em relação ao plano perpendicular a alma.

Figura 5.1 – Perfil CVS 450X116 submetido a flexo-compressão



Fonte: Pfeil (2022)

O código abaixo apresenta a resolução deste exemplo através do pacote Metalpy.

```

1  from metalpy.perfis import PerfilI
2  from metalpy.material import Aco
3
4  #Definindo o aço do tipo MR250 com as propriedades em kN/cm²
5  MR250 = Aco(20000, 0.3, 25, 30, 0.6)
6
7  #Dados do perfil CVS 450 X 16 em cm
8
9  # Altura total
10 d = 45
11 # Largura da mesa superior e inferior
12 bfs=bfi= 30
13 # Espessura da alma
14 tw = 1.25
15 # Espessura da alma superior e inferior
16 tfs=tfi= 1.6
17
18 # Criando o perfil CVS 450 X 116 com métodos de determinação da
19 # capacidade resistente da norma NBR8800:2008
20 CVS450X116 = PerfilI(d, bfs, bfi, tw, tfs, tfi, aco=MR250, norma='NBR8800')
21
22 #Obtendo as respostas com Metalpy
23
24 #Propriedades do perfil
25 A = CVS450X116.A #área do perfil
26 Wx = CVS450X116.Wx #módulo elástico em relação ao eixo X
27 rx = CVS450X116.rx #raio de giração em relação ao eixo X
28 Zx = CVS450X116.Zx #módulo plástico em relação ao eixo X
29 Iy = CVS450X116.Iy #momento de inércia em relação ao eixo Y
30 Wy = CVS450X116.Wy #módulo elástico em relação ao eixo Y
31 ry = CVS450X116.ry #módulo plástico em relação ao eixo Y
32 J = CVS450X116.J #inércia a torção
33 Cw = CVS450X116.Cw #constante de empenamento
34 esb_mesa = CVS450X116.esb_mesa #esbeltez da mesa
35 esb_alma = CVS450X116.esb_alma #esbeltez da alma
36
37 # Verificação à flexo-compressão
38
39 #Solicitações
40 Nsd = -1.4 * 800
41 Msdx = 1.4 * 50 * 1.039 * 100 # Considerando os efeitos de 2ª ordem
42 Msdy = 0
43
44 # Obtendo os valores da resistência do perfil
45
46 #Resistência a compressão
47 Nrd, dados_Ncrd = CVS450X116.Ncrd(klx=600, kly=0.1, klz=0.1, data=True)
48 ier = dados_Ncrd.ier #Índice de esbeltez reduzido
49
50 #Momento resistente em relação ao eixo X
51 Mrd, dados_Mrd = CVS450X116.Mrdx(Lb=0, Cb=1, data=True)
52
53 #Verificação a flexo-compressão
54 verif= CVS450X116.verif_NM(Nsd, Msdx, Msdy, Lbx=0, Cb=1, klx=600, kly=1, klz=1)

```

Nas linhas 1 e 2, ocorre a importação das classes `PerfilI` e `Aco` do módulo `perfis` e do módulo `material`, respectivamente. Na linha 5, é criado um objeto da classe `Aco`, que representa o aço MR250, a partir das propriedades do material: módulo de elasticidade, coeficiente de Poisson, tensão de escoamento, tensão de ruptura e o valor de 0.6 que representa a relação entre a resistência ao cisalhamento e a resistência ao escoamento. Na linha 20, é criado um objeto da classe `PerfilI` que representa um perfil do tipo I soldado CVS450X16, com as características geométricas declaradas da linha 10 à linha 16, constituído do aço MR250 (criado na linha 5) e que apresentará os métodos de determinação da capacidade resistente da NBR8800, indicado pelo parâmetro `norma = 'NBR8800'`.

Da linha 25 à linha 35, são obtidas as propriedades geométricas do perfil CVS450X16, calculadas pelos métodos da classe `PerfilI` do pacote `Metalpy`. Nas linhas 47 e 51 são calculadas as capacidades resistentes a compressão e ao momento fletor do perfil, respectivamente. O argumento `data=True` indica ao método que retornem dados intermediários utilizados nos cálculos e sejam armazenados nas variáveis `dados_Ncrd` e `dados_Mrd`. Na linha 48, o valor do índice de esbeltez reduzido utilizado no cálculo da capacidade resistente do perfil é obtido da variável `dados_Ncrd`. Por fim a linha 54 realiza a verificação à flexo compressão da coluna formada pelo perfil CVS450X16. Cabe ressaltar que para a realização da verificação à flexo-compressão, o único passo necessário é o apresentado na linha 54 do código, o código apresentado da linha 25 à linha 51 serve apenas para apresentar alguns valores intermediários e para demonstrar como utilizar algumas funcionalidades do pacote.

A figura 5.2 apresenta um quadro comparativo entre as respostas apresentadas por Pfeil (2022) e os valores calculados com `Metalpy`. O resultado da verificação à flexo-compressão está indicado pelo termo `NM`.

Figura 5.2 – Repostas do exemplo 1

	Referência	Metalpy
A(cm²)	148.30	1.482500e+02
Wx(cm³)	2348.00	2.348164e+03
rx(cm)	18.88	1.887809e+01
Zx(cm³)	2629.00	2.629213e+03
Iy(cm⁴)	7207.00	7.206803e+03
Wy(cm³)	480.00	4.804536e+02
ry(cm)	6.97	6.972267e+00
J(cm³)	109.00	1.101752e+02
Cw(cm⁶)	3393704.00	3.390408e+06
bf/2tf	9.40	9.375000e+00
h/tw	33.00	3.344000e+01
lambda_0	0.36	3.576830e-01
Nrd(kN)	3192.00	3.193643e+03
Mrdx(kN.cm)	597.50	5.975483e+04
NM	0.47	4.588869e-01

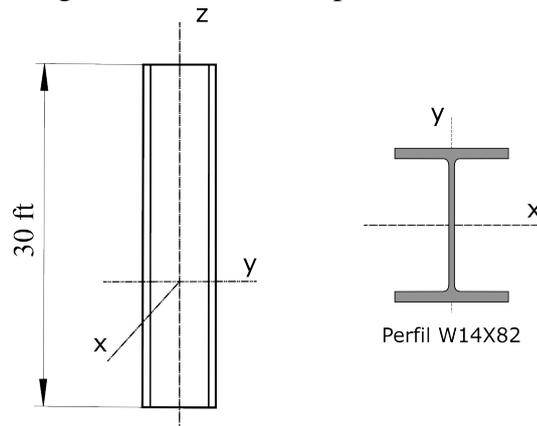
Fonte: Elaborado pelo Autor

5.1.2 Perfil I laminado submetido a flexo-tração

Este exemplo, apresentado em ANSI (2015) (ver exemplo H.1), traz a verificação à flexo-tração de uma barra formada por um perfil W14X82 (equivalente ao W360X122 em mm e kg/m), de acordo com os critérios da AISC360-16. A barra, representada na figura 5.3, apresenta um comprimento de 30ft (914.4 cm) e está submetida a tração de 29 kips (128.89 kN), devido a uma carga permanente, e uma tração de 87kips (386.4) devido uma carga de utilização. Além dos esforços de tração, a barra também está submetida a momentos fletores nas direções x e y devido a carregamentos distribuídos na barra. Os valores dos momentos solicitantes devido as cargas permanentes e variáveis estão apresentados abaixo:

- $M_{sxp} = 32kip.ft$ (4334.93kN.cm) - Momento em X devido a carga permanente
- $M_{sxa} = 96kip.ft$ (13004.8kN.cm) - Momento em X devido a carga variável
- $M_{syp} = 11.3kip.ft$ (1530.77kN.cm) - Momento em Y devido a carga permanente
- $M_{sya} = 33.8kip.ft$ (4578.77kN.cm) - Momento em Y devido a carga variável

Figura 5.3 – Coluna de perfil W14X82



Fonte: Elaborado pelo Autor

As solicitações de cálculo, após a combinação dos esforços, estão apresentadas a seguir:

- $N_{tsd} = 174 \text{kips}$ (773.33kN)
- $M_{xsd} = 192 \text{kip.ft}$ (26009.6kN.cm)
- $M_{ysd} = 67.6 \text{kip.ft}$ (9157.55kN.cm)

A barra é composta por aço ASTM A992, que apresenta $f_y = 50 \text{ksi}$ (34.5kN/cm^2) e $f_u = 65 \text{ksi}$ (44.8kN/cm^2). O coeficiente C_b utilizado no exemplo é igual a 1.41. O exemplo será resolvido em unidades de comprimento em cm e de força em kN e, posteriormente, os resultados serão convertidos para unidades de in e kips, conforme está apresentado em ANSI (2015). O código a seguir apresenta a resolução deste exemplo.

```

1  from metalpy.perfis import PerfilILam
2  from metalpy.material import Aco
3
4  #Unidades para conversão
5  cm_to_in = 0.394
6  ft_to_cm = 30.48
7  ft_to_cm = 30.48
8  ksi_to_kN_cm2 = 0.6894
9  kN_to_kips = 0.225
10 kNcm_to_kipft = kN_to_kips * (1/ft_to_cm)
11
12 #Definição do Aço com as propriedades em ksi sendo convertidas para kN/cm²
13 E = 29000 * ksi_to_kN_cm2
14 fy = 50 * ksi_to_kN_cm2
15 fu = 65 * ksi_to_kN_cm2
16 v = 0.3
17
18 #Definindo um aço ASTM A992 em unidades de kN/cm²
19 A992 = Aco(E, v, fy, fu, 0.6)

```

```

20
21 #Definindo um perfil W360X122 com as propriedades em cm
22 W360X122 = PerfilILam('W360X122', A992, und='cm', norma='AISC360')
23
24 #Respostas com Metalpy
25
26 #Propriedades geométricas do perfil
27 A = W360X122.A * cm_to_in ** 2 #área do perfil
28 Wx = W360X122.Wx * cm_to_in**3 #módulo elástico em relação ao eixo X
29 Zx = W360X122.Zx * cm_to_in**3 #módulo plástico em relação ao eixo X
30 Wy = W360X122.Wy * cm_to_in**3 #módulo elástico em relação ao eixo Y
31 Zy = W360X122.Zy * cm_to_in**3 #módulo plástico em relação ao eixo X
32 Iy = W360X122.Iy * cm_to_in**4 #momento de inércia em relação ao eixo Y
33
34 # Calculando a força de tração resistente de cálculo de acordo com o ELU
35 # de escoamento da seção bruta
36 mp_Ntrd = W360X122.Ntrd_brt() * kN_to_kips
37
38 # Calculando o momento resistente Mx e obtendo os dados de cálculo
39 Cbx = 1.41
40 Lbx = 30 * ft_to_cm
41
42 Mrdx, dados_Mrdx = W360X122.Mrdx(Lb = Lbx, Cb = Cbx, data=True)
43 Mrdx = Mrdx * kNcm_to_kipft
44 Lp = dados_Mrdx.FLT.Lp * cm_to_ft
45 Lr = dados_Mrdx.FLT.Lr * cm_to_ft
46
47 # Calculando o momento resistente My
48 Mrdy = W360X122.Mrdy(Lb = 0, Cb=1) * kNcm_to_kipft
49
50 # Verificação a flexo-tração
51 Nsd = 174 / kN_to_kips
52 Msdx = 192 / kNcm_to_kipft
53 Msdy = 67.6 / kNcm_to_kipft
54
55 mp_verif = W360X122.verif_NM(Nsd, Msdx, Msdy, Lbx= 30 * ft_to_cm, Cb=1.41,\
56                               klx=1, kly=Lbx, klz=1)

```

O código acima apresenta os mesmos passos apresentados no exemplo 1, com a diferença que na linha 1 ocorre a importação da classe `PerfilILam`.

O objeto que representa o perfil W360X122 é criado na linha 21 a partir da classe `PerfilILam`. Os parâmetros de entrada para a criação do perfil I laminado diferem dos parâmetros necessários para criação de um perfil I soldado, apresentado em 5.1.1. Para a criação do objeto que representa o perfil W360X122 os dados de entrada necessários são: o nome do perfil, conforme a base de dados de perfis I laminados, o objeto que representa o aço que o constitui (criado na linha 18), a unidade base das propriedades geométricas do perfil e a norma da qual serão utilizados os critérios de verificação.

Da linha 26 à linha 31, estão sendo obtidas as propriedades geométricas do perfil com os seus

valores sendo multiplicados por fatores de conversão.

A força resistente a tração, de acordo com o estado limite de escoamento da seção bruta, é obtido na linha 35. Os momentos resistentes do perfil e a verificação à flexo-tração, que estão apresentados da linha 37 à linha 54, seguem o mesmo roteiro do exemplo do item 5.1.1. A figura 5.4 apresenta o quadro comparativo dos valores apresentados em (ANSI, 2015) e os obtidos com Metalpy.

Figura 5.4 – Respostas do Exemplo 2

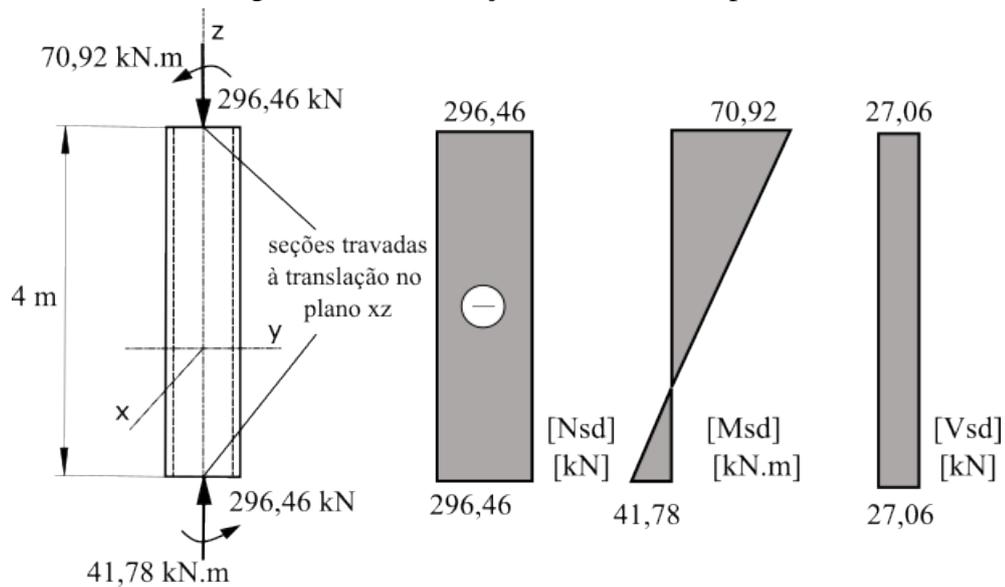
	Referência	Metalpy
A(in²)	24.000	24.061580
Wx(in³)	123.000	123.549228
Zx(in³)	139.000	139.451604
Wy(in³)	29.300	29.358232
Zy(in³)	44.800	44.893630
Iy(in⁴)	148.000	148.445009
Lp(ft)	8.760	8.766701
Lr(ft)	33.200	33.020622
Ntrd(kips)	1080.000	1082.029095
Mrdx(kips.ft)	492.000	491.455393
Mrdy(kips.ft)	168.000	168.107983
NM	0.873	0.873203

Fonte: Elaborado pelo Autor

5.1.3 Perfil tubo retangular submetido a flexo-compressão

Este exemplo foi apresentado por Araújo (2016) (ver exemplo 3.7.8) e nele é realizada a verificação de um pilar flexo-comprimido, representado na figura 5.5, formado por um perfil quadrado TQ220X220X16 e aço VMB 350 ($f_y = 350MPa$). O pilar tem um comprimento de 4m e está submetido a um força de compressão $N_{csd} = 296,46kN$ e um momento fletor $M_{xsd} = 70.92kN.m$, sendo essas, solicitações de cálculo.

Figura 5.5 – Solicitações de cálculo do pilar



Fonte: Araújo (2016)

O código a seguir demonstra a obtenção das resistências a compressão e flexão do pilar, e realiza a verificação para os esforços combinados.

```

1  from metalpy.perfis import TuboRet
2  from metalpy.material import Aco
3
4  #Definição do Aço com a propriedades em kN/cm2
5  VMB350 = Aco(20000, 0.3, 35, 45.0, 0.6)
6  L = 400 #cm
7
8  # Definindo um perfil do tipo tubo retangular TQ 220X220X16 com as
9  # propriedades em cm
10 TQ220X16 = TuboRet('TQ220X220X16', VMB350, und='cm', norma='NBR8800')
11
12 #Respostas com Metalpy
13 A = TQ220X16.A # área total do perfil
14 IX = TQ220X16.Ix # momento de inércia em relação ao eixo X
15 Iy = TQ220X16.Iy # momento de inércia em relação ao eixo Y
16 Wx = TQ220X16.Wx # módulo elástico em relação ao eixo X
17 Wy = TQ220X16.Wy # módulo elástico em relação ao eixo Y
18 Zx = TQ220X16.Zx # módulo plástico em relação ao eixo X
19 rx = TQ220X16.rx # raio de giração em relação ao eixo X
20 ry = TQ220X16.ry # raio de giração em relação ao eixo Y
21 J = TQ220X16.J # inércia a torção
22 esb_mesa = TQ220X16.esb_mesa # esbeltez das mesas
23 esb_alma = TQ220X16.esb_alma # esbeltez das almas
24
25 # Calculando as cargas críticas de flambagem através do método
26 # par_estabilidade()
27
28 par_estabilidade = TQ220X16.par_estabilidade(Lx=400, Ly=400, Lz=400)
29
30 Nex = par_estabilidade.Nex
31 Ney = par_estabilidade.Ney

```

```

32 Nez = par_estabilidade.Nez
33
34 # Calculando a força de compressão resistente de cálculo através do método
35 # Ncrd e obtendo os dados do cálculo
36 Ncrd, dados_Ncrd = TQ220X16.Ncrd(Lx=400, Ly=400, Lz=400 , data=True)
37
38 Q = dados_Ncrd.Q
39 ier = dados_Ncrd.ier
40 Chi = dados_Ncrd.Chi
41
42 #Calculando o momento resistente de cálculo
43
44 Mrdx = TQ220X16.Mrdx()
45
46 # Verificação a flexo-compressão
47 Ncsd = -296.46
48 Mxsd = 7092
49 Mysd = 0
50
51 verif = TQ220X16.verif_NM(Ncsd, Mxsd, Mysd, klx=Lx, kly=Ly, klz=Lz)

```

Na linha 10, do código acima é criado o objeto que representa o perfil TQ220X220X16, a partir da classe `TuboRet`, importada na linha 1. O perfil é criado de forma similar ao perfil I laminado do exemplo apresentado em 5.1.2.

O código deste exemplo apresenta, como novidade, o uso do método `par_estabilidade()` utilizado na linha 31. Este método retorna os parâmetros de estabilidade do perfil (cargas críticas de flambagem, tensões críticas de flambagem e momento crítico de flambagem lateral com torção), de acordo com os comprimentos de flambagem fornecidos. Nas linhas 33, 34 e 35, são obtidas as cargas críticas de flambagem do perfil. O restante do código apresenta a obtenção da resistência a compressão, do momento resistente e a verificação à flexo-compressão, da mesma forma que nos exemplos anteriores. A figura 5.6 apresenta um quadro com os valores obtidos por Araújo (2016) e os obtidos com Metalpy.

Figura 5.6 – Respostas do Exemplo 3

	Referência	Metalpy
A(cm²)	129.00	129.000000
Ix(cm⁴)	7812.00	7812.000000
Iy(cm⁴)	7812.00	7812.000000
Wx(cm³)	710.00	710.000000
Wy(cm³)	710.00	710.000000
Zx(cm³)	881.00	881.000000
rx(cm)	7.78	7.781911
ry(cm)	7.78	7.781911
J(cm³)	13970.00	13970.000000
bf/2tf	10.75	10.750000
h/tw	10.75	10.750000
Q	1.00	1.000000
Nex (kN)	9638.00	9637.668698
Ney(kN)	9638.00	9637.668698
Nez(kN)	889000.00	887259.246130
lambda_0	0.68	0.684452
Chi	0.93	0.927738
Ncrd(kN)	3817.00	3807.943833
Mrdx(kN.cm)	28032.00	28031.818182
NM	0.29	0.291925

Fonte: Elaborado pelo Autor

5.1.4 Força resistente ao corte de perfil tubo circular

Este exemplo foi apresentado por ANSI (2015) (ver exemplo G.5) e nele é determinada a força cortante resistente de uma viga formada por um perfil tubo circular HSS 16X0.75 (equivalente ao HSS406.4X9.5 em mm), através do pacote Metalpy. O aço que compões a viga é o ASTM A500 grau C e apresenta as seguintes propriedades:

- $A = 17.2 \text{ in}^2$
- $D/t = 45.8$

Assim como no exemplo do item 5.1.1, este exemplo será resolvido em unidades de comprimento

em cm e de força em kN. Os resultados são convertidos ,posteriormente, para unidades de in e kips. O código a seguir apresenta a obtenção da força resistente de corte da viga descrita.

```

1  from metalpy.perfis import TuboCir
2  from metalpy.material import Aco
3
4  #Definindo variáveis para conversão
5  in_to_cm = 2.54
6  cm_to_in = 0.394
7  kN_cm2_to_ksi = 1.45
8  kN_to_kips = 0.225
9
10 #Definição do Aço com as propriedades em kN/cm2
11 A500 = Aco(20000, 0.3, 31.7, 43.0, 0.6)
12
13 #Criando uma instancia do perfil TuboCir
14 tubo40 = TuboCir('HSS406.4X9.5', A500, und='cm', norma='AISC360')
15
16 #Respostas com Metalpy
17 A = tubo40.A * cm_to_in ** 2
18 esb = tubo40.esb
19
20 # Calculando o momento resistente de cálculo e obtendo os dados do cálculo
21 Lv = 192 * in_to_cm
22 Vrd, dados = tubo40.Vrdy(Lv = Lv, data=True)
23 Fcr = dados.Fcr * kN_cm2_to_ksi
24 Vrd = Vrd * kN_to_kips

```

O objeto que representa o tubo circular HSS406.4X9.5 é criado, na linha 14 do código acima, através da classe `TuboCir`, de forma similar aos objetos que representam os perfis I laminados e tubos retangulares, apresentados nos exemplos 5.1.2 e 5.1.3. A força cortante resistente do perfil é obtida na linha 21 através do método `Vrdy()`, que determina a resistência de corte na direção do eixo Y. Como se trata de um perfil tubo circular, o mesmo valor poderia ser obtido pelo método `Vrdx()`, que determina a resistência na direção do eixo X.

A figura 5.7 apresenta um quadro comparativo com os resultados apresentados por ANSI (2015) e os valores obtidos com Metalpy.

Figura 5.7 – Respostas do Exemplo 4

	Referência	Metalpy
A (in²)	17.2	17.231196
D/t	45.8	45.800000
Fcr	27.6	27.579000
Vrd (kips)	213.0	213.761025

Fonte: Elaborado pelo Autor

5.2 APLICAÇÕES

Esta seção apresenta exemplos de aplicações dos recursos do pacote Metalpy. Estes exemplos tem o objetivo demonstrar as potencialidades do pacote e apresentar algumas vantagens em se ter acesso a um pacote em código aberto.

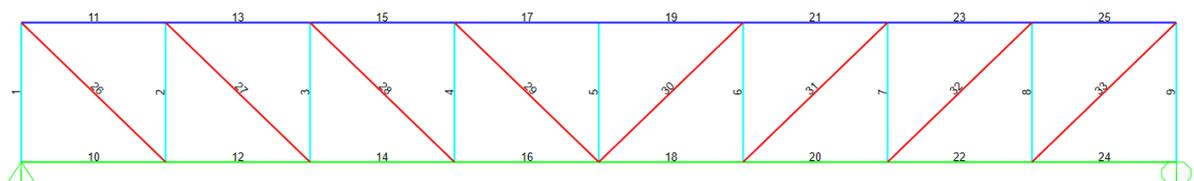
Nos exemplos que serão apresentados nesta seção, buscou-se abordar as possibilidades de integração do pacote Metalpy com outros softwares de engenharia. Para isto foi utilizado o software comercial de análise estrutural SAP2000 (CSI, 2016) e o software de modelagem Revit (AUTODESK, 2022). Nestes exemplos o pacote Metalpy é utilizado na criação de ferramentas que auxiliam na verificação de segurança e na escolha de perfis.

5.2.1 Rotina de verificação integrada ao SAP2000

O SAP2000 (CSI, 2016) é um software comercial de análise e dimensionamento de estruturas bastante difundido no mercado. Ele conta com recursos para dimensionamento de estruturas de aço, podendo ser realizado segundo os critérios de dimensionamento de diversas normas, como: a AISC, o eurocode, a norma chinesa e a norma italiana, mas não conta com os critérios da normativa brasileira. Este exemplo de aplicação integra o SAP2000 com o pacote Metalpy em um rotina que possibilita o dimensionamento da estrutura de acordo com os critérios estabelecidos pela norma NBR8800:2008. É importante salientar que esta rotina não é integrada diretamente ao software SAP2000 e não faz parte do pacote Metalpy. Ela utiliza, simultaneamente, os recursos de ambos em um código independente que realiza a verificação dos perfis.

Neste exemplo o programa SAP2000 é utilizado para modelar a treliça pratt, que faz parte da estrutura da passarela proposta por Henrich (2019), apresentada na figura 5.8, e obter os esforços solicitantes de cada uma das barras. O pacote Metalpy é utilizado para fazer a verificação de segurança das barras, de acordo com os estados limites últimos de barras solicitadas axialmente por tração e compressão.

Figura 5.8 – Modelo estrutural da treliça analisada



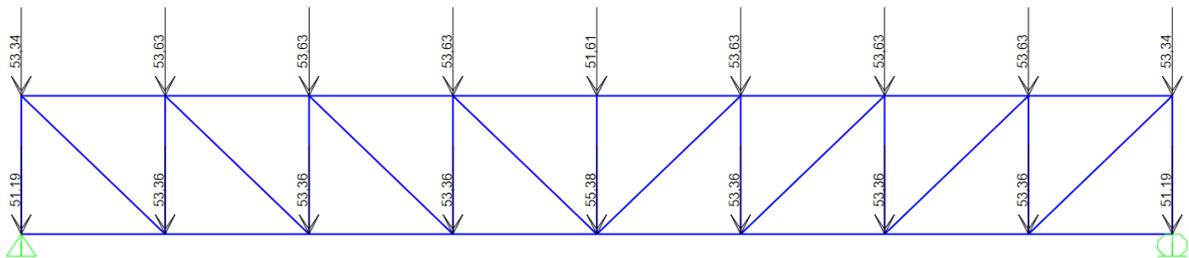
Fonte: Elaborada pelo autor

A treliça da figura 5.8 é formada por 33 barras de perfis tubulares retangulares e quadrados do catalogo da Vallourec. As barras estão agrupadas de forma que, ao todo, a estrutura apresente 4 seções transversais diferentes. Os grupos estão indicados pelas cores das barras da figura 5.8. As

diagonais da treliça (em vermelho) são formadas por tubos de perfil TR100x80x8,8, os montantes (em ciano) são formados por perfis TR150x120x6,4, as barras do banzo superior (em azul) são formadas por perfis TQ160X160X8 e as barras do banzo inferior (em verdes) são formadas por perfis TQ100X80X8,8.

A estrutura apresenta uma altura total de 5m e um vão de 41,35m. Para este exemplo foi adicionado um carregamento de 50kN em cada um dos nós da estrutura, mais o peso próprio da estrutura, resultando no carregamento apresentado na figura 5.9.

Figura 5.9 – Cargas do modelo



Fonte: Elaborada pelo autor

Como este é um exemplo que tem o intuito de ilustrar como os módulos do pacote Metalpy podem ser utilizados, as barras foram analisadas apenas para o caso de cargas mostrada na figura 5.9 e foram realizadas somente a verificação de segurança a compressão e a tração das barras. A rotina utilizada para obter as solicitações e realizar a verificação das barras está apresentada a seguir.

```

1  #Importação do scrip de inicialização do SAP2000
2  from iniciar_sap import IniciarSAP
3
4  #Importando as classes necessárias do pacote Metalpy
5  from metalpy.perfis import TuboRet
6  from metalpy.material import Aco
7  from metalpy.normas import NBR8800
8
9  # Iniciando o SAP2000
10 modelo_sap = IniciarSAP()
11 modelo_sap.InitializeNewModel()
12
13 #Abrindo o Modelo de Analise no Sap
14 FileName = 'C:\CSiAPIexample\TrelizaModelo.sdb'
15 ret = modelo_sap.File.OpenFile(FileName)
16
17 # Verificando o se o modelo está com a análise processada
18 if not modelo_sap.GetModelIsLocked():
19
20     # Rodando a análise caso ela não esteja processada
21     modelo_sap.Analyze.RunAnalysis()
22
23     #Obtendo o nome das barras da treliça do SAP2000

```

```

24 barras_sap = modelo_sap.FrameObj.GetNameList() [1]
25
26 for barra in barras_sap:
27     # -----
28     # OBTENÇÃO DOS DADOS DO SAP2000
29     # -----
30
31     # Obtendo o nome do perfil da barra
32     perfil = modelo_sap.FrameObj.GetSection(barra) [0]
33
34     # Obtendo as propriedades do material da barra
35     mat_perfil_sap = modelo_sap.PropFrame.GetMaterial(perfil) [0]
36
37     E, v = modelo_sap.PropMaterial.GetMPIsotropic(mat_perfil_sap) [:2]
38     fy, fu = modelo_sap.PropMaterial.GetOSteel_1(mat_perfil_sap) [:2]
39
40     # Obtendo o comprimento a as solicitações das barras
41     modelo_sap.Results.Setup.SetCaseSelectedForOutput(barra)
42     resultados = modelo_sap.Results.FrameForce(barra, 0)
43
44     L = resultados[2] [-1] # Comprimento da barra
45     Nsd = max(resultados[8]) # Esforço axial na barra
46
47     # -----
48     # REALIZANDO A VERIFICAÇÃO COM METALPY
49     # -----
50     # Criando o aço correspondente a barra a ser verificada
51     aco = Aco(E, v, fy, fu, 0.6)
52
53     # Criando o perfil correspondente a barra a ser verificada
54     perfil_mp = TuboRet(perfil, aco, und='cm')
55
56     if Nsd > 0:
57         Nrd = NBR8800.Ntrd_brt(perfil_mp)
58
59     elif Nsd <= 0:
60         Nrd = NBR8800.Ncrd(perfil_mp, klx=L, kly=L, klz=L)
61
62     if Nrd >= abs(Nsd):
63         print('Barra ' + barra + ': Ok!', ' - Nsd/Nrd=', abs(Nsd / Nrd))
64
65     else:
66         print('Barra ' + barra + ': ** NÃO PASSOU NA VERIFICAÇÃO**', \
67             ' - Nsd/Nrd=', abs(Nsd / Nrd))

```

Da linha 2 à linha 7 do código acima, ocorre a importação dos módulos e scripts necessários para a criação da rotina. A linha 2 importa o script responsável por iniciar o programa SAP2000 e as linhas 5, 6 e 7 importam os módulos do pacote Metalpy.

Nas linhas 10 e 11, o software SAP2000 é inicializado, e na linha 15 é aberto o modelo estrutural da treliça descrita anteriormente. Na linha 18, a rotina verifica se o modelo já está com a análise estrutural processada, caso não esteja, ele é processado pelo comando da linha 21.

Processada a análise estrutural do modelo, a rotina lê cada uma das barras existentes na estrutura, obtendo as propriedades do perfil (linha 33) e as propriedades do aço que constitui o perfil (linha 36, 38 e 39). A linha 42 indica para o programa SAP2000 de qual caso de cargas serão obtidos os esforços solicitante das barras. Na linha 43, são obtidas informações das barras, como comprimento e esforços solicitantes para o caso indicado anteriormente. O comprimento é repassado para a variável `L` e o esforço axial solicitante da barra é repassado para a variável `Nsd` na linha 46.

Obtida todas as informações necessárias do SAP2000, é realizada a verificação de segurança das barras de acordo com a NBR8800:2008, através do pacote Metalpy. Na linha 52 é criado o aço a partir da classe `ACO` com os parâmetros obtidos do SAP2000. O objeto que representa o perfil da barra é criado na linha 55 utilizando a classe `TuboRet`. Pode se observar que neste exemplo não foi indicado a norma como um parâmetro de inicialização do objeto que representa o perfil.

Após a criação do perfil, a rotina verifica se a barra está sendo solicitada por tração ou compressão e realiza a verificação de acordo com a natureza da solicitação. As barras tracionadas são identificadas pela rotina por apresentarem a magnitude do esforço solicitante positiva e as barras comprimidas são reconhecidas por apresentarem a magnitude do esforço solicitante negativa. A obtenção da capacidade resistente a tração é obtida na linha 58 pelo método `Ntrd_brt()`, chamado diretamente pela classe `NBR8800`, onde o perfil é passado como um dado de entrada do método. O mesmo ocorre para a obtenção da capacidade resistente a compressão do perfil, que é obtida na linha 61 pelo método `Ncrd`.

A verificação de segurança é realizada comparando a capacidade resistente das barras com o esforço solicitante. Este processo é realizado pelo bloco de código da linha 64 à linha 69. Para cada barra é impresso na tela se barra atende ou não os critérios de segurança. A figura 5.10 apresenta o resultado obtido na verificação de segurança de cada uma das barras da estrutura apresentada neste exemplo.

Figura 5.10 – Resultados da verificação de segurança

	Status	Nsd/Nrd		Status	Nsd/Nrd
Barra 1	Ok!	92.31 %	Barra 17	Ok!	93.49 %
Barra 2	Ok!	69.28 %	Barra 18	Ok!	96.61 %
Barra 3	Ok!	46.2 %	Barra 19	Ok!	93.49 %
Barra 4	Ok!	23.11 %	Barra 20	Ok!	77.29 %
Barra 5	Ok!	11.14 %	Barra 21	Ok!	87.64 %
Barra 6	Ok!	23.11 %	Barra 22	Ok!	45.08 %
Barra 7	Ok!	46.2 %	Barra 23	Ok!	70.12 %
Barra 8	Ok!	69.28 %	Barra 24	Ok!	0.0 %
Barra 9	Ok!	92.31 %	Barra 25	Ok!	40.9 %
Barra 10	Ok!	0.0 %	Barra 30	Ok!	11.06 %
Barra 11	Ok!	40.9 %	Barra 31	Ok!	33.18 %
Barra 12	Ok!	45.08 %	Barra 32	Ok!	55.3 %
Barra 13	Ok!	70.12 %	Barra 33	Ok!	77.42 %
Barra 14	Ok!	77.29 %	Barra 26	Ok!	77.42 %
Barra 15	Ok!	87.64 %	Barra 27	Ok!	55.3 %
Barra 16	Ok!	96.61 %	Barra 28	Ok!	33.18 %
			Barra 29	Ok!	11.06 %

Fonte: Elaborada pelo autor

Existem alguns pontos a serem destacados deste exemplo. A rotina foi criada de forma genérica para ler e obter informações de um modelo qualquer do SAP2000, portanto a mesma rotina pode ser utilizada para diversos modelos de treliças. O único requisito para que ele funcione adequadamente é que o nome do perfil do modelo esteja igual ao nome do perfil da base de dados utilizado pelo pacote. Outro ponto, é que com mais algumas linhas de código poderia ser verificada a segurança para todas os tipos de solicitações e para inúmeros casos de cargas, tornando a rotina mais genérica.

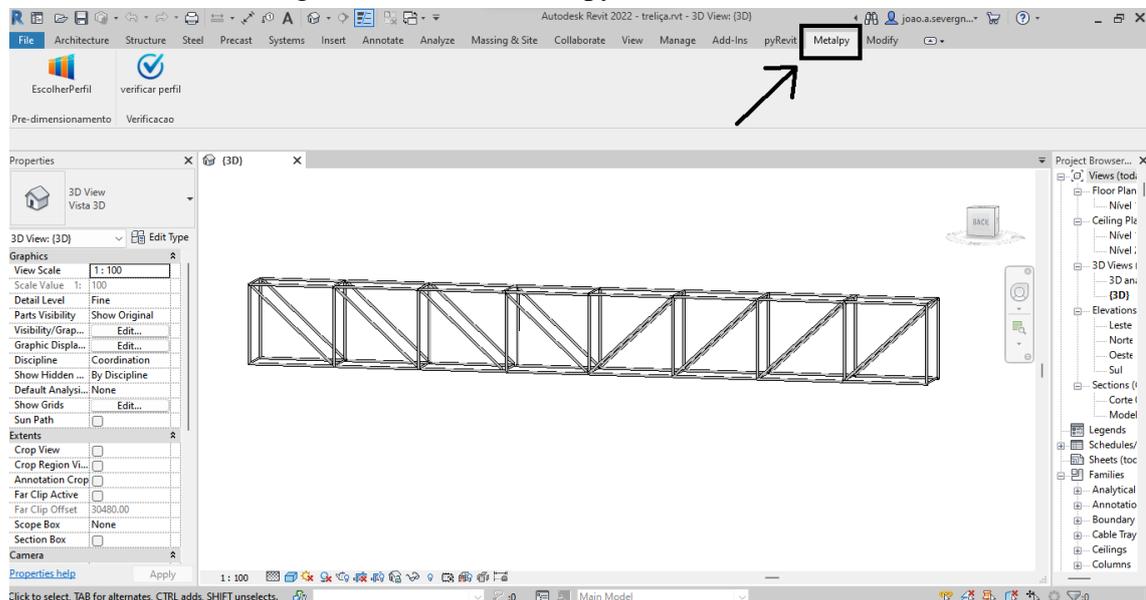
5.2.2 Plugins integrados ao Revit

Revit (AUTODESK, 2022) é um software de criação de modelos BIM (sigla do inglês building information modeling). Ele fornece recursos para modelagem de projetos de diversas disciplinas, como: arquitetura, estrutura, instalações elétricas e hidráulica. Os recursos para criação de modelos estruturais no Revit possibilita que os elementos de um modelo de estrutura armazenem todas as informações referente a ele. Assim, em um único arquivo são armazenadas as informações

dos materiais e geométricas da estrutura e seus componentes. Apesar de não contar com ferramentas para a análise estrutural, o Revit também apresenta recursos para criação de modelos de cálculo da estrutura. Com estes recursos podem ser adicionados os apoios, os carregamentos e armazenada as solicitações obtidas externamente por outros softwares.

Com a API do Revit é possível ter acesso as informações contidas em um modelo (arquivo) e utilizar as funcionalidades do programa de forma automatizada. Isso possibilita a criação de novas ferramentas, que podem ser acessadas pela interface interna do programa. Utilizando esses recursos, foi criada uma aba na interface do Revit com ferramentas que integram o Revit o pacote Metalpy, como mostra a figura 5.11.

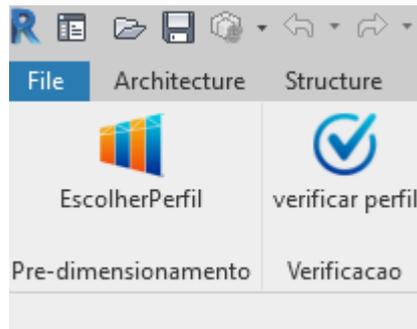
Figura 5.11 – Aba Metalpy na interface do Revit



Fonte: Elaborada pelo autor

A figura 5.12 apresenta os dois plugins desenvolvidos com o pacote Metalpy. Assim como a rotina apresentada na seção anterior, estes plugins são programas desenvolvidos separadamente que utilizam os recursos e os dados armazenados no Revit, juntamente com os recursos do pacote Metalpy, para realizar o função que se propõem. O plugin nomeado como "Escolher perfil" encontra o perfil mais leve que pode ser utilizado em uma barra. E o plugin nomeado como "Verificar perfil" realiza a verificação de segurança para o estado limite último de uma barra.

Figura 5.12 – Plugins adicionados ao Revit



Fonte: Elaborada pelo autor

Para utilização destes dois plugins a estrutura deve ter seus esforços determinados por um programa de análise estrutural, e estes esforços devem ser repassados para o Revit. Como não faz parte do foco deste trabalho, este procedimento não será abordado. As seções a seguir detalham o funcionamento do dois plugins apresentados.

5.2.2.1 Plugin de escolha de perfis

Este plugin tem a finalidade de auxiliar no pré-dimensionamento de treliças de perfis tubulares retangulares. Ele retorna o nome do perfil mais leve entre os perfis disponíveis no banco de dados do Revit que atenda os critérios de segurança para barras solicitadas por tração e compressão.

Para utiliza-lo, é preciso selecionar a barra a ser pré-dimensionada e acionar o plugin. Ele obtém as solicitações, as propriedades do material e a comprimento da barra do software Revit e procura o perfil mais leve que pode ser utilizado. O código que realiza este procedimento está apresentado a seguir.

```

1  from metalpy.perfis import Caixaao
2  from metalpy.material import Aco
3  from metalpy.normas import NBR8800
4
5  from Autodesk.Revit.DB import *
6  from Autodesk.Revit.DB.Structure import *
7
8  app = __revit__.Application
9  doc = __revit__.ActiveUIDocument.Document
10 uidoc = __revit__.ActiveUIDocument
11 selection = uidoc.Selection.GetElementIds()
12 acview = uidoc.ActiveView
13
14 if len(selection) > 1:
15     ArgumentError('Apenas uma barra deve ser selecionada')
16
17 perfis = FilteredElementCollector(doc) \
18     .OfCategory(BuiltInCategory.OST_StructuralFraming) \
19     .WhereElementIsElementType().ToElements()
20

```

```

21
22 # -----
23 #  OBTENDO AS SOLICITAÇÕES DA BARRA SELECIONADA
24 #  -----
25
26 barra = doc.GetElement(selection[0])
27 barra_analitica = barra.GetAnalyticalModel()
28
29 solic = barra_analitica.GetMemberForces()
30 sol_no1 = solic[0]
31 sol_no2 = solic[1]
32
33 Fx1 = sol_no1.Force[0] * 0.0003048 #kN
34 Fx2 = sol_no2.Force[0] * 0.0003048 #kN
35
36 Nsk = max(Fx1, Fx2) if Fx1 > 0 else min(Fx1, Fx2)
37
38 #-----
39 #  OBTENDO AS INFORMAÇÕES DO MATERIAL
40 #  -----
41
42 mat = doc.GetElement(barra.StructuralMaterialId)
43
44 mat_prop = doc.GetElement(mat.StructuralAssetId).GetStructuralAsset()
45
46 fy = mat_prop.MinimumYieldStress / 3047375.16 #kN/cm2
47 fu = mat_prop.MinimumTensileStrength / 3047375.16 #kN/cm2
48 v = mat_prop.PoissonRatio[0]
49 E = mat_prop.YoungModulus[0] / 3047375.16 #kN/cm2
50
51 #Comprimento da barra em cm
52 L = barra.get_Parameter(BuiltInParameter.INSTANCE_LENGTH_PARAM).\
53 AsDouble() * 30.48
54
55 mp_aco = Aco(E, v, fy, fu, 0.6)
56
57 for perfil in perfis:
58     #Largura do perfil
59     b = perfil.get_Parameter(BuiltInParameter.\
60         STRUCTURAL_SECTION_COMMON_WIDTH).AsDouble() * 30.48 #cm
61     #Altura do perfil
62     h = perfil.get_Parameter(BuiltInParameter.\
63         STRUCTURAL_SECTION_COMMON_HEIGHT).AsDouble() * 30.48 #cm
64     #Espessura do perfil
65     t = perfil.get_Parameter(BuiltInParameter.\
66         STRUCTURAL_SECTION_PIPESTANDARD_WALLNOMINALTHICKNESS).\
67     AsDouble() * 30.48 #cm
68
69     mp_perfil = Caixao(h, b, t, t, mp_aco)
70
71     if Nsk > 0:
72         Nrd, dados = NBR8800.Ntrd_brt(mp_perfil, data=True)
73
74     elif Nsk <= 0:
75         try:
76
77             Nrd, dados = NBR8800.Ncrd(mp_perfil, klx = L, kly=L, klz=L,\
78                 data=True)

```

```

79
80     except ValueError:
81         Nrd = 0
82
83     if Nrd >= abs( 1.4 * Nsk):
84         print('Perfil Encontrado')
85         break
86
87     perfil.get_Parameter(BuiltInParameter.SYMBOL_NAME_PARAM).AsString()
88     print(dados)
89     print('Nsd = ' + str(1.4 * Nsk))
90     print('Nsd/Nrd = ' + str( round(abs(1.4 * Nsk / Nrd) * 100, 2)) + '%')

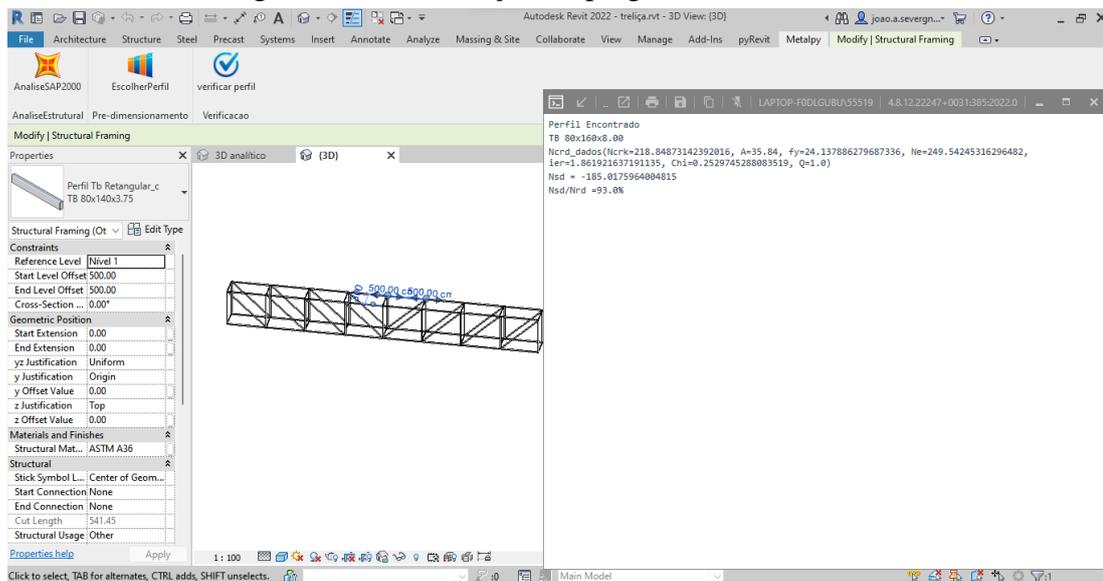
```

Nas primeiras linhas do código, são importados os módulos necessários utilizados na rotina. Da linha 8 a linha 12, são criados as variáveis que permitem a obtenção das informações do modelo Revit. A linha 14 identifica se a barra foi selecionada antes de se acionar o plugin, caso o contrário ela retorna um mensagem de erro para o usuário.

Após esta verificação são coletados todos os perfis tubulares disponíveis da biblioteca de perfis do Revit e armazenados na variável `perfil` (linha 18). O bloco de código da linha 25 à linha 35 obtém os esforços axiais solicitantes das duas extremidades da barra. É atribuído a variável `Nsk` o maior valor em módulo entre os dois valores obtidos. A seguir são coletadas as propriedades dos material e o comprimento da barra (linha 35 à linha 52). Todos os valores obtidos do Revit estão sendo ajustados por fatores de conversão que convertem os valores da unidade padrão interna do software para a unidade indicada no código.

Obtidas as informações da barra selecionada na interface do Revit, é realizado o procedimento para encontrar o perfil mais leve que pode ser utilizado. Na linha 54, é criado uma objeto da classe `ACO`, com as propriedades obtidas do Revit. A partir da linha 56, inicia-se um laço que percorre a lista de perfis criada na linha 17. Cabe ressaltar que os perfis estão ordenados do perfil de menor para o de maior área na lista de perfis, portanto, o primeiro perfil da lista que atende os critérios de segurança é o mais leve. São obtidas do Revit a largura, altura e a espessura do perfil. Como a biblioteca de perfis do Revit não apresenta o mesmo banco de dados do pacote Metalpy, neste exemplo de aplicação foi utilizado a classe `CaixaO` para representar os perfis tubulares (linha 68). A determinação da capacidade resistente a tração e compressão é realizada da mesma forma que na rotina de verificação integrada ao SAP2000. Esta etapa ocorre no bloco de código da linha 70 à linha 80. Na linha 82 é verificado se a capacidade resistente da barra é maior que o esforço solicitante, quando um perfil que atende os critérios de segurança é encontrado o laço é interrompido e o programa retorna o nome do perfil selecionado e os dados da verificação. As figuras 5.13 e 5.14 apresentam a tela final após a utilização do plugin "Escolher Perfil" e a tela de apresentação dos resultados, respectivamente.

Figura 5.13 – Utilização do plugin "Escolher Perfil"



Fonte: Elaborada pelo autor

Figura 5.14 – Apresentação dos resultados do plugin "Escolher Perfil"



Fonte: Elaborada pelo autor

5.2.2.2 Plugin de verificação de perfis

Este plugin realiza a verificação de segurança de uma barra de perfil tubular retangular considerando a geometria do tubo já atribuído a ela no modelo. Neste exemplo é feita a verificação para os esforços de tração, compressão, esforço cortante nas duas direções e momento fletor em relação ao eixo horizontal do perfil.

O plugin é utilizado da mesma forma que o plugin apresentado em 5.2.2.1. Primeiro, a barra a ser verificada deve ser selecionada na interface do programa, e em seguida, o plugin deve ser acionado. Similar aos exemplos apresentados anteriormente, esta rotina obtém os dados necessários do programa e determina a capacidade resistente da barra utilizando o pacote Metalpy. O código que realiza este procedimento está apresentado a seguir.

```

17 # -----
18 # OBTENDO AS SOLICITAÇÕES DO PERFIL
19 # -----
20
21 barra = doc.GetElement(selection[0])
22 barra_analitica = barra.GetAnalyticalModel()
23
24 solic = barra_analitica.GetMemberForces()
25 sol_no1 = solic[0]
26 sol_no2 = solic[1]
27
28 # Solicitações no nó 1 em kN e kN.m
29 Fx1 = sol_no1.Force[0] * 0.0003048
30 Fy1 = sol_no1.Force[1] * 0.0003048
31 Fz1 = sol_no1.Force[2] * 0.0003048
32 Mx1 = sol_no1.Moment[0] / 107.63910416709721
33 My1 = sol_no1.Moment[1] / 107.63910416709721
34 Mz1 = sol_no1.Moment[2] / 107.63910416709721
35
36 # Solicitações no nó 2 em kN e kN.m
37 Fx2 = sol_no2.Force[0] * 0.0003048
38 Fy2 = sol_no2.Force[1] * 0.0003048
39 Fz2 = sol_no2.Force[2] * 0.0003048
40 Mx2 = sol_no2.Moment[0] / 107.63910416709721
41 My2 = sol_no2.Moment[1] / 107.63910416709721
42 Mz2 = sol_no2.Moment[2] / 107.63910416709721
43
44 #Esforços solicitantes dimensionantes
45 Nsd = max(Fx1, Fx2) if Fx1 > 0 else min(Fx1, Fx2) #Solicitação axial
46 VsdY = max(abs(Fz1), abs(Fz2)) #Cortante em relação ao eixo Y
47 VsdX = max(abs(Fy1), abs(Fy2)) #Cortante em relação ao eixo X
48 Msdx = max(abs(My1), abs(My2)) #Momento fletor em relação ao eixo X
49
50 #-----
51 # OBTENDO AS INFORMAÇÕES DO MATERIAL
52 # -----
53
54 mat = doc.GetElement(barra.StructuralMaterialId)
55 mat_prop = doc.GetElement(mat.StructuralAssetId).GetStructuralAsset()
56
57 fy = mat_prop.MinimumYieldStress / 3047375.16 #kN/cm2
58 fu = mat_prop.MinimumTensileStrength / 3047375.16 #kN/cm2
59 v = mat_prop.PoissonRatio[0]
60 E = mat_prop.YoungModulus[0] / 3047375.16 #kN/cm2
61 #Comprimento da barra em cm
62 L = barra.get_Parameter(BuiltInParameter.INSTANCE_LENGTH_PARAM).\
63 AsDouble() * 30.48
64
65 #Largura do perfil em cm
66 b = perfil.get_Parameter(BuiltInParameter.\
67 STRUCTURAL_SECTION_COMMON_WIDTH).AsDouble() * 30.48
68 #Altura do perfil em cm
69 h = perfil.get_Parameter(BuiltInParameter.\
70 STRUCTURAL_SECTION_COMMON_HEIGHT).AsDouble() * 30.48
71 #Espessura do perfil em cm
72 t = perfil.get_Parameter(BuiltInParameter.\
73 STRUCTURAL_SECTION_PIPESTANDARD_WALLNOMINALTHICKNESS).\
74 AsDouble() * 30.48

```

```

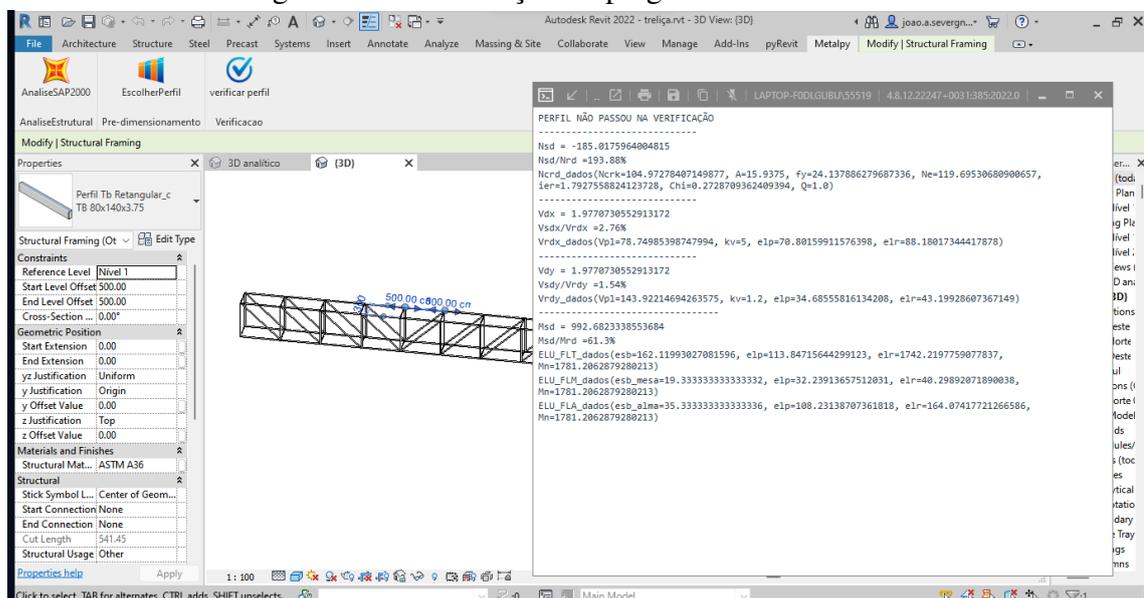
75 #Criação do objeto da classe Aco do pacote Metalpy
76 mp_aco = Aco(E, v, fy, fu, 0.6)
77 #Criação do objeto da classe Caixao do pacote Metalpy
78 mp_perfil = Caixao(h, b, t, t, mp_aco)
79
80 if Nsd > 0:
81     Nrd, dados_Nrd = NBR8800.Ntrd_brt(mp_perfil, data=True)
82 elif Nsd <= 0:
83     try:
84         Nrd, dados_Nrd = NBR8800.Ncrd(mp_perfil, klx, kly, klz, data=True)
85
86     except ValueError:
87         Nrd = 0
88
89 #Força cortante resistente em relação ao eixo Y
90 Vrdy, dados_Vrdy = NBR8800.Vrdy(mp_perfil, data=True)
91 #Força cortante resistente em relação ao eixo X
92 Vrdx, dados_Vrdx = NBR8800.Vrdx(mp_perfil, data=True)
93 #Momento fletor resistente em relação ao eixo X
94 Mrdx, dados_Mrdx = NBR8800.Mrdx(mp_perfil, kly, data=True)
95
96 # Relação entre esforço solicitante e capacidade resistente
97 Nsd_Nrd = abs(Nsd / Nrd)
98 Vsdx_Vrdx = Vsdx / Vrdx
99 Vsdy_Vrdy = Vsdy / Vrdy
100 Msdx_Mrdx = Msdx / Mrdx
101 #Verificação
102 if Nsd_Nrd < 1 and Vsdx_Vrdx < 1 and Vsdy_Vrdy < 1 and Msdx_Mrdx < 1:
103     print('PERFIL OK!')
104 else:
105     print('PERFIL NÃO PASSOU NA VERIFICAÇÃO')
106
107 #-----
108 #Impressão dos resultados
109 #-----
110
111 print('-----')
112 print('Nsd = ' + str(1.4 * Nsk))
113 print('Nsd/Nrd = ' + str( round(Nsd_Nrd * 100, 2)) + '%')
114 print(dados_Nrd)
115 print('')
116 print('-----')
117 print('Vdx = ' + str(1.4 * Vskx))
118 print('Vsdx/Vrdx = ' + str( round(Vsdx_Vrdx * 100, 2)) + '%')
119 print(dados_Vrdx)
120 print('')
121 print('-----')
122 print('Vdy = ' + str(1.4 * Vsky))
123 print('Vsdy/Vrdy = ' + str( round(Vsdy_Vrdy * 100, 2)) + '%')
124 print(dados_Vrdy)
125 print('')
126 print('-----')
127 print('Msdx = ' + str(1.4 * Mskx))
128 print('Msdx/Mrdx = ' + str( round(Msdx_Mrdx * 100, 2)) + '%')
129 print(dados_Mrdx[0])
130 print(dados_Mrdx[1])
131 print(dados_Mrdx[2])

```

O código deste exemplo segue basicamente os mesmos passos do código do exemplo 5.2.2.1. Com a diferença que neste exemplo é verificado somente o perfil atribuído a barra. O primeiro bloco deste exemplo foi omitido por ser idêntico ao bloco da linha 1 à linha 15 do exemplo anterior. Na primeira parte do script apresentado, são obtidos os esforços solicitantes das duas extremidades da barra (linha 21 à linha 42). Da linha 45 à linha 48 são selecionados os esforços que serão utilizados na verificação do perfil. As propriedades do material e o comprimento da barra são obtidos da mesma forma que no exemplo do item 5.2.2.1 (linha 54 à 63). E o bloco de código da linha 66 à linha 74 obtém as propriedades do perfil que será verificado.

Depois de extraídas todas as informações necessárias do programa Revit, o objetos correspondentes ao aço e ao perfil são criados a partir das classes do pacote Metalpy (linha 76 e 78). E a capacidade resistente dos perfis é determinada utilizando os métodos da classe NBR8800 (linha 80 à linha 94). Em seguida, são calculadas as relações entre os esforços solicitantes e resistente (linha 97 à linha 100) e verificado se o perfil está atendendo os critérios de segurança de acordo com estado limite último (linha 102 à 105). Após a verificação, são impressos os resultados na tela. As figuras 5.15 e 5.16 apresentam a tela final após a utilização do plugin "verificar perfil" e a tela de apresentação dos resultados, respectivamente.

Figura 5.15 – Utilização do plugin "Verificar Perfil"



Fonte: Elaborada pelo autor

Figura 5.16 – Apresentação dos resultados do plugin "Verificar Perfil"

```

LAPTOP-F0DLGUBU\55519 | 4.8.12.22247+0031:385-2022.0 | - □ ×
PERFIL NÃO PASSOU NA VERIFICAÇÃO
-----
Nsd = -185.0175964004815
Nsd/Nrd =193.88%
Ncrd_dados(Ncrk=104.97278407149877, A=15.9375, fy=24.137886279687336, Ne=119.69530680900657,
ier=1.7927558824123728, Chi=0.2728709362409394, Q=1.0)
-----
Vdx = 1.9770730552913172
Vsdx/Vrdx =2.76%
Vrdx_dados(Vp1=78.74985398747994, kv=5, elp=70.80159911576398, elr=88.18017344417878)
-----
Vdy = 1.9770730552913172
Vsdy/Vrdy =1.54%
Vrdy_dados(Vp1=143.92214694263575, kv=1.2, elp=34.68555816134208, elr=43.19928607367149)
-----
Msd = 992.6823338553684
Msd/Mrd =61.3%
ELU_FLT_dados(esb=162.11993027081596, elp=113.84715644299123, elr=1742.2197759077837,
Mn=1781.2062879280213)
ELU_FLM_dados(esb_mesa=19.333333333333332, elp=32.23913657512031, elr=40.29892071890038,
Mn=1781.2062879280213)
ELU_FLA_dados(esb_alma=35.333333333333336, elp=108.23138707361818, elr=164.07417721266586,
Mn=1781.2062879280213)

```

Fonte: Elaborada pelo autor

6 CONSIDERAÇÕES FINAIS

Este trabalho aprestou, implementou e avaliou uma proposta de arquitetura para um módulo computacional destinado a determinação da capacidade resistente de perfis laminados e soldados de aço. Foram abordados perfis do tipo I (laminados e soldados), tubo retangular laminado, tubo circular laminado e seção caixão retangular. Para determinação da resistência dos perfis lançou-se mão dos critérios de segurança das normas NBR8800:2008 (ABNT, 2008) e AISC360-16 (ANSI, 2016).

O programa concebido deu origem ao pacote Metalpy. Este pacote foi estruturado em dois módulos independentes. Um módulo de perfis destinado a fornecer ferramentas para armazenar e determinar todos os dados a respeito de um perfil de aço. E um módulo de normas destinado a fornecer funções que possibilite a determinação da capacidade resistente desses perfis. Para avaliar os resultados fornecidos pelo pacote Metalpy, foram utilizados exemplos resolvidos de livros e manuais de estruturas de aço. Os resultados obtidos com o pacote Metalpy se mostraram bastante confiáveis, apresentando pouca ou nenhuma diferença em relação aos resultados apresentados nos textos de referência. De qualquer forma, esta ferramenta deve ser utilizada com parcimônia, sendo responsabilidade do usuário avaliar a confiabilidade dos resultados fornecidos por ela.

Além disso, foram apresentados exemplos de aplicações para utilização do pacote Metalpy. Estes exemplos procuraram explorar a integração do pacote com outros softwares de engenharia. Foram utilizados os softwares comerciais SAP2000 e Revit. Metalpy mostrou-se uma ferramenta bastante eficiente, versátil, intuitiva e de fácil utilização, podendo ser utilizada para promover maior agilidade no processo de realização de projetos.

Como sugestão para futuros trabalhos, recomenda-se a ampliação do pacote. Esta ampliação pode ser realizada de diversas formas. Pode ser realizada expansão do módulo de perfis, criando perfis laminados e soldados ainda não abrangidos pela pacote. Outra alternativa é a expansão do módulo de normas, com a implementação dos métodos de determinação da capacidade resistente por outras normativas não abordadas.

REFERÊNCIAS

- ALMEIDA, F. de. **Notas de Aula: Flambagem local em barras comprimidas segundo a nbr8800:2008 - perfis laminados e soldados.** Porto Alegre, RS, 2020. Citado na página 24.
- AMERICAN INSTITUTE OF STEEL CONSTRUCTION. **AISC 360: Companion to the aisc steel construction manual - volume 1: Design examples.** Chicago Illinois, 2015. Citado 5 vezes nas páginas 44, 45, 47, 50 e 51.
- _____. **AISC 360: Specification for structural steel buildings.** Chicago, Illinois, 2016. Citado 5 vezes nas páginas 14, 15, 24, 25 e 66.
- _____. **AISC: Shapes database v15.0.** Chicago Illinois, 2017. Citado 2 vezes nas páginas 21 e 38.
- ARAÚJO, e. a. A. **Projeto de Estruturas de Edificações com perfis Tubulares em Aço.** Belo Horizonte: [s.n.], 2016. Citado 3 vezes nas páginas 47, 48 e 49.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 8800: Projeto de estruturas de aço e de estruturas mistas de aço e concreto de edificações.** Rio de Janeiro, 2008. Citado 7 vezes nas páginas 14, 15, 17, 22, 24, 25 e 66.
- _____. **NBR 16239: Projeto de estruturas de aço e de estruturas mistas de aço e concreto de edificações com perfis tubulares.** Rio de Janeiro, 2013. Citado na página 23.
- _____. **NBR 5584: Perfil i estrutural de aço soldado por arco elétrico - requisitos gerais.** Rio de Janeiro, 2013. Citado na página 41.
- AUTODESK INC. **Revit: Suporte e aprendizado.** California, USA, 2022. Disponível em: <<https://knowledge.autodesk.com/pt-br/support/revit?sort=score>>. Citado 2 vezes nas páginas 52 e 56.
- AWS. **O que é uma API? — Guia de APIs.** 2022. [Online; acesso 17-Julho-2022]. Disponível em: <<https://aws.amazon.com/pt/what-is/api/>>. Citado na página 13.
- COMPUTERS E STRUCTURES, INC. **CSI Analysis Reference Manual: For sap2000, etabs, safe and csibridge.** Berkeley, California, USA, 2016. Citado na página 52.
- FAKURY, R. **Dimensionamento básico de elementos estruturais de aço e mistos de aço e concreto.** 1. ed. São Paulo: Pearson Education do Brasil, 2016. Citado 4 vezes nas páginas 22, 27, 28 e 30.
- GUANABARA, M. K. **Dimensionamento de estruturas metálicas: Rotina computacional para seleção de perfis metálicos.** Monografia (TCC) — Escola de Engenharia UFRGS, Porto Alegre, RS, 2010. Citado na página 14.
- HENRICH, M. **Projeto Estrutural de Concepção Alternativa para Estrutura Metálica de Passarela Existente.** Monografia (TCC) — Escola de Engenharia UFRGS, Porto Alegre, RS, 2019. Citado na página 52.

MARTHA, L. F. **Análise matricial de estruturas com orientação a objetos**. 1. ed. Rio de Janeiro: Elsevier Editora LTDA, 2019. Citado 2 vezes nas páginas 32 e 33.

PFEIL, W. **Estruturas de aço: dimensionamento prático**. 9. ed. Rio de Janeiro: LTC, 2022. Citado 6 vezes nas páginas 23, 24, 28, 29, 41 e 43.

SEVERGNINI, J. **Metalpy**. 2022. [Online; acesso 18-Setembro-2022]. Disponível em: <<https://github.com/JoaoSevergnini/metalpy.git>>. Citado 2 vezes nas páginas 13 e 41.

SIMSEN, A. R. **Programa computacional para verificação de Perfil I laminado ou Soldado conforme NBR 8800:2008**. Monografia (TCC) — Universidade do vale do taquari, Lajeado, RS, 2014. Citado na página 14.

TEIXEIRA, F. de A. P. **Um programa de verificação para perfis de aço sob esforços usuais de projeto**. Monografia (TCC) — Instituto tecnológico de Aeronáutica, São José dos Campos, 2015. Citado na página 14.

VALLOUREC TUBOS DO BRASIL. **Tubos Esruturais**: Seção circular, quadrada e retangular. Belo Horizonte, MG, s.d. Citado 2 vezes nas páginas 21 e 38.

APÊNDICES

APÊNDICE A – ATRIBUTOS E MÉTODOS DAS CLASSES DO MÓDULO DE PERFIS

Neste anexo estão disponibilizados os diagramas das classes do módulo de perfis. Estes diagramas são divididos em três compartimentos, onde o primeiro deles contém o nome da classe, o segundo compartimento contém os atributos da classe e o terceiro contém os métodos da classe. Os atributos e os métodos listados são antecidos por um sinal de + ou por um sinal -. O sinal de + indica que o atributo ou método pode ser acessado por usuários para criação de rotinas, por exemplo. O sinal de - indica que o atributo ou método só deve ser utilizados dentro da própria classe.

A.1 Classe *PerfilEstrutural*

PerfilEstrutural
<p>+mat: objeto da classe Aco <i>Material</i></p> <p>+A: float <i>Área da seção transversal</i></p> <p>+Ix: float <i>Momento de inércia em relação ao eixo X</i></p> <p>+Iy: float <i>Momento de inércia em relação ao eixo Y</i></p> <p>+J: float <i>Constante de torção pura</i></p> <p>+EA: float <i>Rigidez axial</i></p> <p>+EIx: float <i>Rigidez a flexão em torno do eixo X</i></p> <p>+EIy: float <i>Rigidez a flexão em torno do eixo Y</i></p> <p>+GJ: float <i>Rigidez à torção</i></p> <p>+Wxi: float <i>Módulo de resistência elástico inferior em relação ao eixo X</i></p>

PerfilEstrutural (continuação)
<p>+Wys: float <i>Módulo de resistência elástico superior em relação ao eixo Y</i></p> <p>+Wxs: float <i>Módulo de resistência elástico superior em relação ao eixo X</i></p> <p>+Wxi: float <i>Módulo de resistência elástico inferior em relação ao eixo X</i></p> <p>+Wys: float <i>Módulo de resistência elástico superior em relação ao eixo Y</i></p> <p>+Wys: float <i>Módulo de resistência elástico inferior em relação ao eixo Y</i></p> <p>+Zx: float <i>Módulo de resistência plástico em X</i></p> <p>+Zy: float <i>Módulo de resistência plástico em Y</i></p> <p>+xo: float <i>coordenada x do centro de corte em relação ao centro geométrico</i></p> <p>+yo: float <i>coordenada y do centro de corte em relação ao centro geométrico</i></p> <p>+Cw: float <i>Constante de empenamento da seção transversal</i></p> <p>+raiz_E_fy: float <i>Raiz da razão entre o módulo de elasticidade e tensão de escoamento</i></p> <p>+raiz_fy_E: float <i>Raiz da razão entre a tensão de escoamento e o módulo de elasticidade</i></p>
<p>Métodos de propriedades</p> <hr/> <p>+rx: return float <i>Raio de giração em relação ao eixo X</i></p> <p>+ry: return float <i>Raio de giração em relação ao eixo Y</i></p> <p>+ro: return float <i>Raio de giração polar de inércia</i></p> <p>+Afy: return float <i>Resistência axial do escoamento</i></p> <p>+Afu: return float <i>Resistência axial a ruptura</i></p> <p>+Vplx: return float <i>Força cortante resistente na direção X</i></p>

PerfilEstrutural (continuação)
<p>+Vply: return float <i>Força cortante resistente na direção Y</i></p> <p>+Mplx: return float <i>Momento de plastificação no eixo X</i></p> <p>+Mply: return float <i>Momento de plastificação no eixo Y</i></p> <p>+Mrx: return float <i>Momento de início do escoamento em relação eixo X</i></p> <p>+Mry: return float <i>Momento de início do escoamento em relação eixo Y</i></p> <p>+simetria_x: return bool <i>Simetria em relação ao eixo X</i></p> <p>+simetria_y: return bool <i>Simetria em relação ao eixo y</i></p> <p>+bissimetrico: return list de bool <i>Indica se o perfil apresenta dupla simetria</i></p> <p>Métodos</p> <p>_____</p> <p>+indice_esbeltez(Lx, Ly): return list de float <i>Calcula o índice de esbeltez de uma barra de comprimentos Lx e Ly</i></p> <p>+par_estabilidade(klx, kly, klz): return par_estabilidade (namedtuple) <i>Cálculo as cargas críticas de flambagem e o momento crítico de flambagem lateral com torção</i></p> <p>+definir_norma(norma): return void <i>Define a norma dos métodos de verificação</i></p>

A.2 Classe *Perfil*

Perfil
<p>+d: float <i>Altura total do perfil</i></p> <p>+dl: float <i>Distância interna entre as mesas</i></p> <p>+h: float <i>Largura plana da alma</i></p> <p>+bfs: float <i>Largura da mesa superior</i></p> <p>+bfi: float <i>Largura da mesa inferior</i></p> <p>+tw: float <i>Espessura da alma</i></p> <p>+tfs: float <i>Espessura da mesa superior</i></p> <p>+tfi: float <i>Espessura da mesa inferior</i></p> <p>+esb_alma: float <i>Esbeltez da alma</i></p> <p>+esb_mesa_s: float <i>Esbeltez da mesa superior</i></p> <p>+esb_mesa_i: float <i>Esbeltez da mesa inferior</i></p> <p>+Iys: float <i>Momento de inércia da mesa superior em relação ao eixo y</i></p> <p>+Iyi: float <i>Momento de inércia da mesa inferior em relação ao eixo y</i></p> <p>+hcg: float <i>Altura do centro geométrico</i></p> <p>+hpl: float <i>Altura da linha neutra na plastificação da seção</i></p> <p>+tipo: str, default: 'I SOLDADO' <i>tipo de perfil</i></p>
<p>- _prop_geo(): return dict <i>Calcula as propriedades geométricas do perfil</i></p>

A.3 Classe *PerfilLam*

PerfilLam
<p>+d: float <i>Altura total do perfil</i></p> <p>+dl: float <i>Distância interna as mesas dos perfis</i></p> <p>+h: float <i>Largura plana da alma</i></p> <p>+bf: float <i>Largura da mesa</i></p> <p>+tw: float <i>Espessura da alma</i></p> <p>+tf: float <i>Espessura da mesa</i></p> <p>+r: float <i>Raio de concordância</i></p> <p>+esb_mesa: float <i>Esbeltez da mesa</i></p> <p>+esb_alma: float <i>Esbeltez da alma</i></p> <p>+simetria: list de bool, default: [True, True] <i>Indica se existe simetria em relação aos eixos X e Y.</i></p> <p>+tipo: str, default: 'I LAMINADO' <i>tipo de perfil</i></p>
<p>- _prop_geo(): return dict <i>Calcula as propriedades geométricas do perfil</i></p>

A.4 Classe *TuboCir*

TuboCir
<p>+D: float <i>Diâmetro externo do perfil</i></p> <p>+t: float <i>Espessura do perfil</i></p> <p>+Dint: float <i>Diâmetro interno do perfil</i></p> <p>+esb: float <i>Eslbeltez do perfil</i></p> <p>+Wt: float <i>Módulo de torção</i></p> <p>+simetria: list de bool, default: [True, True] <i>Indica se existe simetria em relação aos eixos X e Y.</i></p> <p>+tipo: str, default: 'TUBO CIR' <i>tipo de perfil</i></p>
<p>-_prop_geo(): return dict <i>Cálcula as propriedades geométricas do perfil</i></p>

A.5 Classe *TuboRet*

TuboRet
<p>+h: float <i>Altura total do perfil</i></p> <p>+b: float <i>Largura total do perfil</i></p> <p>+tw: float <i>Espessura das almas</i></p> <p>+tf: float <i>Espessura das mesas</i></p> <p>+hint: float <i>Altura plana da alma do perfil</i></p> <p>+bint: float <i>Largura plana das mesas do perfil</i></p> <p>+esb_alma: float <i>Esbeltez das almas</i></p> <p>+esb_mesa: float <i>Esbeltez das mesas</i></p> <p>+Wt: float <i>Módulo de torção</i></p> <p>+simetria: list de bool, default: [True, True] <i>Indica se existe simetria em relação aos eixos X e Y.</i></p> <p>+tipo: str, default: 'TUBO RET' <i>tipo de perfil</i></p>
<p>-_prop_geo(): return dict <i>Busca no banco de dados as propriedades geométricas do perfil</i></p>

A.6 Classe *Caixao*

Caixao
<p>+h: float <i>Altura total do perfil</i></p> <p>+b: float <i>Largura total do perfil</i></p> <p>+tw: float <i>Espessura das almas</i></p> <p>+tf: float <i>Espessura das mesas</i></p> <p>+hint: float <i>Distância interna entre as mesas do perfil</i></p> <p>+bint: float <i>Distância interna entre as almas do perfil</i></p> <p>+esb_alma: float <i>Esbeltez das almas</i></p> <p>+esb_mesa: float <i>Esbeltez das mesas</i></p> <p>+tipo: str, default: 'CAIXAO' <i>tipo de perfil</i></p>
<p>-_prop_geo(): return dict <i>Calcula as propriedades geométricas do perfil</i></p>

APÊNDICE B – MÉTODOS DAS CLASSES DO MÓDULO DE NORMAS

B.1 Classe *NBR8800*

NBR8800
<p>Métodos principais</p> <hr style="width: 20%; margin-left: 0;"/> <p>+Ntrd_brt(perfil, gama_a1=1.1, data=False):return float <i>Calcula a força axial de tração resistente para o ELU de escoamento da seção bruta</i></p> <p>+Ncrd(perfil, klx, kly, klz, gama_a1=1.1, data=False):return float <i>Calcula a força axial de compressão resistente</i></p> <p>+Vrdx(perfil, Lv=None, gama_a1=1.1, data=False):return float <i>Calcula a força cortante resistente para flexão em torno no eixo Y</i></p> <p>+Vrdy(perfil, a=None, Lv=None, gama_a1=1.1, data=False):return float <i>Calcula a força cortante resistente para flexão em torno no eixo X</i></p> <p>+Mrdx(perfil, Lb=None, Cb=1.0, gama_a1=1.1, data=False):return float <i>Calcula o momento fletor resistente de acordo com o eixo X</i></p> <p>+Mrdy(perfil, Lb=None, Cb=1.0, gama_a1=1.1, data=False):return float <i>Calcula o momento fletor resistente de acordo com o eixo Y</i></p> <p>+Trd(perfil, L, gama_a1=1.1, data=False):return float <i>Calcula o momento torsor resistente de calculo</i></p> <p>+verif_NM(perfil, Nsd, Msdx, Msdy, gama_a1=1.1, Lbx=None, Lby=None, Cb=1, klx=None, kly=None, klz=None, data=False) return float <i>Realiza a verificação para esforços combinados</i></p> <p>Métodos auxiliares</p> <hr style="width: 20%; margin-left: 0;"/> <p>-_kc(esb): return float <i>Calcula o fator k_c conforme item (c) da seção F.2 do anexo F da NBR8800</i></p> <p>-_Qs_g3(perfil):return float <i>Calcula o Q_s para mesas de perfis do grupo 3 (cantoneiras)</i></p> <p>-_Qs_g4(perfil):return float <i>Calcula o Q_s para mesas de perfis do grupo 4 (I, H, U e T laminados)</i></p> <p>-_Qs_g5(perfil):return float <i>Calcula o Q_s para mesas de perfis do grupo 5 (I, U e T soldados)</i></p>

NBR8800 (continuação)

- _Qs_g6(perfil):return float

Calcula o Q_s para alma de perfis do grupo 6 (perfil T)

- _bef(b, t, elr, E, fy, ca, chi):return float

Calcula a largura efetiva de elementos AA

- _Qa(perfil, chi): return float

Calcula o fator Q_a de elementos AA

- _Qs_g5(perfil):return float

Calcula o Q_s para mesas de perfis do grupo 5 (I, U e T soldados)

- _Qs_g6(perfil):return float

Calcula o Q_s para alma de perfis do grupo 6 (perfil T)

- _bef(b, t, elr, E, fy, ca, chi):return float

Calcula a largura efetiva de elementos AA

- _Qa(perfil, chi): return float

Calcula o fator Q_a de elementos AA

- _Q(perfil, chi):return float

Calcula o fator de redução da resistência a compressão devido a flambagem local (Q)

- _Vrd_tubo(perfil, Lv, gama_a1, data): return float

Calcula a força cortante resistente de cálculo para tubos circulares

- _Mnx_FLT(perfil, Lb, Cb, data=False): return float

Calcula o momento fletor característico para o ELU de FLT para flexão em relação ao eixo x

- _Mnx_FLM(perfil, data=False): return float

Calcula o momento fletor característico para o ELU de FLM para flexão em relação ao eixo x

- _Mnx_FLA(perfil, data=False): return float

Calcula o momento fletor característico para o ELU de FLA para flexão em relação ao eixo x

- _Mnx_VAE(mat, Wxt, Wxc, bfc, tfc, hcg, d, dl, tw, Cb, Lb, data=False)=return float

Calculo do momento resistente em relação a X para perfis de alma esbelta conforme Anexo H da NBR8800:2008

- _Mny_FLT(perfil, Lb, Cb, data=False): return float

Calcula o momento fletor característico para o ELU de FLT para flexão em relação ao eixo y

- _Mny_FLM(perfil, data=False): return float

Calcula o momento fletor característico para o ELU de FLM para flexão em relação ao eixo y

- _Mny_FLA(perfil, data=False): return float

Calcula o momento fletor característico para o ELU de FLA para flexão em relação ao eixo y

- _Wefx(perfil, bef): return float

Calcula o módulo elástico efetivo para perfis tubo retangulares e caixão em relação ao eixo x

- _Wefy(perfil, bef): return float

Calcula o módulo elástico efetivo para perfis tubo retangulares e caixão em relação ao eixo y

B.2 Classe AISC360

AISC360
<p>Métodos principais</p> <hr/> <p>+Ntrd_brt(perfil, phi_s=0.9, data=False):return float <i>Calcula a força axial de tração resistente para o ELU de escoamento da seção bruta</i></p> <p>+Ncrd(perfil, klx, kly, klz, phi_c=0.9, data=False):return float <i>Calcula a força axial de compressão resistente</i></p> <p>+Vrdx(perfil, Lv=None, phi_v=0.90, data=False):return float <i>Calcula a força cortante resistente para flexão em torno no eixo Y</i></p> <p>+Vrdy(perfil, a=None, Lv=None, phi_v=0.90, data=False):return float <i>Calcula a força cortante resistente para flexão em torno no eixo X</i></p> <p>+Mrdx(perfil, Lb=None, Cb=1.0, phi_b=0.90, data=False):return float <i>Calcula o momento fletor resistente de acordo com o eixo X</i></p> <p>+Mrdy(perfil, Lb=None, Cb=1.0, phi_b=0.90, data=False):return float <i>Calcula o momento fletor resistente de acordo com o eixo Y</i></p> <p>+Trd(perfil, L, phi_t=0.9, data=False):return float <i>Calcula o momento torsor resistente de calculo</i></p> <p>+verif_NM(perfil, Nsd, Msdx, Msdy, phi=0.9, Lbx=None, Lby=None, Cb=1, klx=None, kly=None, klz=None, data=False): return float <i>Realiza a verificação para esforços combinados</i></p> <p>Métodos auxiliares</p> <hr/> <p>-_bef(b, c1, elr, esb, Fy, Fcr): return float <i>Calcula a largura efetiva de um elemento em compressão</i></p> <p>-_kc(esb): return float <i>Calcula o fator k_c conforme a tabela B4.1a da AISC360</i></p> <p>-_Aef(perfil, Fcr): return float <i>Calcula a área efetiva de perfis comprimidos</i></p> <p>-_Vrd_tubo(perfil, Lv, phi_v, data): return float <i>Calcula a força cortante resistente de cálculo para tubos circulares</i></p> <p>-_Vn_perfil_IU(perfil, Cv2, a): return float <i>Calcula a força cortante característica para perfis I e U na direção y</i></p> <p>-_Mnx_LTB(perfil, Lb, Cb, data=False): return float <i>Calcula o momento fletor resistente característico para o ELU de FLT para flexão em relação ao eixo x</i></p>

AISC360(continuação)

- $M_{nx_LTB_IU_CW}$ (perfil, Lb, Cb, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLT para flexão em relação ao eixo x, para perfis I e U de alma compacta

- $M_{nx_LTB_I_NCW}$ (perfil, Lb, Cb, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLT para flexão em relação ao eixo x, para perfis I e U de alma não compacta

- $M_{nx_LTB_I_SW}$ (perfil, Lb, Cb, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLT para flexão em relação ao eixo x, para perfis I e U de alma esbelta

- M_{nx_FLB} (perfil, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLM para flexão em relação ao eixo x

- M_{nx_WLB} (perfil, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLA para flexão em relação ao eixo x

- M_n _Tubo

Calcula o momento fletor resistente característico para perfis tubo circulares

- M_n _Tubo(perfil, data=False): return float

Calcula o momento fletor resistente característico para perfis tubo circulares

- M_{ny_LTB} (perfil, Lb, Cb, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLT para flexão em relação ao eixo y

- M_{ny_FLB} (perfil, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLM para flexão em relação ao eixo y

- M_{ny_WLB} (perfil, data=False): return float

Calcula o momento fletor resistente característico para o ELU de FLA para flexão em relação ao eixo y

- R_{pc} (perfil, dados=False): return float

Calcula o fator de plastificação da alma correspondente a compressão da mesa

- R_{pt} (perfil, dados=False): return float

Calcula o fator de plastificação da alma correspondente a tração da mesa

- F_l (F_y , S_{xc} , S_{xi}): return float

- R_{pg} (aw, esb, E, f_y): return float

Calcula o fator de redução da resistencia a flexão

- S_{ex} (perfil, bef): return float

Calcula o módulo resistente elástico efetivo em relação ao eixo y, considerando possível flambagem local

- S_{ey} (perfil, bef): return float

Calcula o módulo resistente elástico efetivo em relação ao eixo y, considerando possível flambagem local