



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO DE ENGENHARIA CIVIL

JÚLIO ÍCARO QUIRINO TAVARES

**FERRAMENTA COMPUTACIONAL EM PYTHON PARA ANÁLISE E
DIMENSIONAMENTO À FLEXÃO DE ESCADAS PLISSADAS DE CONCRETO
ARMADO: INTEGRAÇÃO DAS BIBLIOTECAS ANASTRUCT E
CONCRETEPROPERTIES**

PAU DOS FERROS

2026

JÚLIO ÍCARO QUIRINO TAVARES

**FERRAMENTA COMPUTACIONAL EM PYTHON PARA ANÁLISE E
DIMENSIONAMENTO À FLEXÃO DE ESCADAS PLISSADAS DE CONCRETO
ARMADO: INTEGRAÇÃO DAS BIBLIOTECAS *ANASTRUCT* E
*CONCRETEPROPERTIES***

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

Orientador: Matheus Fernandes de Araújo Silva, Prof. Dr.

PAU DOS FERROS

2026

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

T231f Tavares, Júlio Ícaro Quirino.

Ferramenta computacional em python para análise e dimensionamento à flexão de escadas plissadas de concreto armado: integração das bibliotecas anastruct e concretoproperties / Júlio Ícaro Quirino Tavares. - 2026.
69 f. : il.

Orientador: Matheus Fernandes de Araújo Silva.
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Engenharia Civil, 2026.

1. Escada plissada. 2. Concreto armado. 3. Anastruct. 4. Concretoproperties. 5. Python. I. Silva, Matheus Fernandes de Araújo, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade com AACR2 e os dados fornecidos pelo(a) autor(a).
Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

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

JÚLIO ÍCARO QUIRINO TAVARES

**FERRAMENTA COMPUTACIONAL EM PYTHON PARA ANÁLISE E
DIMENSIONAMENTO À FLEXÃO DE ESCADAS PLISSADAS DE CONCRETO
ARMADO: INTEGRAÇÃO DAS BIBLIOTECAS *ANASTRUCT* E
*CONCRETEPROPERTIES***

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

Defendida em: 17 / 06 / 2026.

BANCA EXAMINADORA

Matheus Fernandes de Araújo Silva, Prof. Dr. (UFERSA)
Presidente

Gustavo Luz Xavier da Costa, Prof. Dr. (UFERSA)
Membro Examinador

Gilvan Bezerra dos Santos Junior, Prof. Dr. (UFERSA)
Membro Examinador

RESUMO

Este trabalho teve como objetivo geral desenvolver uma ferramenta computacional em Python para automatizar a análise estrutural e o dimensionamento à flexão de lances de escadas plissadas. A metodologia consistiu em três etapas principais: (i) a implementação de uma classe utilizando a biblioteca *Anastruct* para gerar a geometria, calcular cargas e realizar a análise matricial linear, obtendo os momentos fletores máximos característicos; (ii) o desenvolvimento de um código iterativo para dimensionar a armadura longitudinal no Estado Limite Último (ELU), e (iii) o uso da biblioteca *Concreteproperties* para verificar a seção dimensionada. Os resultados da análise estrutural, realizados para uma escada de referência de 120 cm de largura, 30 cm de passo e 17,5 cm de espelho, resultaram em um momento fletor de cálculo de 2.141 kN·cm. O dimensionamento iterativo indicou a necessidade de uma armadura dupla simétrica composta por 9 barras de 10 mm em ambas as faces da seção. A verificação com o *Concreteproperties* demonstrou que esta configuração resiste ao momento solicitante. Conclui-se que a ferramenta integrada desenvolvida é eficaz e de código aberto, representando uma alternativa acessível a softwares comerciais para o pré-dimensionamento e verificação de escadas plissadas.

Palavras-chave: escada plissada; concreto armado; *Anastruct*; *Concreteproperties*; Python.

ABSTRACT

This work aimed to develop a computational tool in Python to automate the structural analysis and flexural design of folded plate stair flights. The methodology consisted of three main steps: (i) the implementation of a class using the *Anastruct* library to generate the geometry, calculate loads, and perform linear matrix analysis, obtaining the characteristic maximum bending moments; (ii) the development of an iterative code to design the longitudinal reinforcement at the Ultimate Limit State (ULS); and (iii) the use of the *Concreteproperties* library to verify the designed section. The structural analysis results, carried out for a reference staircase with a width of 120 cm, a tread of 30 cm, and a riser of 17.5 cm, resulted in a design bending moment of 2.141 kN·cm. The iterative design indicated the need for a symmetrical double reinforcement consisting of 9 bars of 10 mm diameter on both faces of the section. Verification with *Concreteproperties* demonstrated that this configuration withstands the design moment. It is concluded that the integrated open-source tool developed is effective, representing an accessible alternative to commercial software for the preliminary design and verification of folded plate stairs.

Keywords: folded plate stair; reinforced concrete; *Anastruct*; *Concreteproperties*; Python.

LISTA DE FIGURAS

Figura 1 - Diagrama retangular simplificado para distribuição de tensões de compressão no concreto, para concretos do Grupo I de resistência ($f_{ck} \leq 50$ MPa).	24
Figura 2 - Distribuição de tensões de compressão segundo os diagramas parábola-retângulo e retangular simplificado, para concretos com $f_{ck} \leq 40$ MPa.	25
Figura 3 - Método construtor da classe ' <i>EscadaPlissada</i> ', mostrando a inicialização dos parâmetros e o cálculo das propriedades geométricas e dos carregamentos.	32
Figura 4 - Método de geração da geometria: laço que cria os nós inferior e superior de cada espelho, resultando na linha poligonal da escada plissada.	33
Figura 5 - Representação da modelagem da estrutura da escada	34
Figura 6 - Construção do modelo no Anastruct: adição de elementos, definição de apoios e aplicação de carga distribuída nos pisos e no patamar.	35
Figura 7 - Rotina de análise e extração dos momentos fletores máximos.....	37
Figura 8 - Método <i>mostrar_resultados</i> gerando o relatório textual com dados de entrada, coordenadas dos nós, tabela de momentos por elemento e identificação do elemento crítico.	39
Figura 9 - Métodos <i>plotar_geometria</i> e <i>plotar_diagramas</i>	40
Figura 10 - Função da interface interativa para entrada de dados	41
Figura 11 - Exemplo de aplicação	42
Figura 12 - Relatório textual final gerado pelo método <i>mostrar_resultados</i>	43
Figura 13 - Gráfico da geometria da escada plissada	44
Figura 14 - Esquema estático com cargas distribuídas.....	45
Figura 15 - Diagrama de momento fletor	45
Figura 16 - Diagrama de esforço normal.....	46
Figura 17 - Diagrama de deslocamentos	46
Figura 18 - Modelo da escada plissada reproduzido no Ftool.....	48
Figura 19 - Diagrama de momento fletor obtido no Ftool	49
Figura 20 - Diagrama de esforço normal obtido no Ftool	49
Figura 21 - Diagrama de deslocamentos obtido no Ftool.....	50
Figura 22 - Definição dos materiais concreto C30 e aço CA-50.....	53
Figura 23 - Criação da seção retangular bruta.....	54
Figura 24 - Adição das armaduras à seção retangular	55
Figura 25 - Cálculo dos momentos resistentes último para as quatro combinações de solicitação	57

Figura 26 - Tabelas de momentos resistentes para as combinações $N=0$ e $N=50$ kN.....	58
Figura 27 - Resumo dos momentos resistentes e representação da seção armada	59
Figura 28 - Diagrama de interação momento-força normal	60
Figura 29 - Fluxograma do procedimento iterativo para dimensionamento da armadura longitudinal à flexão	63
Figura 30 - Diagrama de interação momento-força normal para a seção do degrau adotado ..	67
Figura 31 - Seção de concreto armado com $9\varnothing 10$	67

LISTA DE TABELAS

Tabela 1 - Comparação quantitativa entre os resultados obtidos no Python e no Ftool.....	50
Tabela 2 - Opções de armadura geradas pelo dimensionamento.....	65
Tabela 3 - Comparação entre momento solicitante (MSd) e momento resistente (MRd).....	66

LISTA DE QUADROS

Quadro 1 - Parâmetros de entrada da função <i>dimensionar_armadura</i>	64
Quadro 2 - Parâmetros de saída da função <i>dimensionar_armadura</i>	64

LISTA DE SÍMBOLOS

A_c	-	Área bruta da seção transversal
A_s	-	Área de aço da armadura longitudinal
$A_{s,ef}$	-	Área de aço efetiva (armadura escolhida)
$A_{s,máx}$	-	Área de aço máxima (4% de A_c)
$A_{s,mín}$	-	Área de aço mínima (0,15% de A_c)
$A_{s,nec}$	-	Área de aço necessária calculada
b_w	-	Largura da seção transversal
d	-	Altura útil da seção
E	-	Módulo de elasticidade do concreto
E_s	-	Módulo de elasticidade do aço
f_{cd}	-	Resistência de cálculo do concreto à compressão
f_{ck}	-	Resistência característica do concreto à compressão
f_{yd}	-	Resistência de cálculo do aço ao escoamento
f_{yk}	-	Resistência característica do aço ao escoamento
h	-	Espessura total da laje (altura da seção)
I	-	Momento de inércia da seção bruta
K_c	-	Coeficiente tabelado para dimensionamento (concreto)
K_s	-	Coeficiente tabelado para dimensionamento (aço)
M_k	-	Momento fletor característico
M_{Rd}	-	Momento fletor resistente de cálculo
M_{Sd}	-	Momento fletor solicitante de cálculo
p	-	Passo do degrau

e	- Espelho do degrau
n	- Número de espelhos
x	- Profundidade da linha neutra
α_c	- Fator de redução do diagrama retangular (0,85)
β_x	- Relação x/d
γ_c	- Coeficiente de ponderação do concreto (1,4)
γ_f	- Coeficiente de ponderação das ações (1,4)
γ_s	- Coeficiente de ponderação do aço (1,15)
λ	- Relação entre a altura da zona comprimida e x (0,8)
ρ	- Taxa geométrica de armadura
ρ'	- Taxa de armadura de compressão
σ_{sd}	- Tensão de cálculo na armadura tracionada
ϕ	- Diâmetro nominal da barra de aço

SUMÁRIO

1	INTRODUÇÃO.....	15
2	PROBLEMATIZAÇÃO.....	17
2.1	Lacuna de projeto abordadas	17
2.2	Necessidade de ferramentas computacionais acessíveis	17
2.3	Justificativa do trabalho.....	18
3	OBJETIVOS.....	19
3.1	Objetivo geral.....	19
3.2	Objetivos específicos	19
4	FUNDAMENTAÇÃO TEÓRICA	20
4.1	Características das escadas plissadas.....	20
4.2	Considerações sobre carregamentos e segurança estrutural segundo a NBR 6118:2026.....	20
4.3	Conceitos gerais do dimensionamento à flexão simples	24
4.4	Modelagem computacional de escadas plissadas	27
4.5	A linguagem Python e seu uso em engenharia estrutural	27
4.6	A biblioteca <i>Anastruct</i> – análise matricial de pórticos planos	28
4.7	A biblioteca <i>Concreteproperties</i> – verificação de seções de concreto armado	28
5	DESCRIÇÃO DA ANÁLISE ESTRUTURAL EM PYTHON	30
5.1	Escada de referência adotada	30
5.2	Célula 1 – DEFINIÇÃO DA CLASSE	30
5.2.1	Estrutura da classe EscadaPlissada	31
5.2.2	Geração automática da geometria.....	32
5.2.3	Definição das cargas e propriedades da seção	33
5.2.4	Análise estrutural e extração de momentos	35
5.2.5	Pós-processamento e visualização	37
5.2.5.1	Relatório textual	37

5.2.5.2	Visualização gráfica	39
5.3	Célula 2 – INTERFACE DO USUÁRIO	41
5.4	Célula 3 - EXEMPLO.....	41
5.5	Comparação dos resultados com o Ftool	46
6	APLICAÇÃO PRELIMINAR DA BIBLIOTECA CONCRETEPROPERTIES EM SEÇÃO RETANGULAR DE CONCRETO ARMADO	52
6.1	Definição dos materiais	53
6.2	Geometria da seção e criação do modelo	53
6.3	Adição das armaduras.....	54
6.4	Criação da seção de concreto armado e cálculo da capacidade última	55
6.5	Resultados.....	57
6.5.1	Momentos resistentes para as combinações de carregamento	57
6.5.2	Representação da seção transversal e resumo dos momentos resistentes.....	58
6.5.3	Diagrama de interação momento-força normal	59
7	DIMENSIONAMENTO DA ARMADURA LONGITUDINAL DA ESCADA PLISSADA	61
7.1	Algoritmo de dimensionamento iterativo	61
7.2	Implementação em Python.....	64
7.3	Exemplo de aplicação para a escada analisada.....	65
7.4	Verificação da armadura dimensionada com a biblioteca <i>Concreteproperties</i> ..	65
8	CONSIDERAÇÕES FINAIS	69
	REFERÊNCIAS	70

1 INTRODUÇÃO

As escadas plissadas de concreto armado – também denominadas escadas em “dente de serra” ou *slabless staircases* – têm ganhado espaço em projetos arquitetônicos contemporâneos devido à sua estética moderna e à ausência da laje de fundo contínua, conferindo leveza visual e integração com ambientes de alto padrão (Malysz, 2007; Simões, 2008). Diferentemente das escadas convencionais, cuja estrutura se assemelha a uma laje plana inclinada, a escada plissada é composta exclusivamente por pisos (partes horizontais dos degraus) e espelhos (partes verticais), que trabalham em conjunto como uma chapa dobrada (*folded plate*) ou casca facetada. Esse arranjo estrutural mobiliza mecanismos de resistência próximos aos de pórticos múltiplos, mas também introduz altas concentrações de tensão nos cantos reentrantes – as quinas internas entre piso e espelho –, que se tornam pontos críticos para o surgimento de fissuras (Trautwein et al., 2010).

Apesar de seu apelo estético, a escada plissada apresenta desafios significativos para o projeto estrutural, especialmente no que se refere ao controle de deformações e vibrações em serviço. Sua geometria esbelta torna a verificação do Estado Limite de Serviço de Deformações Excessivas (ELS-DEF) uma etapa de grande responsabilidade. A literatura técnica tem demonstrado que a percepção humana de vibrações pode ser mais restritiva do que a própria flecha estática, e os limites normativos tradicionais ($L/250$ para aceitabilidade visual, $L/500$ para danos a elementos não estruturais) podem ser insuficientes para garantir o conforto dos usuários em escadas plissadas (Araújo, 2003; Way et al., 2025). Além disso, os efeitos diferidos da fluência do concreto – que, sob carga permanente, pode triplicar os deslocamentos imediatos – agravam a incerteza no dimensionamento (Ghali; Favre; Elbadry, 2002).

A dificuldade não se limita aos fenômenos físicos. Do ponto de vista normativo, tanto a ABNT NBR 6118:2026 quanto o Eurocode 2 (CEN, 2023) não apresentam capítulos ou anexos específicos para o cálculo de escadas plissadas, enquadrando-as nas regras gerais de lajes e vigas planas. Consequentemente, os projetistas recorrem a modelos simplificados de viga equivalente, que não capturam adequadamente a flexibilidade rotacional dos nós nem a perda de rigidez causada pela fissuração precoce nos cantos reentrantes (Trautwein et al., 2010; Oyakojo, 2025). A adaptação de relações de esbeltez (L/d) de vigas convencionais para escadas plissadas pode levar a resultados antieconômicos ou, no extremo oposto, a problemas de vibrações excessivas e deslocamentos visíveis.

Diante desse contexto, a comunidade científica tem apontado a necessidade de ferramentas computacionais acessíveis que permitam a análise não linear de escadas plissadas,

incorporando a fissuração e a fluência, e que sejam capazes de gerar critérios de pré-dimensionamento baseados em análises paramétricas (Malysz, 2007; Sabeeh; Jaafer, 2024). Nesse sentido, a linguagem Python, combinada com bibliotecas de código aberto como *Anastruct* (análise matricial de pórticos) e *Concreteproperties* (propriedades de seções de concreto armado), oferece um ambiente flexível e gratuito para o desenvolvimento de soluções customizadas, sem depender de softwares comerciais fechados.

O presente trabalho tem como objetivo principal desenvolver uma classe em Python – denominada EscadaPlissada – que automatiza a geração geométrica, o cálculo de cargas, a análise estrutural linear (Estádio I) e a extração de momentos fletores máximos para escadas plissadas. Embora a versão atual não contemple ainda a não linearidade física (fissuração) nem os efeitos diferidos (fluência), ela constitui a base computacional sobre a qual essas extensões serão construídas. A ferramenta é validada qualitativamente pela reprodução de geometrias típicas e pela coerência dos diagramas de esforços. Ao final, discutem-se as limitações atuais e propõe-se um roteiro para trabalhos futuros, incluindo o acoplamento com *Concreteproperties* e a validação com ensaios experimentais da literatura (Oezbek et al., 2021; Simões, 2008).

2 PROBLEMATIZAÇÃO

2.1 Lacuna de projeto abordadas

Embora escadas plissadas sejam amplamente utilizadas em projetos arquitetônicos contemporâneos, o projeto estrutural desse tipo de elemento ainda carece de ferramentas computacionais acessíveis e integradas. As normas brasileiras (ABNT NBR 6118:2026) e internacionais, como a Eurocode 2 (CEN, 2023), não trazem procedimentos específicos para o cálculo de escadas plissadas, de modo que os engenheiros frequentemente recorrem a analogias com lajes ou vigas, ou então a softwares comerciais de elementos finitos de uso geral. No entanto, esses programas, embora sejam bastante consolidados, exigem aquisição de licenças e conhecimento especializado para modelagem, o que limita seu uso em pequenos escritórios e em contextos acadêmicos com recursos reduzidos.

Além disso, não se tem conhecimento, até o momento, de uma rotina computacional de código aberto que integre, em um mesmo ambiente, a geração automática da geometria da escada plissada, a análise estrutural linear e o dimensionamento à flexão da armadura longitudinal. A ausência de uma ferramenta integrada e transparente dificulta tanto o aprendizado do comportamento estrutural dessas escadas quanto a realização de análises paramétricas rápidas para diferentes geometrias e carregamentos.

Diante desse contexto, este trabalho identifica como lacunas a serem preenchidas: a ausência de uma implementação computacional aberta e documentada para a análise estrutural de escadas plissadas utilizando o método dos elementos finitos (pórtico plano); a inexistência de um código que automatize o dimensionamento iterativo da armadura longitudinal no Estado Limite Último (ELU), integrado à análise de esforços; e a carência de uma ferramenta que permita, em um mesmo fluxo, verificar a seção dimensionada por meio de uma biblioteca de cálculo de seções de concreto armado. Essas lacunas motivam o desenvolvimento da presente ferramenta computacional em Python, que busca oferecer uma alternativa acessível, integrada e didática para o pré-dimensionamento e verificação de escadas plissadas.

2.2 Necessidade de ferramentas computacionais acessíveis

Softwares comerciais de elementos finitos, como o ANSYS, ABAQUS ou SAP2000, permitem análises refinadas, mas sua utilização envolve a aquisição de licenças e, muitas vezes, conhecimentos especializados para a modelagem de geometrias não convencionais, como as

escadas plissadas. Essas barreiras limitam o acesso a essas ferramentas por parte de pequenos escritórios de projeto, profissionais autônomos e estudantes.

Por outro lado, bibliotecas de código aberto como *Anastruct* e *Concreteproperties* têm amadurecido nos últimos anos, oferecendo funcionalidades de análise matricial de pórticos e cálculo de seções de concreto armado (incluindo momento de fissuração, inércia fissurada e diagrama momento-curvatura). A integração dessas bibliotecas em uma classe dedicada a escadas plissadas permitirá que engenheiros e pesquisadores realizem análises paramétricas de forma ágil e transparente, sem dependências comerciais.

2.3 Justificativa do trabalho

Diante do exposto, este trabalho se justifica por três frentes:

- Acadêmica: Preenche uma lacuna na literatura nacional ao propor uma implementação computacional aberta para o pré-dimensionamento de escadas plissadas, servindo como ponto de partida para futuras pesquisas que incorporem não linearidades e validação experimental.
- Prática: Oferece aos engenheiros calculistas uma ferramenta gratuita para a análise linear de escadas plissadas, com código transparente e passível de adaptação a diferentes geometrias e carregamentos.
- Didática: A documentação detalhada da classe ‘EscadaPlissada’ – incluindo a descrição de cada método e sua correspondência com os conceitos de resistência dos materiais – contribui para o ensino de análise estrutural computacional em cursos de graduação e pós-graduação.

3 OBJETIVOS

3.1 Objetivo geral

Desenvolver uma ferramenta computacional em Python para auxiliar a análise estrutural e o dimensionamento à flexão de lances de escadas plissadas de concreto armado, a partir da determinação dos momentos fletores solicitantes por análise global linear e da estimativa da armadura longitudinal necessária conforme os critérios do Estado Limite Último da ABNT NBR 6118:2026.

3.2 Objetivos específicos

- Desenvolver uma ferramenta parametrizada que permita a representação e análise preliminar de diferentes geometrias de escadas plissadas de concreto armado.
- Possibilitar a obtenção automatizada dos momentos fletores solicitantes, reduzindo o esforço de modelagem manual e favorecendo a padronização do processo de análise.
- Fornecer subsídios para o dimensionamento da armadura longitudinal à flexão, conforme os critérios do Estado Limite Último previstos na ABNT NBR 6118:2023.
- Permitir a verificação da segurança da seção dimensionada por meio da comparação entre o momento solicitante de cálculo e o momento resistente de cálculo.

4 FUNDAMENTAÇÃO TEÓRICA

4.1 Características das escadas plissadas

As escadas plissadas, caracterizam-se pela ausência de uma laje de fundo contínua, sendo constituídas exclusivamente pelos pisos e espelhos, que trabalham de forma integrada (Malysz, 2007). Essa configuração confere à estrutura um comportamento de chapa dobrada (folded plate), no qual os pisos e espelhos atuam como uma casca facetada, mobilizando mecanismos de resistência semelhantes aos de pórticos múltiplos (Simões, 2008). Diferentemente das escadas convencionais com laje de fundo plana, a escada plissada não possui uma superfície contínua sob os degraus, o que lhe confere leveza visual e apelo arquitetônico, mas também introduz desafios estruturais significativos.

Devido à descontinuidade geométrica, ocorrem elevadas concentrações de tensão nos cantos reentrantes – as quinas internas entre piso e espelho –, que se tornam pontos críticos para o surgimento de fissuras (Trautwein et al., 2010). Estudos experimentais indicam que a fissuração costuma iniciar nessas regiões, reduzindo a rigidez global da estrutura mais precocemente do que o previsto por modelos de viga equivalente (Sabeeh; Jaafer, 2024). Os modos de falha mais comuns observados em ensaios incluem a flexão excessiva dos pisos – com a formação de rótulas plásticas –, o cisalhamento nos espelhos e, em casos extremos, o esmagamento do concreto na zona comprimida dos cantos reentrantes.

Apesar dessas dificuldades, as escadas plissadas apresentam vantagens relevantes: (i) estética moderna e integração com projetos arquitetônicos contemporâneos; (ii) maior rigidez longitudinal quando comparadas a uma laje plana equivalente, devido ao aumento do braço de alavanca interno proporcionado pela geometria dos degraus; (iii) potencial para vãos maiores com espessuras reduzidas (Malysz, 2007). Entretanto, sua execução exige fôrmas especiais e detalhamento cuidadoso das armaduras nos cantos reentrantes, o que aumenta o custo e a complexidade construtiva.

4.2 Considerações sobre carregamentos e segurança estrutural segundo a NBR 6118:2026

O projeto de estruturas de concreto armado deve atender aos requisitos de segurança, desempenho em serviço e durabilidade estabelecidos pela ABNT NBR 6118:2026. Essa norma define os procedimentos gerais para o projeto de estruturas de concreto simples, armado e

protendido, devendo ser utilizada em conjunto com normas complementares relacionadas às ações, segurança estrutural e uso das edificações.

No contexto deste trabalho, a análise da escada plissada é realizada com foco no Estado Limite Último (ELU), associado à segurança contra ruína ou colapso estrutural. Para isso, devem ser consideradas as ações que produzem esforços internos relevantes no elemento estrutural, bem como os coeficientes de ponderação aplicáveis às ações e às resistências dos materiais. A verificação no ELU é necessária para garantir que a capacidade resistente da seção de concreto armado seja superior aos esforços solicitantes de cálculo.

As ações atuantes em uma escada plissada podem ser classificadas, de forma geral, em ações permanentes e ações variáveis. As ações permanentes correspondem àquelas que atuam com intensidade praticamente constante ao longo da vida útil da estrutura, como o peso próprio do concreto armado e o peso dos revestimentos. Já as ações variáveis são aquelas cuja intensidade pode se modificar ao longo do tempo, como a sobrecarga de uso decorrente da circulação de pessoas. A ABNT NBR 8681:2025 estabelece critérios gerais para a verificação da segurança das estruturas usuais da construção civil, incluindo a definição das ações e de suas combinações.

Neste trabalho, foram consideradas como ações permanentes o peso próprio da estrutura de concreto armado e o peso do revestimento, enquanto a ação variável corresponde à sobrecarga de uso da escada. O peso próprio é determinado a partir do peso específico do concreto armado e das dimensões geométricas da seção. A carga de revestimento e a sobrecarga de uso são adotadas como cargas superficiais e convertidas em cargas lineares conforme a largura da escada. Essa conversão é necessária porque o modelo estrutural adotado representa a escada plissada por elementos lineares de pórtico plano.

Para o dimensionamento em Estado Limite Último, as ações características devem ser transformadas em ações de cálculo por meio dos coeficientes de ponderação. De forma geral, a combinação última normal pode ser expressa por:

$$F_d = \sum_{i=1}^m \gamma_{g,i} F_{Gk,i} + \gamma_q F_{Qk,1} + \sum_{j=2}^n \gamma_{q,j} \psi_{0,j} F_{Qk,j} \quad \text{Eq. (1)}$$

em que:

F_d é o valor de cálculo da combinação de ações;

$F_{Gk,i}$ representa o valor característico da ação permanente i ;

$F_{Qk,1}$ representa o valor característico da ação variável principal;

$F_{Qk,j}$ representa o valor característico das demais ações variáveis concomitantes;

$\gamma_{g,i}$ é o coeficiente de ponderação das ações permanentes;

γ_q é o coeficiente de ponderação da ação variável principal;

$\psi_{0,j}$ é o fator de combinação das ações variáveis secundárias.

No caso analisado neste trabalho, considera-se apenas uma ação variável principal, correspondente à sobrecarga de uso da escada. Assim, a expressão geral pode ser simplificada para:

$$F_d = \gamma_g F_{Gk} + \gamma_q F_{Qk} \quad \text{Eq. (2)}$$

em que F_{Gk} representa o conjunto das ações permanentes características e F_{Qk} representa a ação variável característica. Para situações usuais de dimensionamento em concreto armado, considerando ações permanentes e variáveis desfavoráveis, são frequentemente adotados coeficientes de ponderação iguais a 1,4, conforme os critérios da ABNT NBR 6118:2023 e da ABNT NBR 8681:2025. Dessa forma, para a escada analisada, a combinação de cálculo pode ser representada por:

$$F_d = 1,4 \cdot (F_{Gk} + F_{Qk}) \quad \text{Eq. (3)}$$

Essa combinação é aplicada para a obtenção dos esforços solicitantes de cálculo. No caso da análise à flexão, o momento fletor característico obtido no modelo estrutural é majorado para determinar o momento fletor solicitante de cálculo:

$$M_{Sd} = \gamma_f \cdot M_k \quad \text{Eq. (4)}$$

em que M_{Sd} é o momento fletor solicitante de cálculo, M_k é o momento fletor característico obtido na análise estrutural e γ_f é o coeficiente de ponderação das ações. Neste trabalho, adotou-se $\gamma_f = 1,4$, considerando a combinação última normal com ações desfavoráveis.

A análise estrutural da escada plissada é realizada em regime elástico linear, considerando a seção no Estádio I para a obtenção dos esforços internos. Essa hipótese é adequada ao objetivo de pré-dimensionamento, pois permite determinar os momentos fletores solicitantes a partir da geometria da escada, das condições de apoio e dos carregamentos aplicados. Entretanto, é importante destacar que essa simplificação não representa a perda de rigidez associada à fissuração do concreto, nem os efeitos diferidos decorrentes da fluência e da retração. Portanto, os resultados obtidos devem ser interpretados como uma etapa preliminar de análise e dimensionamento.

Após a determinação dos esforços solicitantes de cálculo, procede-se ao dimensionamento da seção de concreto armado à flexão. Para isso, a ABNT NBR 6118:2023 adota os princípios gerais de segurança baseados nos estados limites, nos quais a resistência de cálculo da seção deve ser maior ou igual ao efeito das ações de cálculo. No caso específico da flexão simples, a verificação fundamental pode ser escrita como:

$$M_{Rd} \geq M_{Sd} \quad \text{Eq. (5)}$$

em que M_{Rd} é o momento fletor resistente de cálculo da seção de concreto armado; M_{Sd} é o momento fletor solicitante de cálculo obtido a partir da combinação de ações.

Essa desigualdade expressa a condição básica de segurança estrutural: a seção dimensionada deve possuir capacidade resistente suficiente para suportar o momento fletor atuante no Estado Limite Último. Caso $M_{Rd} < M_{Sd}$, a seção ou a armadura adotada não atende à segurança requerida, sendo necessário aumentar a área de aço, alterar a geometria da seção, modificar a classe dos materiais ou rever a concepção estrutural.

O cálculo do momento resistente da seção é realizado com base nas hipóteses clássicas do concreto armado, incluindo a manutenção das seções planas após a deformação, a aderência perfeita entre aço e concreto e a desconsideração da resistência à tração do concreto no ELU. Para concretos de classe até C50, a NBR 6118:2023 permite a utilização do diagrama retangular simplificado de tensões no concreto comprimido, adotando parâmetros normativos para representar a resultante de compressão. A resistência de cálculo do concreto é dada por:

$$f_{cd} = \frac{f_{ck}}{\gamma_c} \quad \text{Eq. (6)}$$

em que f_{ck} é a resistência característica à compressão do concreto e γ_c é o coeficiente de ponderação da resistência do concreto. Para o aço, a resistência de cálculo ao escoamento é dada por:

$$f_{yd} = \frac{f_{yk}}{\gamma_s} \quad \text{Eq. (7)}$$

em que f_{yk} é a resistência característica de escoamento do aço e γ_s é o coeficiente de ponderação da resistência do aço.

Neste trabalho, o dimensionamento da armadura longitudinal é realizado a partir do maior momento fletor solicitante obtido na análise estrutural da escada plissada. Em seguida, a armadura necessária é estimada conforme os critérios de dimensionamento à flexão no ELU, respeitando os limites normativos de armadura mínima, armadura máxima e ductilidade. Por fim, a seção dimensionada é verificada por meio da comparação entre M_{Rd} e M_{Sd} , garantindo que a armadura adotada seja suficiente para resistir ao momento solicitante de cálculo.

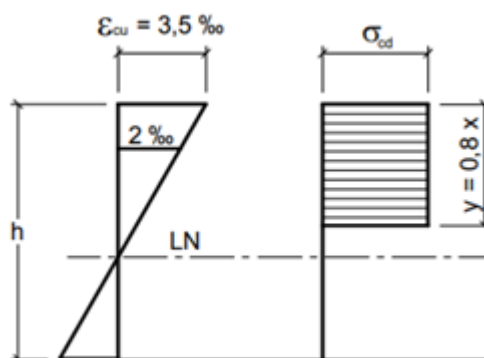
Dessa forma, a consideração das combinações de ações e da verificação resistente permite estabelecer uma ligação direta entre o modelo computacional desenvolvido e os critérios de segurança estrutural prescritos pela ABNT NBR 6118:2026. Essa etapa é essencial para que a ferramenta proposta não apenas determine esforços internos, mas também forneça subsídios técnicos para o dimensionamento seguro da armadura longitudinal de escadas plissadas de concreto armado.

4.3 Conceitos gerais do dimensionamento à flexão simples

O dimensionamento de elementos de concreto armado sujeitos à flexão simples no Estado Limite Último fundamenta-se nas hipóteses básicas adotadas pela ABNT NBR 6118:2026. Entre essas hipóteses, destacam-se a manutenção das seções planas após a deformação, a aderência perfeita entre o aço e o concreto e a desconsideração da resistência à tração do concreto no ELU. Dessa forma, admite-se que a força de tração seja resistida pela armadura longitudinal, enquanto a força de compressão seja resistida pelo concreto comprimido.

Para concretos de classe até C50, a ABNT NBR 6118:2026 permite a utilização do diagrama retangular simplificado de tensões no concreto comprimido, apresentado na Figura 1.

Figura 1 - Diagrama retangular simplificado para distribuição de tensões de compressão no concreto, para concretos do Grupo I de resistência ($f_{ck} \leq 50$ MPa).



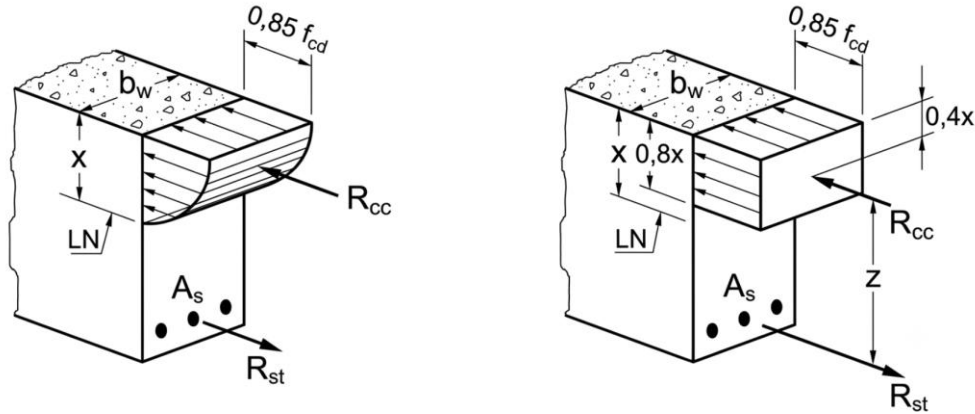
Fonte: Adaptado de Bastos (2023, p. 12).

No procedimento adotado neste trabalho, a resultante de compressão no concreto é representada pela expressão:

$$R_{cc} = 0,68 \cdot b_w \cdot x \cdot f_{cd} \quad \text{Eq. (8)}$$

em que R_{cc} é a resultante de compressão no concreto (demonstrado na Figura 2), b_w é a largura da seção, x é a profundidade da linha neutra e f_{cd} é a resistência de cálculo do concreto à compressão.

Figura 2 - Distribuição de tensões de compressão segundo os diagramas parábola-retângulo e retangular simplificado, para concretos com $f_{ck} \leq 40$ MPa.



Fonte: Bastos (2023, p.14).

O coeficiente 0,68 resulta do produto entre os parâmetros do bloco retangular equivalente adotados para concretos de classe até C50, considerando $\alpha_c = 0,85$ e $\lambda = 0,80$.

A posição da resultante de compressão é considerada a uma distância $0,4x$ da fibra mais comprimida da seção, uma vez que a altura do bloco retangular equivalente é igual a $0,8x$. Assim, o braço de alavanca interno entre a resultante de compressão do concreto e a armadura tracionada é dado por:

$$z = d - 0,4x \quad \text{Eq. (9)}$$

em que z é o braço de alavanca e d é a altura útil da seção.

A partir do equilíbrio de momentos, o momento resistente da seção pode ser expresso por:

$$M_{Rd} = 0,68 \cdot b_w \cdot x \cdot f_{cd} \cdot (d - 0,4x) \quad \text{Eq. (10)}$$

Para o dimensionamento, admite-se que a seção deve resistir ao momento solicitante de cálculo. Assim, igualando-se o momento resistente ao momento solicitante, obtém-se a equação utilizada para determinar a profundidade da linha neutra:

$$M_{Sd} = 0,68 \cdot b_w \cdot x \cdot f_{cd} \cdot (d - 0,4x) \quad \text{Eq. (11)}$$

Essa equação é resolvida em função de x , sendo adotada a raiz fisicamente válida, isto é, positiva e compatível com a altura útil da seção.

Após a determinação da linha neutra, a área de aço necessária é obtida pelo equilíbrio de forças entre a compressão no concreto e a tração na armadura. A força resistente de tração na armadura é dada por:

$$R_{st} = A_s \cdot f_{yd} \quad \text{Eq. (12)}$$

Como o equilíbrio interno exige:

$$R_{cc} = R_{st} \quad \text{Eq. (13)}$$

tem-se:

$$A_s = \frac{0,68 \cdot b_w \cdot x \cdot f_{cd}}{f_{yd}} \quad \text{Eq. (14)}$$

em que A_s é a área de aço necessária e f_{yd} é a resistência de cálculo do aço ao escoamento.

Além da obtenção da área de aço, devem ser verificados os limites normativos de armadura mínima, armadura máxima e ductilidade. A armadura mínima tem a função de garantir capacidade resistente após a fissuração do concreto, evitando ruptura frágil. A armadura máxima limita a quantidade de aço na seção, contribuindo para condições adequadas de concretagem, aderência e comportamento dúctil.

No procedimento computacional adotado, também se verifica a posição da linha neutra em relação à altura útil da seção. Para garantir comportamento dúctil, a profundidade da linha neutra deve permanecer dentro dos limites associados aos domínios de deformação adequados. Neste trabalho, adotou-se como limite de ductilidade:

$$x \leq 0,45 \cdot d \quad \text{Eq. (15)}$$

Quando essa condição é atendida, a seção permanece em domínio compatível com comportamento dúctil. Caso contrário, a seção é classificada como superarmada, indicando a necessidade de alteração da geometria, dos materiais ou da solução de armadura.

Por fim, a segurança da seção é verificada pela comparação entre o momento resistente de cálculo e o momento solicitante de cálculo, dado pela Eq. (5).

Portanto, o dimensionamento à flexão simples adotado neste trabalho é baseado na resolução direta das equações de equilíbrio da seção retangular no Estado Limite Último. Essa

formulação serve de base para o procedimento iterativo implementado em Python, no qual a área de aço calculada é posteriormente convertida em combinações viáveis de barras comerciais.

4.4 Modelagem computacional de escadas plissadas

A análise computacional de escadas plissadas pode ser realizada em diferentes níveis de complexidade. Modelos de pórtico plano são bastante utilizados em pré-dimensionamento devido à sua simplicidade e baixo custo computacional, mas não capturam fielmente a distribuição de esforços nos cantos reentrantes nem a flexibilidade rotacional dos nós (Trautwein et al., 2010). Modelos de elementos finitos com elementos de casca (*shell*) fornecem resultados mais precisos, especialmente para concentrações de tensão e rigidez global, mas exigem maior tempo de modelagem e conhecimento especializado (Oyakojo, 2025).

Neste trabalho, optou-se pela biblioteca *Anastruct* (Van Der Meer, 2022) para a análise global linear da escada plissada. Trata-se de uma ferramenta de código aberto, escrita em Python, que implementa a análise matricial de pórticos planos com eficiência. Ela permite definir nós, elementos, apoios, carregamentos e propriedades da seção (área, momento de inércia, módulo de elasticidade), monta a matriz de rigidez global, resolve o sistema linear de equações e fornece deslocamentos, reações e esforços internos (momento fletor, força cortante, força normal). A simplicidade do modelo de pórtico é adequada para o objetivo de pré-dimensionamento e para a demonstração do fluxo de análise.

Para a verificação e análise da seção transversal do degrau, utilizou-se a biblioteca *Concreteproperties* (Robbins, 2023), também desenvolvida em Python. Entre muitos recursos disponíveis, essa biblioteca calcula a capacidade última à flexão de seções de concreto armado com base nas hipóteses do diagrama retangular de tensões e da compatibilidade de deformações. Para isso, o usuário fornece a geometria da seção, as armaduras (posição, diâmetro, quantidade), propriedades mecânicas dos materiais, e o programa retorna o momento fletor resistente. É importante destacar que *Concreteproperties* é uma ferramenta de verificação (dada uma armadura, informa se a seção resiste), não de dimensionamento automático. O dimensionamento iterativo será implementado no Capítulo 7, integrando os momentos solicitantes obtidos da análise estrutural de 1ª ordem com um procedimento de busca da armadura necessária.

4.5 A linguagem Python e seu uso em engenharia estrutural

Python é uma linguagem de programação de alto nível, interpretada, multiplataforma e de código aberto, amplamente adotada nas comunidades científica e de engenharia devido à sua sintaxe clara e ao enorme ecossistema de bibliotecas especializadas (Python Software Foundation, 2024). No contexto da engenharia estrutural, Python permite a implementação de rotinas de análise matricial, processamento de resultados, geração de gráficos e automação de tarefas repetitivas, sem a necessidade de licenças comerciais onerosas. Diferentemente de softwares fechados, o código em Python pode ser adaptado e estendido livremente, o que o torna particularmente adequado para pesquisas acadêmicas e para o desenvolvimento de ferramentas de projeto customizadas (Sebastian et al., 2024). Neste trabalho, a linguagem foi utilizada para implementar uma classe dedicada à análise de escadas plissadas, integrando as bibliotecas *Anastruct* e *Concreteproperties*.

4.6 A biblioteca *Anastruct* – análise matricial de pórticos planos

A *Anastruct* é uma biblioteca Python de código aberto desenvolvida para a análise matricial de pórticos planos lineares (Van Der Meer, 2022). O usuário define nós, elementos (barras), condições de contorno (apoios), carregamentos e propriedades dos materiais (área, momento de inércia, módulo de elasticidade). A biblioteca monta automaticamente a matriz de rigidez global do sistema, aplica as condições de contorno, resolve o sistema linear de equações e retorna os deslocamentos nodais, reações de apoio e esforços internos (momento fletor, força cortante, força normal) ao longo de cada elemento. A biblioteca *Anastruct* também oferece funções de pós-processamento para a visualização dos diagramas de esforços e da configuração deformada. Sua simplicidade e transparência a tornam adequada para fins didáticos e para a prototipagem rápida de modelos estruturais, sendo utilizada neste trabalho para a análise global da escada plissada.

4.7 A biblioteca *Concreteproperties* – verificação de seções de concreto armado

A *concreteproperties* é uma biblioteca Python especializada no cálculo de propriedades de seções de concreto armado, incluindo momentos de inércia (bruto, fissurado, efetivo), diagramas momento-curvatura e capacidade última de flexão (Robbins, 2023). O usuário define as geometrias da seção (retangular, circular, vazada, etc.), as armaduras (posição, diâmetro, quantidade) e os materiais (concreto com diagrama retangular simplificado, aço com comportamento elastoplástico). A biblioteca resolve o equilíbrio de forças e a compatibilidade de deformações com base nas hipóteses de seções planas e nos diagramas tensão-deformação

prescritos por normas como a ABNT NBR 6118:2026 e o Eurocode 2. É importante destacar que *concreteproperties* é uma ferramenta de verificação, na qual o usuário fornece uma seção e uma armadura candidatas, e o programa informa se essa seção resiste aos momentos solicitantes (momento último resistente $\geq$ momento atuante). Ela não realiza o dimensionamento automático, ou seja, não determina qual armadura é necessária. Essa distinção será fundamental para o entendimento do Capítulo 6, onde se aplica a verificação a um degrau típico, e do Capítulo 0, onde se implementará um procedimento de dimensionamento iterativo.

5 DESCRIÇÃO DA ANÁLISE ESTRUTURAL EM PYTHON

Neste capítulo, descreve-se a implementação computacional da classe *EscadaPlissada*, desenvolvida em Python para automatizar a geração da geometria, o cálculo das cargas, a análise estrutural linear (pórtico plano) e a extração dos momentos fletores máximos em escadas plissadas. O código foi organizado em células (blocos executáveis) dentro do ambiente Google Colab, e a descrição a seguir acompanha essa estrutura celular, explicando cada etapa de forma sequencial. A classe principal e seus métodos são apresentados na primeira célula, denominada “DEFINIÇÃO DA CLASSE”. Em seguida, uma segunda célula, “INTERFACE DO USUÁRIO”, reúne os parâmetros de entrada (passo, espelho, número de degraus, etc.) e instancia a classe. Por fim, a terceira célula, “EXEMPLO”, executa a análise, exibe os resultados numéricos e gera os gráficos da geometria, dos diagramas de momento fletor, força normal e deslocamento. A intenção é que até mesmo um leitor com pouca familiaridade com programação possa compreender o fluxo de dados e a lógica estrutural por trás do código.

5.1 Escada de referência adotada

Para a validação e demonstração do funcionamento da classe *EscadaPlissada*, bem como para as análises paramétricas subsequentes, foi definida uma escada de exemplo com as seguintes características geométricas e de carregamento, que atendem às prescrições da NBR 9050:2020 (Abnt, 2020) e à Fórmula de Blondel: passo de 30 cm, espelho de 17,5 cm, largura de 120 cm, espessura da laje de 12 cm, patamar de 120 cm e totalizando 9 espelhos (8 degraus). Aplicando a fórmula de Blondel, temos $2E + P = 65$ cm, um valor que se encontra dentro da faixa recomendada de 63 a 65 cm, garantindo o conforto e a segurança dos usuários.

A altura do espelho de 17,5 cm está em conformidade com o intervalo de 16 a 18 cm recomendado pela NBR 9050:2020, e o piso de 30 cm atende aos limites de 28 a 32 cm, assegurando também a acessibilidade.

A largura de 120 cm foi adotada por ser um valor comum para escadas residenciais e estar de acordo com as dimensões mínimas frequentemente recomendadas por normas de segurança.

5.2 Célula 1 – DEFINIÇÃO DA CLASSE

A primeira célula contém a definição completa da classe *EscadaPlissada*, que encapsula todos os parâmetros geométricos, de carregamento e de análise. Esta classe será detalhada nas subseções posteriores.

5.2.1 Estrutura da classe EscadaPlissada

A classe *EscadaPlissada* (Figura 3) foi desenvolvida para encapsular todos os parâmetros geométricos e mecânicos de uma escada plissada. O método construtor `__init__` recebe: passo (p), espelho (e), número de espelhos (n), largura (b_w), espessura da laje (h), comprimento do patamar (L_{pat}) e módulo de elasticidade do concreto (E). A partir desses valores, calcula-se o ângulo de inclinação ($angulo = np \cdot \arctan(e/p)$), o momento de inércia da seção bruta ($I = b_w \cdot h^3/12$) e a área da seção ($A = b_w \cdot h$). As cargas distribuídas são determinadas segundo a ABNT NBR 6120:2021: peso próprio do concreto (25 kN/m^3), revestimento (1 kN/m^2) e sobrecarga de uso ($2,5 \text{ kN/m}^2$). O objeto *SystemElements* da biblioteca *Anastruct* é instanciado com as rigidezes iniciais EA e EI , assumindo comportamento elástico linear (Estádio I).

Figura 3 - Método construtor da classe ‘EscadaPlissada’, mostrando a inicialização dos parâmetros e o cálculo das propriedades geométricas e dos carregamentos.

```

8 class EscadaPlissada:
9     """
10     Classe para geração automática de escadas plissadas
11     e análise estrutural usando o método dos elementos finitos pelo Anastruct
12     """
13
14     def __init__(self, p, e, n, bw, h, L_pat, E):
15         """
16         Inicializa os parâmetros da escada
17
18         Parâmetros:
19         -----
20         p : float
21             Passo do degrau (cm)
22         e : float
23             Espelho do degrau (cm)
24         n : int
25             Número de espelhos no lance
26         bw : float
27             largura da escada (cm)
28         h : float
29             espessura da escada (cm)
30         L_pat : float
31             Comprimento do patamar (cm)
32         E : float
33             Módulo de elasticidade do concreto (kN/cm²)
34         """
35         self.p = p
36         self.e = e
37         self.n = n
38         self.bw = bw
39         self.h = h
40         self.L_pat = L_pat
41         self.E = E
42
43         # Cálculo do ângulo da escada
44         self.angulo = np.arctan(e/p)
45
46         # Propriedades geométricas da seção bruta da escada
47         self.I = bw*(h**3)/12 #cm4
48         self.A = bw*h #cm2
49         self.ss = SystemElements(EA=E*self.A, EI=E*self.I)
50
51         # Carga no lance
52         self.glance=((p+h+e*h)*bw/1e6)*25/(p/100) #kN/m peso próprio
53         self.rev=1 * bw/100 #kN/m revestimento
54         self.variavel = 2.5 * bw/100 #kN/m uso e ocupação
55         self.cargalance = self.glance + self.rev + self.variavel #somatório
56
57         # Carga no patamar
58         self.gpat=h*25*bw/10000 #kN/m peso próprio
59         self.cargapat = self.gpat + self.rev + self.variavel #somatório
60
61         # Listas para armazenar dados
62         self.nos = []
63         self.elementos = []
64
65         # Array para momentos absolutos máximos característicos
66         self.momentos_absolutos = None
67
68         #=====

```

Fonte: Google Colab (2026).

5.2.2 Geração automática da geometria

O método *gerar_geometria* (Figura 4) constrói as coordenadas cartesianas de todos os nós da escada. A sequência inicia na origem (0,0) e, para cada espelho, adiciona-se primeiro o

ponto inferior (se $i > 1$) e depois o ponto superior. Essa estratégia produz uma poligonal que descreve fielmente o perfil em “dente de serra”. Se o patamar for maior que zero, um segmento horizontal é acrescentado ao final. As coordenadas são armazenadas na lista *self.coordenadas* e serão usadas para definir os elementos de pórtico.

Figura 4 - Método de geração da geometria: laço que cria os nós inferior e superior de cada espelho, resultando na linha poligonal da escada plissada.

```

70 def gerar_geometria(self):
71     """
72     Gera automaticamente as coordenadas dos nós da escada plissada
73     """
74     # Inicializa lista para armazenar coordenadas
75     coordenadas = []
76     x, y = 0, 0
77     coordenadas.append((x, y))
78
79
80     # Geração dos degraus: "i" é a ordem dos espelhos: 1º, 2º, 3º ... espelho
81     for i in range(1, self.n+1):
82         # Ponto inferior do espelho
83         if i <= self.n and i>1:
84             x = (i-1)*self.p
85             y = (i-1)* self.e
86             coordenadas.append((x, y))
87
88         # Ponto superior do espelho
89         x = (i-1)*self.p
90         y =i* self.e
91         coordenadas.append((x, y))
92
93     # Adiciona patamar no topo
94     if self.L_pat > 0:
95         x += self.L_pat
96         coordenadas.append((x, y))
97
98     self.coordenadas = coordenadas
99     return coordenadas
100
101     #=====

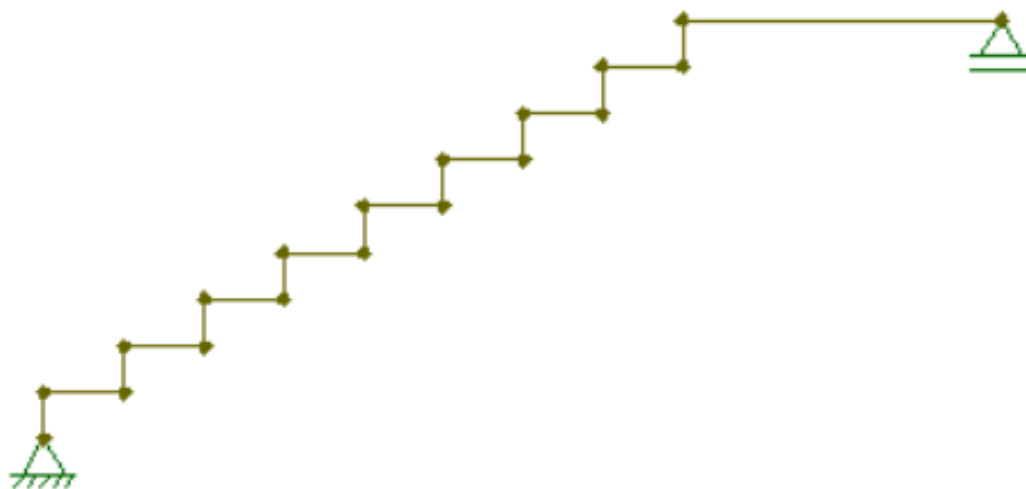
```

Fonte: Google Colab (2026).

5.2.3 Definição das cargas e propriedades da seção

O método *criar_modelo* (Figura 6) instancia o modelo de elementos finitos no *Anastruct*. Para cada par de nós consecutivos, um elemento de barra é adicionado com as rigidezes *EA* e *EI* previamente calculadas. A escada é modelada como uma viga contínua sobre dois apoios: o primeiro nó (base) recebe um apoio fixo (rótula), e o último nó (topo) recebe um apoio móvel na direção horizontal, permitindo a livre deformação axial, apresentado na Figura X.

Figura 5 - Representação da modelagem da estrutura da escada



Fonte: Ftool (2026).

As cargas distribuídas são aplicadas exclusivamente sobre os elementos horizontais (pisos), identificados pelos índices pares ($\text{elemento_id} \% 2 == 0$), e sobre o patamar (último elemento). Essa modelagem reflete a hipótese de que os espelhos atuam apenas como elementos de transmissão de esforços, sem carregamento direto.

Figura 6 - Construção do modelo no Anastruct: adição de elementos, definição de apoios e aplicação de carga distribuída nos pisos e no patamar.

```

103 def criar_modelo(self):
104     """
105     Cria o modelo estrutural no Anastruct
106     """
107     # Gera a geometria
108     self.gerar_geometria()
109
110     # Adiciona elementos entre os nós
111     for i in range(len(self.coordenadas) - 1):
112         no_inicio = self.coordenadas[i]
113         no_fim = self.coordenadas[i + 1]
114
115         # Adiciona elemento ao sistema
116         self.ss.add_element(
117             location=[no_inicio, no_fim],
118             EA=self.E * self.A,
119             EI=self.E * self.I
120         )
121
122         # Armazena informação do elemento
123         self.elementos.append({
124             'id': i + 1,
125             'inicio': no_inicio,
126             'fim': no_fim,
127             'comprimento': np.sqrt(
128                 (no_fim[0] - no_inicio[0])**2 +
129                 (no_fim[1] - no_inicio[1])**2
130             )
131         })
132
133     # Adiciona apoios
134     # Base da escada: apoio fixo (primeiro nó)
135     self.ss.add_support_hinged(node_id=1)
136
137     # Topo da escada: apoio móvel na direção x (último nó)
138     self.ss.add_support_roll(
139         node_id=len(self.coordenadas),
140         direction='x'
141     )
142
143     # Aplica carregamento
144     # Carga distribuída em todos os elementos horizontais do lance
145     for elemento_id in range(1, len(self.elementos)):
146         if elemento_id % 2 == 0: #os elementos de piso são pares
147             self.ss.q_load(
148                 q=-self.cargalance/100, # para baixo, kN/m -> kN/cm
149                 element_id=elemento_id,
150                 direction='y'
151             )
152     # Carga distribuída no patamar
153     self.ss.q_load(
154         q=-self.cargapat/100, # para baixo, kN/m -> kN/cm
155         element_id=len(self.elementos),
156         direction='y'
157     )
158     return self.ss

```

Fonte: Google Colab (2026).

5.2.4 Análise estrutural e extração de momentos

O método *analisar* (Figura 7) executa a análise estrutural: após criar o modelo, chama *solve()* para resolver o sistema linear de equações de equilíbrio e calcula as reações de apoio. O método *momentos_caracteristicos* percorre todos os elementos, acessa o *array* de momentos fletores ao longo do comprimento de cada barra (*elemento.bending_moment*) e

armazena o valor máximo em módulo para aquele elemento. Essa abordagem conservadora é adequada para o dimensionamento à flexão, pois considera tanto momentos positivos quanto negativos. O método *obter_momento_maximo* complementa a análise: identifica o elemento mais solicitado de toda a escada, aquele com o maior momento absoluto, retornando um dicionário com o identificador do elemento, o valor do momento, o comprimento do elemento e as coordenadas de seus nós inicial e final. Essa informação é útil para localizar a seção crítica e para verificar se o momento solicitante ultrapassa o momento resistente da seção (calculado separadamente com a biblioteca *concreteproperties*). Após a solução do sistema, cada elemento tem seu vetor de momentos percorrido para obter o maior valor absoluto, e o elemento mais solicitado é identificado.

Figura 7 - Rotina de análise e extração dos momentos fletores máximos

```

160     def analisar(self):
161         """
162         Realiza a análise estrutural
163         """
164         # Cria o modelo
165         self.criar_modelo()
166
167         # Resolve o sistema
168         self.ss.solve()
169
170         # Calcula reações de apoio
171         self.reacoes = self.ss.get_node_results_system()
172
173         # Calcula momentos absolutos
174         self.momentos_caracteristicos()
175
176         return self.ss
177
178     def momentos_caracteristicos(self):
179         """
180         Calcula e armazena os momentos absolutos máximos de cada elemento
181         Retorna array com os valores em kN.cm
182         """
183         n_elementos = len(self.ss.element_map)
184         momentos_abs = np.zeros(n_elementos)
185
186         for i in range(1, n_elementos + 1):
187             elemento = self.ss.element_map[i]
188
189             if hasattr(elemento, 'bending_moment'):
190                 momentos = elemento.bending_moment
191                 if len(momentos) > 0:
192                     # Valor absoluto máximo
193                     max_abs = np.max(np.abs(momentos))
194                     momentos_abs[i-1] = max_abs
195
196                     # Armazena no dicionário do elemento
197                     if i-1 < len(self.elementos):
198                         self.elementos[i-1]['momento_max_abs'] = max_abs
199
200         self.momentos_absolutos = momentos_abs
201         return momentos_abs
202
203     def obter_momento_maximo(self):
204         """
205         Retorna informações sobre o momento máximo geral
206         """
207         if self.momentos_absolutos is None:
208             self.momentos_caracteristicos()
209
210         idx_max = np.argmax(self.momentos_absolutos)
211         momento_max = self.momentos_absolutos[idx_max]
212
213         return {
214             'elemento_id': idx_max + 1,
215             'momento_max': momento_max,
216             'comprimento_elemento': self.elementos[idx_max]['comprimento'],
217             'inicio_elemento': self.elementos[idx_max]['inicio'],
218             'fin_elemento': self.elementos[idx_max]['fin']
219         }

```

Fonte: Google Colab (2026).

5.2.5 Pós-processamento e visualização

5.2.5.1 Relatório textual

O método *mostrar_resultados* (Figura 8 - Método *mostrar_resultados* gerando o relatório textual com dados de entrada, coordenadas dos nós, tabela de momentos por elemento e identificação do elemento crítico.) gera um relatório textual completo, impresso diretamente no console. Esse relatório é organizado em blocos: (i) dados de entrada (passo, espelho, número de espelhos, comprimento do patamar, cargas distribuídas, módulo de elasticidade, área e inércia da seção); (ii) geometria gerada (número de nós e elementos); (iii) coordenadas de todos os nós, permitindo a verificação visual da sequência de pontos; (iv) tabela dos momentos absolutos máximos para cada elemento, listando o comprimento do elemento e o momento fletor máximo correspondente em $kN.cm$; (v) identificação do elemento crítico, com seu momento máximo geral. Esse relatório é essencial para a validação dos resultados e para a documentação das análises paramétricas, pois permite ao usuário confrontar rapidamente os valores com limites normativos ou com resultados esperados.

Figura 8 - Método *mostrar_resultados* gerando o relatório textual com dados de entrada, coordenadas dos nós, tabela de momentos por elemento e identificação do elemento crítico.

```

223     def mostrar_resultados(self):
224         """
225         Exibe os resultados da análise
226         """
227         print("=" * 60)
228         print("ANÁLISE ESTRUTURAL DE ESCADA PLISSADA")
229         print("=" * 60)
230
231         # Dados de entrada
232         print("\nDADOS DE ENTRADA:")
233         print(f"Passo (p): {self.p:.3f} cm")
234         print(f"Espelho (e): {self.e:.3f} cm")
235         print(f"Número de espelhos (n): {self.n}")
236         print(f"Comprimento do patamar: {self.l_pat:.3f} cm")
237         print(f"Carga distribuída no lance: {self.cargalance:.2f} kN/m")
238         print(f"Carga distribuída no patamar: {self.cargapat:.2f} kN/m")
239         print(f"Módulo de elasticidade: {self.E:.0f} kN/cm²")
240         print(f"Área da seção: {self.A:.2f} cm²")
241         print(f"Momento de inércia: {self.I:.2f} cm⁴")
242
243         # Geometria gerada
244         print(f"\nGEOMETRIA GERADA:")
245         print(f"Número total de nós: {len(self.coordenadas)}")
246         print(f"Número total de elementos: {len(self.elementos)}")
247
248         print("\nCOORDENADAS DOS NÓS:")
249         for i, (x, y) in enumerate(self.coordenadas, 1):
250             print(f"Nó {i:2d}: ({x:6.2f}, {y:6.2f}) cm")
251         print("=" * 60)
252
253         # Mostra momentos a lista de momentos em cada elemento
254         print("\nMOMENTOS ABSOLUTOS MÁXIMOS POR ELEMENTO (kN.cm):")
255         print("-" * 50)
256         print(f"{'Elem':<6} {'Comp (cm)':<10} {'Mmax (kN.cm)':<12}")
257         print("-" * 50)
258
259         for i, elem in enumerate(self.elementos):
260             momento = self.momentos_absolutos[i] if self.momentos_absolutos is not None else 0
261             print(f"{i+1:<6} {elem['comprimento']:<10.3f} {momento:<12.3f}")
262
263         # Momento máximo geral
264         if self.momentos_absolutos is not None:
265             momento_max = self.obter_momento_maximo()
266             print("-" * 50)
267             print(f"\nMOMENTO MÁXIMO GERAL:")
268             print(f"Elemento crítico: {momento_max['elemento_id']}")
269             print(f"Momento máximo: {momento_max['momento_max']:.3f} kN.cm")
270             print(f"Comprimento do elemento: {momento_max['comprimento_elemento']:.3f} cm")
271
272         print("=" * 60)

```

Fonte: Google Colab (2026).

5.2.5.2 Visualização gráfica

Os métodos gráficos (Figura 9) complementam o relatório textual com representações visuais da estrutura e dos esforços. O método *plotar_geometria* desenha a poligonal da escada, numerando cada nó e mantendo a escala igual nos eixos (axis('equal')), o que permite verificar a proporção real dos degraus e a correta inclinação do lance. O método *plotar_diagramas* produz uma figura com quatro subgráficos alinhados verticalmente: (a) esquema estático, mostrando os apoios e a configuração deformada (amplificada); (b) diagrama de momento fletor, com cores diferenciadas para momentos positivos e negativos; (c)

diagrama de força normal; (d) diagrama de deslocamentos (flecha). Essas visualizações são fundamentais para a validação qualitativa do modelo, pois permitem identificar inconsistências como apoios mal posicionados, cargas aplicadas incorretamente ou comportamentos anômalos dos esforços.

Figura 9 - Métodos *plotar_geometria* e *plotar_diagramas*

```

275     def plotar_geometria(self):
276         """
277         Plota a geometria da escada
278         """
279         fig, ax = plt.subplots(figsize=(12, 6))
280
281         # Extrai coordenadas
282         x_coords = [coord[0] for coord in self.coordenadas]
283         y_coords = [coord[1] for coord in self.coordenadas]
284
285         # Plota a escada
286         ax.plot(x_coords, y_coords, 'b-o', linewidth=2, markersize=8)
287
288         # Configurações do gráfico
289         ax.set_xlabel('Comprimento (m)', fontsize=12)
290         ax.set_ylabel('Altura (m)', fontsize=12)
291         ax.set_title('Geometria da Escada Plissada', fontsize=14, fontweight='bold')
292         ax.grid(True, alpha=0.3)
293         ax.axis('equal')
294
295         # Adiciona anotações dos nós
296         for i, (x, y) in enumerate(self.coordenadas, 1):
297             ax.annotate(f'N{i}', (x, y), xytext=(5, 5),
298                         textcoords='offset points', fontsize=10)
299
300         plt.tight_layout()
301         plt.show()
302
303     def plotar_diagramas(self):
304         """
305         Plota os diagramas de esforços solicitantes
306         """
307         fig, axes = plt.subplots(4, 1, figsize=(14, 12))
308
309         # Esquema estático
310         self.ss.show_structure()
311         axes[0].set_title('Esquema estático', fontsize=12, fontweight='bold')
312         axes[0].grid(True, alpha=0.3)
313
314         # Diagrama de momento fletor
315         self.ss.show_bending_moment()
316         axes[1].set_title('Diagrama de Momento Fletor', fontsize=12, fontweight='bold')
317         axes[1].grid(True, alpha=0.3)
318
319         # Diagrama de força normal
320         self.ss.show_axial_force()
321         axes[2].set_title('Diagrama de Força Normal', fontsize=12, fontweight='bold')
322         axes[2].grid(True, alpha=0.3)
323
324         # Diagrama de deslocamento
325         self.ss.show_displacement()
326         axes[3].set_title('Diagrama de Deslocamento', fontsize=12, fontweight='bold')
327         axes[3].grid(True, alpha=0.3)
328
329         plt.tight_layout()
330         plt.show()

```

Fonte: Google Colab (2026).

5.3 Célula 2 – INTERFACE DO USUÁRIO

A segunda célula implementa uma função interativa denominada *interface_usuario()*, cujo objetivo é coletar os parâmetros da escada diretamente do usuário durante a execução do programa, sem necessidade de editar o código-fonte. A Figura 10 apresenta o trecho dessa função.

Figura 10 - Função da interface interativa para entrada de dados

```

1 # @title
2 #=====
3 def interface_usuario():
4     """
5     Interface interativa para entrada de dados
6     """
7     print("=" * 60)
8     print("FERRAMENTA DE ANÁLISE DE ESCADAS PLISSADAS")
9     print("=" * 60)
10    print("\nDigite os parâmetros da escada:")
11
12    # Entrada de dados
13    p = float(input("Passo do degrau (p) em [cm]: ") or "28")
14    e = float(input("Espelho do degrau (e) em [cm]: ") or "18")
15    n = int(input("Número de espelhos no lance: ") or "10")
16    bw = float(input("Largura da escada em [cm]: ") or "100")
17    h = float(input("Espessura da escada em [cm]: ") or "12")
18    L_pat = float(input("Comprimento do patamar em [cm]: ") or "120")
19
20
21    # Parâmetros de material (opcionais)
22    print("\nParâmetros do material conforme NBR 6118:")
23    E = float(input("Módulo de elasticidade E (kN/cm²) [25000]: ") or "25000")
24    #cargalance = float(input("Carga distribuída no lance (kN/m): ") or "6.0")
25    #cargapat = float(input("Carga distribuída no patamar (kN/m) :") or "5.0")
26
27
28    return p, e, n, bw, h, L_pat, cargalance, cargapat, E

```

Fonte: Google Colab (2026).

A função exibe mensagens orientativas e utiliza o comando *input()* para receber valores em *cm*, *kN/cm²* e *kN/m*, conforme as unidades adotadas na classe. Para cada parâmetro, é fornecido um valor padrão que será utilizado caso o usuário simplesmente pressione “Enter”. Essa estratégia agiliza os testes e evita a necessidade de digitar repetidamente os mesmos valores. Os parâmetros retornados são: passo (*p*), espelho (*e*), número de espelhos (*n*), largura (*bw*), espessura da laje (*h*), comprimento do patamar (*L_pat*) e módulo de elasticidade (*E*).

5.4 Célula 3 - EXEMPLO

A terceira célula contém o bloco principal de execução (*if __name__ == "__main__":*), que demonstra como utilizar a classe e a interface. A Figura 11 mostra o trecho correspondente.

Figura 11 - Exemplo de aplicação

```

1 # Exemplo de uso=====
2 if __name__ == "__main__":
3     """
4     Modo de execução principal
5     """
6
7     # Modo 1: Valores padrão (para testes rápidos)
8     usar_padrao = input("\nUsar valores padrão? (s/n) [s]: ") or "s"
9
10    if usar_padrao.lower() == 's':
11        # Exemplo com valores típicos
12        p = 30      # Passo [cm]
13        e = 17.5    # Espelho [cm]
14        n = 9      # Número de degraus
15        bw = 120   # Largura da escada [cm]
16        h = 12     # Espessura da escada [cm]
17        L_pat = 120 # Comprimento do Patamar [cm]
18        E = 2500   # Módulo de elasticidade do concreto (kN/cm²)
19
20    else:
21        # Modo 2: Entrada interativa
22        p, e, n, bw, h, L_pat, E = interface_usuario()
23
24    # Cria e analisa a escada
25    escada = EscadaPlissada(p, e, n, bw, h, L_pat, E)
26    escada.analisar()
27
28    # Mostra resultados
29    escada.mostrar_resultados()
30
31    # Plota resultados
32    print("\nGerando visualizações gráficas...")
33    escada.plotar_geometria()
34    escada.plotar_diagramas()

```

Fonte: Google Colab (2026).

O código oferece dois modos de operação:

- Modo padrão: se o usuário responder “s” (ou simplesmente pressionar “Enter”), são atribuídos valores típicos (passo 30 cm, espelho 17,5 cm, 9 degraus, largura 120 cm, espessura 12 cm, patamar 120 cm, $E = 2500 \text{ kN/cm}^2$). Este modo é útil para demonstrações rápidas e para verificar o funcionamento geral do programa.
- Modo interativo: se o usuário responder “n”, a função *interface_usuario()* é chamada, permitindo que ele digite seus próprios valores.

Para o exemplo, foi utilizado o modo padrão para demonstração da execução do programa.

Após a instância da classe (*escada = EscadaPlissada(...)*), são executados sequencialmente:

- *analisar()* – realiza a montagem do modelo, a solução do sistema e a extração dos momentos fletores (conforme detalhado nas seções anteriores).
- *mostrar_resultados()* – exibe um relatório textual completo, incluindo dados de entrada, coordenadas dos nós e uma tabela com os momentos máximos por elemento, como demonstrado na Figura 12.

Figura 12 - Relatório textual final gerado pelo método *mostrar resultados*

```

=====
ANÁLISE ESTRUTURAL DE ESCADA PLISSADA
=====

DADOS DE ENTRADA:
Passo (p): 30.000 cm
Espelho (e): 17.500 cm
Número de espelhos (n): 9
Comprimento do patamar: 120.000 cm
Carga distribuída no lance: 9.90 kN/m
Carga distribuída no patamar: 7.80 kN/m
Módulo de elasticidade: 2500 kN/cm²
Área da seção: 1440.00 cm²
Momento de inércia: 17280.00 cm⁴

GEOMETRIA GERADA:
Número total de nós: 19
Número total de elementos: 18

COORDENADAS DOS NÓS:
Nó 1: ( 0.00,  0.00) cm
Nó 2: ( 0.00, 17.50) cm
Nó 3: (30.00, 17.50) cm
Nó 4: (30.00, 35.00) cm
Nó 5: (60.00, 35.00) cm
Nó 6: (60.00, 52.50) cm
Nó 7: (90.00, 52.50) cm
Nó 8: (90.00, 70.00) cm
Nó 9: (120.00, 70.00) cm
Nó 10: (120.00, 87.50) cm
Nó 11: (150.00, 87.50) cm
Nó 12: (150.00, 105.00) cm
Nó 13: (180.00, 105.00) cm
Nó 14: (180.00, 122.50) cm
Nó 15: (210.00, 122.50) cm
Nó 16: (210.00, 140.00) cm
Nó 17: (240.00, 140.00) cm
Nó 18: (240.00, 157.50) cm
Nó 19: (360.00, 157.50) cm
=====

MOMENTOS ABSOLUTOS MÁXIMOS POR ELEMENTO (kN.cm):
-----
Elen  Comp (cm)  Mmax (kN.cm)
-----
1      17.500      0.000
2      30.000      477.450
3      17.500      477.450
4      30.000      865.800
5      17.500      865.800
6      30.000      1165.050
7      17.500      1165.050
8      30.000      1375.200
9      17.500      1375.200
10     30.000      1496.250
11     17.500      1496.250
12     30.000      1529.091
13     17.500      1528.200
14     30.000      1528.200
15     17.500      1471.050
16     30.000      1471.050
17     17.500      1324.800
18     120.000     1324.800
-----

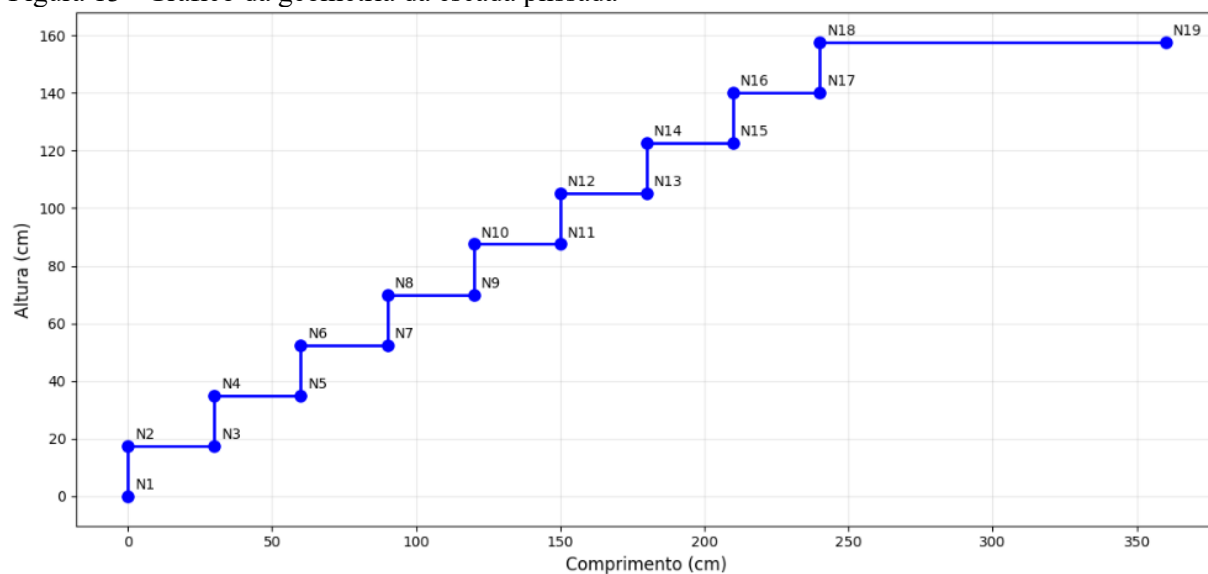
MOMENTO MÁXIMO GERAL:
Elemento crítico: 12
Momento máximo: 1529.091 kN.cm
Comprimento do elemento: 30.000 cm
=====

```

Fonte: Google Colab (2026).

- `plotar_geometria()` – gera um gráfico da linha da escada, numerando os nós e mantendo a escala igual nos eixos, o que permite verificar a proporção real dos degraus (Figura 13).

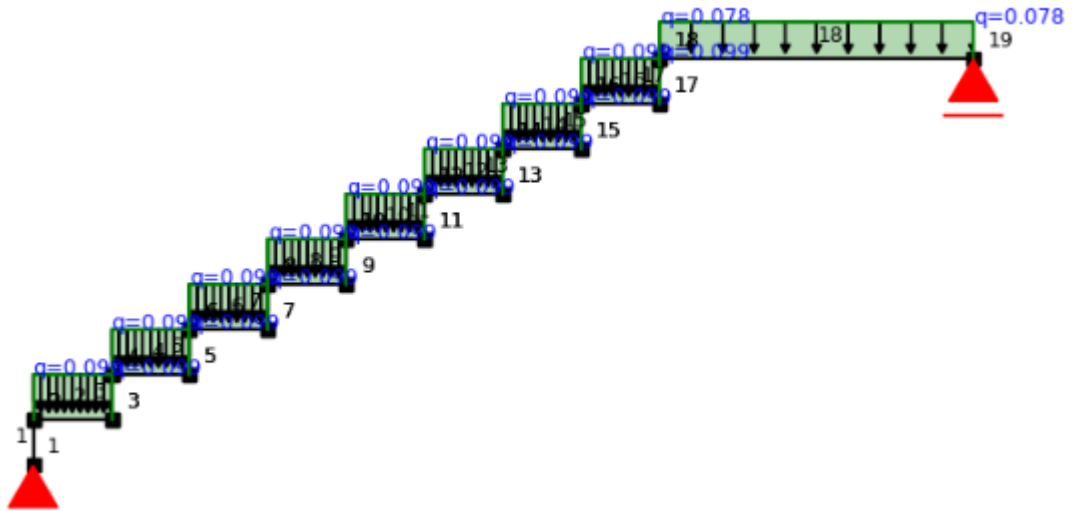
Figura 13 - Gráfico da geometria da escada plissada



Fonte: Anastruct (2022).

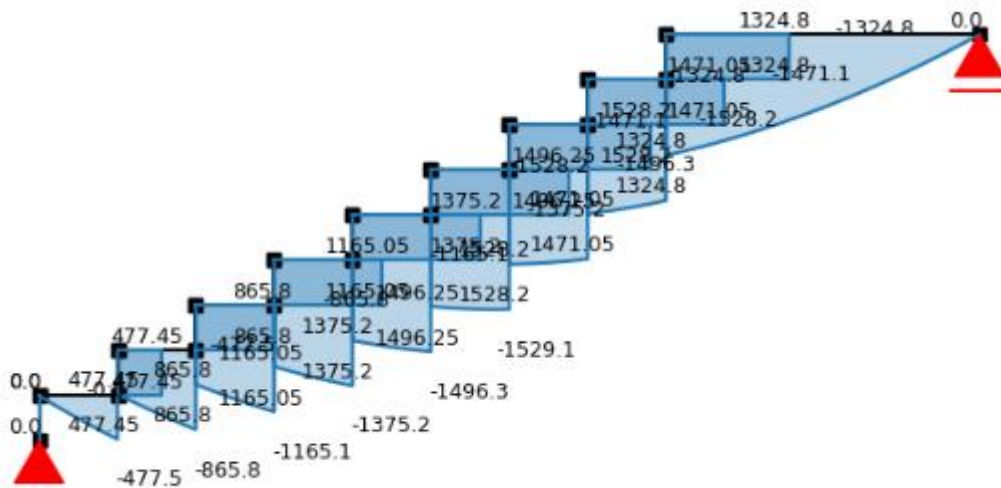
- *plotar_diagramas()* – produz quatro subgráficos alinhados verticalmente: esquema estático com as cargas distribuídas no lance e no patamar, diagrama de momento fletor, diagrama de força normal e diagrama de deslocamento. As Figuras de Figura 14 a Figura 17 exemplificam essa saída para o caso padrão.

Figura 14 - Esquema estático com cargas distribuídas



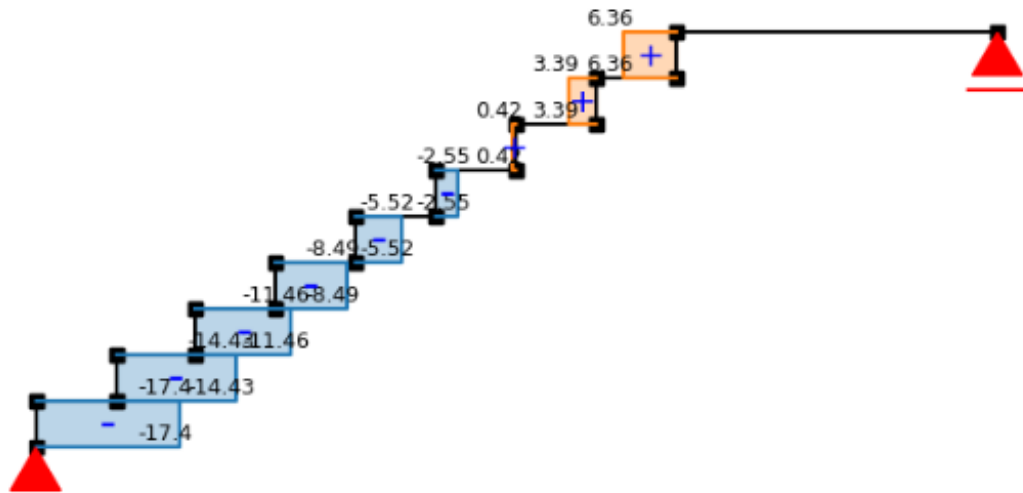
Fonte: Anastruct (2022).

Figura 15 - Diagrama de momento fletor



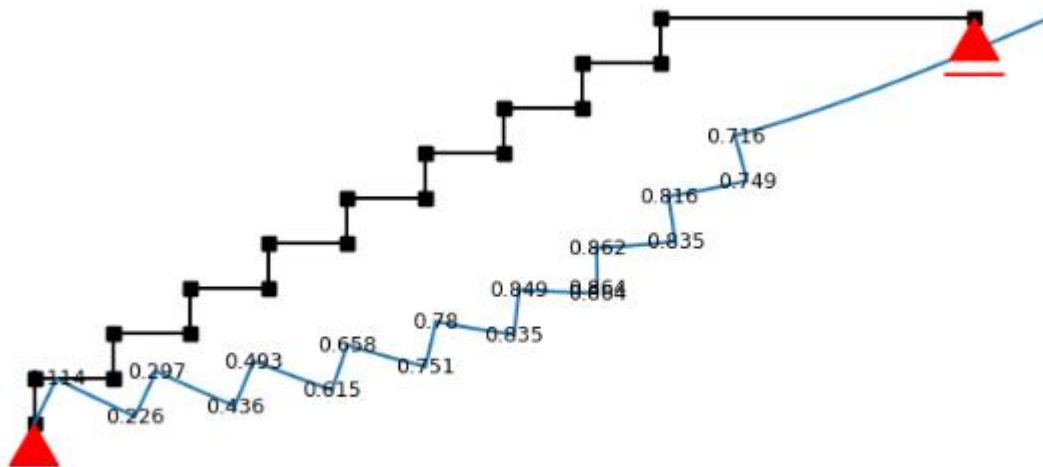
Fonte: Anastruct (2022).

Figura 16 - Diagrama de esforço normal



Fonte: Anastruct (2022).

Figura 17 - Diagrama de deslocamentos



Fonte: Anastruct (2022).

Essa organização permite que o usuário modifique os parâmetros na célula de interface e reexecute a célula de exemplo para observar a variação dos resultados, sem necessidade de reiniciar todo o ambiente.

5.5 Comparação dos resultados com o Ftool

Com o objetivo de verificar a coerência dos resultados obtidos pela ferramenta desenvolvida em Python, realizou-se uma comparação entre os diagramas gerados pela classe *EscadaPlissada* e os resultados fornecidos pelo software Ftool. Essa comparação foi realizada

para a mesma escada de referência apresentada na Seção 5.1, mantendo-se as mesmas dimensões geométricas, condições de apoio, propriedades mecânicas e carregamentos adotados no modelo implementado com a biblioteca *Anastruct*.

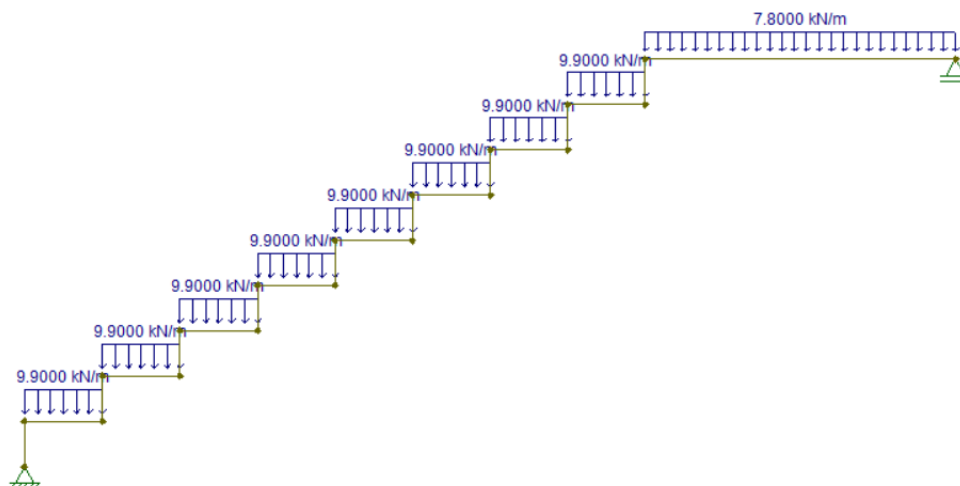
O Ftool foi escolhido como ferramenta de comparação por ser um programa amplamente utilizado no ensino e na prática da análise estrutural de pórticos planos. Como o modelo estrutural adotado neste trabalho representa a escada plissada por elementos lineares de pórtico plano, a comparação com o Ftool permite verificar se a montagem automática da geometria, a aplicação dos carregamentos e a extração dos esforços internos no código Python estão coerentes com uma ferramenta consolidada de análise matricial.

Para possibilitar a comparação, a geometria da escada foi reproduzida manualmente no Ftool, utilizando as coordenadas dos nós geradas pela classe *EscadaPlissada*. Foram considerados os mesmos elementos horizontais e verticais, representando respectivamente os pisos e espelhos da escada. As condições de contorno também foram mantidas: apoio articulado na base e apoio móvel no topo, conforme adotado no modelo computacional desenvolvido em Python. Além disso, foram inseridas as mesmas propriedades de rigidez axial e à flexão dos elementos, definidas pelos parâmetros (EA) e (EI), calculados a partir da área da seção transversal, do momento de inércia e do módulo de elasticidade do concreto.

Os carregamentos foram aplicados no Ftool de forma equivalente ao modelo desenvolvido em Python. Assim, as cargas distribuídas foram atribuídas aos elementos horizontais correspondentes aos pisos e ao patamar, mantendo-se a hipótese de que os espelhos atuam como elementos de transmissão de esforços, sem carregamento distribuído direto. Essa equivalência é fundamental para que a comparação entre os dois modelos seja tecnicamente válida, uma vez que pequenas diferenças na aplicação das cargas, nas unidades ou nas condições de apoio podem alterar os diagramas de esforços e deslocamentos.

A Figura 18 apresenta o modelo da escada plissada reproduzido no Ftool, com a mesma geometria, apoios e carregamentos utilizados na análise realizada em Python. Essa representação permite verificar visualmente a correspondência entre o modelo construído manualmente no Ftool e a geometria gerada automaticamente pela classe *EscadaPlissada*.

Figura 18 - Modelo da escada plissada reproduzido no Ftool



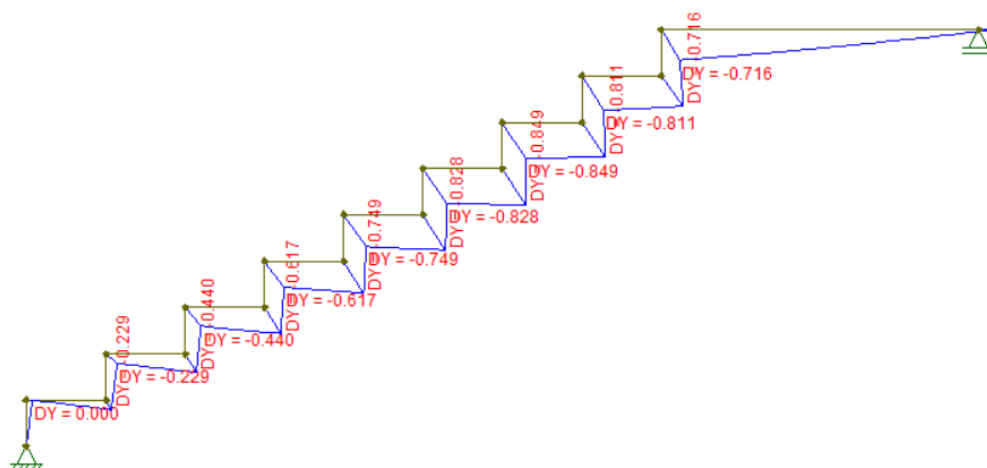
Fonte: Ftool (2026).

Após a montagem do modelo, foram comparados os principais resultados fornecidos pelas duas ferramentas: o diagrama de momento fletor, o diagrama de esforço normal e o diagrama de deslocamentos. A comparação dos diagramas permite avaliar não apenas os valores máximos obtidos, mas também a forma geral da distribuição dos esforços ao longo dos elementos da escada.

A Figura 19 apresenta o diagrama de momento fletor obtido no Ftool para a escada analisada. Observa-se que a distribuição dos momentos ao longo dos elementos apresenta comportamento compatível com o diagrama obtido pelo código Python, com concentração dos maiores valores na mesma região crítica da estrutura. Essa concordância indica que a rotina de geração da geometria, definição dos apoios e aplicação dos carregamentos está compatível com o modelo de pórtico plano reproduzido no Ftool.

A Figura 21 apresenta o diagrama de deslocamentos obtido no Ftool. Os deslocamentos devem ser interpretados considerando que a análise foi realizada em regime elástico linear, com rigidez bruta da seção de concreto e sem consideração da fissuração. O comportamento deformado apresentado pelo Ftool mostrou-se condizente com o diagrama de deslocamentos gerado anteriormente pela ferramenta desenvolvida em Python, reforçando a compatibilidade entre os modelos analisados.

Figura 21 - Diagrama de deslocamentos obtido no Ftool



Fonte: Ftool (2026).

Para complementar a avaliação visual dos diagramas, a Tabela 1 apresenta a comparação quantitativa entre os principais valores obtidos nas duas ferramentas. Foram considerados o momento fletor máximo, o esforço normal máximo e o deslocamento máximo.

Tabela 1 - Comparação quantitativa entre os resultados obtidos no Python e no Ftool

Resultado analisado	Python/Anastruct	Ftool	Diferença (%)
Momento fletor máximo (kN.m)	15,29	15,29	0
Esforço normal máximo (kN)	17,4	17,4	0
Deslocamento máximo (cm)	0,864	0,849	1,77

Fonte: Autoria Própria (2026).

A partir da comparação realizada, observa-se que os resultados obtidos pelo código Python apresentam concordância com aqueles fornecidos pelo Ftool, tanto em relação à forma geral dos diagramas quanto aos valores máximos dos esforços e deslocamentos. Eventuais

diferenças numéricas podem estar associadas a critérios internos de discretização gráfica, convenções de sinal, arredondamentos numéricos ou à forma de apresentação dos resultados em cada programa. Ainda assim, desde que mantidas as mesmas hipóteses de modelagem, apoios, propriedades dos elementos e carregamentos, a comparação confirma a consistência da análise estrutural implementada na classe *EscadaPlissada*.

Dessa forma, a comparação com o Ftool reforça a confiabilidade da ferramenta desenvolvida para a etapa de análise global linear da escada plissada. Ressalta-se, entretanto, que essa verificação não substitui uma validação experimental ou uma análise não linear mais refinada, mas constitui uma etapa importante de conferência computacional dos resultados obtidos pelo modelo de pórtico plano adotado neste trabalho.

6 APLICAÇÃO PRELIMINAR DA BIBLIOTECA CONCRETEPROPERTIES EM SEÇÃO RETANGULAR DE CONCRETO ARMADO

Antes da aplicação da biblioteca *Concreteproperties* à seção da escada plissada, apresenta-se neste capítulo um exemplo preliminar de verificação de uma seção retangular de concreto armado. Esse exemplo inicial possui caráter didático e demonstrativo, tendo como finalidade apresentar o funcionamento básico da biblioteca, a definição dos materiais, a criação da geometria da seção, a inserção das armaduras e a obtenção dos momentos resistentes últimos.

Assim, a seção retangular analisada inicialmente não corresponde à seção dimensionada da escada plissada estudada neste trabalho. Trata-se de uma seção genérica, semelhante à de uma viga retangular de concreto armado, adotada apenas para demonstrar a rotina de utilização da biblioteca e interpretar suas principais saídas. Dessa forma, os resultados obtidos nesse primeiro exemplo não devem ser comparados diretamente com os esforços solicitantes da escada, nem utilizados como solução de dimensionamento para o caso estudado.

A aplicação da biblioteca à seção efetiva da escada plissada será realizada posteriormente, após a determinação da armadura longitudinal necessária pelo procedimento iterativo de dimensionamento. Nesse segundo momento, a seção da escada será modelada com as dimensões, materiais e armaduras correspondentes ao exemplo analisado, permitindo verificar se o momento fletor resistente de cálculo é superior ao momento fletor solicitante de cálculo.

Portanto, para avaliar a capacidade resistente da seção transversal típica de um degrau (pisso) da escada plissada, utilizou-se a biblioteca *Concreteproperties* (Robbins, 2023), desenvolvida em Python para análise de seções de concreto armado segundo o diagrama retangular simplificado e as hipóteses de seções. A seção adotada é retangular, com largura $b = 150$ mm e altura total $h = 300$ mm. Foram adicionadas armadura inferior ($3\phi 12,5$ mm – tração) e superior ($2\phi 10$ mm – compressão), com cobrimento de 30 mm. O concreto é da classe C30 ($f_{ck} = 30$ MPa) e o aço é CA-50 ($f_{yk} = 500$ MPa). O método *ultimate_bending_capacity* calculou os momentos resistentes para quatro combinações de carregamento (positivo/negativo, com/sem força axial de compressão de 50 kN). O código foi executado no ambiente Google Colab, e os trechos principais são discutidos a seguir.

6.1 Definição dos materiais

A geometria da seção retangular foi criada com auxílio das rotinas geométricas associadas ao ambiente da biblioteca *Concreteproperties*, que utiliza recursos do pacote *sectionproperties* (Van Leeuwen; Ferster, 2024) para a construção de seções transversais. A biblioteca SciPy foi atualizada para garantir compatibilidade com a versão do NumPy utilizada. Todas as análises foram executadas em ambiente Python (PYTHON SOFTWARE FOUNDATION, 2024).

O concreto é modelado pela classe *Concrete*, com resistência característica à compressão $f_{ck} = 30$ MPa e módulo de elasticidade linear $E_c = 34,8$ GPa. O perfil tensão-deformação último adota os parâmetros $\alpha = 0,85$ e $\gamma = 0,8$, que é equivalente ao prescrito na ABNT NBR 6118:2023 para concretos até 50 MPa. A resistência à tração na flexão é fixada em 4,2 MPa. O aço CA-50 é definido pela classe *SteelBar*, com tensão de escoamento $f_y = 500$ MPa, módulo de elasticidade $E_s = 200$ GPa e comportamento elasto-plástico perfeito, conforme Figura 22.

Figura 22 - Definição dos materiais concreto C30 e aço CA-50

```

1 # -----
2 # 1. DEFINIÇÃO DOS MATERIAIS
3 # -----
4 concrete = Concrete(
5     name="Concreto C30",
6     density=2.4e-6, # kg/mm³
7     stress_strain_profile=ssp.ConcreteLinear(elastic_modulus=34.8e3), # MPa
8     ultimate_stress_strain_profile=ssp.RectangularStressBlock(
9         compressive_strength=30/1.4, # f'c = 30 MPa
10        alpha=0.85, # fator de intensidade
11        gamma=0.8, # fator de altura do bloco
12        ultimate_strain=0.0035,
13    ),
14    flexural_tensile_strength=4.2, # MPa (resistência à tração na flexão)
15    colour="lightgrey",
16 )
17
18 steel = SteelBar(
19     name="Aço CA-50",
20     density=7.85e-6, # kg/mm³
21     stress_strain_profile=ssp.SteelElasticPlastic(
22         yield_strength=500/1.15, # fy = 500 MPa
23         elastic_modulus=200e3, # MPa
24         fracture_strain=0.01,
25     ),
26     colour="grey",
27 )

```

Fonte: Google Colab (2026).

6.2 Geometria da seção e criação do modelo

A Figura 23 ilustra a criação da seção bruta de concreto. As dimensões nominais do degrau são largura $b = 150$ mm e altura total $h = 300$ mm. A função *rectangular_section* da

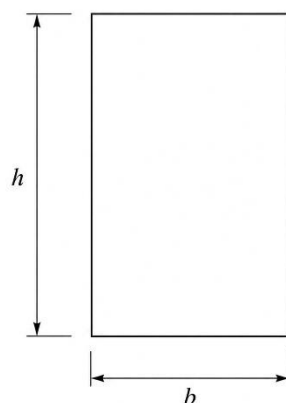
biblioteca *sectionproperties* gera a geometria e atribui automaticamente o material concreto definido anteriormente. Esse objeto geométrico servirá de base para a adição das armaduras.

Figura 23 - Criação da seção retangular bruta

```

29 # -----
30 # 2. GEOMETRIA DA SEÇÃO RETANGULAR (DEGRAU)
31 # -----
32 # Dimensões do degrau: largura (b) e altura total (h)
33 b = 150.0      # mm
34 h = 300.0      # mm
35
36 # Cria a seção de concreto (material atribuído diretamente)
37 geom = rectangular_section(d=h, b=b, material=concrete)

```



Fonte: Adaptado de Google Colab (2026).

6.3 Adição das armaduras

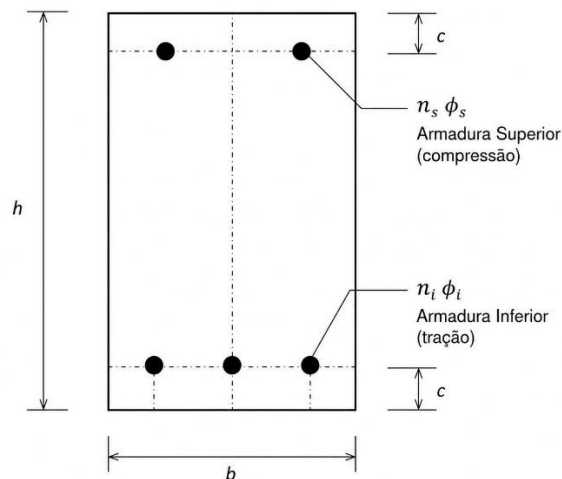
A Figura 24 apresenta a rotina de inclusão das barras de aço na seção. O cobrimento nominal adotado é 30 mm, medido até o centro das barras, valor compatível com a classe de agressividade ambiental II (NBR 6118:2023). A armadura inferior (tração) é composta por três barras de 12,5 mm de diâmetro, resultando em área total aproximada de 380 mm². As barras são distribuídas horizontalmente de forma simétrica, com espaçamento calculado automaticamente. A armadura superior (compressão) é constituída por duas barras de 10 mm, totalizando 160 mm², posicionadas a uma distância do topo igual ao cobrimento. A função `add_bar_rectangular_array` insere as barras na geometria da seção, respeitando as coordenadas.

Figura 24 - Adição das armaduras à seção retangular

```

39 # -----
40 # 3. ADIÇÃO DAS ARMADURAS (dupla: inferior e superior)
41 # -----
42 # Cobrimento nominal (considerando estribos)
43 cover = 30.0      # mm (cobrimento até o centro das barras)
44
45 # --- Armadura Inferior (tração) ---
46 # 3 barras de 12.5 mm (área total aproximada = 3 * pi*(12.5/2)^2 = 368 mm²)
47 area_bottom = 368 # mm²
48 n_bottom = 3
49 # Posição vertical: a partir da base (y = cover)
50 y_bottom = cover
51 # Distribuição horizontal: simétrica dentro da largura
52 x_start = cover
53 x_end = b - cover
54 x_spacing = (x_end - x_start) / (n_bottom - 1) if n_bottom > 1 else 0
55
56 geom = add_bar_rectangular_array(
57     geometry=geom,
58     area=area_bottom / n_bottom,    # área por barra
59     material=steel,
60     n_x=n_bottom,
61     x_s=x_spacing,
62     anchor=(x_start, y_bottom),
63 )
64
65 # --- Armadura Superior (compressão) ---
66 # 2 barras de 10 mm (área total aproximada = 2 * pi*(10/2)^2 = 157 mm²)
67 area_top = 157 # mm²
68 n_top = 2
69 y_top = h - cover
70 x_start_top = cover
71 x_end_top = b - cover
72 x_spacing_top = (x_end_top - x_start_top) / (n_top - 1) if n_top > 1 else 0
73
74 geom = add_bar_rectangular_array(
75     geometry=geom,
76     area=area_top / n_top,
77     material=steel,
78     n_x=n_top,
79     x_s=x_spacing_top,
80     anchor=(x_start_top, y_top),
81 )

```



Fonte: Adaptado de Google Colab (2026).

6.4 Criação da seção de concreto armado e cálculo da capacidade última

A Figura 25 exibe a criação da seção de concreto armado (*ConcreteSection*) e a determinação dos momentos fletores resistentes de cálculo. O

método *ultimate_bending_capacity* resolve o equilíbrio de forças e deformações na seção, adotando as hipóteses de compatibilidade de deformações e os diagramas tensão-deformação dos materiais. Foram analisadas quatro situações:

1. Momento positivo sem força axial;
2. Momento negativo sem força axial;
3. Momento positivo com força axial de compressão de 50 kN;
4. Momento negativo com a mesma força axial.

Os resultados são impressos em unidades de $kN.m$ e também armazenados nos objetos *sagging_res* e *hogging_res*. A extração simplificada dos valores é feita diretamente pelo atributo *m_xy*. Os momentos obtidos servem como referência para a verificação da segurança da seção do degrau quando submetida aos esforços provenientes da análise global realizada pela classe *EscadaPlissada*.

Figura 25 - Cálculo dos momentos resistentes último para as quatro combinações de solicitação

```

83 # -----
84 # 4. CRIAÇÃO DA SEÇÃO DE CONCRETO ARMADO
85 # -----
86 conc_sec = ConcreteSection(geom)
87
88 # Visualização da seção (opcional, requer matplotlib)
89 # conc_sec.plot_section()
90
91 # -----
92 # 5. CÁLCULO DA CAPACIDADE ÚLTIMA DE FLEXÃO
93 # -----
94 # Sem força axial (N=0)
95 print("\n--- RESULTADOS PARA N = 0 ---")
96 sagging_res = conc_sec.ultimate_bending_capacity() #  $\theta = 0$  (momento positivo)
97 hogging_res = conc_sec.ultimate_bending_capacity(theta=np.pi) #  $\theta = \pi$  (momento negativo)
98
99 print("\nMomento Positivo (tração na fibra inferior):")
100 sagging_res.print_results(units=si_kn_m)
101
102 print("\nMomento Negativo (tração na fibra superior):")
103 hogging_res.print_results(units=si_kn_m)
104
105 # Exemplo adicional: com força axial de compressão (simulando protensão ou carga permanente)
106 N_axial_kN = 50.0 # kN (força axial compressiva)
107 N_axial_N = N_axial_kN * 1000.0
108
109 print(f"\n--- RESULTADOS PARA N = {N_axial_kN} kN (compressão) ---")
110 sag_axial_res = conc_sec.ultimate_bending_capacity(n=N_axial_N)
111 hog_axial_res = conc_sec.ultimate_bending_capacity(theta=np.pi, n=N_axial_N)
112
113 print("\nMomento Positivo com compressão axial:")
114 sag_axial_res.print_results(units=si_kn_m)
115
116 print("\nMomento Negativo com compressão axial:")
117 hog_axial_res.print_results(units=si_kn_m)
118
119 # -----
120 # (Opcional) Extração simplificada dos momentos resistentes
121 # -----
122 print("\n--- RESUMO DOS MOMENTOS RESISTENTES (kN.m) ---")
123 print(f"Momento positivo (N=0) : {sagging_res.m_xy / 1e6:.1f} kN.m")
124 print(f"Momento negativo (N=0) : {hogging_res.m_xy / 1e6:.1f} kN.m")
125 print(f"Momento positivo (N={N_axial_kN} kN): {sag_axial_res.m_xy / 1e6:.1f} kN.m")
126 print(f"Momento negativo (N={N_axial_kN} kN): {hog_axial_res.m_xy / 1e6:.1f} kN.m")
127 conc_sec = ConcreteSection(geom)
128 conc_sec.plot_section()

```

Fonte: Google Colab (2026).

6.5 Resultados

Neste item, apresenta-se a saída do código implementado na biblioteca *Concreteproperties* para a verificação da seção transversal do degrau. São exibidos os valores dos momentos resistentes calculados, bem como a representação geométrica da seção armada. Os resultados são organizados nas subseções seguintes.

6.5.1 Momentos resistentes para as combinações de carregamento

A Figura 26 reproduz a saída do código com as tabelas de momentos resistentes últimos para a seção retangular com armadura dupla, considerando as quatro combinações de carregamento: momento positivo e negativo sem força axial e momento positivo e negativo

com força axial de compressão de 50 kN. Os valores numéricos são apresentados diretamente na figura, permitindo a comparação entre as diferentes solicitações.

Figura 26 - Tabelas de momentos resistentes para as combinações N=0 e N=50 kN

--- RESULTADOS PARA N = 0 ---

Momento Positivo (tração na fibra inferior):
Ultimate Bending Results

Property	Value
Bending Angle - theta	0.000 rads
Neutral Axis Depth - d_n	52.80 x 10 ⁻³ m
Neutral Axis Parameter - k_u	195.6 x 10 ⁻³
Axial Force	-1.764 x 10 ⁻³ kN
Bending Capacity - m_x	39.42 kN.m
Bending Capacity - m_y	2.095 x 10 ⁻¹⁵ kN.m
Bending Capacity - m_xy	39.42 kN.m

Momento Negativo (tração na fibra superior):
Ultimate Bending Results

Property	Value
Bending Angle - theta	3.142 rads
Neutral Axis Depth - d_n	30.27 x 10 ⁻³ m
Neutral Axis Parameter - k_u	112.1 x 10 ⁻³
Axial Force	5.302 x 10 ⁻³ kN
Bending Capacity - m_x	-17.57 kN.m
Bending Capacity - m_y	-1.164 x 10 ⁻¹⁵ kN.m
Bending Capacity - m_xy	17.57 kN.m

--- RESULTADOS PARA N = 50.0 kN (compressão) ---

Momento Positivo com compressão axial:
Ultimate Bending Results

Property	Value
Bending Angle - theta	0.000 rads
Neutral Axis Depth - d_n	68.97 x 10 ⁻³ m
Neutral Axis Parameter - k_u	255.5 x 10 ⁻³
Axial Force	50.00 kN
Bending Capacity - m_x	44.76 kN.m
Bending Capacity - m_y	2.794 x 10 ⁻¹⁵ kN.m
Bending Capacity - m_xy	44.76 kN.m

Momento Negativo com compressão axial:
Ultimate Bending Results

Property	Value
Bending Angle - theta	3.142 rads
Neutral Axis Depth - d_n	35.87 x 10 ⁻³ m
Neutral Axis Parameter - k_u	132.9 x 10 ⁻³
Axial Force	50.00 kN
Bending Capacity - m_x	-23.60 kN.m
Bending Capacity - m_y	-465.7 x 10 ⁻¹⁸ kN.m
Bending Capacity - m_xy	23.60 kN.m

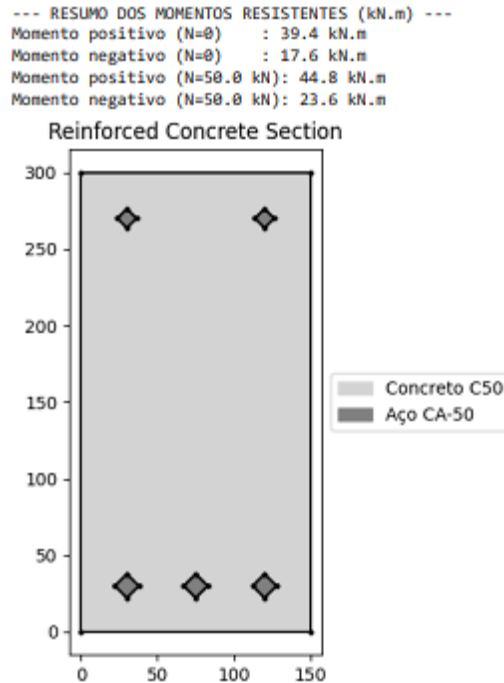
Fonte: Google Colab (2026).

6.5.2 Representação da seção transversal e resumo dos momentos resistentes

A Figura 27 apresenta a representação gráfica da seção transversal do degrau, com a disposição das armaduras inferior (3φ12,5 mm) e superior (2φ10 mm), gerada pelo método `plot_section()`. Além disso, a mesma figura inclui um resumo simplificado dos

momentos resistentes já apresentados na subseção anterior, consolidando as informações de forma visual para rápida consulta.

Figura 27 - Resumo dos momentos resistentes e representação da seção armada



Fonte: Adaptado de Robbins (2023).

6.5.3 Diagrama de interação momento-força normal

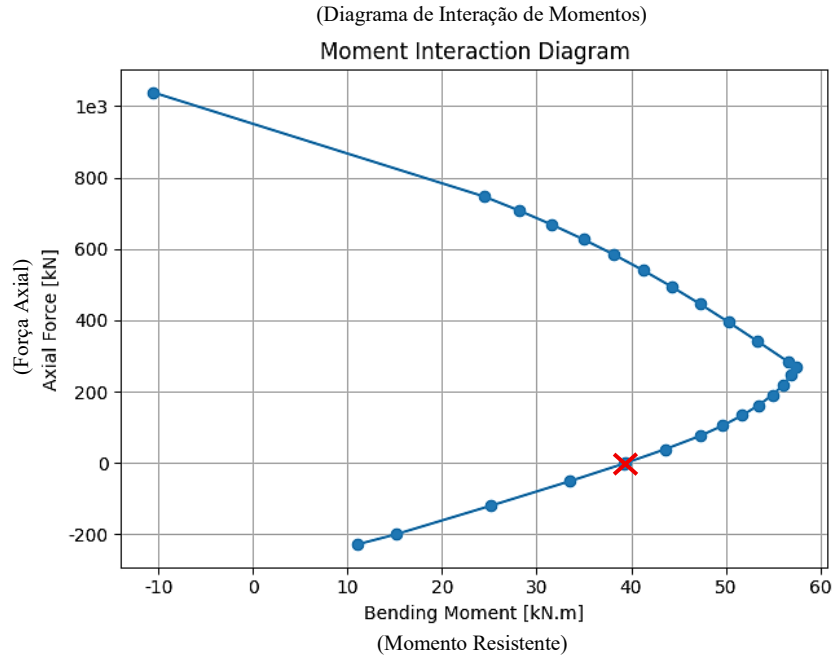
Além dos momentos resistentes obtidos para as combinações específicas analisadas, a biblioteca *Concreteproperties* permite gerar o diagrama de interação entre força normal e momento fletor da seção. Esse diagrama representa os pares $(N; M)$ associados à capacidade resistente última da seção para uma determinada direção de flexão, permitindo avaliar a variação do momento resistente em função da força normal aplicada.

A Figura 28 apresenta o diagrama de interação $N \times M$ para a seção retangular analisada anteriormente na Seção 6.5.2, com dimensões de 15 cm $\times$ 30 cm, concreto C30 e aço CA-50, obtido por meio do método *moment_interaction_diagram()* da biblioteca. Para efeito demonstrativo, foi considerada a armadura dupla adotada anteriormente, composta por 3 ϕ 12,5 mm na face inferior e 2 ϕ 10 mm na face superior. O diagrama indica, para cada nível de força normal, o momento fletor resistente correspondente. O ponto marcado no diagrama corresponde à combinação sem força axial, que é a situação mais crítica para a seção.

Para a combinação sem força axial ($N = 0$), o diagrama fornece o momento resistente da seção à flexão pura. Quando se considera uma força axial de compressão de 50 kN, observa-se a alteração do momento resistente, permitindo comparar numericamente o efeito da

compressão normal sobre a capacidade resistente da seção. Portanto, o diagrama permite verificar diretamente se o par solicitante (N_{sd} ; M_{sd}) está contido no domínio resistente da seção.

Figura 28 - Diagrama de interação momento-força normal



Fonte: Adaptado de Robbins (2023).

Com base nesse exemplo preliminar, observa-se que a biblioteca *Concreteproperties* é adequada para verificar a capacidade resistente de seções de concreto armado previamente definidas. Entretanto, como a biblioteca não determina automaticamente a armadura necessária, torna-se necessário desenvolver um procedimento próprio de dimensionamento. Esse procedimento é apresentado no Capítulo 0, no qual a armadura longitudinal da escada plissada é calculada a partir do momento fletor solicitante obtido na análise estrutural e, posteriormente, verificada com auxílio da biblioteca *Concreteproperties*.

7 DIMENSIONAMENTO DA ARMADURA LONGITUDINAL DA ESCADA PLISSADA

7.1 Algoritmo de dimensionamento iterativo

O dimensionamento da armadura longitudinal da escada plissada foi realizado por meio de um algoritmo iterativo desenvolvido em Python. O procedimento utiliza como base as equações de equilíbrio da seção retangular no Estado Limite Último, apresentadas na Seção 4.3, e tem como finalidade determinar combinações de barras comerciais capazes de atender à área de aço necessária à flexão.

Inicialmente, são fornecidos ao algoritmo os parâmetros geométricos e mecânicos da seção analisada, incluindo a largura da seção b_w , a altura útil d , a resistência característica do concreto f_{ck} , a resistência característica do aço f_{yk} , o momento fletor solicitante de cálculo M_{Sd} e a lista de bitolas comerciais disponíveis. Com esses dados, o programa calcula as resistências de cálculo do concreto e do aço, f_{cd} e f_{yd} , utilizando os coeficientes de ponderação dos materiais.

Em seguida, o algoritmo resolve a equação de equilíbrio de momentos para determinar a profundidade da linha neutra x . Para isso, a equação é organizada na forma de uma equação quadrática, e são calculadas suas raízes reais. Entre as raízes obtidas, adota-se aquela que possui significado físico, isto é, a raiz positiva e compatível com a altura útil da seção.

Após a determinação da profundidade da linha neutra, calcula-se a área de aço necessária por meio do equilíbrio entre a força resultante de compressão no concreto e a força resultante de tração na armadura. O valor obtido é então comparado com os limites de armadura mínima e máxima. Caso a área calculada seja inferior à armadura mínima, o algoritmo adota a área mínima exigida. Caso a área necessária ultrapasse a armadura máxima, a seção é considerada inviável para as condições analisadas, indicando a necessidade de alteração da geometria da seção, das propriedades dos materiais ou da concepção estrutural.

O algoritmo também verifica o domínio de deformação da seção por meio da relação entre a profundidade da linha neutra e a altura útil. Neste trabalho, considera-se adequada a solução em que a linha neutra permanece nos domínios associados ao comportamento dúctil. Caso a linha neutra ultrapasse o limite de ductilidade adotado, a seção é classificada como superarmada, não sendo considerada adequada para o dimensionamento proposto.

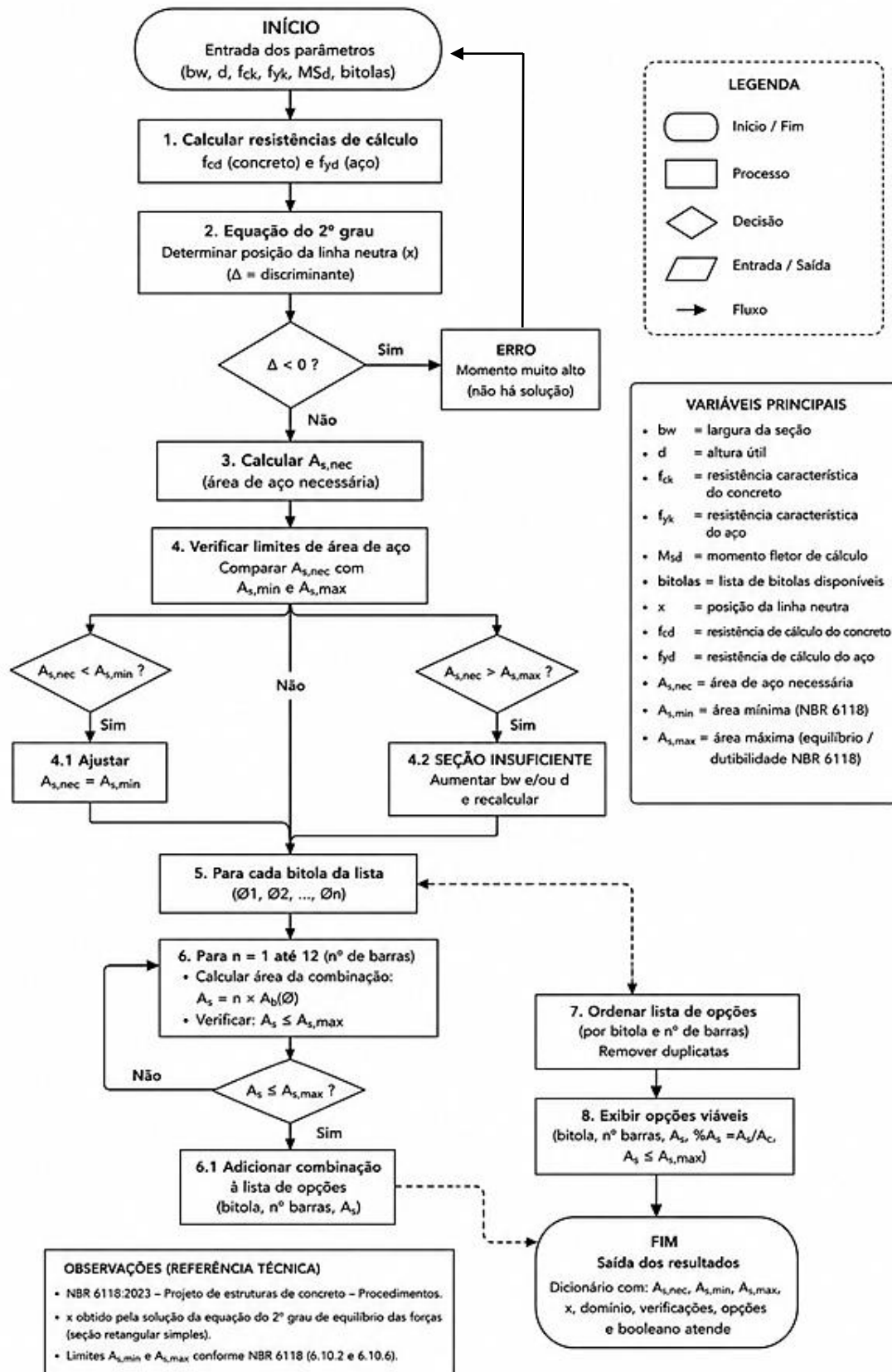
Após a definição da área de aço necessária, o algoritmo percorre as bitolas comerciais disponíveis e calcula a área individual de cada barra. Em seguida, determina o número mínimo

de barras necessário para que a área efetiva seja igual ou superior à área requerida. Para evitar soluções com baixa viabilidade construtiva, foi adotado um limite prático de até 12 barras por face.

As combinações que atendem à área de aço necessária e não ultrapassam a armadura máxima são armazenadas em uma lista de soluções viáveis. Cada solução contém a bitola adotada, o número de barras e a área de aço efetiva correspondente. Ao final, o algoritmo retorna as opções disponíveis, permitindo a escolha da configuração mais adequada em função dos critérios de segurança, execução e espaçamento.

A sequência lógica do procedimento iterativo de dimensionamento é apresentada na Figura 29, desde a entrada dos parâmetros geométricos e mecânicos até a geração das combinações viáveis de armadura.

Figura 29 - Fluxograma do procedimento iterativo para dimensionamento da armadura longitudinal à flexão



Fonte: Autoria própria (2026).

Dessa forma, o procedimento implementado não se limita ao cálculo da área teórica de aço, mas também transforma esse resultado em alternativas práticas de detalhamento com barras comerciais. Posteriormente, a armadura escolhida é verificada com auxílio da biblioteca *Concreteproperties*, por meio da comparação entre o momento resistente de cálculo M_{Rd} e o momento solicitante de cálculo M_{Sd} .

7.2 Implementação em Python

A função *dimensionar_armadura* foi implementada em Python. Os parâmetros de entrada e saída são descritos nos Quadro 1 e Quadro 2.

Quadro 1 - Parâmetros de entrada da função *dimensionar_armadura*

Parâmetro	Descrição
b_w	Largura da seção transversal (cm)
d	Altura útil (cm)
f_{ck}	Resistência característica do concreto (MPa)
f_{yk}	Resistência característica do aço (MPa)
M_{Sd}	Momento fletor solicitante de cálculo (kN·cm)
$\emptyset$	Lista opcional com os diâmetros nominais das barras a considerar (mm). Por padrão, utiliza [6.3, 8.0, 10.0]

Fonte: Autoria Própria (2026).

Quadro 2 - Parâmetros de saída da função *dimensionar_armadura*

Parâmetro	Descrição
A_{s_nec}	Área de aço necessária (cm ²) - já ajustada à armadura mínima se necessário
A_{s_min}	Armadura mínima exigida pela NBR 6118:2023 (cm ²)
A_{s_max}	Armadura máxima permitida (4% da área bruta) (cm ²)
x	Profundidade da linha neutra (cm)
$dominio$	Domínio de deformação conforme a posição da linha neutra
$verificacoes$	Lista de mensagens textuais indicando o atendimento dos limites normativos
$opcoes$	Lista de dicionários, cada um contendo $bitola_mm$, n_barras e A_{s_ef} (cm ²), representando todas as combinações viáveis de barras
$atende$	Valor booleano: <i>True</i> se a seção for viável (ductilidade ok e $A_{s_nec} \leq A_{s_max}$), <i>False</i> caso contrário

Fonte: Autoria Própria (2026).

A seguir, discutem-se os resultados obtidos para a escada de referência.

7.3 Exemplo de aplicação para a escada analisada

Aplicou-se a função *dimensionar_armadura* à escada de referência, cujos parâmetros geométricos e mecânicos foram estabelecidos na Seção 5.1. A seção transversal do degrau apresenta largura de 120 cm e altura total de 12 cm, resultando em uma altura útil $d = 8,5$ cm (descontando-se o cobrimento de 3 cm e a metade do diâmetro da barra de 10 mm). O concreto adotado é da classe C25 ($f_{ck} = 25$ MPa), e o aço é CA-50 ($f_{yk} = 500$ MPa). O momento fletor característico máximo obtido da análise estrutural foi $M_k = 1529,09$ kN.cm, que, majorado pelo coeficiente de ponderação $\gamma_f = 1,4$, resultou em $M_{Sd} = 2141$ kN.cm.

O programa foi executado considerando as bitolas de 6,3 mm, 8,0 mm e 10,0 mm. As únicas combinações viáveis geradas foram com a bitola de 10 mm, com números de barras variando de 9 a 12 (áreas efetivas entre 7,07 e 9,42 cm²). Não surgiram opções com 6,3 mm ou 8,0 mm porque, para a área de aço necessária (aproximadamente 6,9 cm²), o número de barras necessário seria, respectivamente, 21 e 13 barras – valores que superam o limite prático de 12 barras adotado no algoritmo para garantir espaçamento mínimo e exequibilidade construtiva. A Tabela 2 apresenta a saída do programa com a listagem dessas opções, dispensando-se uma tabela textual.

Tabela 2 - Opções de armadura geradas pelo dimensionamento

Bitola (mm)	Número de barras	Área efetiva (cm²)
10	9	7,07
10	10	7,85
10	11	8,64
10	12	9,42

Fonte: Autoria Própria (2026).

Em escadas plissadas, a armadura longitudinal é constituída por estribos fechados que envolvem toda a seção transversal, exigindo uma armadura dupla e simétrica (mesma área nas faces inferior e superior). Por essa razão, a escolha recaiu sobre a combinação de 9 barras de 10 mm em ambas as faces. Essa configuração fornece uma área efetiva de 7,07 cm² em cada face, superior à área necessária, e respeita amplamente os limites normativos: a armadura mínima exigida é de 2,43 cm², e a máxima é de 64,8 cm². A verificação complementar dessa armadura simétrica, por meio da biblioteca *Concreteproperties*, é apresentada na seção 7.4.

7.4 Verificação da armadura dimensionada com a biblioteca *Concreteproperties*

Para validar o dimensionamento realizado, utilizou-se a biblioteca *Concreteproperties* (apresentada no Capítulo 6) como ferramenta de verificação. A seção do degrau da escada de referência foi modelada com as seguintes características:

- Largura: 120 cm
- Espessura (altura): 12 cm
- Concreto: C25 ($f_{ck} = 25 \text{ MPa}$)
- Aço: CA-50 ($f_{yk} = 500 \text{ MPa}$)
- Armadura dupla simétrica: 9 barras de 10 mm (área efetiva = 7,07 cm²) em cada face, conforme opção escolhida entre as geradas pelo algoritmo iterativo.

O momento fletor resistente de cálculo (M_{Rd}) foi obtido por meio do método *ultimate_bending_capacity()*, resultando em $M_{Rd} = 25,9 \text{ kN}\cdot\text{m}$ para a combinação sem força axial ($N=0$), que é a mais crítica. A Tabela 3 compara o momento solicitante com o momento resistente.

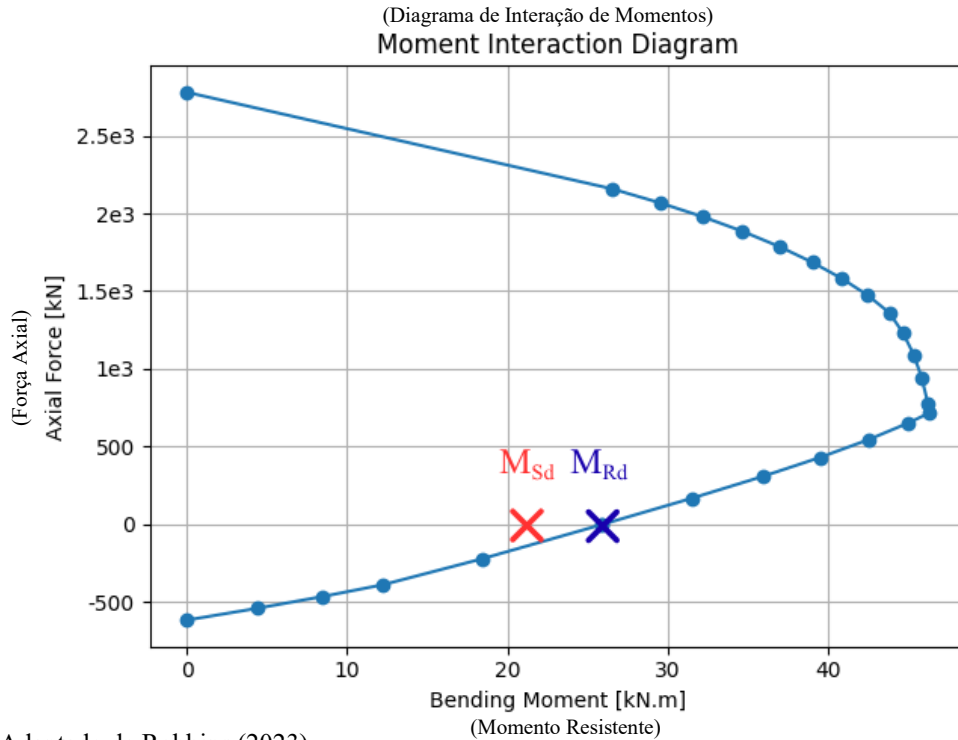
Tabela 3 - Comparação entre momento solicitante (M_{Sd}) e momento resistente (M_{Rd})

Parâmetro	Valor (kN·m)
M_{Sd}	21,41
M_{Rd}	25,90

Fonte: Autoria Própria (2026).

Observa-se que $M_{Rd} > M_{Sd}$ (aproximadamente 21% superior), confirmando que a armadura dupla simétrica é suficiente para garantir a segurança da seção no Estado Limite Último. A Figura 30 apresenta o diagrama de interação momento–força normal da seção dimensionada, com a marcação dos pontos correspondentes ao momento solicitante de cálculo M_{Sd} e ao momento resistente de cálculo M_{Rd} . Observa-se que mesmo na situação de flexão pura, a capacidade resistente da seção é superior à solicitação atuante.

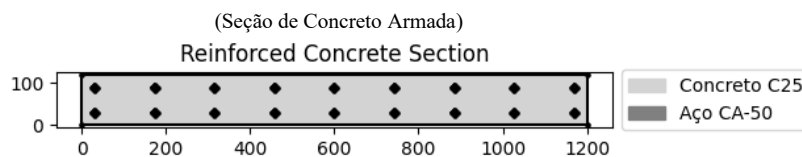
Figura 30 - Diagrama de interação momento-força normal para a seção do degrau adotado



Fonte: Adaptado de Robbins (2023).

Dessa forma, o *Concreteproperties* mostra-se uma ferramenta eficaz e confiável para a verificação de seções de concreto armado, podendo ser utilizada como validação complementar aos métodos analíticos de dimensionamento. A Figura 31 exibe a representação gráfica da seção armada com as 9 barras de 10 mm (gerada pelo método *plot_section()*).

Figura 31 - Seção de concreto armado com 9Ø10



Fonte: Gerado pelo Concreteproperties (ROBBINS, 2023) no ambiente Google Colab (2026).

Como referência externa, os resultados obtidos podem ser comparados qualitativamente ao estudo experimental de Özbek et al. (2021), que avaliaram o comportamento à flexão de escadas de concreto armado sem laje de fundo contínua (*slabless reinforced concrete staircases*). Os autores ensaiaram modelos reduzidos com diferentes configurações de armadura e diferentes espessuras de piso e espelho, observando que a espessura do piso exerce influência significativa sobre a capacidade resistente e o comportamento à flexão da escada. Esse resultado é compatível com a abordagem adotada neste trabalho, uma vez que a verificação resistente foi realizada a partir da seção transversal do degrau submetida ao maior momento fletor solicitante obtido na análise global. Assim, embora não se trate de uma validação experimental direta, a

comparação com a literatura reforça a coerência da metodologia adotada para verificar a segurança à flexão da seção dimensionada.

8 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo principal o desenvolvimento e a documentação de uma ferramenta computacional em Python para a análise estrutural e o dimensionamento à flexão de escadas plissadas. A pesquisa partiu da premissa da ausência de recomendações normativas específicas para este tipo estrutural, que frequentemente leva a projetos baseados em analogias imprecisas com lajes ou vigas. Para atender a esse objetivo, foi implementada a classe *EscadaPlissada*, que automatiza a geração da geometria, o cálculo de cargas, a análise matricial de pórticos planos via *Anastruct* e a extração de momentos fletores máximos. A validação qualitativa da classe foi realizada com sucesso para uma escada de referência, demonstrando coerência na geometria gerada e nos diagramas de esforços.

Adicionalmente, desenvolveu-se uma função iterativa para o dimensionamento da armadura longitudinal no Estado Limite Último (ELU), que, aplicada à escada de referência, resultou na escolha de uma armadura dupla simétrica de 9 $\phi 10$ mm. A biblioteca *Concreteproperties* foi empregada como ferramenta de verificação, confirmando a segurança da seção dimensionada, uma vez que seu momento resistente ($M_{Rd} = 2590 \text{ kN}\cdot\text{cm}$) superou o momento solicitante de cálculo ($M_{Sd} = 2141 \text{ kN}\cdot\text{cm}$). Essa integração entre o dimensionamento analítico iterativo e a verificação numérica demonstra a aplicabilidade da metodologia proposta.

É importante destacar que a ferramenta desenvolvida se apresenta como uma solução de código aberto e de fácil acesso, contribuindo tanto para a prática profissional de engenheiros quanto para o ensino da análise estrutural computacional. No entanto, a versão atual apresenta limitações, pois considera a análise global no regime elástico linear (Estádio I). Esta simplificação, embora prática, não captura a perda de rigidez causada pela fissuração do concreto e ignora os efeitos diferidos da fluência e retração.

Como sugestões para trabalhos futuros, recomenda-se a implementação de uma análise incremental não linear, acoplando o *Anastruct* ao *Concreteproperties* para atualizar a rigidez dos elementos a cada passo de carga. Outra importante linha de pesquisa é a incorporação de um módulo para análise de vibrações, fundamental para garantir o conforto humano, um fator crítico para estruturas esbeltas como as escadas plissadas. Por fim, a validação da ferramenta com dados experimentais de ensaios laboratoriais, como os de Ozbek et al. (2021) e Simões (2008), permanece como um passo essencial para a consolidação da ferramenta. Acredita-se que, com tais extensões, a ferramenta poderá se tornar uma referência para o projeto seguro e econômico de escadas plissadas.

REFERÊNCIAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6118:2026**: projeto de estruturas de concreto — procedimento. Rio de Janeiro: ABNT, 2026.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 8681:2025**: ações e segurança nas estruturas — procedimento. Rio de Janeiro: ABNT, 2025.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 9050:2020**: **Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos**. Rio de Janeiro: ABNT, 2020. Versão corrigida: 2021.
- ANSYS, INC. **Ansys® Academic Research Mechanical**, Release 18.1. [Software]. Canonsburg, PA: ANSYS, Inc., 2017.
- ARAÚJO, J. M. Processos simplificados para cálculo de flechas de vigas de concreto armado. **Revista Engenharia, Ciência e Tecnologia**, Vitória, v. 6, n. 2, p. 3-13, 2003.
- BASTOS, Paulo Sérgio. **Flexão normal simples: vigas**. Bauru: Universidade Estadual Paulista, Departamento de Engenharia Civil e Ambiental, 2023. Notas de aula da disciplina Estruturas de Concreto I. Disponível em: <https://wwwp.feb.unesp.br/pbastos>. Acesso em: 18 jun. 2026.
- EUROPEAN COMMITTEE FOR STANDARDIZATION — CEN. **EN 1992-1-1**: Eurocode 2 — Design of concrete structures — Part 1-1: General rules and rules for buildings, bridges and civil engineering structures. Brussels: CEN, 2023.
- COMPUTERS AND STRUCTURES, INC. **SAP2000: Integrated Software for Structural Analysis and Design**, Version 14. [Software]. Berkeley: Computers and Structures, Inc., 2011.
- DASSAULT SYSTÈMES. **Abaqus/CAE Standard**, Version 6.14. [Software]. Providence: Dassault Systèmes, 2014.
- GHALI, A.; FAVRE, R.; ELBADRY, M. **Concrete structures: stresses and deformations**. 3. ed. London: Spon Press, 2002.
- GOOGLE. **Google Colaboratory**. [Software]. 2026. Disponível em: <https://colab.research.google.com/>. Acesso em: 10 abr. 2026.
- MALYSZ, S. T. **Análise de escadas usuais de edifícios em concreto armado**. 2007. Dissertação (Mestrado em Engenharia Civil) — Universidade Federal de Santa Catarina, Florianópolis, 2007.
- ÖZBEK, E. et al. Flexural behavior of slabless reinforced concrete staircases. **ACI Structural Journal**, v. 118, n. 6, 2021. DOI: 10.14359/51732849.
- OYAKOJO, E. O. Numerical Investigations on Folded Plate Staircases. **Open Journal of Civil Engineering**, v. 15, n. 1, p. 45-58, 2025.

PYTHON SOFTWARE FOUNDATION. **Python Language Reference**, Version 3.12. [Software]. 2024. Disponível em: <https://www.python.org/>. Acesso em: 10 abr. 2026.

ROBBINS, C. **concreteproperties: A Python Library for Reinforced Concrete Section Analysis**, Version 0.1.2. [Software]. 2023. Disponível em: <https://pypi.org/project/concreteproperties/>. Acesso em: 10 abr. 2026.

SABEEH, R. S.; JAAFER, A. A. Slabless Staircase Flexural Behavior with Multi Reinforcement Configuration. **Misan Journal of Engineering Sciences**, v. 3, n. 2, p. 167-181, 2024.

SEBASTIAN, K. et al. Application of AI for modelling and structural analysis of a parametric 2D frame with voice assistant. **E3S Web of Conferences**, v. 586, 02003, 2024. DOI: 10.1051/e3sconf/202458602003.

SIMÕES, M. L. F. **Análise numérica e experimental de escadas autoportantes de concreto armado**. 2008. Dissertação (Mestrado em Estruturas e Construção Civil) — Universidade de Brasília, Brasília, 2008.

TRAUTWEIN, L. M. et al. Análise estrutural de escadas autoportantes: recomendações de projeto. **Revista IBRACON de Estruturas e Materiais**, v. 3, n. 4, p. 396-415, 2010.

VAN DER MEER, R. **anaStruct: 2D structural analysis in Python**, Version 1.0.5. [Software]. 2022. Disponível em: <https://pypi.org/project/anastruct/>. Acesso em: 10 abr. 2026.

VAN LEEUWEN, R.; FERSTER, C. sectionproperties: a Python package for the analysis of arbitrary cross-sections using the finite element method. **Journal of Open Source Software**, v. 9, n. 96, 6105, 2024. DOI: 10.21105/joss.06105.

WAY, A. et al. Target reliability for serviceability limit states. **Structural Safety**, v. 114, 102572, 2025.