



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA AGROALIMENTAR  
UNIDADE ACADÊMICA DE CIÊNCIAS E TECNOLOGIA AMBIENTAL  
CURSO DE GRADUAÇÃO EM ENGENHARIA CIVIL**

**FERRAMENTA COMPUTACIONAL PARA DIMENSIONAMENTO,  
VERIFICAÇÃO, DETALHAMENTO E GERAÇÃO DE QUANTITATIVOS  
DE VIGAS DE CONCRETO ARMADO**

**DIEGO ARMANDO DE SOUSA CARNEIRO**

**POMBAL – PB**

**2023**

DIEGO ARMANDO DE SOUSA CARNEIRO

FERRAMENTA COMPUTACIONAL PARA DIMENSIONAMENTO,  
VERIFICAÇÃO, DETALHAMENTO E GERAÇÃO DE QUANTITATIVOS  
DE VIGAS DE CONCRETO ARMADO

Trabalho de Conclusão de Curso  
apresentado à Unidade Acadêmica de  
Ciências e Tecnologia Ambiental da  
Universidade Federal de Campina Grande,  
como parte dos requisitos necessários  
para obtenção do título de Engenheiro  
Civil.

Orientador(a): Prof.(a) Eric Mateus  
Fernandes Bezerra

POMBAL – PB

2023

C289f Carneiro, Diego Armando de Sousa.  
Ferramenta computacional para dimensionamento, verificação, detalhamento e  
geração de quantitativos de vigas de concreto armado / Diego Armando de  
Sousa Carneiro. – Pombal, 2023.  
23 f. il. color.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) –  
Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia  
Agroalimentar, 2023.

“Orientação: Prof. Me. Eric Mateus Fernandes Bezerra”.  
Referências.

1. Concreto armado. 2. Dimensionamento de estruturas. 3. Ferramenta  
computacional - Linguagem python. I. Bezerra, Eric Mateus Fernandes. II. Título.

CDU 691.328 (043)

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA AGROALIMENTAR  
UNIDADE ACADÊMICA DE CIÊNCIAS E TECNOLOGIA AMBIENTAL  
CURSO DE GRADUAÇÃO EM ENGENHARIA CIVIL

PARECER DA COMISSÃO EXAMINADORA DE TRABALHO DE CONCLUSÃO DE CURSO.

DIEGO ARMANDO DE SOUSA CARNEIRO

**FERRAMENTA COMPUTACIONAL PARA DIMENSIONAMENTO,  
VERIFICAÇÃO, DETALHAMENTO E GERAÇÃO DE QUANTITATIVOS  
DE VIGAS DE CONCRETO ARMADO**

Trabalho de Conclusão de Curso do discente Diego Armando de Sousa Carneiro **APROVADO** em 10 de fevereiro de 2023 pela comissão examinadora composta pelos membros abaixo relacionados como requisito para obtenção do título de ENGENHEIRO CIVIL pela Universidade Federal de Campina Grande

Registre-se e publique-se.



---

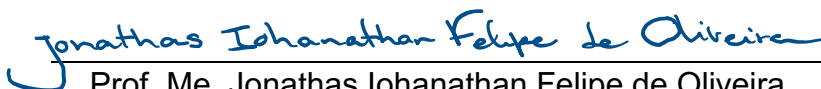
Prof. Me. Eric Mateus Fernandes Bezerra  
(Orientador – UFCG)



Assinado digitalmente por LEOVEGILDO DOUGLAS PEREIRA DE SOUZA:  
DN: cn=LEOVEGILDO DOUGLAS PEREIRA DE SOUZA:08419377457,  
ou=UFCG - Universidade Federal de Campina Grande, o=ICPEdu, c=BR  
SuaZiG: Eu sou o autor deste documento  
Localização: Pombal - PB  
Data: 2023.02.16 20:56:56-03'00"  
Foxit PDF Reader Versão: 11.2.1

---

Prof. Dr. Leovegildo Douglas Pereira de Souza  
(Membro Interno – UFCG)



---

Prof. Me. Jonathas Iohanathan Felipe de Oliveira  
(Membro Externo – IFCE)

## **AGRADECIMENTOS**

Quero agradecer ao corpo docente da UFCG que foi fundamental para integrar conhecimento a mim e me capacitar tanto com conhecimento teórico quanto prático, agradeço também ao professor orientador Eric Mateus Fernandes Bezerra, que me ajudou tanto na confecção deste trabalho, quanto na formação como profissional da engenharia civil.

## Ferramenta computacional para o dimensionamento, verificação, detalhamento e geração de quantitativos de vigas de concreto armado

*Computational tool for design, verification, detailing and generating quantities of reinforced concrete beams*

Diego Armando de Sousa Carneiro, diego.armando@estudante.ufcg.edu.br

Eric Mateus Fernandes Bezerra, e-mail: eric.mateus@professor.ufcg.edu.br

---

### RESUMO

Grande parte das edificações demandam projetos para serem concebidas, dentre eles, o projeto estrutural, que tem como etapas a concepção, a análise, o dimensionamento e detalhamento e a emissão de desenhos e documentos. Diversos sistemas em concreto armado são criados a partir de objetos que podem ser analisados como barras, tais como as vigas que são elementos lineares submetidos a momento fletor, torsor e esforço cortante. O dimensionamento desses componentes se torna menos produtivo e mais susceptível a erros se feito manualmente, por isso a utilização de ferramentas computacionais para o auxílio no cálculo de projetos estruturais é recomendável, visto que, se utilizadas corretamente, essas ferramentas promovem produtividade. Sendo assim, o objetivo do presente trabalho é desenvolver uma ferramenta computacional para o dimensionamento à flexão e ao cisalhamento, verificação do estado limite de deformações excessivas e geração do detalhamento em arquivo compatível com o Autodesk AutoCAD de vigas de concreto armado de seção retangular submetidas a flexão e ao cisalhamento utilizando a linguagem de programação Python que é baseada principalmente nos paradigmas Funcional e Orientação a Objetos. Os resultados mostraram que a ferramenta é aplicável tanto para cálculos rápidos, quanto para o dimensionamento das vigas de uma estrutura de concreto armado, assim como, é susceptível à utilização para auxiliar docentes em atividades acadêmicas.

**Palavras-chave:** Concreto Armado, software, dimensionamento, detalhamento, Python,

### ABSTRACT

Much of the buildings require projects to be conceived, among them, the structural project, which has the stages of conception, analysis, dimensioning and detailing and the issuance of drawings and documents. Most systems in reinforced concrete are created from objects that can be analyzed as bars, such as beams, which are linear elements subjected to bending moments, torsion and shear forces. The structural design of these components becomes less productive and more susceptible to errors if done manually, so the use of computational tools to aid in the calculation of structural projects is recommended, since, if used correctly, these tools promote productivity. Therefore, the objective of the present work is to develop a computational tool for bending and shear design, verification of the limit state of excessive deformations and generation of the detailing in a file that is compatible with Autodesk AutoCAD of reinforced concrete beams of rectangular section subjected to bending and shearing stress using the Python programming language, which is based mainly on the Functional and Object Oriented paradigms. The results showed that the tool is applicable both for quick calculations and for the dimensioning of the beams of a reinforced concrete structure, as well as being susceptible to use for assist teachers in academic activities.

**Keywords:** Reinforced concrete, software, desing, detailing, Python.

## 1. INTRODUÇÃO

É fato que grande parte dos empreendimentos da engenharia civil devem ser executados a partir de projetos específicos previamente confeccionados, dentre estes, podemos citar os projetos arquitetônico, hidrossanitário, elétrico, combate a incêndio e, como enfoque deste trabalho, o estrutural. Este último, em grande parte das obras, é fundamental, visto que todos os outros subsequentes em uma edificação dependem da execução da estrutura.

É comum dividir o projeto estrutural em quatro etapas, são elas, concepção, análise, dimensionamento detalhamento e emissão de pranchas de uma estrutura (KIMURA, 2018). A primeira está relacionada com a decisão preliminar sobre qual sistema estrutural deve ser escolhido para constituir o empreendimento de modo a fazer uma idealização do comportamento da estrutura real (MARTHA, 2017, p. 3). A segunda é voltada para a análise de esforços internos e deformações nos elementos estruturais. De acordo com Martha (2017), a análise estrutural “é a etapa do projeto estrutural na qual é realizada uma previsão do comportamento da estrutura. Nela são utilizadas todas as teorias físicas e matemáticas resultantes da formalização da engenharia estrutural como ciência”. O terceiro passo é o dimensionamento e o detalhamento de cada elemento estrutural com base em normas técnicas vigentes. Por fim, são emitidas pranchas, que é o produto final projeto, onde são impressos desenhos contendo todo o detalhamento estrutura, especificação de montagem e avisos.

Para o estado em que a sociedade atualmente se encontra no quesito tecnologia digital, faz-se necessário, à medida do possível, desenvolver ferramentas que automatizem tarefas, principalmente na engenharia, pois “O advento computacional e a sua popularização justificaram o desenvolvimento de *softwares* para auxílio no projeto, em especial o de estruturas” (MONTEIRO et al, 2020). Diante do cenário marcado por demandas cada vez mais urgentes de velocidade de projeto, se torna ineficiente desenvolver projetos de engenharia sem o auxílio de um *software* ou rotina específica. (SOUZA et al, 2019). Isso se aplica ao projeto estrutural de vigas, haja vista que o dimensionamento e o detalhamento desse tipo de elemento se tornam dispendiosos e, por essa razão, podem potencializar falhas humanas. A utilização consciente de computadores pode melhorar a eficiência do dimensionamento de estruturas, pois, o tempo despendido com a realização de operações matemáticas repetitivas pode ser destinado ao teste de diferentes situações de projeto para obter a melhor situação possível para o problema (LOPES, CHAVES e BEZERRA, 2018). Diversas vezes o prazo de entrega de um projeto é o fator decisivo para um profissional aceitar ou rejeitar uma proposta.

Existem no mercado diversas ferramentas computacionais para auxiliar no projeto de estruturas de concreto, tais como o TQS® e o EBERICK®. Entretanto, esses *softwares* são distribuídos em conjuntos mais completos, tornando o custo de aquisição elevado, e, muitas vezes, injustificáveis. Além disso, “algumas aplicações corriqueiras não justificam a concepção de um modelo estrutural tão sofisticado” (LIMA et al, 2020, p.3).

Portanto, é justificável a criação de ferramentas computacionais que resolvam problemas isolados mais simples e que, por essa razão, possam ser úteis tanto para verificações rápidas quanto para dimensionamento mais elaborados, as ferramentas sofisticadas que existem no mercado hoje partiram de *softwares* básicos para elementos isolados, por exemplo o Pro-Vigas e o Pro-Pilar que evoluíram para o Eberick. O TQS utiliza atualmente várias ferramentas separadas destinadas a funções específicas, por exemplo, o TQS-Vigas, TQS-Lajes, entre outros. A ferramenta desenvolvida nesse trabalho. Destaca-se também o potencial de emprego dessas ferramentas no processo de ensino aprendizagem do comportamento de estruturas.

Diante do exposto, esse trabalho tem como objetivo geral mostrar a criação de uma ferramenta computacional desenvolvida na linguagem de programação Python capaz de dimensionar, detalhar e exportar os quantitativos de vigas de concreto armado. A linguagem Python foi escolhida por ser gratuita, de fácil instalação e possuir um campo vasto de bibliotecas para soluções de problemas gráficos e matemáticos.

## 2. FUNDAMENTAÇÃO TEÓRICA

As vigas são elementos responsáveis por suportar esforços aplicados por lajes, paredes, entre outros, para o modelo de viga abordado nesse trabalho, as vigas são tratadas como elementos lineares, que são “elementos com uma de suas dimensões significativamente superior às demais” (LIMA et al, 2020, p.1).

O dimensionamento de uma viga decorre de uma série de etapas, a princípio, devem ser conhecidas as solicitações de projeto aplicadas sobre a estrutura, com isso, deve-se analisar os diagramas de momento fletor, esforço cortante e momento de torção a fim de encontrar as seções críticas em que se deve direcionar o dimensionamento. Para este caso, a ferramenta é limitada a vigas que não estão submetidas a esforço de torção e para concreto com o  $f_{ck}$  menor que 50MPa. É possível admitir que o modelo de viga utilizado é baseado no conceito da viga de Euler, em que a “seção transversal ao eixo do elemento estrutural indeformado, que inicialmente era plana e normal a esse eixo, permanece nessa condição após as deformações do elemento” (ARAÚJO, 2014).

Para a flexão e o cisalhamento, as combinações utilizadas são as normais do Estado Limite Último, e para o Estado Limite de Serviço, são utilizadas as combinações quase permanentes.

### 2.1. Dimensionamento à flexão

Para prosseguir, é de fundamental entender o que a NBR 6118 trata como domínios de dimensionamento de uma estrutura submetida a um momento fletor. Na Figura 1 estão dispostos esses domínios.

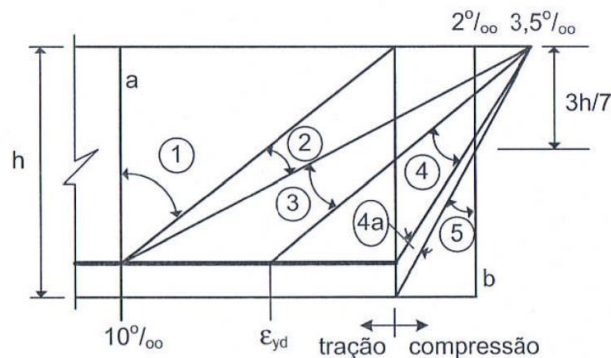


Figura 1 - Domínios de dimensionamento do concreto submetido à flexão. (Fonte: ARAÚJO, 2014).

- O domínio 1 corresponde à falha por deformação máxima por tração de 1% da armadura longitudinal e não existe compressão na peça;
- No domínio 2, a armadura está sempre submetida à deformação por tração de 1% e o concreto pode deformar em compressão entre zero e 0,35‰;
- No domínio 3, tanto a armadura longitudinal, quanto o concreto trabalham no limite, com uma deformação de 1% em tração e 0,35‰ em compressão respectivamente;
- No domínio 4, o concreto trabalha na deformação máxima de compressão (0,35‰) e o aço não chega a escoar.
- Para o restante dos domínios, a peça trabalha sob flexo compressão ou compressão.



Dentre estes domínios, o que é recomendado para dimensionar peças submetidas à flexão é o 3, pois, obtém-se a maior eficiência possível dos materiais.

Nos domínios 2, 3 e 4, devido aos esforços de tração e compressão que equilibram a seção transversal, forma-se um binário de forças que resulta em um momento equivalente ao módulo de  $M_{sd}$ , desta forma, a ideia do dimensionamento da flexão em vigas é que a seção transversal tenha aço suficiente para resistir à tração e concreto e aço suficientes para resistir à compressão, de modo que o binário de esforços resistentes seja capaz de anular o binário de esforços solicitantes.

A Figura 2 abaixo mostra nomenclatura de elementos da seção transversal de uma viga de concreto.

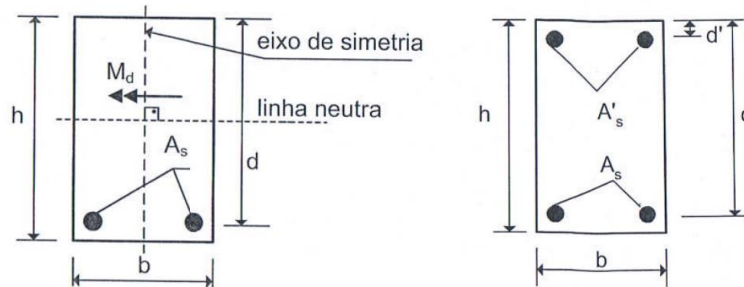


Figura 2 - Disposições dos elementos da seção transversal de uma viga de concreto armado. (Fonte: ARAÚJO, 2014).

Em que:

- $h$  é altura total da viga;
- $b$  ou  $b_w$  é a largura;
- $d$  é a altura útil;
- $A_s$  é a área de aço de armadura positiva;
- $M_d$  ou  $M_{sd}$  é o momento fletor de projeto;
- $A'_s$  é a área de aço para a armadura negativa;
- $d'$  é a distância entre o topo da seção e o centroide da armadura negativa.

No Quadro 1 estão dispostas formulações matemáticas utilizadas para o dimensionamento à flexão de uma viga de concreto armado.

O momento fletor reduzido ( $\mu$ ) é o $M_{sd}$ adimensional	$\mu = \frac{M_{sd}}{bd^2 0,85 f_{cd}}$	(1)
O $\xi$ é a posição da linha neutra adimensionalizada.	$\xi = 1,25(1 - \sqrt{1 - 2\mu})$	(2)
Área de aço ( $A_s$ ) para $\xi \leq 0,45$	$A_s = 0,8\xi bd 0,85 \frac{f_{cd}}{f_{yd}}$	(3)
Parâmetro geométrico ( $\delta$ )	$\delta = \frac{d'}{d}$	(4)
Tensão na armadura de compressão ( $\sigma'_{sd}$ )	$\sigma'_{sd} = E_s \frac{3,5}{1000} \left( \frac{0,45 - \delta}{0,45} \right) \leq f_{yd}$	(5)
Área de aço da armadura dupla ( $A'_s$ )	$A'_s = \frac{(\mu - \mu_{lim}) bd 0,85 f_{cd}}{(1 - \delta) \sigma'_{sd}}$	(6)
Área de aço positiva para $\xi > 0,45$	$A_s = \left( 0,8 \cdot 0,45 + \frac{\mu - \mu_{lim}}{1 - \delta} \right) \cdot \frac{bd \cdot 0,85 \cdot f_{cd}}{f_{yd}}$	(7)

Quadro 1 - Equações necessárias para o dimensionamento à flexão de vigas de seção retangular. (Fonte: Adaptado de ARAÚJO, 2014).

Onde,  $f_{yd}$  e  $f_{cd}$  são, respectivamente a tensão de escoamento do aço e tensão de compressão do concreto de projeto,  $\mu_{lim}$  é o momento fletor limite adimensionalizado para uma região segura do domínio 3, para valores de  $\mu$  maiores que  $\mu_{lim}$  torna-se necessária a utilização de armadura dupla para flexão e  $E_s$  é o módulo de elasticidade do aço.

A princípio, é necessário obter o  $M_{sd}$  a partir de combinações para o Estado Limite Último, em seguida, deve-se calcular o  $\mu$  utilizando a equação (1), então, inserir esse valor na equação (2) para encontrar a posição da linha neutra adimensionalizada. Caso o valor de  $\mu$  seja menor ou igual a  $\mu_{lim}$ , deve-se usar a equação (3) para obter o valor da área de aço longitudinal ( $A_s$ ), caso contrário, parte do esforço solicitante de compressão é absorvido pela armadura longitudinal, é necessário calcular o parâmetro geométrico  $\delta$  pela equação (4) que é fundamental para calcular a tensão solicitante de projeto na armadura de compressão (equação 5). De posse desses valores, é viável calcular a área de aço para a tração ( $A_s$ ) e para a compressão ( $A'_s$ ) pelas equações (6) e (7) respectivamente.

## 2.2. Dimensionamento ao cisalhamento

O esforço cortante na viga provoca o cisalhamento, neste sentido, deve-se levar em conta as seções em que esse esforço é máximo no trecho da viga em análise. O modelo utilizado na ferramenta para este dimensionamento foi o da treliça de Morsch.

*“Nesse modelo imagina-se que, após a fissuração, o esforço cortante é equilibrado pela associação de bielas comprimidas de concreto e de diagonais tracionadas acompanhando as trajetórias das tensões principais. As tensões de compressão nas bielas principais devem ser limitadas, para não haver ruptura por esmagamento do concreto. As diagonais tracionadas são formadas por estribos convenientemente dimensionados para o esforço de tração de cálculo.” (ARAÚJO, 2010, p.178).*

Para ser mais explicativo, a Figura 3 mostra a ideia da analogia de treliça de Morsch aplicada ao dimensionamento ao cisalhamento de uma viga de concreto armado utilizando o Modelo I.

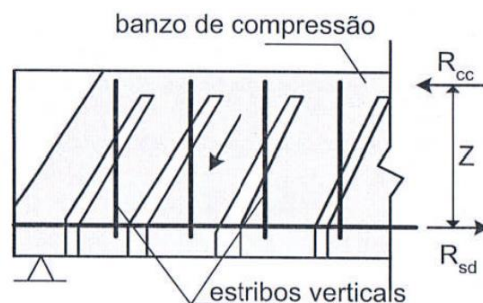


Figura 3 - Analogia de treliça de Morsch. (Fonte: ARAÚJO, 2014).

O Quadro 2 dispõe das equações necessárias para o dimensionamento ao cisalhamento de uma viga de concreto armado.

Tensão de cisalhamento de projeto ( $\tau_{wd}$ )	$\tau_{wd} = \frac{V_d}{b_w d}$	(8)
Tensão última do concreto ( $\tau_{wu}$ )	$\tau_{wu} = 0,27 \alpha_v f_{cd}$	(9)
Constante $\alpha_v$	$\alpha_v = 1 - \frac{f_{ck}}{250}$	(10)
Tensão convencional de cisalhamento reduzida ( $\tau_d$ )	$\tau_d = 1,11(\tau_{wd} - \tau_c)$	(11)
Fator de redução ( $\tau_c$ )	$\tau_c = 0,09 \cdot f_{ck}^{2/3}$	(12)
Área de aço da armadura transversal por cm ( $A_{sw}$ )	$A_{sw} = 100 b_w \cdot \frac{\tau_d}{f_{ywd}}$	(13)

Quadro 2 - Principais equações para o dimensionamento ao cisalhamento de uma viga de concreto armado. (Fonte: Adaptado de ARAÚJO, 2014).

Onde,  $V_d$  é o esforço cortante de projeto,  $f_{ck}$  é a tensão de compressão característica do concreto aos 28 dias de cura, e  $f_{ywd}$  é a tensão de escoamento do aço da armadura transversal.

A princípio, para dimensionar uma viga ao cisalhamento, é necessário obter o esforço cortante de projeto na seção por meio de combinações no Estado Limite Último, em seguida, a equação (8) deve ser utilizada para obter a tensão de cisalhamento de projeto ( $\tau_{wd}$ ). Deve-se analisar as bielas de compressão, de modo que  $\tau_{wd}$  não deve exceder  $\tau_{wu}$  (Equações 8 e 9). Utilizando a equação (11), calcula-se a tensão convencional de cisalhamento reduzida ( $\tau_d$ ) que se utiliza da equação (12) para obter o valor do fator de redução  $\tau_c$ . Por fim, a área de aço da armadura transversal é calculada pela equação (13) em que o resultado representa a área de aço necessária para combater os esforços de cisalhamento por metro de viga. De posse desse valor, deve-se usar relações matemáticas para obter os valores do espaçamento e do total de estribos.

### 2.3. Estado Limite de Serviço para deformações excessivas

Vigas deformam ao serem submetidas a solicitações, essas deformações podem causar transtornos para a utilização da edificação, são exemplos disso, fissuras na alvenaria, emperramento de portas, alteração no sentido das quedas de água, entre outros. A NBR 6118 trata com boas aproximações as deformações desses elementos estruturais.

O Quadro 3 dispõe de equações para a verificação do estado limite de deformações excessivas.

Momento de inércia para o estágio I ( $I_I$ )	$I_I = \frac{b_w h^3}{12} + b_w h \left( x_I - \frac{h}{2} \right)^2 + \alpha_e A_s (d - x_I)^2$	(14)
Posição da linha neutra para o estágio I ( $x_I$ )	$x_I = \frac{\frac{b_w h^2}{2} + \alpha_e A_s d}{b_w h + \alpha_e A_s}$	(15)
Momento de inércia para o estágio II ( $I_{II}$ )	$I_{II} = \frac{b_w x_{II}^3}{3} + \alpha_e A_s (d - x_{II})^2$	(16)
Posição da linha neutra para o estágio II ( $x_{II}$ )	$x_{II} = \frac{\alpha_e A_s}{b_w} \left( -1 + \sqrt{1 + \frac{2b_w d}{\alpha_e A_s}} \right)$	(17)
Momento de fissuração ( $M_r$ )	$M_r = \frac{\alpha f_{ct} I_c}{y_t}$	(18)

Quadro 3 - Equações para verificação do ELS DEF. (Fonte: Adaptado de ARAÚJO, 2014).

Os valores calculados pelas equações do Quadro 3 são aplicados na equação (19) proposta por Branson para calcular o momento de inércia equivalente da seção ( $I_{eq}$ ) e então, calcula-se a flecha imediata utilizando equações provindas da análise estrutural (NBR 6118, 2014).

$$I_{eq} = \left( \frac{M_r}{M_a} \right)^3 I_I + \left( 1 - \left( \frac{M_r}{M_a} \right)^3 \right) I_{II} \leq I_I \quad (19)$$

Em que,  $M_a$  é o momento na seção crítica do vão considerado.

Ao longo do tempo de aplicação de uma solicitação, o material sofre fluência (NBR 6118, 2014) e ocasiona a flecha diferida ( $\delta_{t,\infty}$ ), essa deformação é calculada pelo produto do  $\alpha_f$  equação (20) e a flecha imediata ( $\delta_{t,0}$ ).

$$\alpha_f = \frac{\xi(t) - \xi(t_0)}{1 + 50\rho'} \quad (19)$$

Em que  $\xi$  é um coeficiente função do tempo que pode ser obtido pela equação (20) e  $\rho'$  é obtido pela equação (21).

$$\xi(t) = 0,68 \cdot 0,996^t \cdot t^{0,32} \text{ se } t \leq 70 \text{ meses, caso contrário, } 2. \quad (20)$$

Onde  $t$  é o tempo em meses de quando se deseja o valor da flecha e  $t_0$  é a data em meses do início da aplicação da carga.

$$\rho' = \frac{A'_s}{b_w d} \quad (21)$$

De posse desses valores, é possível calcular a flecha diferida pela equação (22).

$$\delta_{t,\infty} = \delta_{t,0}(1 + \alpha_f) \quad (22)$$

$$I_{eq} = \left(\frac{M_r}{M_a}\right)^3 I_I + \left(1 - \left(\frac{M_r}{M_a}\right)^3\right) I_{II} \leq I_I \quad (19)$$

Em que,  $M_a$  é o momento na seção crítica do vão considerado.

Ao longo do tempo de aplicação de uma solicitação, o material cede e aumenta a deformação, esse fenômeno é denominado fluência (NBR 6118, 2014) e ocasiona a flecha diferida ( $\delta_{t,\infty}$ ), essa deformação é calculada pelo produto do  $\alpha_f$  equação (20) e a flecha imediata ( $\delta_{t,0}$ ).

$$\alpha_f = \frac{\xi(t) - \xi(t_0)}{1 + 50\rho'} \quad (19)$$

Em que  $\xi$  é um coeficiente função do tempo que pode ser obtido pela equação (20) e  $\rho'$  é obtido pela equação (21).

$$\xi(t) = 0,68 \cdot 0,996^t \cdot t^{0,32} \text{ se } t \leq 70 \text{ meses, caso contrário, } 2. \quad (20)$$

Onde  $t$  é o tempo em meses de quando se deseja o valor da flecha e  $t_0$  é a data em meses do início da aplicação da carga.

$$\rho' = \frac{A'_s}{b_w d} \quad (21)$$

De posse desses valores, é possível calcular a flecha diferida pela equação (22).

$$\delta_{t,\infty} = \delta_{t,0}(1 + \alpha_f) \quad (22)$$

## 2.4. A linguagem de programação Python e as principais bibliotecas utilizadas

Essa linguagem segue o padrão da maioria das linguagens, para essa ferramenta, são utilizadas principalmente variáveis, listas, tuplas, funções e classes.

Uma variável é um espaço alocado na memória RAM do computador para que seja inserido um valor que “tem a possibilidade de ser alterado em algum instante ao decorrer do tempo, ou seja, durante a execução do algoritmo em que é utilizado” (FORBELLONE & EBERSPÄCHER, 2022, p.21), variáveis formam a base de uma linguagem de programação, elas dão a possibilidade de armazenamento de valores ou objetos.

As listas são variáveis que armazenam uma coleção dados em uma ordem particular (MATTHEUS, 2016, P.37, tradução própria) e que permite repetição, todos os elementos de uma lista são indexados a partir do zero. Já uma tupla é semelhante a uma

lista, nela é possível guardar dados indexados, visto que, “Uma tupla nada mais é do que uma lista que você não pode modificar” (SHAW, 2014, p.195, tradução própria).

Python é uma linguagem de programação com mais de um paradigma, os dois principais são o funcional e o de orientação a objetos. Um paradigma indica a estrutura em que uma linguagem de programação é baseada. Na programação orientação a objetos, “você escreve classes que representam as coisas e situações do mundo real e cria objetos baseados nessas classes” (MATTHEUS, 2016, P.161, tradução própria). Ao criar uma classe, você está criando um tipo de objetos que tem atributos e comportamentos para posteriormente instanciá-los para alguma aplicação, dispondo de uma função construtora para a classe que atribui valores iniciais para o objeto instanciado. A programação funcional é baseada na utilização de funções, que são blocos de código nomeados que são feitos para um trabalho específico (MATTHEUS, 2016, P.134, tradução própria). Ou seja, é possível reunir várias linhas de código em um bloco que é executado toda vez que é chamado, algumas das principais vantagens de funções são o reuso do código e a facilidade de manutenção. (PAIVA et al. 2020, p.160).

Para a criação da interface gráfica para o usuário, foi utilizada a *PyQT6*, que é uma biblioteca que disponibiliza classes de elementos da interface de usuário, como, botões, caixas de texto, janelas, entre outros. Já para exportar o detalhamento das vigas em arquivo tipo *DXF* é utilizada a *Ezdxflib*, essa biblioteca disponibiliza possibilidades de criar desenhos em CAD.

### 3. APRESENTAÇÃO DA FERRAMENTA

A ferramenta se divide em três arquivos: o *modelo.py*, que contém todas as classes; o *interface\_vigaR.py*, que é responsável por gerenciar a interface gráfica utilizando o *PyQT6*; e o *main.py*, que é responsável por iniciar o programa. Do arquivo *modelo.py* são descritas as principais classes ao longo dessa seção.

#### 3.1. A classe VigaR

O modelo de viga é implementado como uma classe (*VigaR*) que recebe características da viga em seu construtor, tais como, nome,  $b_w$ ,  $h$ ,  $I$ ,  $CAA$  (classe de agressividade do ambiente),  $f_{ck}$ ,  $V_d$ ,  $f_{yw}$ ,  $M_d$ ,  $f_{yk}$ ,  $E_s$ ,  $E_{cs}$  e  $M_a$  (momento crítico para a seção). Essa classe conta com as funções *obter\_dLinha*, *dimensionar\_flexao*, *dimensionar\_cisalhamento*, *contar\_barras\_h* (essa função conta a quantidade de barras de cada diâmetro que são necessárias), *verificar\_ELSDEF*, *verificar\_ELSW*, *calc\_leq* (calcula o momento de inércia equivalente da seção composta), *detalhar*. A Figura 4 dispõe da estrutura geral da classe *VigaR*.

```
8 class VigaR:
9     def __init__(self, nome, bw, h, l, caa, fck, vd, fywk, md, fyk, Es, Ecs=0, ma=0):...
58
59     def detalhar(self):...
112
113     def obter_dLinha(self, h, caa):...
124
125     def dimensionar_flexao(self):...
168
169     def calc_Ieq(self, areaSef, areaSLinhaEf, momentoA):...
188
189     def verificar_ELSDEF(self):...
207
208     def verificar_ELSW(self):...
213
214     def contar_barras_h(self, areaDeAco, agregadoG = 19):...
244
245     def dimensionar_cisalhamento(self):...
```

Figura 4 - Estrutura geral da classe VigaR. (Fonte: Autoria própria, 2023).

As funções *dimensionar\_flexão*, *dimensionar\_cisalhamento*, *verificar\_ELSW* e *verificar\_ELSDEF* foram implementadas seguindo os passos e a utilização das equações para o dimensionamento e verificação de vigas de concreto armado abordados no tópico de revisão bibliográfica.

Foram criadas funções auxiliares para realizar papéis específicos, como, por exemplo, a função *calc\_leq*, que tem o papel de calcular o momento de inércia equivalente na seção transversal da viga, visto que o concreto fissa quando tracionado e a região fissurada deixa de contribuir para o momento de inércia. Essa função calcula também a equivalência do aço para o momento de inércia (*I*) da seção e retorna o valor do *I* equivalente.

A função *contar\_barras\_h* é responsável por receber uma área de aço e retornar a contagem de barras de diferentes diâmetros comerciais. Essa função só retorna os arranjos de barras possíveis para a seção transversal em análise.

### 3.2. A classe barraaco

A classe *BarraAco* é um molde para instanciar cada seguimento de aço utilizado para constituir a armadura longitudinal da viga. Como parâmetros para o construtor dessa classe, são solicitados o *pontoZero* (tupla contendo as coordenadas do ponto zero global da barra), *comprimento*, *diametro*, *dobraEsq* (verdadeiro ou falso), *dobraDir* (verdadeiro ou falso), *justificacao* (centro, esquerda ou direita), *posicao*, *layer* (camada CAD), *contavel* (variável para verificar se a barra deve ser contabilizada nos quantitativos), *quantidade*, *nome*.

A classe *BarraAco* contém a função *posicionarBarra*, quemonta uma lista de entidades em CAD contendo linhas e textos e exporta para ser usada pela biblioteca *Ezdx* para desenhar as barras no arquivo de saída em formato *DXF*. A Figura 5 mostra o corpo da classe *BarraAco*.

```
296 class BarraAco:
297     def __init__(self, pontoZero, comprimento, diametro,
298                 dobraEsq, dobraDir, justificacao, posicao,
299                 layer, contavel, quantidade=1, nome=""):...
314
315     def posicionarBarra(self):...
```

Figura 5 - Corpo da classe BarraAco. (Fonte: Autoria própria, 2023).

### 3.3. Classes complementares

Dentre as classes complementares, foi implementada a *TipoAco*, que armazena as características do aço, tais como  $f_{yk}$ ,  $E_s$  e classificação, essa classe é utilizada apenas para instanciar objetos que representam o aço. A classe *ClassesAA* instancia objetos com informações sobre a classe de agressividade ambiental. Ambas são utilizadas em uma caixa de seleção.

A classe *SecaoTransv* é responsável por gerar o desenho da seção transversal da viga. Essa classe recebe como argumentos um ponto com coordenadas iniciais, largura e altura, arranjos positivo e negativo, cobertura, diâmetro e a quantidade de estribos da viga. Todo o processamento é feito dentro da função *gerar\_secao*, que retorna uma lista contendo entidades com as informações necessárias para serem desenhadas usando a biblioteca *Ezdxflib*. A Figura 6 mostra o corpo dessa classe. A Figura 7 mostra o resultado exportado pela classe supracitada em um desenho aberto no *software* Autodesk AutoCAD.

```
522 class SecaoTransv:
523     def __init__(self, pontoZero, base, altura, arranjoNeg,
524                 arranjoPos, cobertura, diametro, quantidadeEst):...
533
534     def gerar_secao(self):...
```

Figura 6 - Classe SecaoTransv. (Fonte: Autoria própria, 2023).

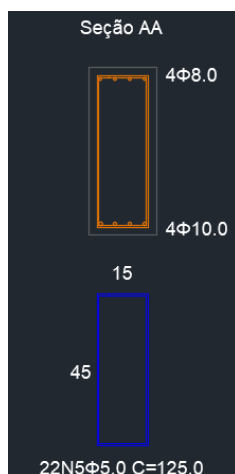


Figura 7 - Detalhamento de seção transversal exportado pela classe SecaoTransv (Fonte: Autoria própria, 2023).

Para aumentar o reaproveitamento de código, foi criada a classe *EstriboSecao*, que retorna entidades que juntas formam o desenho de um estribo com os textos relacionados. Quanto ao perfil da viga, a função *detalhar* junta as entidades recebidas das classes responsáveis pelos desenhos, reúne todas as entidades em uma lista e a retorna.

### 3.4. Gerenciamento de interface gráfica, dimensionamento e detalhamento

O arquivo responsável por gerenciar a interface gráfica e a exportação do arquivo em DXF é o *interface\_vigaR.py*. De início, esse arquivo conta com o código necessário para criar a janela principal em que está disposta toda a interface gráfica da ferramenta. A Figura 8 mostra a disposição do código para tornar isso possível.

```
8 app = QtWidgets.QApplication([])
9 formulario = uic.loadUi('JanelaPrincipal.ui')
```

Figura 8 - Criação de app e formulário usando PyQt6. (Fonte: Autoria própria, 2022).

Em seguida, são criadas e preenchidas as caixas de seleção (*combobox*) destinadas à seleção da classe de agressividade ambiental e o tipo de aço das armaduras transversais e longitudinais. A Figura 9 mostra como essas funções foram implementadas.

```
12 formulario.comboCAA.addItem("I", "II", "IV", "V")
13
14 listaTipoAco = [
15     TipoAco("CA-50", 500, 210),
16     TipoAco("CA-60", 600, 210),
17 ]
18 classesAG = [...]
23 formulario.comboAL.addItem(
24     ...
26 )
27 formulario.comboAT.addItem(
28     ...
30 )
```

Figura 9 - Caixas de seleção para armaduras longitudinais e transversais. (Fonte: Autoria própria, 2023).

A função *dimensionarViga* é responsável por fazer a integração entre a interface gráfica e a parte de lógica do programa. Ao clicar no botão “Calcular área de aço” na interface de usuário, essa função é chamada e recolhe todas as informações inseridas por meio de *LineEdits* (locais para inserção de texto) e cria um objeto a partir da classe *VigaR*, acessa as funções internas à classe que são responsáveis pelo dimensionamento da viga.

Nesse mesmo arquivo existe a função *addEntidades* que é responsável por criar o documento em CAD, camadas, estilos e formas geométricas. Por fim, essa função salva o detalhamento no arquivo “detalhamento.dxf”.

### 3.5. Interface gráfica e funcionamento da ferramenta

A Figura 10 mostra a interface gráfica da ferramenta apresentada nesse documento. A princípio, o usuário deve inserir os parâmetros da seção da viga. Em seguida, deve preencher as informações sobre o  $f_{ck}$ , a classe de agressividade ambiental, o tipo de aço, tanto para a armadura longitudinal quanto para a transversal, e os esforços de projeto.



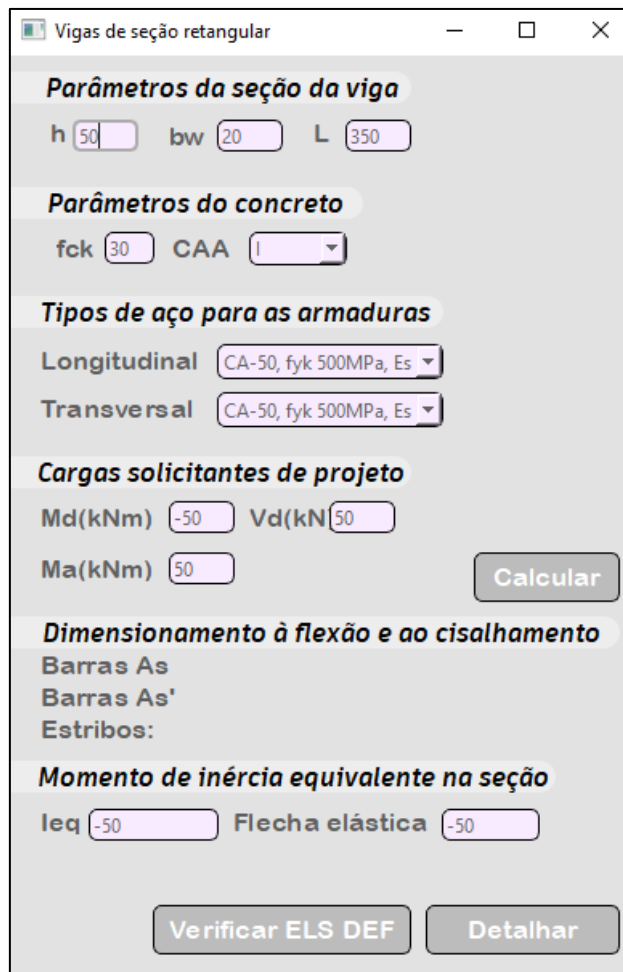


Figura 10 – Janela principal da ferramenta. (Fonte: Autoria própria, 2023).

Ao clicar no botão “Calcular” o *software* calcula as áreas de aço. Após isso, o usuário deve selecionar os arranjos de barras tanto para a armadura positiva quanto para a negativa na janela que é aberta (Figura 11). Os arranjos mostrados pelo *software* são filtrados e disponibilizados apenas os que respeitam as restrições da norma. Tendo em mãos os arranjos de barras, deve-se clicar no botão “Obter  $l_{eq}$ ” para que o *software* calcule o momento de inércia equivalente ( $l_{eq}$ ) para a seção composta e disponibilize para o usuário copiar e colar em algum *software* de análise estrutural como, por exemplo, o Ftool.

De posse da flecha imediata, o usuário deve inserir na caixa de texto referida ao  $l_{eq}$  (é utilizada uma caixa de texto para facilitar o processo de copiar e colar) e clicar no botão “Verificar ELS DEF”. A partir disso, o *software* retorna o valor da flecha total para aquela viga na condição de carregamento especificada. Por fim, clicando no botão “detalhar”, é gerado um arquivo em *.dxf* contendo todo o detalhamento da viga.

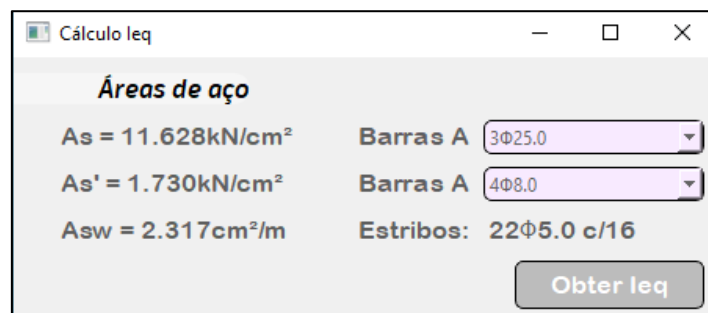


Figura 11 - Janela para calcular o momento de inércia equivalente. (Fonte: Autoria própria, 2023).

## 4. RESULTADOS E DISCUSSÕES

### 4.1. Dimensionamento à flexão

Para verificar o correto funcionamento da ferramenta, foi utilizado o Exemplo 1 do tópico 3.9 do livro do Araújo (2014) que está disposto na Figura 12. Nesse problema, é adotado um  $f_{ck} = 20$  MPa,  $p_k = 15$  kN/m, aço da classe CA-50 e classe de agressividade ambiental II. Por se tratar de uma viga biapoiada, o momento fletor máximo de projeto ( $M_{Sd}$ ) será igual a 42 kNm.

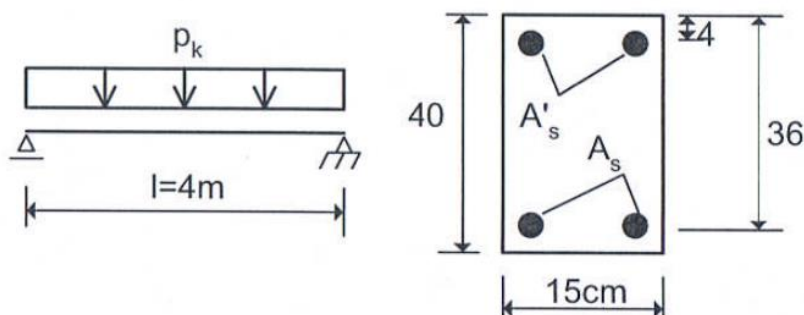


Figura 12 - Exemplo de viga para testar a ferramenta. (Fonte: ARAÚJO, 2014).

O resultado do cálculo da armadura longitudinal e o arranjo de barras é apresentado na Figura 13. Em Araújo (2014), a armadura de tração calculada é de  $2,98\text{cm}^2$ , que é a mesma obtida pelo *software*. A armadura de compressão ( $A'_s$ ) calculada pelo *software*, porém, difere da indicada pelo autor porque a ferramenta calcula a taxa mínima de aço e este não definiu o limite mínimo para a armadura longitudinal. Em uma seção posterior, o autor retoma o mesmo exemplo e acrescenta o limite mínimo de  $0,9\text{cm}^2$  para a armadura de compressão.

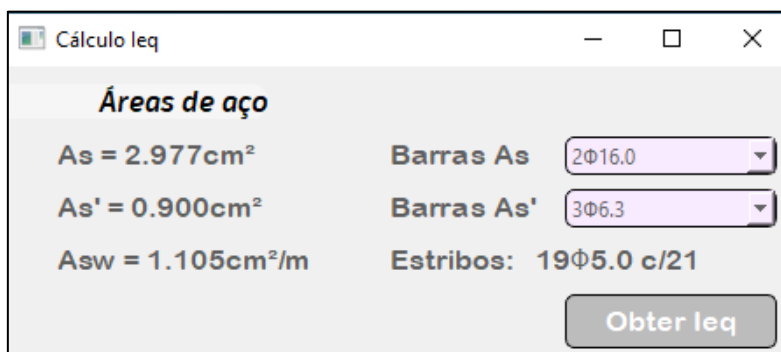


Figura 13 - Resultado do dimensionamento à flexão. (Fonte: Autoria própria, 2023).

### 4.2. Dimensionamento ao cisalhamento

No dimensionamento à força cortante, foi utilizado o exemplo resolvido de Araújo (2014), cujos dados são apresentados na Figura 14. Foi considerado concreto de classe C20, classe de agressividade ambiental II e estribos de aço CA-50. Por ser uma viga biapoiada, a força cortante solicitante de cálculo será de 42 kN.

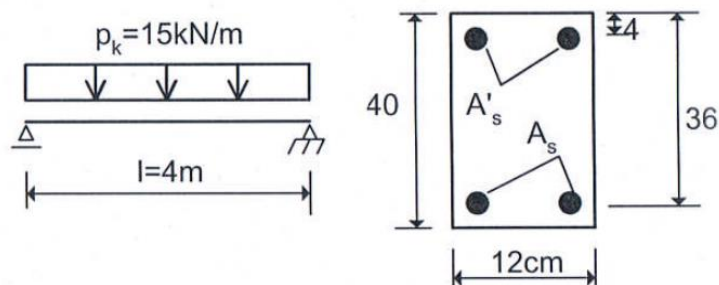


Figura 14 - Exemplo sobre dimensionamento à cortante. (Fonte: ARAÚJO, 2014).

A Figura 15 apresenta o resultado apresentado pela ferramenta. Araújo (2014) apresenta uma área de aço para a armadura transversal ( $A_{sw}$ ) de  $0,94\text{cm}^2$ , que é praticamente igual à obtida pela ferramenta.

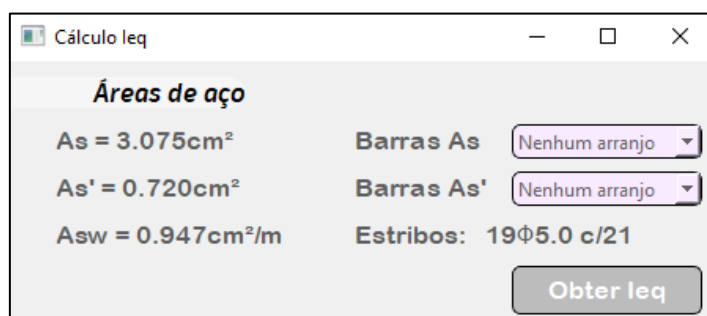


Figura 15 - Resultado do dimensionamento. (Fonte: Autoria própria, 2023).

### 4.3. Estado Limite de Deformações Excessivas

Para esse caso, o exemplo utilizado foi o da seção 6.7 de Araújo (2014). A Figura 16 apresenta a seção da viga. Neste, uma viga biapoiada de 5 metros de comprimento é submetida a uma carga de serviço de  $17\text{ kN/m}$  que tem início aos 28 dias. Para o caso, o  $f_{ck} = 20\text{MPa}$ ,  $M_{sd} = 74,37\text{MPa}$ ,  $M_a = 53,125\text{MPa}$  e classe de agressividade ambiental II.

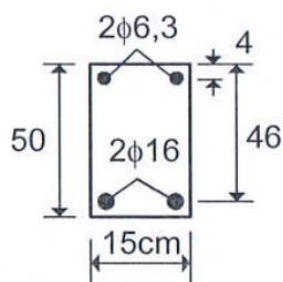


Figura 16 - Exemplo sobre Estado Limite de serviço. (Fonte: Autoria própria, 2023).

No problema, o autor não considera a armadura mínima na seção e seleciona duas barras de 16 para a armadura de tração, pois o valor da área de aço calculada para esse, de acordo com a norma, seria de  $4,16\text{cm}^2$ , sendo, portanto, necessárias mais de 2 barras de 16 para combater o esforço, visto que a área de aço de duas barras de 16 é igual a  $4,02\text{cm}^2$ . Entretanto, devido à proximidade no resultado, o autor optou por usar duas barras de 16 mm, diferindo do *software* abordado por esse trabalho, que tem em sua configuração base a adoção de área de aço efetiva maior que a área de aço calculada. Para o caso, o resultado do *software* foi que a área de aço calculada é de  $4,16\text{cm}^2$  e o arranjo seria de duas barras de 20mm (é o único arranjo que cabe na seção seguindo as recomendações

da NBR 6118/2014). Para o resultado do *software* não divergir do resultado do autor por essa adoção, foi utilizado um momento de projeto de 72,02kNm e um momento atuante de 51,43kNm. Sendo assim, o resultado mostrado na Figura 17 é obtido já com o momento de inércia equivalente calculado, com o valor de 61881,06 cm<sup>4</sup> (divergente do autor, pois no livro não foi considerada a parcela do momento de inércia provocado pela armadura mínima de compressão).

Modelando uma viga no *software* Ftool para a combinação do Estado Limite de Serviço, temos uma flecha imediata de 1,05cm. Inserindo essa flecha na caixa de texto relacionada no *software*, obtém-se uma flecha total de 2,44 cm, conforme pode ser visto na Figura 17.

Vigas de seção retangular

**Parâmetros da seção da viga**

h 50 bw 15 L 500

**Parâmetros do concreto**

fck 20 CAA II

**Tipos de aço para as armaduras**

Longitudinal CA-50, fyk 500MPa, Es

Transversal CA-60, fyk 600MPa, Es

**Esforços solicitantes de projeto**

Md(kNm) 72.02 Vd(kN) 42

Ma(kNm) 51.43

Calcular

**Dimensionamento à flexão e ao cisalhamento**

As => 2Φ16.0

As' => 3Φ8.0

Estribos: 19Φ5.0 c/27

**Momento de inércia equivalente na seção**

leq 61881.06 Flecha elástica 1,05

Flecha diferida = 24.39mm

Verificar ELS DEF

Detalhar

Figura 17 - Resolução do exemplo abordado. (Fonte: Autoria própria, 2023).

#### 4.4. Dimensionamento e detalhamento de viga

Com intuito de mostrar a aplicabilidade da ferramenta com relação ao detalhamento e exportação de planilha com quantitativos, foi proposta uma viga biapoiada com 4 m de comprimento submetida a uma carga uniformemente distribuída de 20 kN/m. A viga possui seção de 20 x 40 cm e classe de agressividade ambiental II.

A Figura 18 mostra a inserção e o cálculo dos valores a viga proposta, enquanto a Figura 19 mostra o resultado do cálculo das áreas de aço e solicita a escolha do arranjo para calcular o momento de inércia equivalente ( $I_{eq}$ ). De posse do valor de  $I_{eq}$ , foi utilizado o Ftool para obter a flecha imediata para as configurações adotadas no exemplo.

The screenshot shows a software window titled "Vigas de seção retangular". It contains several sections for input and output:

- Parâmetros da seção da viga:** h (40), bw (20), L (400)
- Parâmetros do concreto:** fck (30), CAA (II)
- Tipos de aço para as armaduras:** Longitudinal (CA-50, fyk 500MPa, Es), Transversal (CA-60, fyk 600MPa, Es)
- Esforços solicitantes de projeto:** Md(kNm) (56), Vd(kN) (40), Ma(kNm) (40)
- Dimensionamento à flexão e ao cisalhamento:** As => 2Φ16.0, As' => 2Φ10.0, Estribos: 20Φ5.0 c/20
- Momento de inércia equivalente na seção:** Ieq (62432.42), Flecha elástica (0,41)
- Flecha diferida = 8.99mm**

Buttons include "Calcular", "Verificar ELS DEF", and "Detalhar".

Figura 18 - Cálculo da viga em questão. (Fonte: Autoria própria, 2023).

The screenshot shows a software window titled "Cálculo Ieq". It displays the results of the steel area calculation:

- Áreas de aço:** As = 3.820cm<sup>2</sup>, As' = 1.384cm<sup>2</sup>, Asw = 1.931cm<sup>2</sup>/m
- Barras As:** 2Φ16.0
- Barras As':** 2Φ10.0
- Estribos:** 20Φ5.0 c/20

A button labeled "Obter Ieq" is visible at the bottom.

Figura 19 - Resultado da área de aço e arranjos para a viga proposta. (Fonte: Autoria própria, 2023).

Então, ao clicar no botão "Detalhar" é gerado o detalhamento da armadura que pode ser aberto no Autodesk AutoCAD, conforme apresentado na Figura 20. A ferramenta também gera uma planilha no Microsoft Excel com os dados dos quantitativos do aço utilizado (Figura 21).

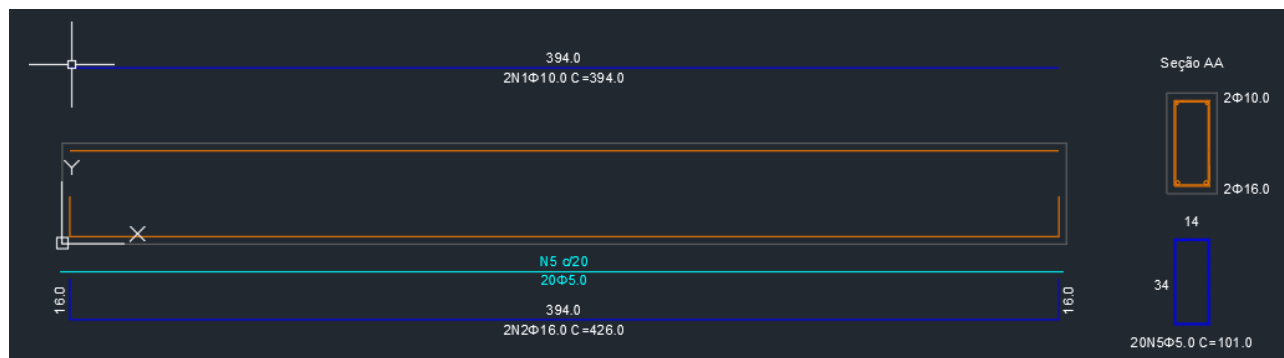


Figura 20 - Detalhamento da viga proposta. (Fonte: Autoria própria, 2023).

	A	B	C	D
1	Nome	Diâmetro	Comprimento	Quantidade
2	N1	10.0	394.0	2
3	N2	16.0	426.0	2
4	N5	5.0	101.0	20

Figura 21 - Planilha de quantitativo de aço exportada. (Fonte: Autoria própria, 2023).

## 5. CONCLUSÕES

Pode-se afirmar que o trabalho obteve êxito nas metas propostas, visto que a ferramenta foi capaz de calcular de maneira correta os dimensionamentos abordados na literatura utilizada, gerar o detalhamento em formato *.dxf* e exportar os quantitativos em um arquivo compatível com o Microsoft Excel. Além disso, a ferramenta possui aplicação didática, visto que oferece, de maneira prática, o dimensionamento, verificação e detalhamento de vigas submetidas à flexão e ao cisalhamento e, por essa razão, pode ser utilizada em sala de aula como auxílio para a compreensão do comportamento das estruturas de concreto armado.

Podemos citar como vantagens da ferramenta a possibilidade de escolha de diferentes arranjos de barras de aço na seção transversal filtrados pelas restrições impostas pela norma, a praticidade no dimensionamento e utilização no dia a dia. A ausência de custos para sua utilização e a facilidade para a execução do código, bem como o próprio fácil acesso ao código fonte, permite que qualquer pessoa adepta à programação em Python e capacitada em engenharia de estruturas tem a possibilidade de aperfeiçoar o *software*. Convém mencionar que, para além dos resultados discutidos, esse trabalho apresenta uma metodologia simples que pode ser empregada na construção de outras ferramentas computacionais.

## REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6118:2014. **Projeto de estruturas de concreto**: Procedimento. Rio de Janeiro, 2014.

FORBELLONE, A. L. V.; EBERSPÄCHER H. F. **Lógica de Programação**: A construção de algoritmos e estruturas de dados com aplicações em Python. 4. ed. São Paulo: Pearson; Porto Alegre: Bookman, 2022.

KIMURA, A. **Informática Aplicada a Estruturas de Concreto Armado**. 2. ed. São Paulo SP: Oficina de Textos, 2018.

LIMA, P. V. N.; BEZERRA, E. M. F.; PEIXOTO, L. K. S.; MONTEIRO, J. V. S.; BEZERRA, J. E. **Desenvolvimento de uma Planilha Eletrônica para Dimensionamento de Seções de Concreto Armado Sujeitas à Flexocompressão Oblíqua** In: Congresso Brasileiro de Concreto, 62., 2020, Florianópolis SC.

LOPES, H. M. T.; CHAVES, W. K. C.; BEZERRA, E. M. F. **Aplicativo para dispositivos móveis para o dimensionamento de peças de madeira sujeitas à tração e compressão**. In: Encontro Brasileiro em Madeiras e Congresso Latino-Americano de Estruturas de Madeira, 16, 3. São Carlos, EBRAMEM e CLEM, 2018.

MARTHA, L. F. **Análise de Estruturas**. 2. ed. Rio de Janeiro: Campus, 2017.

MATTHEUS, E. **Python Crash Course: A Hands-On, Project-Based Introduction to Programming**. San Francisco: No Starch Press, 2016.

MONTEIRO, J. V. S.; BEZERRA, E. M. F.; PEIXOTO, L. K. S.; MATIAS, B. S. **Concepção de Aplicação Android para Dimensionamento de Sapatas Rígidas Isoladas** In: Congresso Brasileiro de Concreto, 62., 2020, Florianópolis SC.

PAIVA, F. A. P.; NASCIMENTO, J. M. A.; MARTINS, R. S.; SOUZA, G. R. **Introdução a Python Com Aplicações de Sistemas Operacionais**. Natal RN: Editoraifrn, 2020.

PORTO, T. B.; FERNANDES, D. S. G. **Curso Básico de Concreto Armado: conforme a NBR 6118/2014**. São Paulo SP: Oficina de Textos, 2015.

SHAW, Z. A. **Learn Python The Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code**. 3. ed. Crawfordsville, Indiana: Pearson, 2014.

SOUZA, P. H. E.; FREITAS, I. L. B.; HOLANDA, F. A.; OLIVEIRA, J. I. F., BEZERRA, E. M. F. **MIT APP INVENTOR: UMA APLICAÇÃO NO DESENVOLVIMENTO DE UM APLICATIVO PARA CÁLCULO DA ARMADURA DE VIGAS DE CONCRETO** In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA E SIMPÓSIO INTERNACIONAL DE EDUCAÇÃO EM ENGENHARIA ABENGE, 47., 2019, Fortaleza CE.

SORIANO, H. L.; LIMA, S. S. **Análise de estruturas: método das forças e método dos deslocamentos**. 2. ed. Rio de Janeiro: Ciência Moderna, 2006.