

FACULDADE GALILEU

ENGENHARIA CIVIL

MÁRCIO ANTÔNIO DA SILVA JÚNIOR

**DESENVOLVIMENTO DE SOFTWARE PARA CÁLCULO
DE SEÇÕES USUAIS DE CONCRETO ARMADO**

BOTUCATU - SP
2020

FACULDADE GALILEU

ENGENHARIA CIVIL

MÁRCIO ANTÔNIO DA SILVA JÚNIOR

**DESENVOLVIMENTO DE SOFTWARE PARA CÁLCULO
DE SEÇÕES USUAIS DE CONCRETO ARMADO**

Trabalho de conclusão de curso
apresentado à Faculdade Galileu como
parte dos requisitos necessários para
obtenção do grau no Curso de
Engenharia Civil.

Orientador: Prof. Leandro Corrêa Martins

BOTUCATU - SP
2020

FACULDADE GALILEU

FOLHA DE APROVAÇÃO

**DESENVOLVIMENTO DE SOFTWARE PARA CÁLCULO
DE SEÇÕES USUAIS DE CONCRETO ARMADO**

MÁRCIO ANTONIO DA SILVA JÚNIOR

Trabalho de conclusão de curso
apresentado à Faculdade Galileu como
parte dos requisitos necessários para
obtenção do grau no Curso de
Engenharia Civil.

Data de aprovação: ____/____/____

Banca examinadora:

Prof. Leandro Corrêa Martins

Avaliador 1

Avaliador 2

DEDICATÓRIA

À minha família, que é meu alicerce e sempre me incentivou em tudo na vida, sem a qual nada disso seria possível, e aos meus professores e amigos.

*“Grandes realizações não são feitas
por impulso, mas por uma soma de
pequenas realizações. ”
(Vincent Van Gogh)*

RESUMO

O presente trabalho se propõe a investigar processos simplificados de automatização de marchas de cálculo utilizadas na obtenção de áreas de aço de armaduras de seções de vigas e pilares usuais de concreto armado, submetidos a esforços oriundos da flexão simples e de flexão composta reta.

Após revisão literária, os cálculos serão traduzidos em linhas de código fonte da linguagem de programação C# e, por meio do uso de um ambiente integrado de desenvolvimento e bibliotecas gráficas, os mesmos serão utilizados na criação de um exemplo de programa de computador para o sistema operacional Windows®.

Palavras-chave: Software, Concreto Armado, Dimensionamento, Desenvolvimento.

ABSTRACT

This paper aims to investigate simplified processes for the automation of calculation routines used for obtaining steel areas of reinforcement for usual beams and columns cross sections made of reinforced concrete, submitted to efforts originated from simple bending and straight compound bending.

After literary review, the calculations will be translated into C# programming language source code lines so that, through the use of integrated development environment and graphic libraries, it will be applied into the creation of an example of computer program for the Windows® operating system.

Keywords: Software, Reinforced Concrete, Sizing, Development.

SUMÁRIO

1. INTRODUÇÃO.....	6
1.1 OBJETIVOS GERAIS	7
1.2 OBJETIVOS ESPECÍFICOS	8
2. REVISÃO DA LITERATURA.....	9
2.1. CONCRETO ARMADO	9
2.2. FLEXÃO SIMPLES.....	11
2.3. FLEXÃO COMPOSTA RETA.....	14
2.4. PROGRAMAÇÃO DE SOFTWARE.....	19
2.5. AMBIENTE DE DESENVOLVIMENTO.....	20
3. DESENVOLVIMENTO DA APLICAÇÃO	25
3.1. FLEXÃO SIMPLES	25
3.2. FLEXÃO COMPOSTA RETA.....	26
4. RESULTADOS	29
4.1. EXEMPLO 1	29
4.2. EXEMPLO 2	30
5. CONCLUSÃO.....	33
REFERÊNCIAS BIBLIOGRÁFICAS.....	34
ANEXO I.....	36

1. INTRODUÇÃO

Ao entrar no mercado de trabalho, o trabalho do engenheiro projetista será o de solucionar diversos problemas em que se fará necessário a realização de grandes quantidades de cálculos matemáticos para a determinação de soluções para a estabilidade estrutural de empreendimentos, cuja responsabilidade do engenheiro é garantir a segurança e correta utilização das instalações para qual a estrutura será projetada.

“Engenheiros, matemáticos e outros profissionais das áreas de ciências exatas se deparam diariamente com problemas numéricos complexos, que exigem grande esforço de cálculo.” (ARAÚJO, 2018, p.2).

As marchas para o cálculo de estruturas em geral podem ser muito trabalhosas, longas e dispendiosas. Para algumas vigas contínuas de concreto armado, por exemplo, há um número elevado de pontos críticos em que se fazem presentes diferentes intensidades de momento fletor e forças cortantes. Como consequência, são necessárias dezenas de repetições envolvendo os cálculos de área de aço para obtenção das armaduras, o que é apenas um dos passos no processo completo de dimensionamento e detalhamento de uma única viga contínua de concreto armado, por exemplo.

Quando o calculista se depara com uma estrutura de edifício composta por dezenas de vigas, lajes, pilares, blocos e paredes, o cálculo manual pode se tornar inviável comercialmente, devido ao tempo necessário para sua realização ser muito elevado.

Nos dias atuais tornou-se comum o emprego de softwares que automatizam os processos de cálculo e detalhamento que envolvem o projeto de estruturas em geral, buscando a rápida execução dos cálculos necessários para a obtenção das dimensões dos diversos componentes que compõem as estruturas em desenvolvimento. Para as estruturas de concreto armado, destacam-se alguns nomes como: TQS, AltoQi Eberick, Multiplus CypeCAD, Autodesk Robot, dentre outros.

De acordo com Kimura (2007), atualmente todas as partes do projeto de um edifício são influenciadas pela agilidade que o uso da informática proporciona, possibilitando que o projetista, em mãos de um computador pessoal, realize

detalhamentos e cálculos com grande rapidez e precisão, o que para algumas situações até algum tempo atrás era inviável de ser feito manualmente.

Há décadas, as réguas de cálculo (que hoje mais parecem objetos pré-históricos perante as máquinas atuais), os computadores que ocupavam uma sala inteira e as calculadoras programáveis auxiliaram, e muito, os Engenheiros a automatizar simples contas e tarefas isoladas menos complexas. Nessa época, os cálculos levavam dias para serem processados, havia uma enorme limitação de memória e somente modelos mais simples podiam ser analisados. (KIMURA, 2007, p.24).

O estudo do processo simplificado de obtenção de um sistema automatizado de cálculo pela criação de software, que pode se utilizar das marchas de cálculo manuais para criação de seus algoritmos, pode ser útil para o desenvolvimento de habilidades na criação de soluções que agilizem processos de concepção e detalhamento de estruturas. Tais práticas podem conferir mais tranquilidade e confiança ao projetista na hora de conceber suas estruturas, de maneira ágil, pois podem reduzir drasticamente o tempo necessário à concepção e detalhamento de estruturas usuais de concreto armado, em contraste com a utilização de processos manuais repetitivos. Criar o próprio software, pode proporcionar ao projetista maior intimidade com os resultados obtidos, visto que o programa se utiliza de seus próprios critérios.

1.1 OBJETIVOS GERAIS

Realizar revisão literária para obtenção de marchas de cálculo de concreto armado e conceitos de programação de computadores, para sua aplicação na criação de um projeto exemplo, sendo que este possa ser utilizado no processo de dimensionamento de seções usuais retangulares de concreto armado, bem como no estudo do dimensionamento das mesmas por alunos de cursos de graduação em engenharia. O trabalho também se propõe a servir como iniciação ao estudo da criação de software em geral, com foco na automatização de tarefas e cálculos manuais de engenharia com o uso de computadores pessoais e dispositivos móveis.

1.2 OBJETIVOS ESPECÍFICOS

O foco deste trabalho está na realização de revisão literária para obtenção de marchas de cálculo manuais de seções de vigas retangulares de concreto armado de classes até C50 e aço CA-50, submetidas a solicitações oriundas da flexão simples.

Também serão compreendidas seções submetidas a esforços de flexão composta reta, com base no trabalho de Venturini e Rodrigues (1987) e seus ábacos.

Após a revisão literária, com o intuito de aplicar o conhecimento adquirido, será desenvolvido um exemplo de programa de computador (software) com interface gráfica para o sistema operacional Windows® utilizando-se da linguagem de programação “C#” (Plataforma .NET) em conjunto com o as bibliotecas gráficas “WinForms” e o ambiente de design “Visual Studio 2019 Community”, plataforma de desenvolvimento de software oferecida gratuitamente pela empresa Microsoft.

O programa será escrito em forma de sequências de linhas de código (linguagem de programação, que são linhas de comando para o processador do computador realizar tarefas) com base nas marchas de cálculo encontradas na revisão literária, frequentemente usadas por engenheiros e acadêmicos no cálculo de vigas e pilares usuais de concreto armado.

Não serão compreendidas seções submetidas à flexão composta oblíqua neste trabalho, bem como tópicos relacionados à algoritmos e lógica de programação.

2. REVISÃO DA LITERATURA

2.1. CONCRETO ARMADO

De acordo com Bastos (2014), o concreto é um material composto, constituído por cimento, água, agregado miúdo, agregado graúdo e em alguns casos, aditivos para conseguir determinados efeitos na mistura. Devido à baixa capacidade do concreto resistir a tensões de tração, são adicionadas barras de aço no interior da peça de concreto, aumentando significativamente a capacidade do material em vencer cargas oriundas de grandes vãos livres, estas que o concreto sozinho não seria capaz de suportar.

Figura 1 – Representação gráfica de uma peça de concreto armado.



Fonte: NICOLETTI (2020).

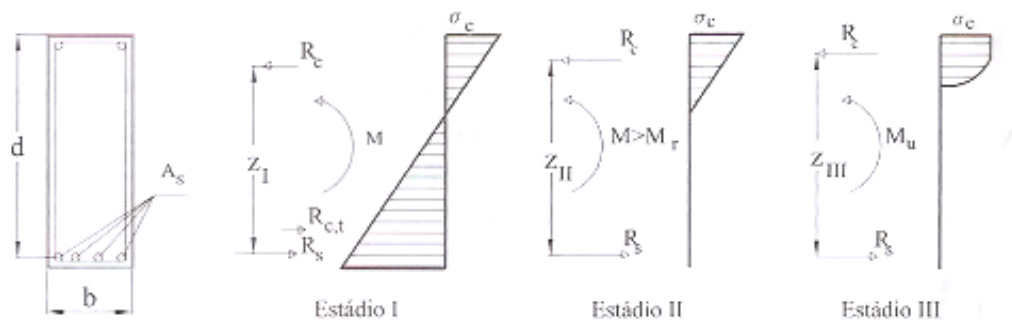
Figura 2 – Exemplo de estrutura feita em concreto armado.



Fonte: MOURA (2018).

Segundo Carvalho e Filho (2014) o cálculo e dimensionamento de estruturas de concreto armado deve ser realizado no estágio III (Figura 3), que corresponde ao estado limite último das seções, sendo o objetivo projetar estruturas de forma econômica e que resistam ao colapso.

Figura 3 – Comportamento da seção transversal de uma viga de concreto armado.



Fonte: CARVALHO E FILHO (2014, p.114).

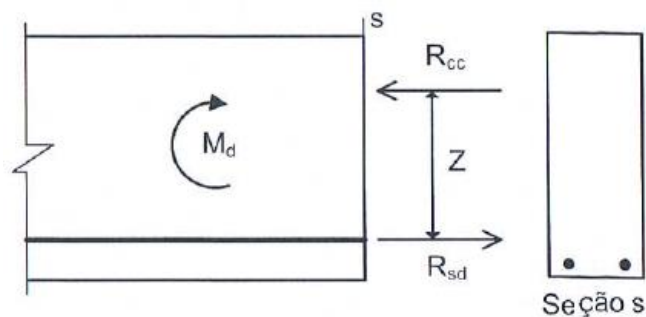
As hipóteses básicas para o cálculo de peças de concreto armado, de acordo com Carvalho e Filho (2014) são as seguintes:

- As seções transversais são consideradas de acordo com a hipótese de Bernoulli, em que as seções permanecem planas após o início da deformação até o estado limite último. As deformações, em cada ponto, são proporcionais à sua distância até a linha neutra da seção.
- Admite-se a solidariedade perfeita entre o concreto e a armadura, sendo iguais as deformações entre o concreto e a armadura adjacente.
- São desprezadas as tensões de tração no concreto.
- O valor máximo permitido para deformação de encurtamento em concretos de classes até C50 é de 2,0‰ em seções totalmente comprimidas e de 3,5‰ em seções sob flexão.
- O valor máximo permitido para deformação de alongamento nas armaduras é de 10,0‰, de modo a prevenir contra deformação plástica excessiva.
- A distribuição de tensões em concretos de classes até C50 se dá por meio de um diagrama retangular de altura $0,8x$ (Figura 5d), onde x é a profundidade da linha neutra.

2.2. FLEXÃO SIMPLES

De acordo com Araújo (2010), para seções de concreto armado submetidas à flexão simples, é necessário ao equilíbrio que uma parte da seção esteja comprimida e a outra parte esteja tracionada. O concreto será responsável por absorver as tensões de compressão e a armadura longitudinal de aço será responsável em absorver as tensões de tração.

Figura 4 – Equilíbrio em flexão simples.



Fonte: ARAÚJO (2010, p.90).

Segundo Araújo (2010), a resultante de compressão no concreto e a resultante de tração no aço, R_{cc} e R_{sd} respectivamente, podem ser consideradas multiplicando-se pelo braço de alavanca Z , formando um binário de momentos que se equilibra com o momento fletor solicitante de cálculo M_d , levando às equações:

$$M_d = R_{cc} \cdot Z \quad (2.1.1)$$

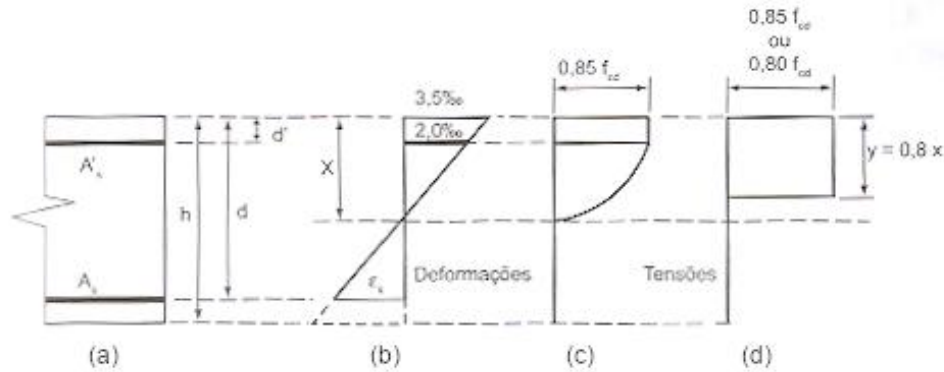
e

$$M_d = R_{sd} \cdot Z \quad (2.1.2)$$

em que, para simplificação dos cálculos, admite-se a situação que pode ser observada na Figura 5d, em que é feita uma aproximação ao considerar as tensões de compressão no concreto como constantes, o que é uma simplificação válida para casos recorrentes de flexão simples em concretos de classes até C50, de modo que a situação da Figura 5c ficará restrita às situações de maior complexidade, como o

caso dos concretos de classes superiores a C50, bem como casos de flexão composta oblíqua.

Figura 5 – Diagramas de tensões no ELU para concretos até classe C50



Fonte: CARVALHO E FILHO (2014, p.117).

Bastos (2013) desenvolve as equações decompondo R_{cc} , R_{sd} e Z da seguinte forma:

$$\sigma = \frac{F}{A} \quad \text{ou} \quad F = \sigma \cdot A \quad (2.1.3)$$

portanto pode-se escrever R_{cc} , R_{sd} como:

$$R_{cc} = 0,85 f_{cd} \cdot 0,8x \cdot bw \quad (2.1.4)$$

e

$$R_{sd} = \sigma_{sd} \cdot A_s \quad (2.1.5)$$

em que $0,85 f_{cd}$ corresponde à tensão constante de cálculo no concreto, $0,80x \cdot bw$ às dimensões da parte comprimida de concreto da seção, σ_{sd} à tensão de cálculo na armadura e sua área de aço A_s .

Bastos (2013) define o valor de Z como a distância entre o centro de gravidade do trecho comprimido de concreto até o centro de gravidade da armadura tracionada, podendo ser expresso pela seguinte equação:

$$Z = d - 0,4x \quad (2.1.6)$$

Estas definições possibilitam que as equações 2.1.1 e 2.1.2 sejam reescritas da seguinte maneira:

$$Md = 0,68fcd \cdot bw \cdot x(d - 0,4x) \quad (2.1.7)$$

e

$$As = \frac{Md}{\sigma_{sd}(d - 0,4x)} \quad (2.1.8)$$

sendo a equação 2.1.7 utilizada para encontrar o valor de x , que corresponde à altura da linha neutra na seção. Uma vez encontrada a altura da linha neutra, aplica-se o valor na equação 2.1.8 para obter-se o valor da área de aço (As) necessária ao equilíbrio da seção.

Para a obtenção da altura da linha neutra, Bastos (2013) iguala a equação 2.1.7 a zero e a reescreve em forma de uma equação do segundo grau:

$$-0,4x^2 + d \cdot x - \frac{Md}{0,68 \cdot bw \cdot fcd} = 0 \quad (2.1.9)$$

possibilitando a obtenção da raiz positiva da equação, cujo valor é o desejado, por meio de uma equação direta usando o teorema de Bhaskara:

$$x = \frac{-d + \sqrt{(d)^2 - 4 \cdot (-0,4) \cdot \left(\frac{Md}{0,68bw \cdot fcd}\right)}}{2(-0,4)} \quad (2.1.10)$$

em que, uma vez fixados valores para a altura útil da armadura d , momento fletor solicitante de cálculo Md , largura da seção de concreto bw e a tensão resistente de cálculo no concreto fcd , as equações 2.1.10 e 2.1.8 podem ser utilizadas para a realização do dimensionamento da seção, seja por meio de cálculo manual, planilhas eletrônicas ou até mesmo por meio de um programa de computador elaborado utilizando-se destas equações.

Antes da realização do detalhamento final, Bastos (2013) ressalta a importância da verificação dos resultados de x e As quanto ao domínio de deformação e às

exigências da norma (atualmente NBR 6118:2014), quanto às áreas de aço mínima e máxima, bem como os espaçamentos mínimos entre as barras acomodadas na seção transversal. Faz-se também necessária a verificação para a necessidade de armadura de pele.

2.3. FLEXÃO COMPOSTA RETA

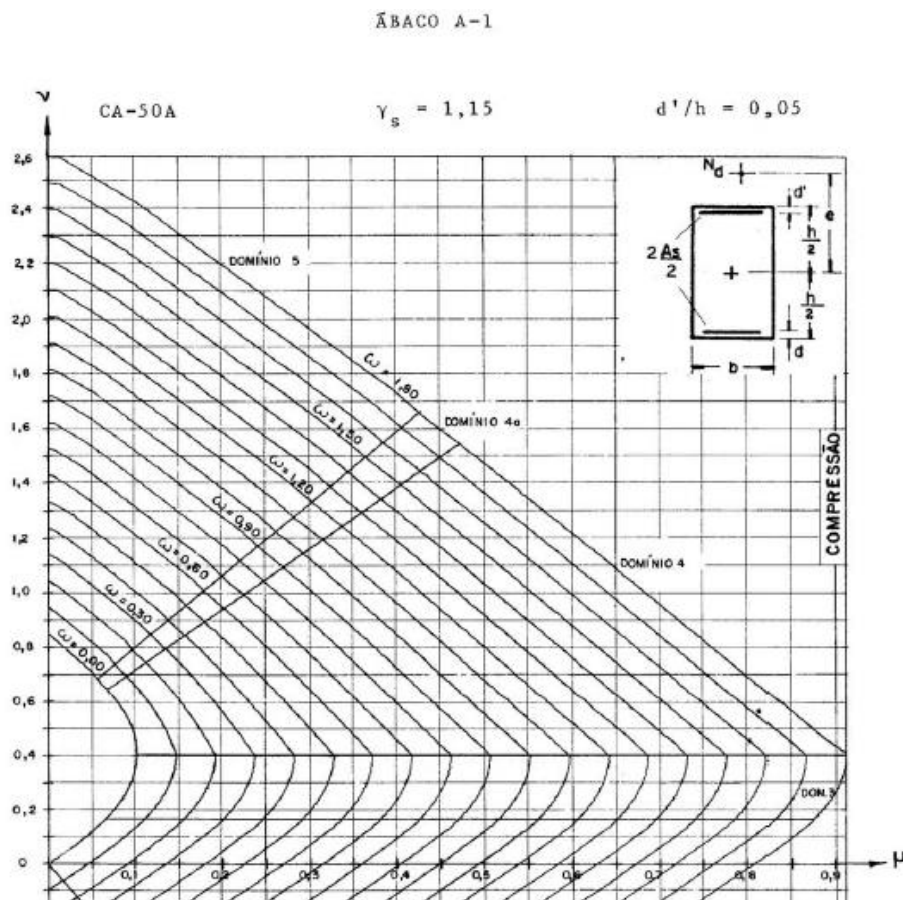
Segundo Bastos (2005), para o dimensionamento manual de seções de concreto armado submetidas à flexão composta, é imprescindível o uso de ábacos pois estes permitem a rápida determinação da quantidade de armadura necessária, isto é, a taxa de armadura e consequentemente a área de aço, eliminando assim a necessidade da aplicação de equações teóricas.

Bastos (2005) recomenda o uso dos ábacos de Venturini e Rodrigues (1987) para os casos de flexão composta reta em seções retangulares de concreto armado.

Segundo Venturini e Rodrigues (1987), ábacos são conjuntos de múltiplas curvas denominadas diagramas de interação, que relacionam combinações de valores de momento fletor e força normal para uma determinada taxa de armadura, atuando em uma seção transversal.

Em seu trabalho, Venturini e Rodrigues (1987) apresentam as equações, restrições e hipóteses de cálculo utilizadas na elaboração de seus ábacos, possibilitando a obtenção das coordenadas exatas dos pontos utilizados na confecção dos diagramas de interação dos ábacos contidos no trabalho. Estas coordenadas podem ser usadas para recriar os ábacos dentro de um programa de computador, ou até mesmo criar mais curvas para os ábacos existentes. O uso da formulação também pode ser usado para a criação de ábacos diferentes dos contidos no trabalho de Venturini e Rodrigues (1987), desde que obedeçam às restrições quanto aos materiais e a forma da seção.

Figura 6 – Exemplo de ábaco adimensional para dimensionamento.



Fonte: Adaptado de VENTURINI E RODRIGUES (1987, p.36).

De acordo com Venturini e Rodrigues (1987), para a confecção de um diagrama de interação adimensional, é necessário encontrar o valor da força normal e momento fletor adimensionais, v e μ respectivamente, para uma série de valores da altura adimensional da linha neutra ξ diferentes, variando de $-\infty$ até $+\infty$.

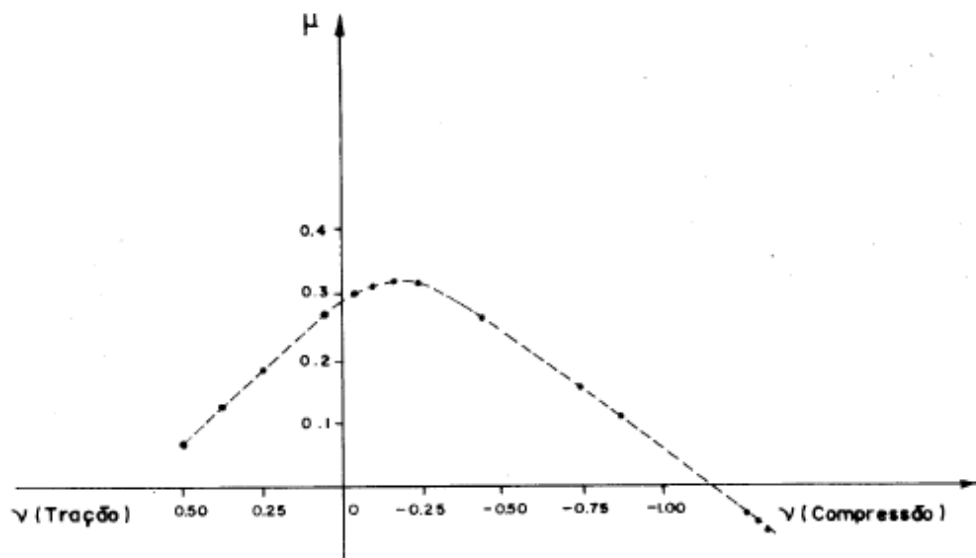
Tabela 1 – Valores de esforços adimensionais para plotagem de curva de interação.

ξ	β_1	β_2	β_3	ξ	ϵ_{lim}	ϵ_{s1}	ϵ_{c2}	σ_{s1}	σ_{s2}	ν	μ
-5,0	-0,5	-0,50	-0,5	0,95	0,0100	0,0100	0,0085	435	435	0,50	0,07
0,0	-0,5	-0,50	-0,5	0,95	0,0100	0,0100	0,0005	435	105	0,38	0,13
0,1	-0,4	-0,50	-0,5	0,95	0,0100	0,0100	0,0017	435	-370	0,25	0,19
0,2	-0,3	-0,45	-0,5	0,95	0,0100	0,0100	0,0020	435	-420	-0,05	0,27
0,3	-0,2	-0,37	-0,5	0,00	-0,0035	0,0076	-0,0029	435	-435	-0,04	0,30
0,4	-0,1	-0,33	-0,5	0,00	-0,0035	0,0048	-0,0031	435	-435	-0,10	0,32
0,5	0,0	-0,29	-0,5	0,00	-0,0035	0,0032	-0,0032	435	-435	-0,17	0,32
0,6	0,1	-0,24	-0,5	0,00	-0,0035	0,0020	-0,0032	429	-435	-0,24	0,33
0,7	0,2	-0,20	-0,5	0,00	-0,0035	0,0013	-0,0035	263	-435	-0,44	0,26
0,8	0,3	-0,16	-0,5	0,00	-0,0035	0,0007	-0,0033	138	-435	-0,60	0,21
0,9	0,4	-0,11	-0,5	0,00	-0,0035	0,0002	-0,0033	041	-435	-0,75	0,16
1,0	0,5	-0,07	-0,5	0,00	-0,0035	-0,0002	-0,0033	-35	-435	-0,88	0,12
3,0	0,5	-0,07	-0,5	0,43	-0,0020	-0,0016	-0,0023	-335	-435	-1,27	-0,04
5,0	0,5	-0,07	-0,5	0,43	-0,0020	-0,0018	-0,0022	-372	-435	-1,30	-0,05
∞	0,5	-0,07	-0,5	0,43	-0,0020	-0,0020	-0,0020	-420	-420	1,33	-0,07

Fonte: VENTURINI E RODRIGUES (1987, p.28).

Venturini e Rodrigues (1987) afirmam que com um número suficiente de pares de esforços adimensionais, pode-se usar estes valores como coordenadas de pontos para a plotagem dos diagramas de interação para a taxa de armadura considerada.

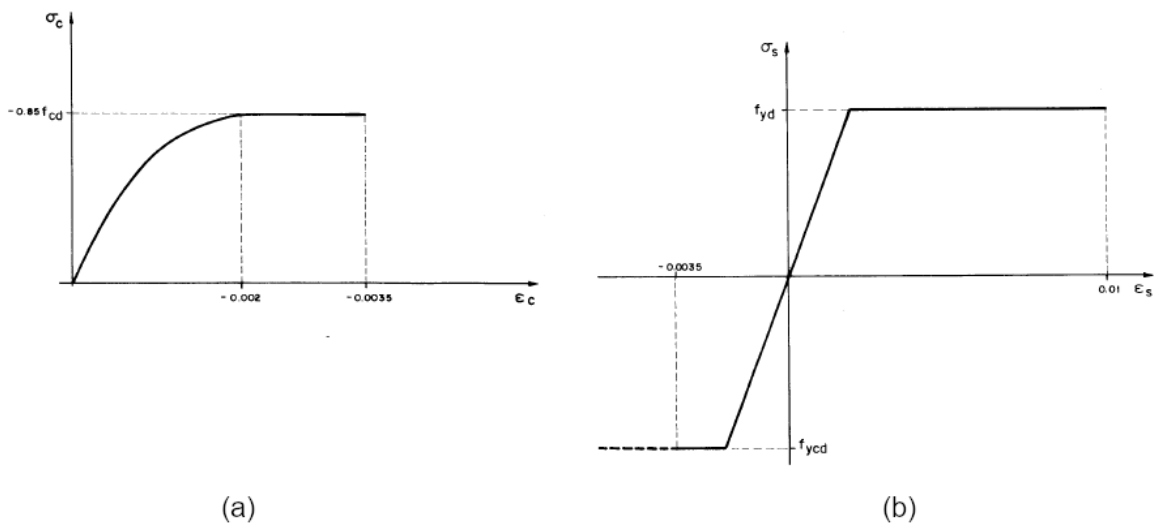
Figura 7 – Diagrama de interação elaborado à partir da Tabela 1.



Fonte: VENTURINI E RODRIGUES (1987, p.29).

Segundo Venturini e Rodrigues (1987), cada valor de ξ deve ser analisado quanto aos limites adimensionais da zona de compressão do concreto β_1 , β_2 e β_3 , quanto à constante adimensional da seção ε_0 e quanto à deformação limite ε_{lim} . Para cada uma das armaduras devem ser calculados os valores relativos à posição relativa η , a deformação ε e o valor da tensão atuante de cálculo σ .

Figura 8 – Diagramas *tensão x deformação* utilizados no cálculo das tensões.



Fonte: Adaptado de VENTURINI E RODRIGUES (1987).

As equações para a determinação das tensões no concreto (Figura 8a), segundo Venturini e Rodrigues (1987) são:

$$\sigma_c = -0,85 f_{cd} \quad \text{para} \quad -0,002 \geq \varepsilon_c \geq -0,0035 \quad (2.2.1)$$

e

$$\sigma_c = 850 f_{cd} (1 + 250 \varepsilon_c) \cdot \varepsilon_c \quad \text{para} \quad 0 \geq \varepsilon_c \geq -0,002 \quad (2.2.2)$$

Para a determinação das tensões no aço CA-50, com base no diagrama apresentado na Figura 8b, as equações ficam definidas como:

$$\sigma_s = E \cdot \varepsilon_s \quad \text{sendo} \quad E = 210 \text{ GPa} \quad (2.2.3)$$

e

$$\sigma_s = f_{yd} \quad \text{para} \quad |\varepsilon_s| \geq |\varepsilon_{yd}| \quad (2.2.4)$$

Por fim, o cálculo dos esforços adimensionais referentes ao ponto ξ estudado se dá pelo uso das equações elaboradas por Venturini e Rodrigues (1987), podendo estas serem observadas na Figura 9.

Figura 9 – Equações para cálculo dos esforços adimensionais.

$$\begin{aligned}
 v = \frac{850\epsilon_u}{\xi - \xi_o} \left\{ \frac{250\epsilon_u}{\xi - \xi_o} \left[\frac{(\beta_1^3 - \beta_2^3)}{3} - (\xi - 0,5)(\beta_1^2 - \beta_2^2) + \right. \right. \\
 \left. \left. + (\xi - 0,5)^2(\beta_1 - \beta_2) \right] - \frac{(\beta_1^2 - \beta_2^2)}{2} + \right. \\
 \left. + (\xi - 0,5)(\beta_1 - \beta_2) \right\} - 0,85(\beta_2 - \beta_3) + \\
 + \frac{\omega}{f_{yd}} \sum_{i=1}^N \sigma_{si} \eta_{si} \quad (6.8a)
 \end{aligned}$$

$$\begin{aligned}
 \mu = \frac{850\epsilon_u}{\xi - \xi_o} \left\{ - \frac{(\beta_1^3 - \beta_2^3)}{3} + (\xi - 0,5)(\beta_1^2 - \beta_2^2)/2 + \frac{250\epsilon_u}{\xi - \xi_o} \right. \\
 \left[- 2(\xi - 0,5)(\beta_1^3 - \beta_2^3)/3 + (\xi - 0,5)^2(\beta_1^2 - \beta_2^2)/2 + \right. \\
 \left. + \frac{(\beta_1^4 - \beta_2^4)}{4} \right] \left. \right\} - 0,425(\beta_2 - \beta_3) + \frac{\omega}{f_{yd}} \sum_{i=1}^N \sigma_{si} \beta_{si} \eta_{si} \quad (6.8b)
 \end{aligned}$$

Fonte: Adaptado de VENTURINI E RODRIGUES (1987).

Segundo Venturini e Rodrigues (1987), o uso dos ábacos tem como objetivo a obtenção da taxa de armadura ω , para então a aplicação deste valor na equação da taxa de armadura (Figura 10c), obtendo-se então o tão desejado valor de A_s .

Figura 10 – Equações para utilização dos ábacos adimensionais.

$$v = \frac{N_d}{A_c f_{cd}} \quad (a)$$

$$\mu = \frac{M_d}{A_c h f_{cd}} \quad (b)$$

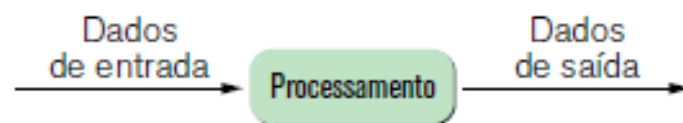
$$\omega = \frac{A_s f_{yd}}{A_c f_{cd}} \quad (c)$$

Fonte: Adaptado de VENTURINI E RODRIGUES (1987).

2.4. PROGRAMAÇÃO DE SOFTWARE

De acordo com Gotardo (2015), um computador é basicamente um processador de dados que transforma dados de entrada em dados finais (dados de saída). Como pode ser visto na Figura 11, os dados entram, são processados e saem do nosso computador.

Figura 11 – Computador de único propósito.

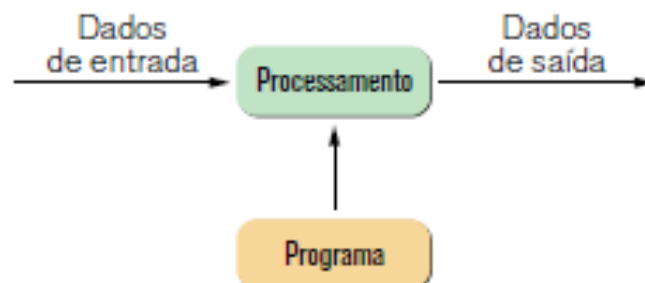


Fonte: GOTARDO (1987, p.11).

Gotardo (2015) afirma que para controlar o processamento é necessário um conjunto de instruções de controle denominado programa.

“Um programa é um conjunto de instruções a serem realizadas pelo nosso “computador programável”. Vamos chamar então (assim como os cientistas chamaram) este computador de: computador de propósito geral. É geral, pois podemos trocar o programa e resolver problemas diferentes com instruções diferentes.” (GOTARDO, 2015, p.12).

Figura 12 – Computador de propósito geral.



Fonte: GOTARDO (1987, p.13).

“Uma linguagem de programação é um método padronizado que usamos para expressar as instruções de um programa a um computador programável. Ela segue

um conjunto de regras sintáticas e semânticas para definir um programa de computador. “ (GOTARDO, 2015, p.17).

Através da especificação de uma linguagem de programação você pode especificar quais dados um computador vai usar; como estes dados serão tratados, armazenados, transmitidos; quais ações devem ser tomadas em determinadas circunstâncias.

Ao usarmos uma linguagem de programação você cria o chamado “Código Fonte”. Um código fonte é um conjunto de palavras escritas de acordo com as regras sintáticas e semânticas de uma linguagem. (GOTARDO, 2015, p.12).

Figura 13 – Exemplo de código fonte em linguagem de programação C.

Código 1.1

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    printf("Hello C World\n");
    return 0;
}
```

Fonte: GOTARDO (1987, p.23).

2.5. AMBIENTE DE DESENVOLVIMENTO

Segundo Rouse (2018), é comum dentre os desenvolvedores de software o emprego de ferramentas que auxiliem o processo de desenvolvimento de programas em geral. Dentre as ferramentas existentes, destacam-se os ambientes integrados de desenvolvimento, ou como são conhecidas do inglês, IDE's, que são plataformas que integram uma série de ferramentas necessárias ao desenvolvimento de software como editor de texto, bibliotecas de classes, compiladores, ferramentas de depuração e interfaces gráficas.

Neste trabalho optou-se pela utilização da plataforma de desenvolvimento Visual Studio Community 2019, oferecida gratuitamente pela empresa Microsoft para desenvolvedores individuais ou pequenas equipes.

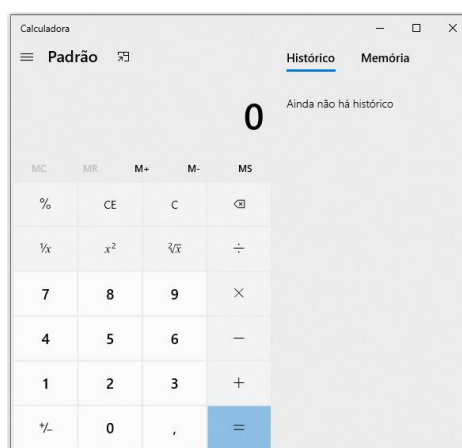
De acordo com a Microsoft (2019b), o Visual Studio Community 2019 oferece uma série de opções de linguagens de programação para execução em diversas plataformas, com centenas de bibliotecas de classes para adição de funcionalidades adicionais disponíveis para incorporação no código fonte dos projetos desenvolvidos na IDE.

Um IDE (ambiente de desenvolvimento integrado) é um programa repleto de recursos que pode ser usado por muitos aspectos do desenvolvimento de software. Além do editor e do depurador padrão fornecidos pela maioria dos IDEs, o Visual Studio inclui compiladores, ferramentas de preenchimento de código, designers gráficos e muitos outros recursos para facilitar o processo de desenvolvimento de software. MICROSOFT (2019b).

Dentre as opções oferecidas pelo Visual Studio Community 2019, optou-se pela criação de um programa de computador para o sistema operacional Windows, por este ser o sistema operacional mais encontrado nos computadores pessoais e já difundido amplamente nos escritórios brasileiros. A linguagem de programação utilizada será a C#, linguagem contida no ambiente de execução .NET, escolhida por ser uma linguagem orientada a objeto, moderna e possuidora de diversas funcionalidades que se mostram úteis ao desenvolvedor.

Nos dias atuais, a grande maioria dos programas de computador possuem interface gráfica (ex. Figura 14), que possibilita a interação do usuário com o programa de forma visual e intuitiva.

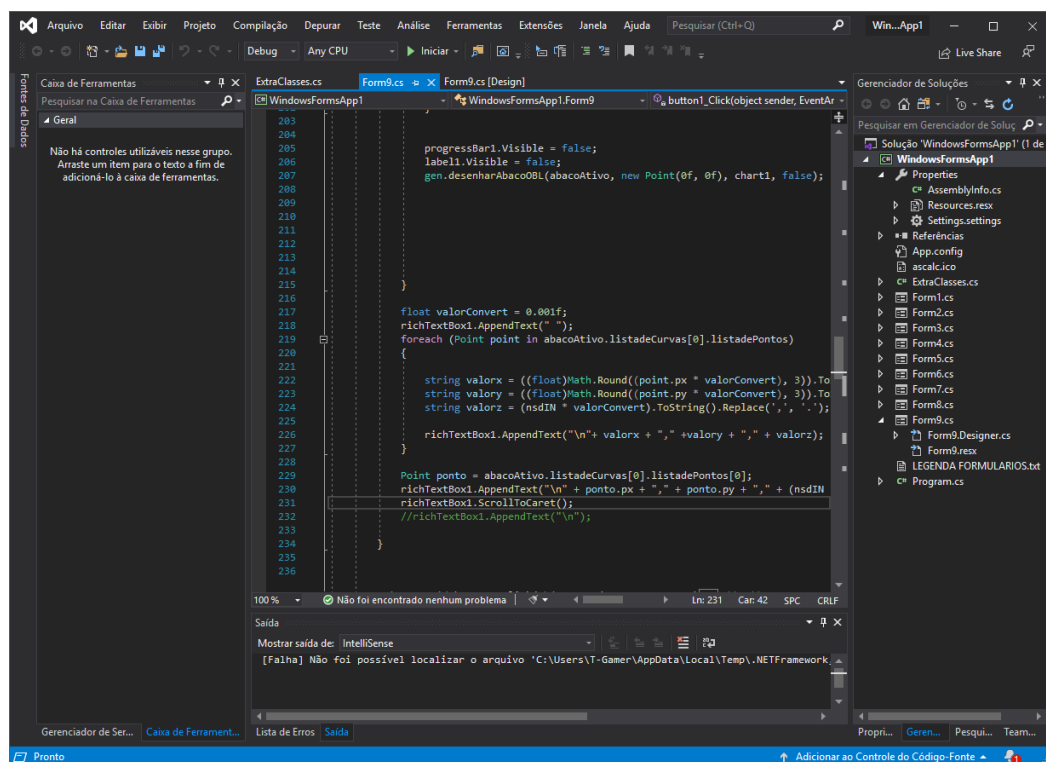
Figura 14 – Exemplo de software com interface gráfica para Windows.



Fonte: Adaptado de MICROSOFT (<https://support.microsoft.com>).

Segundo a Microsoft (2019b), para facilitar o desenvolvimento de programas (ou aplicativos) com interface gráfica para Windows, o Visual Studio Community 2019 disponibiliza uma biblioteca de classes gráficas chamada WinForms que, combinada com um poderoso editor visual e uma biblioteca de controles, possibilita que o desenvolvedor crie facilmente soluções gerais para aplicativos das mais diversas áreas que necessitem processamento de dados. O programa apresentado na Figura 14 é um exemplo do que pode ser feito utilizando-se de tais recursos.

Figura 15 – Ambiente de desenvolvimento Visual Studio Community 2019.



Fonte: Adaptado de MICROSOFT (2019b).

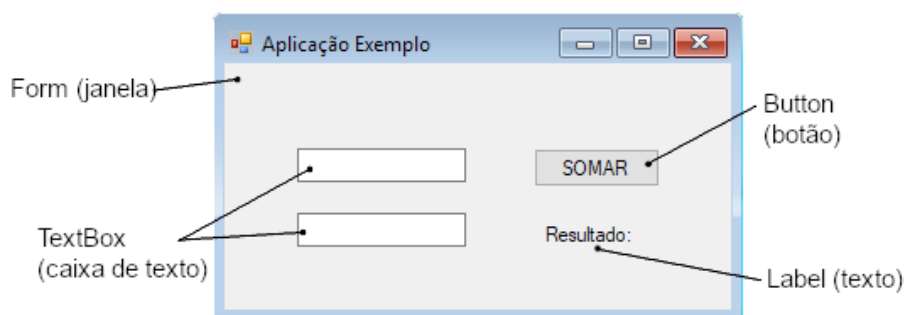
De acordo com a documentação referente à linguagem C#, disponibilizada pela empresa Microsoft (2019a), esta trata-se de uma linguagem orientada a objetos, que se baseiam em classes e que podem ser instanciados em declarações de variáveis ou listas. Ao declarar uma variável, aloca-se um campo de memória no sistema referente ao tipo de variável declarada.

A documentação disponibilizada pela Microsoft (2019a) apresenta os tipos de dados aceitos pela linguagem, sua sintaxe referente a declarações variáveis e objetos de classes, de estruturas de repetição, classes, funções, controles e eventos que, uma vez combinados, podem ser utilizados para a criação de códigos fonte complexos para aplicação no processamento de dados de qualquer natureza.

Este trabalho terá como foco o uso destes recursos para a obtenção de um programa de computador que processe dados de entrada referentes às características de seções retangulares de concreto armado, para então fornecer valores referentes à quantidade de aço que estas seções devem possuir para o devido equilíbrio estrutural junto às solicitações impostas.

Na criação da interface visual da aplicação, será empregado o uso do conjunto de bibliotecas gráficas e controles disponibilizados pela Microsoft denominada WinForms. De acordo com a Microsoft (2019c), seu uso se baseia no uso de controles e eventos, de modo a facilitar a criação das mais diversas disposições de elementos e funcionalidades envolvendo os mesmos.

Figura 16 – Exemplo de aplicativo e os tipos de controle que o compõem.

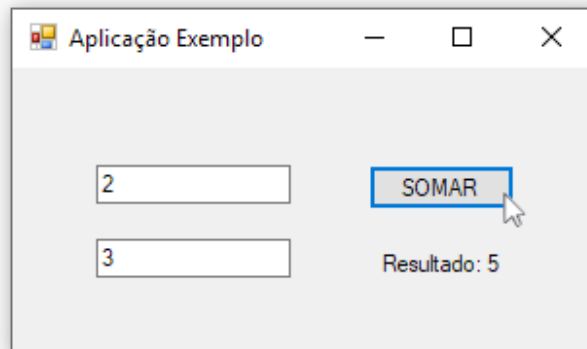


Fonte: Adaptado de MICROSOFT (2019c).

A Figura 16 ilustra o funcionamento do sistema de controles do WinForms através de um aplicativo exemplo, cuja única funcionalidade é a de somar dois valores e exibir seu resultado. O usuário deve entrar com os valores nos campos de texto e clicar no botão *SOMAR*. O botão *SOMAR* ativa um evento que declara campos numéricos na memória (variáveis), lê os caracteres das caixas de texto e os converte em valores numéricos a serem atribuídos às variáveis declaradas. Os valores das

variáveis são então somados e adicionados à propriedade de texto do controle *Label* para a exibição do resultado da soma.

Figura 17 – Funcionamento do aplicativo exemplo.



Fonte: Adaptado de MICROSOFT (2019c).

Na Figura 18 pode-se observar o código fonte que dá funcionalidade ao programa exemplo (Figura 17), cujo arranjo de controles pode ser observado na Figura 16.

Figura 18 – Código fonte do aplicativo exemplo, escrito em linguagem C#.

```

1  using System;
2  using System.Windows.Forms;
3
4
5  namespace WindowsFormsApp1
6  {
7      3 referências
8      public partial class FormExemplo : Form
9      {
10         1 referência
11         public FormExemplo()
12         {
13             InitializeComponent();
14         }
15
16         1 referência
17         private void button1_Click(object sender, EventArgs e) //evento acionado
18         {
19             //declaração das variáveis
20             float valor1, valor2;
21
22             //leitura das caixas de texto e atribuição de valores às variáveis
23             valor1 = (float)Convert.ToDouble(textBox1.Text.Replace('.', ','));
24             valor2 = (float)Convert.ToDouble(textBox2.Text.Replace('.', ','));
25
26             //Exibição do resultado
27             label1.Text = "Resultado: " + (valor1 + valor2).ToString();
28         }
29     }
30 }

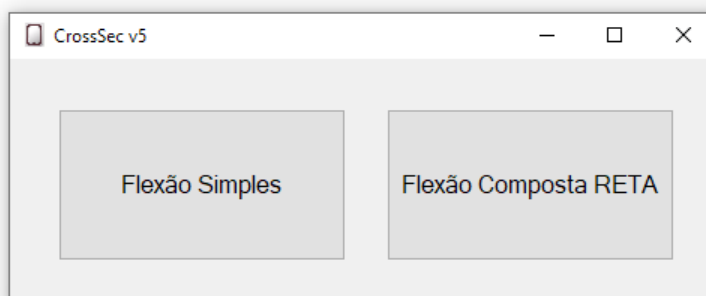
```

Fonte: Adaptado de MICROSOFT (2019c).

3. DESENVOLVIMENTO DA APLICAÇÃO

A aplicação desenvolvida neste trabalho, nomeada *CrossSec*, possui uma interface de usuário composta de três janelas, sendo uma janela inicial e duas para a realização dos cálculos de área de aço contidas nas seções de concreto armado aqui estudadas. A janela inicial é formada por dois controles *Button* (botão), possibilitando ao usuário escolher qual janela de cálculo abrir.

Figura 19 – Programa *CrossSec*, janela inicial.



Fonte: O Autor.

3.1. FLEXÃO SIMPLES

Esta janela de cálculo é formada por uma série de controles utilizados para a entrada de dados e disposição dos resultados dos cálculos, estando todo o código fonte responsável por este cálculo e processamento contido em um evento que é acionado quando o usuário clica no botão *CALCULAR*. Este evento reúne todas as linhas de código necessárias à leitura e processamento dos dados, conforme ilustrado à partir da linha 14 no código fonte exposto na Figura 18. Pode-se observar o arranjo de controles que compõe a janela em pleno funcionamento na Figura 20, em que, uma vez fixados os valores correspondentes às dimensões e características da seção, bem como as solicitações, o botão *CALCULAR* fornece os resultados dos cálculos e verificações e os exibe agrupados na aba *Resultados*.

As operações realizadas no evento do botão *CALCULAR* são baseadas nas equações 2.1.10 e 2.1.8, realizando também as verificações da norma exigidas pela

norma NBR 6118:2014, sendo as equações executadas em código escrito conforme os nomes das variáveis declaradas, tudo na sintaxe da linguagem C#.

Figura 20 – Programa *CrossSec*, janela para cálculo de flexão simples.

CrossSec v5 - Flexão Simples

Armadura: **Simples**

CALCULAR

Informações

Mk: kN.cm

Concreto C:

Aço CA-:

bw: cm

h: cm

acg,est: cm

C. nom: cm

Brita:

φt (estribo): mm

Resultados

d,est = 45 cm

Md = 14000 kN.cm

x2lim = 11,7 cm

x3lim = 28,35 cm

LN = 14,74234 cm

x/d = 0,3276075

As = 8,234649 cm²

As,min = 1,5 cm²

As,max = 40 cm²

Área (Seção) = 1000 cm²

Fonte: O Autor.

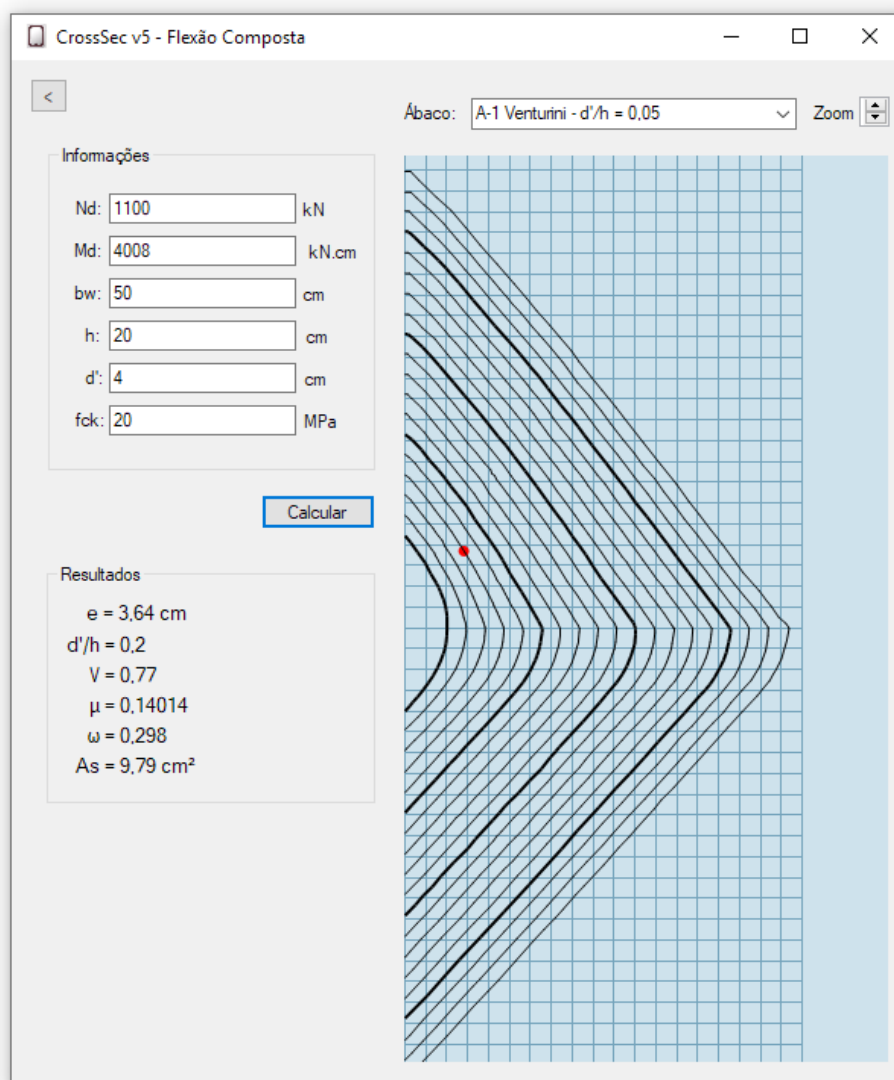
3.2. FLEXÃO COMPOSTA RETA

Similar à janela de flexão simples, esta janela de cálculo é formada por uma série de controles utilizados na entrada de dados e exibição dos resultados dos cálculos, com a diferença de que esta janela possui um controle do tipo *Panel*, que é utilizado para desenhar as linhas necessárias à exibição dos ábacos de Venturini e Rodrigues (1987) contidos no código fonte do programa. As coordenadas dos pontos das linhas que formam os ábacos foram obtidas em um programa escrito à parte

utilizando-se das equações exibidas na Figura 9. A janela possui também um controle do tipo *Dropdown*, utilizado na seleção do ábaco desejado.

Todo o código fonte responsável por esta leitura dos ábacos, cálculo e processamento dos dados está, de maneira similar à da janela de flexão simples, contido em uma única função evento, que é acionado quando o usuário clica no botão *Calcular*. Pode-se observar o arranjo de controles que compõe esta janela, bem como seu funcionamento na Figura 21, em que, uma vez fixados os valores correspondentes às dimensões e características da seção, bem como as solicitações, o botão *Calcular* fornece os resultados dos cálculos e os exibe agrupados na aba *Resultados*.

Figura 21 – Programa *CrossSec*, janela para cálculo de flexão composta reta.



As operações realizadas no evento do botão *Calcular* são baseadas nas equações apresentadas na Figura 10, escritas conforme os nomes das variáveis declaradas, tudo conforme a sintaxe da linguagem C#.

As coordenadas dos pontos das curvas, utilizadas na recriação dos ábacos de Venturini e Rodrigues (1987) foram incorporadas ao código em forma de listas de pares de coordenadas, sendo a leitura dos ábacos realizada no programa *CrossSec* baseada na comparação das coordenadas dos pontos das curvas com a coordenada do ponto analisado, este composto pelos valores da força normal adimensional e do momento fletor adimensional. Os pares de coordenadas podem ser observados na Figura 22, em que cada linha com coordenadas se refere a uma curva de um ábaco.

Figura 22 – Trecho de código fonte responsável por conter o ábaco A1 de Venturini.

```
abacoAtivo = new Abaco();
switch (comboBox1.SelectedIndex)
{
    case 0: //abaco A1
        abacoAtivo.addCurve("0.00,0.0000,0.8431,0.0140,0.8098,0.0298,0.7641,0.0482,0.7123,0.06
        abacoAtivo.addCurve("0.10,0.0000,0.9413,0.0072,0.9280,0.0153,0.9082,0.0308,0.8658,0.04
        abacoAtivo.addCurve("0.20,0.0000,1.0410,0.0136,1.0150,0.0283,0.9770,0.0424,0.9405,0.05
        abacoAtivo.addCurve("0.30,0.0000,1.1362,0.0108,1.1203,0.0258,1.0848,0.0392,1.0520,0.05
        abacoAtivo.addCurve("0.40,0.0000,1.2349,0.0082,1.2224,0.0240,1.1917,0.0382,1.1542,0.05
        abacoAtivo.addCurve("0.50,0.0000,1.3304,0.0167,1.3047,0.0390,1.2573,0.0583,1.2064,0.07
        abacoAtivo.addCurve("0.60,0.0000,1.4283,0.0044,1.4283,0.0118,1.4114,0.0305,1.3719,0.04
        abacoAtivo.addCurve("0.70,0.0000,1.5234,0.0059,1.5234,0.0153,1.5045,0.0404,1.4567,0.06
        abacoAtivo.addCurve("0.80,0.0000,1.6175,0.0072,1.6175,0.0327,1.5674,0.0463,1.5411,0.06
        abacoAtivo.addCurve("0.90,0.0000,1.7181,0.0060,1.7181,0.0226,1.684,0.0554,1.6209,0.078
        abacoAtivo.addCurve("1.00,0.0000,1.8129,0.0064,1.8129,0.0184,1.7934,0.0416,1.7501,0.05
        abacoAtivo.addCurve("1.10,0.0000,1.9073,0.0088,1.9073,0.0279,1.8727,0.0406,1.8439,0.07
        abacoAtivo.addCurve("1.20,0.0000,2.0068,0.0093,2.0068,0.0213,1.9841,0.0409,1.9458,0.06
        abacoAtivo.addCurve("1.30,0.0000,2.1041,0.0110,2.1041,0.0228,2.0805,0.0357,2.0547,0.05
        abacoAtivo.addCurve("1.40,0.0000,2.2037,0.0084,2.2037,0.0279,2.1666,0.0611,2.1051,0.08
        abacoAtivo.addCurve("1.50,0.0000,2.3045,0.0082,2.3045,0.0221,2.2799,0.0448,2.2335,0.06
        abacoAtivo.addCurve("1.60,0.0000,2.4041,0.0089,2.4041,0.0181,2.3852,0.0483,2.3274,0.08
        abacoAtivo.addCurve("1.70,0.0000,2.4947,0.0135,2.4947,0.0269,2.4682,0.0458,2.433,0.062
        abacoAtivo.addCurve("1.80,0.0000,2.5936,0.0132,2.5936,0.0329,2.553,0.0754,2.4692,0.1101,2.4
        abacoAtivo.refine(7);

        break;

    case 1: //abaco A2
        abacoAtivo.addCurve("0.00,0.0000,0.8492,0.0215,0.7918,0.0454,0.7233,0.0671,0.6442,0.0856,0.0
```

Não foi encontrado nenhum problema | Ln: 49 Car: 13 SPC CR

Fonte: O Autor.

4. RESULTADOS

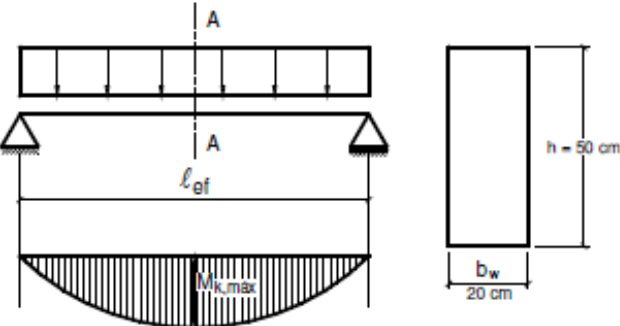
4.1. EXEMPLO 1

O exemplo aqui estudado, que pode ser visualizado na Figura 23, foi elaborado por Bastos (2013) e ilustra a aplicação da formulação utilizada no dimensionamento de vigas de concreto armado submetidas à flexão simples. O objetivo do autor, neste exemplo, é o de mostrar a aplicação das equações teóricas na obtenção da área de aço necessária ao dimensionamento da seção.

Figura 23 – Exemplo numérico de flexão simples formulado por Bastos (2013).

1º) Para a viga indicada na Figura 12, calcular para o momento fletor máximo: a área de armadura longitudinal de flexão. São conhecidos:

$M_{k,max} = 10.000 \text{ kN.cm}$
 $\gamma_c = \gamma_f = 1,4$; $\gamma_s = 1,15$
 concreto C20 ($f_{ck} = 20 \text{ MPa}$)
 aço CA-50
 $\phi_t = 5 \text{ mm}$ (diâmetro do estribo)
 concreto com brita 1 ($d_{max} = 19 \text{ mm}$)
 $h = 50 \text{ cm}$
 $b_w = 20 \text{ cm}$
 $d = 47 \text{ cm}$ (altura útil)
 $c = 2,0 \text{ cm}$ (cobrimento nominal)



Resolução:

$$M_d = \gamma_f \cdot M_k = 1,4 \cdot 10000 = 14.000 \text{ kN.cm}$$

$$x_{2lim} = 0,26 d = 0,26 \cdot 47 = 12,2 \text{ cm}$$

$$x_{3lim} = 0,63 d = 0,63 \cdot 47 = 29,6 \text{ cm}$$

$$x^2 - 117,5 x + 1801,8 = 0 \rightarrow x = 18,1 \text{ cm}$$

$$A_s = \frac{14000}{\frac{50}{1,15} (47 - 0,4 \cdot 18,1)} = 8,10 \text{ cm}^2$$

Fonte: Adaptado de BASTOS (2013, p.13).

O programa *CrossSec* apresentou resultados idênticos aos obtidos por Bastos (2013), como pode ser observado na Figura 24.

Figura 24 – Resultados obtidos com o uso do programa *CrossSec*.

The screenshot shows the 'CrossSec v5 - Flexão Simples' window. It is divided into two main sections: 'Informações' (Information) on the left and 'Resultados' (Results) on the right. A 'CALCULAR' button is located at the top right of the 'Informações' section.

Informações (Input Parameters):

- Armadura: Simples (dropdown)
- Mk: 10000 kN.cm
- Concreto C: 20
- Aço CA: 50
- bw: 20 cm
- h: 50 cm
- acg.est: 3 cm
- C. nom: 2 cm
- Brita: 1 - (19mm) (dropdown)
- φt (estribo): 5 mm

Resultados (Calculated Results):

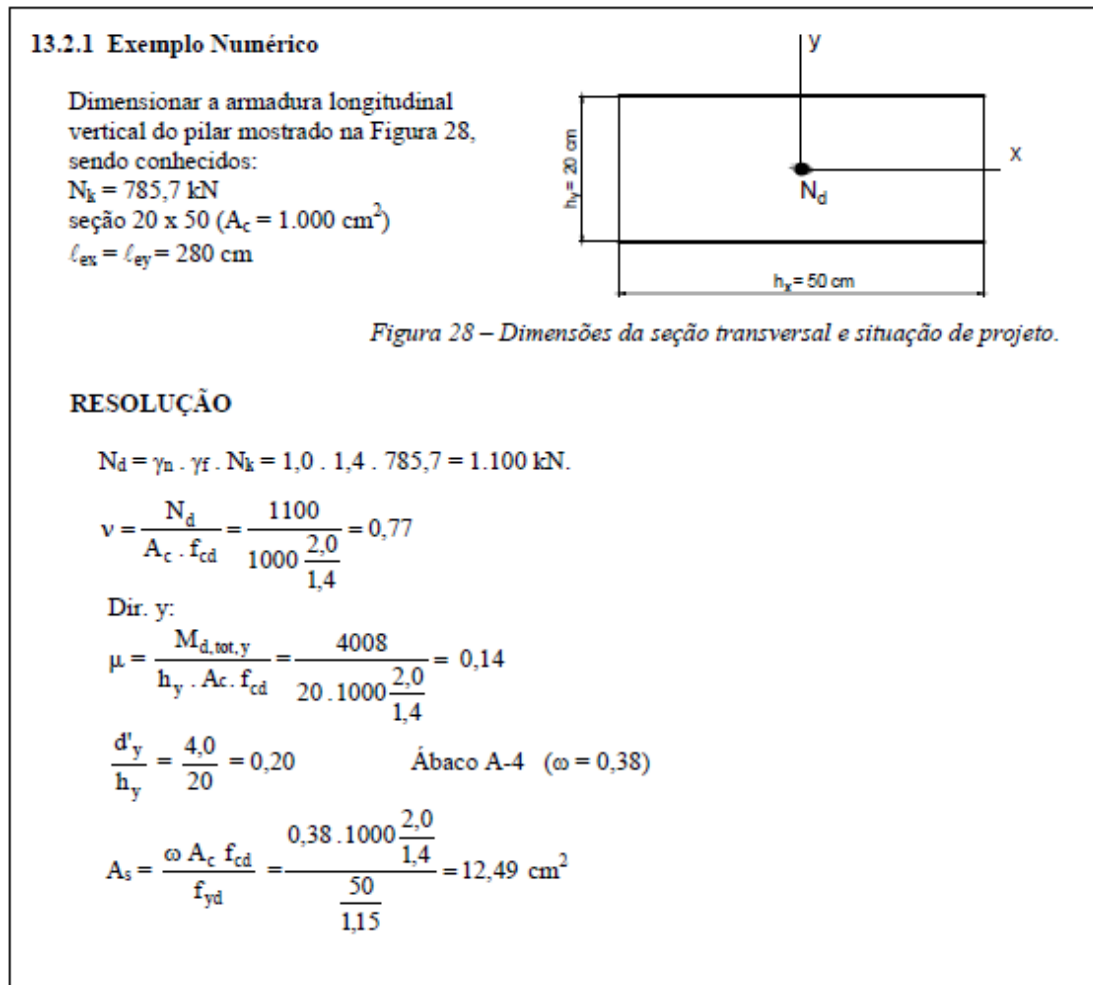
- d,est = 47 cm
- Md = 14000 kN.cm
- x2lim = 12,22 cm
- x3lim = 29,61 cm
- LN = 18,12868 cm
- x/d = 0,3857165
- As = 8,100928 cm²
- As,min = 1,5 cm²
- As,max = 40 cm²
- Área (Seção) = 1000 cm²

Fonte: O Autor.

4.2. EXEMPLO 2

O exemplo aqui estudado pode ser observado na Figura 25, foi elaborado por Bastos (2015) para ilustrar a aplicação da formulação e leitura de ábacos utilizada no dimensionamento de seções retangulares de concreto armado submetidas à flexão composta reta.

Figura 25 – Exemplo numérico de flexão composta reta formulado por Bastos (2005).

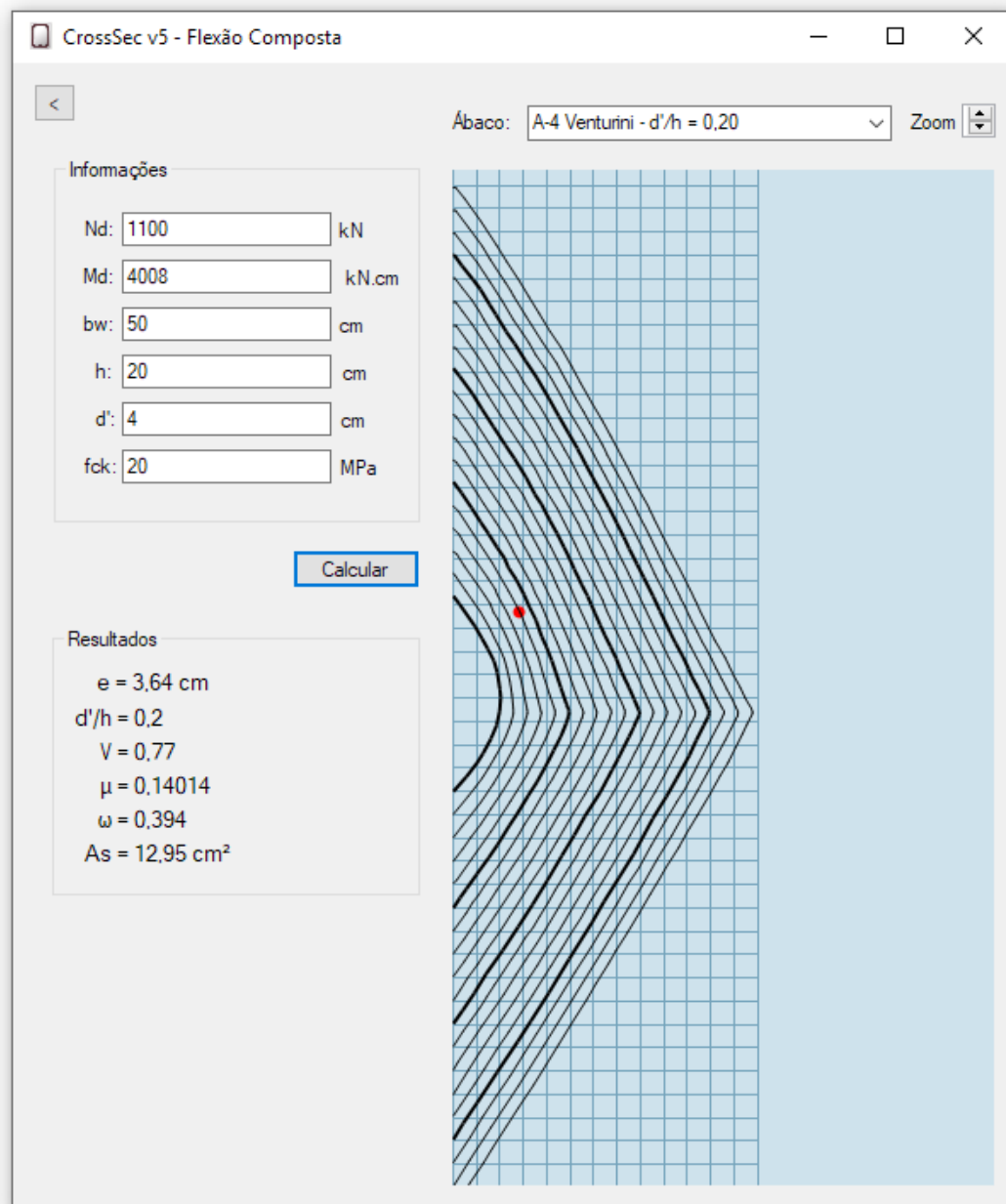


Fonte: Adaptado de BASTOS (2005, p.30).

Os resultados obtidos com o programa *CrossSec* foram muito próximos aos encontrados por Bastos (2005) na resolução de seu exemplo. As pequenas diferenças se devem provavelmente às adaptações envolvidas na transformação de um processo manual de leitura de ábacos em um sistema de leitura automatizada de ábacos.

Como pode-se observar na Figura 26, os resultados obtidos com o programa apresentaram um valor de 3,68% a mais em relação à leitura manual da taxa de armadura ω , realizada por Bastos (2005) em seu exemplo. Esta diferença resultou em uma variação também de 3,68% a mais no valor da área de aço (A_s) calculada pelo programa.

Figura 26 – Resultados obtidos com o uso do programa CrossSec.



Fonte: O Autor.

5. CONCLUSÃO

Os resultados obtidos neste trabalho se mostraram satisfatórios, sendo o conhecimento adquirido muito útil ao trabalho de um engenheiro civil, pois pode ser aplicado em qualquer área da engenharia, servindo como base para trabalhos futuros e estudos considerando o desenvolvimento de aplicações mais complexas.

Através da revisão literária e de estudos realizados, foi possível concluir que é perfeitamente possível a construção de pequenas aplicações pessoais sem conhecimento profundo na área de desenvolvimento de software, habilidade esta que pode se mostrar de grande valia na realização de tarefas manuais e repetitivas com o auxílio de um computador pessoal. Nos dias atuais, não tirar proveito das facilidades que a tecnologia pode oferecer pode resultar em perda de tempo e ineficiência na execução de trabalhos a serem realizados.

O processo de pensamento ao redor do tema se mostrou universal, o que possibilita que a lógica aplicada no desenvolvimento deste trabalho possa ser aplicada na criação de soluções para outras plataformas, como dispositivos móveis e aplicações para web.

REFERÊNCIAS BIBLIOGRÁFICAS

ARAÚJO, J.M. **Programas para dimensionamento e verificação de concreto armado**. 1. ed. Rio Grande: Dunas, 2018.

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

BASTOS, P.S.S **Estruturas de Concreto Armado**. Disciplina 2117 – Estruturas de Concreto I. Bauru/SP, Faculdade de Engenharia - Universidade Estadual Paulista - UNESP, 2014, 9p.

BASTOS, P.S.S. **Flexão normal simples - Vigas**. Disciplina 2117 – Estruturas de Concreto I. Bauru/SP, Faculdade de Engenharia - Universidade Estadual Paulista - UNESP, 2013, 88p.

BASTOS, P.S.S. **Pilares de Concreto Armado**. Disciplina 1309 – Estruturas de Concreto II. Bauru/SP, Faculdade de Engenharia - Universidade Estadual Paulista - UNESP, 2005, 93p.

CARVALHO, R. C.; FILHO, J. R. F. **Cálculo e Detalhamento de Estruturas Usuais de Concreto Armado Segundo a NBR 6118:2014**. v. 1. São Carlos: EdUFSCar, 2014.

GOTARDO, R. **Linguagem de programação I**. Rio de Janeiro: SESES, 2015.

KIMURA, A. **Informática aplicada em estruturas de concreto armado: cálculos de edifícios com o uso de sistemas computacionais**. São Paulo: Pini, 2007.

MICROSOFT, CORP. **Referência C#**. Microsoft, 2019a. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/>. Acesso em: 03 nov. 2019a.

MICROSOFT, CORP. **Documentos do Visual Studio**. Microsoft, 2019c. Disponível em: <https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019>. Acesso em: 03 dez. 2019b.

MICROSOFT, CORP. **Windows Forms**. Microsoft, 2019b. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/framework/winforms/>. Acesso em: 03 nov. 2019c.

MOURA, J. **Pré-dimensionamento de pilares, vigas e lajes de concreto**. Guia da Engenharia, 2018. Disponível em: <https://www.guiadaengenharia.com/pre-dimensionamento-concreto-armado/> - Acesso em: 02 jul. 2020.

NICOLETTI, R.S. **4 formas de reduzir flechas em vigas de concreto armado**. Portal Civil, 2020. Disponível em: <https://portalcivil.com.br/4-formas-de-reduzir-flechas-em-vigas-de-concreto-armado/> - Acesso em: 01 jul. 2020.

VENTURINI, W.S.; RODRIGUES, R.O. **Dimensionamento de peças retangulares de concreto armado solicitadas à flexão reta**. São Carlos, Departamento de Engenharia de Estruturas, Escola de Engenharia de São Carlos – USP, 1987.

ANEXO I

Exemplo de código fonte – Programa *CrossSec* – Janela de flexão composta reta – Função evento do Botão *Calcular*.

```

47     private void Button1_Click(object sender, EventArgs e)
48     {
49         try
50         {
51             Nd = (float)Convert.ToDouble(textBox1.Text.Replace('.', ','));
52             bw = (float)Convert.ToDouble(textBox2.Text.Replace('.', ','));
53             he = (float)Convert.ToDouble(textBox3.Text.Replace('.', ','));
54             Md = (float)Convert.ToDouble(textBox4.Text.Replace('.', ','));
55             dl = (float)Convert.ToDouble(textBox5.Text.Replace('.', ','));
56             fck = (float)Convert.ToDouble(textBox6.Text.Replace('.', ','));
57
58             ex = (float)Math.Round((Md/Nd), 2);
59             dlh = (float)Math.Round((dl / he), 2);
60
61             label20.Text = "d'/h = " + dlh.ToString();
62             label21.Text = "e = " + ex.ToString() + " cm";
63
64             ni = Nd / (bw * he * ((fck / 10f) / 1.4f));
65             mi = (Nd * ex) / (bw * he * he * ((fck / 10f) / 1.4f));
66
67             label8.Text = "v = " + ni.ToString();
68             label7.Text = "μ = " + mi.ToString();
69
70             analyzedPoint.px = mi;
71             analyzedPoint.py = ni;
72
73             w = (float)Math.Round( abacoAtivo.returnW(new Point(mi, ni)) , 3 ) ;
74
75             label3.Text = "w = " + w.ToString();
76
77             As = ( w * (bw * he * ((fck / 10f) / 1.4f))) / (50f/1.15f) ;
78             As = (float)Math.Round(As, 2);
79
80             label19.Text = "As = " + As.ToString() + " cm²";
81
82
83
84             panel1.Refresh();
85         }
86         catch
87         {
88
89         }
90     }
91 }

```