

**UNIVERSIDADE DE CAXIAS DO SUL**

**OSCAR ANTÔNIO TRICHES JUNIOR**

**ELEMENTOS ESTRUTURAIS EM MADEIRA: DESENVOLVIMENTO DE  
PROGRAMA COMPUTACIONAL DE DIMENSIONAMENTO**

**CAXIAS DO SUL  
2018**

**OSCAR ANTÔNIO TRICHES JUNIOR**

**ELEMENTOS ESTRUTURAIS EM MADEIRA: DESENVOLVIMENTO DE  
PROGRAMA COMPUTACIONAL DE DIMENSIONAMENTO**

Monografia entregue como parte dos requisitos para obtenção da aprovação na disciplina de Trabalho de Conclusão de Curso II, do curso de Engenharia Civil da Universidade de Caxias do Sul/RS.

Orientador Me. Gustavo Ribeiro da Silva

**CAXIAS DO SUL  
2018**

**OSCAR ANTÔNIO TRICHES JUNIOR**

**ELEMENTOS ESTRUTURAIS EM MADEIRA:  
DESENVOLVIMENTO DE PROGRAMA COMPUTACIONAL DE  
DIMENSIONAMENTO**

Monografia entregue como parte dos requisitos para obtenção da aprovação na disciplina de Trabalho de Conclusão de Curso II, do curso de Engenharia Civil da Universidade de Caxias do Sul/RS.

Aprovado em 01/12/2018

**BANCA EXAMINADORA**

**Prof. Me. Rodrigo Fabiano Montemezzo**

Universidade de Caxias do Sul - UCS

**Prof. Me. Luciano Zatti**

Universidade de Caxias do Sul - UCS

**Prof. Me. Gustavo Ribeiro da Silva**

Universidade de Caxias do Sul - UCS

## RESUMO

A madeira é um material que apresenta diversas características favoráveis, como sua relação resistência/peso, a sua fácil obtenção de forma sustentável, a facilidade de manuseio do material, um bom isolamento térmico, entre outros. Como a madeira é um material natural está sujeita a ataques biológicos e apresenta diversas imperfeições, como nós e fendas. Porém com o avanço da tecnologia estes aspectos negativos são facilmente tratados com produtos industriais adequados para cada situação. Tornando assim a madeira um material durável e de qualidade.

A utilização de madeira como material estrutural ainda é pouco explorada pelo fato de que ainda existe um grande preconceito com o material pelos fatos citados acima e também pela falta de material de pesquisa sobre este assunto.

Desta forma, o presente trabalho tem por objetivo o desenvolvimento de um programa computacional de dimensionamento de elementos estruturais em madeira, seguindo fielmente as exigências da ABNT NBR 7190:97.

O programa computacional foi desenvolvido na IDE NetBeans 8.2, utilizando a linguagem de programação Java. E foi dividido em nove abas diferentes, onde cada uma possui sua função específica para melhorar o uso do usuário. Dentre estas abas estão as abas de dimensionamento de elementos estruturais em madeira.

## **ABSTRACT**

Wood is a material that shows several favorable characteristics, like its relation between endurance and weight, its sustainability, the materials easiness to handle, the fact that it's a good thermal insulator, among other things. Because wood is a natural building material it's susceptible to biological flaws and has several imperfections, like knots and crevices. Although, with the advancement of technology, these negative aspects are easily handled with the use of industrial products for each situation. Thus, making wood a durable and high quality building material.

The use of wood as a structural material is still very little explored because of the fact that there is still prejudice against the material for the reasons stated above and also for the lack of research on this subject.

Therefore, this study has as its objective the development of a computer program for dimensioning structural elements in wood, following accurately the requirements of ABNT NBR 7190:97.

The software was developed on IDE NetBeans 8.2 using Java coding and it was divided in 9 different tabs, where which of the tabs has its specific function to improve the users experience. Among these flaps are the dimensioning flaps of structural elements in wood.

## LISTA DE FIGURAS

Figura 1 – Camada de um tronco.....	17
Figura 2 – Lenho das coníferas.....	18
Figura 3 – Lenho das dicotiledôneas.....	19
Figura 4 – Vigas de madeira e estruturas metálicas submetidas a um severo incêndio.....	24
Figura 5 – Comportamento da madeira a compressão .....	25
Figura 6 – Comportamento da madeira a tração.....	26
Figura 7 – Cisalhamento na madeira .....	27
Figura 8 – Flexão na madeira.....	27
Figura 9 – Presença de nó na madeira .....	28
Figura 10 – Esquema de corte das toras de madeira.....	31
Figura 11 – Detalhes de emendas de lâminas .....	33
Figura 12 – Treliça <i>Howe</i> .....	34
Figura 13 – Vigamento de apoio das terças.....	35
Figura 14 – Treliça de cobertura e sistemas de contraventamento.....	35
Figura 15 – Vigamento para pisos de madeira.....	36
Figura 16 – Pórticos associados e sistemas de contraventamento.....	37
Figura 17– Sistema estrutural pórtico espacial.....	38
Figura 18 – Elementos de contraventamento vertical para edificações .....	38
Figura 19 – Fluxograma geral .....	62
Figura 20 – Fluxograma de tração .....	63
Figura 21 – Fluxograma de compressão .....	64
Figura 22 – Fluxograma de flexão.....	65
Figura 23 – Fluxograma de cisalhamento .....	66
Figura 24 - Interface .....	66
Figura 25 – Página inicial .....	67
Figura 26 – Comprimento da peça horizontal.....	68
Figura 27 – Carga permanente horizontal.....	68
Figura 28 – Carga variável horizontal.....	68
Figura 29 – Combinações horizontal .....	69
Figura 30 – Carga permanente vertical .....	69
Figura 31 – Carga variável vertical.....	70
Figura 32 – Combinações vertical .....	70
Figura 33 – Classe de resistência .....	71
Figura 34 – Coeficiente de modificação .....	71
Figura 35 – Dados tração .....	72
Figura 36 – Dimensionamento à tração .....	73
Figura 37 – Verificação tração.....	73
Figura 38 – Dados compressão .....	74
Figura 39 – Dimensionamento à compressão .....	75
Figura 40 – Verificações compressão .....	75
Figura 41 – Dados flexão .....	76
Figura 42 – Dimensionamento à flexão.....	77
Figura 43 – Verificação flexão .....	77
Figura 44 – Dimensionamento de travamentos.....	78
Figura 45 – Dados cisalhamento.....	79
Figura 46 – Verificação cisalhamento .....	79

Figura 47 – Dimensionamento ao cisalhamento .....	80
Figura 48 – Resultados .....	81

## LISTA DE TABELAS

Tabela 1 - Variação dimensional da madeira .....	21
Tabela 2 – Exemplos de absorção sonora de alguns materiais .....	22
Tabela 3 – Classes de umidade .....	39
Tabela 4 – Classes de resistência das coníferas .....	40
Tabela 5 – Classes de resistência das dicotiledôneas .....	40
Tabela 6 – Classes de carregamentos .....	41
Tabela 7 – Valores de $k_{mod,1}$ .....	42
Tabela 8 – Valores de $k_{mod,2}$ .....	42
Tabela 9 – Valores de $k_{mod,3}$ .....	43
Tabela 10 – Dimensões mínimas .....	44
Tabela 11 – Ações permanentes de pequena variabilidade .....	45
Tabela 12 – Ações permanentes de grande variabilidade .....	46
Tabela 13 – Ações permanentes indiretos .....	46
Tabela 14 – Ações variáveis .....	47
Tabela 15 – Fatores de combinação e de utilização .....	48
Tabela 16 – Valores de $\alpha_n$ .....	53
Tabela 17 – Coeficiente de fluência .....	57



## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 QUESTÃO DE PESQUISA .....	13
1.2 OBJETIVOS DO TRABALHO .....	14
<b>1.2.1 Objetivo principal</b> .....	<b>14</b>
<b>1.2.2 Objetivos específicos</b> .....	<b>14</b>
1.3 PRESSUPOSTOS .....	14
1.4 PREMISSAS / DELIMITAÇÕES .....	14
1.5 LIMITAÇÕES .....	14
<b>2 REVISÃO BIBLIOGRÁFICA</b> .....	<b>15</b>
2.1 A MADEIRA NO BRASIL .....	15
2.2 ANATOMIA DA MADEIRA .....	16
2.3 TIPOS DE MADEIRAS .....	18
<b>2.3.1 Coníferas</b> .....	<b>18</b>
<b>2.3.2 Dicotiledôneas</b> .....	<b>19</b>
2.4 PROPRIEDADES FÍSICAS .....	19
<b>2.4.1 Umidade</b> .....	<b>19</b>
<b>2.4.2 Densidade</b> .....	<b>20</b>
<b>2.4.3 Propriedades térmicas</b> .....	<b>21</b>
<b>2.4.4 Propriedades acústicas</b> .....	<b>22</b>
<b>2.4.5 Resistividade</b> .....	<b>23</b>
<b>2.4.6 Resistência ao fogo</b> .....	<b>23</b>
2.5 PROPRIEDADES MECÂNICAS .....	25
<b>2.5.1 Compressão</b> .....	<b>25</b>
<b>2.5.2 Tração</b> .....	<b>26</b>
<b>2.5.3 Cisalhamento</b> .....	<b>26</b>
<b>2.5.4 Flexão simples</b> .....	<b>27</b>

<b>2.5.5 Torção</b> .....	<b>28</b>
2.6 PRINCIPAIS FALHAS .....	28
<b>2.6.1 Agentes físicos</b> .....	<b>28</b>
2.6.1.1 Nós .....	28
2.6.1.2 Inclinação das fibras .....	29
2.6.1.3 Formação do tronco.....	29
2.6.1.4 Fendas .....	29
<b>2.6.2 Agentes biológicos</b> .....	<b>29</b>
2.6.2.1 Ataque de fungos.....	29
2.6.2.2 Ataque de insetos .....	30
2.7 PRODUTOS DE MADEIRA .....	30
<b>2.7.1 Madeira roliça</b> .....	<b>30</b>
<b>2.7.2 Madeira serrada</b> .....	<b>30</b>
<b>2.7.3 Madeira compensada</b> .....	<b>31</b>
<b>2.7.4 Madeira laminada e colada</b> .....	<b>32</b>
2.8 SISTEMAS ESTRUTURAIS EM MADEIRA .....	34
<b>2.8.1 Treliças de cobertura</b> .....	<b>34</b>
<b>2.8.2 Vigamento para pisos</b> .....	<b>36</b>
<b>2.8.3 Pórticos</b> .....	<b>36</b>
<b>2.8.4 Pontes em madeira</b> .....	<b>37</b>
<b>2.8.5 Estruturas aporricadas para edificações</b> .....	<b>37</b>
<b>3 DIMENSIONAMENTO DE ESTRUTURAS DE MADEIRA CONFORME A ABNT NBR 7190:97</b> .....	<b>39</b>
3.1 CLASSE DE UMIDADE .....	39
3.2 CLASSE DE RESISTÊNCIA.....	39
3.3 COEFICIENTE DE MODIFICAÇÃO.....	41
<b>3.3.1 Classes de carregamento (<math>k_{mod,1}</math>)</b> .....	<b>41</b>
<b>3.3.2 Classe de umidade (<math>k_{mod,2}</math>)</b> .....	<b>42</b>

<b>3.3.3 Classe de categoria (<math>k_{mod,3}</math>)</b> .....	<b>42</b>
3.4 COEFICIENTE DE MINORAÇÃO DA RESISTÊNCIA .....	43
3.5 RESISTÊNCIA DE CÁLCULO .....	43
3.6 DIMENSÕES MÍNIMAS .....	44
3.7 ESFORÇOS SOLICITANTES DE CÁLCULO – ESTADO LIMITE ÚLTIMO .....	45
<b>3.7.1 Ações permanentes</b> .....	<b>45</b>
3.7.1.1 Ações permanentes de pequena variabilidade .....	45
3.7.1.2 Ações permanentes de grande variabilidade .....	46
3.7.1.3 Ações permanentes indiretos .....	46
<b>3.7.2 Ações variáveis</b> .....	<b>47</b>
<b>3.7.3 Combinações de ações em estados limites últimos</b> .....	<b>47</b>
3.8 ESFORÇO NORMAL DE TRAÇÃO RESISTENTE DE CÁLCULO .	49
<b>3.8.1 Tensão característica à tração paralela as fibras</b> .....	<b>50</b>
<b>3.8.2 Tensão característica à tração normal as fibras</b> .....	<b>50</b>
<b>3.8.3 Tensão característica à tração inclinada em relação as fibras</b> .....	<b>51</b>
<b>3.8.4 Índice de esbeltez</b> .....	<b>51</b>
3.9 ESFORÇO NORMAL DE COMPRESSÃO RESISTENTE DE CÁLCULO .....	52
<b>3.9.1 Tensão característica à compressão</b> .....	<b>52</b>
<b>3.9.2 Estabilidade</b> .....	<b>53</b>
3.9.2.1 Peças curtas .....	54
3.9.2.2 Peças medianamente esbeltas .....	54
3.9.2.3 Peças esbeltas .....	56
3.10 MOMENTO FLETOR RESISTENTE DE CÁLCULO .....	57
<b>3.10.1 Instabilidade lateral</b> .....	<b>58</b>
3.11 ESFORÇO CISALHANTE RESISTENTE DE CÁLCULO .....	60

<b>4. O PROGRAMA COMPUTACIONAL DE DIMENSIONAMENTO DE ELEMENTOS ESTRUTURAIS DE MADEIRA.....</b>	<b>61</b>
4.1 FLUXOGRAMAS .....	61
4.1.1 Fluxograma geral.....	62
4.1.2 Fluxograma de tração .....	63
4.1.3 Fluxograma de compressão .....	64
4.1.4 Fluxograma de flexão.....	64
4.1.5 Fluxograma de cisalhamento .....	65
4.2 INTERFACE .....	66
4.3 ABA PÁGINA INICIAL .....	67
4.4 ABA CARREGAMENTOS .....	67
4.4.1 Carregamento horizontal .....	67
4.4.2 Carregamento vertical.....	69
4.5 ABA PROPRIEDADES .....	70
4.5.1 Classe de resistência .....	70
4.5.2 Coeficiente de modificação da resistência.....	71
4.6 ABA TRAÇÃO .....	72
4.7 ABA COMPRESSÃO.....	73
4.8 ABA FLEXÃO .....	75
4.9 ABA CISALHAMENTO .....	78
4.10 ABA RESULTADOS .....	80
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>82</b>
<b>REFERÊNCIAS.....</b>	<b>83</b>
<b>APÊNDICE A – CÓDIGO-FONTE DO PROGRAMA COMPUTACIONAL .....</b>	<b>84</b>

# 1 INTRODUÇÃO

A madeira é um material que é usado na construção civil desde os primórdios até a atualidade. Onde sua função pode variar de diversas formas, desde a madeira como material estrutural, sendo usado em vigas, colunas e treliças até a utilização como formas para estruturas de concreto armado ou até mesmo para a confecção de forros, assoalhos e esquadrias.

A utilização da madeira como material estrutural vem ganhando espaço pelo seu custo benefício, onde o material é encontrado em vasta quantidade e possui uma elevada relação peso-resistência e uma excelente resistência à compressão e a tração. Fazendo assim com o material seja apropriado para elementos sujeitos a esforços de flexão.

A sustentabilidade é um assunto cada vez mais pertinente em nossa realidade e a madeira é um material altamente sustentável. Desde que seja extraído de maneira correta e seja sempre feito o reflorestamento, a madeira existirá em grande escala e sua utilização na construção civil tende a crescer muito com o decorrer do tempo. Tornando assim a madeira um material totalmente renovável de acordo com o que precisamos para um futuro melhor para nosso planeta.

Ainda existe um grande preconceito das pessoas com a utilização da madeira. Diversas pessoas imaginam a madeira como um material frágil e suscetível à umidade e a ataques físicos e biológicos, porém com o avanço das tecnologias os tratamentos em madeira evoluíram muito fazendo com que estes problemas não venham a acontecer. Outro preconceito é em questão com a resistência ao fogo do material, diversas pessoas nem imaginam que a resistência ao fogo da madeira seja superior ao do concreto armado e do metal.

## 1.1 QUESTÃO DE PESQUISA

Qual é o *software* para dimensionar elementos estruturais em madeira?

## 1.2 OBJETIVOS DO TRABALHO

### 1.2.1 Objetivo principal

O objetivo principal do trabalho é o desenvolvimento de um programa computacional que auxilie o engenheiro no dimensionamento de elementos estruturais de madeira.

### 1.2.2 Objetivos específicos

O objetivo específico do trabalho é apresentar as diretrizes para o processo de dimensionamento de elementos de madeira submetidos à esforços de tração, compressão, cisalhamento e flexão.

## 1.3 PRESSUPOSTOS

O trabalho tem como pressuposto a validade das recomendações que constam na ABNT NBR 7190:97 – Projeto de estruturas de madeira.

## 1.4 PREMISSAS / DELIMITAÇÕES

O programa computacional de dimensionamento delimita-se a análise apenas do Estado Limite Último.

## 1.5 LIMITAÇÕES

O programa de dimensionamento limita-se em analisar apenas elementos isolados.

O peso próprio da peça não é calculado automaticamente.

O programa limita-se em utilizar apenas seções retangulares.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 A MADEIRA NO BRASIL

Nos últimos anos o planeta vem sofrendo severas alterações ambientais devido ao aquecimento global. Com isso o fator de como reduzir os efeitos do aquecimento global é mais importante do que discutir se vai ou não acontecer o mesmo (RUSSO, 2016).

Estudos realizados por Russo (2016) mostram que a construção civil é responsável pela utilização de cerca de 40% e 75% dos recursos naturais do mundo. E no Brasil esse setor é responsável pela geração de 25% do total de resíduos da indústria e 60% do lixo sólido das cidades. Segundo a UNEP (*United Nations Environment Programme*), o setor também é responsável por 40% do consumo global de energia e 30% das emissões globais de gases de efeito estufa.

Porém, segundo Russo (2016), o Brasil está com uma carência de moradias enorme e com isso é preciso investir em residências descentes para a população e conciliar com o fato de sustentabilidade, para não afetar ainda mais o nosso planeta.

Com a necessidade de construções com o mínimo de desperdício e a procura de materiais renováveis e que levem em consideração a eficiência do uso dos recursos e principalmente a redução da emissão de CO<sub>2</sub>, a madeira é o material mais indicado, com a sua utilização de forma adequada e com alta tecnologia embarcada (RUSSO, 2016).

A madeira no Brasil ainda sofre muitos preconceitos e por isso não é utilizada em maior escala na construção civil. Um dos maiores motivos para essa ocorrência é a falta de informação sobre o material no mercado. Segundo uma pesquisa realizada pelo programa Madeira é Legal onde 28 empresas participaram, 71% apontam que não há todas as informações técnicas e de mercado para especificar a madeira em seus projetos (RUSSO, 2016).

Russo (2016) explica que é preciso exercer uma mudança da cultura construtiva, fazendo com que todos os profissionais possam ter clareza nas informações sobre o uso da madeira na construção, deve ser feita a padronização da informação, onde todos devem ter acesso ao mesmo conteúdo. Um mecanismo que pode surgir bastante efeito é a divulgação de experiências dos projetos bem sucedidos

e qualificados, para que assim mostre ao consumidor o potencial do material em questão.

Para Zerbini (2014), a procedência da madeira é de extrema importância para a garantia de que o material é de qualidade. O selo FSC garante que a madeira em questão foi extraída legalmente, com todos os direitos trabalhistas em dia, com a preservação das espécies nativas do ecossistema, com o apoio das comunidades vizinhas ao local de extração.

O FSC no Brasil foi formalizado em 2001, através da criação do Conselho Brasileiro de Manejo Florestal. A organização tem como objetivo principal ser a referência em boas práticas do manejo florestal, garantir uniformidade e credibilidade ao sistema e incentivar o consumo responsável de produtos certificados. O Brasil se encontra atualmente como a sexta maior área certificada do mundo (ZERBINI, 2014).

A certificação está totalmente ligada aos problemas citados anteriormente, pois sua prática garante que florestas não sejam desmatadas, colaborando para a diminuição do aquecimento global. O uso controlado das florestas é de suma importância para que este recurso seja renovável e a emissão de CO<sub>2</sub> seja controlada da melhor forma.

## 2.2 ANATOMIA DA MADEIRA

Em seus estudos Pinheiro (2012), descreve a madeira como um ser vivo de anatomia complexa, que é originada de uma árvore. Esta é dividida em três principais elementos:

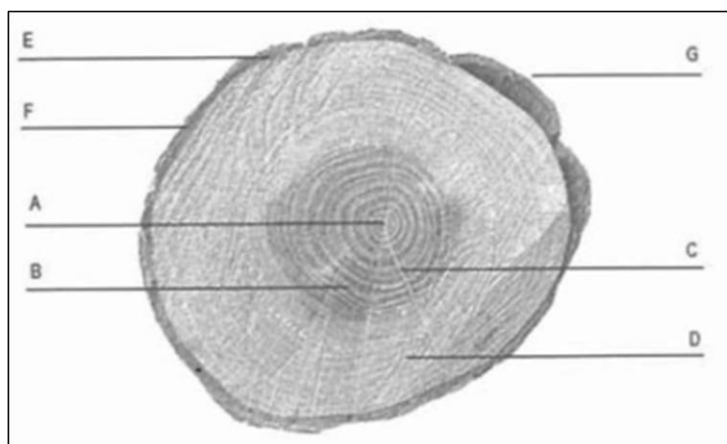
- a) Raiz: é caracterizada como o suporte e também o meio por onde é transportado todos os minerais e água necessária para o crescimento e sobrevivência da árvore;
- b) Caule: Também é conhecido como tronco, que é o elemento vertical e comprido da árvore;
- c) Copa: é o ponto mais alto, onde se encontram os galhos e as folhas da árvore.



Para Pinheiro (2012), o material retirado da árvore com o intuito de uso estrutural é extraído do tronco da árvore. Este é dividido em sete camadas, como mostra a Figura 1. Cada camada tem a sua própria função e característica, são elas:

- a) Medula;
- b) Cerne;
- c) Raio Medular;
- d) Borne;
- e) Cambio;
- f) Liber;
- g) Casca.

Figura 1 – Camada de um tronco



Fonte: Pinheiro (2012).

Ainda segundo Pinheiro (2012), as únicas camadas que possuem as características estruturais são o borne e o cerne. O cerne tem como função a sustentação da árvore, onde não apresenta mais atividades fisiológicas por isso possui uma melhor qualidade. Já o borne tem como principal funcionalidade o transporte de seiva bruta, por isso possui uma menor resistência aos ataques biológicos.

## 2.3 TIPOS DE MADEIRAS

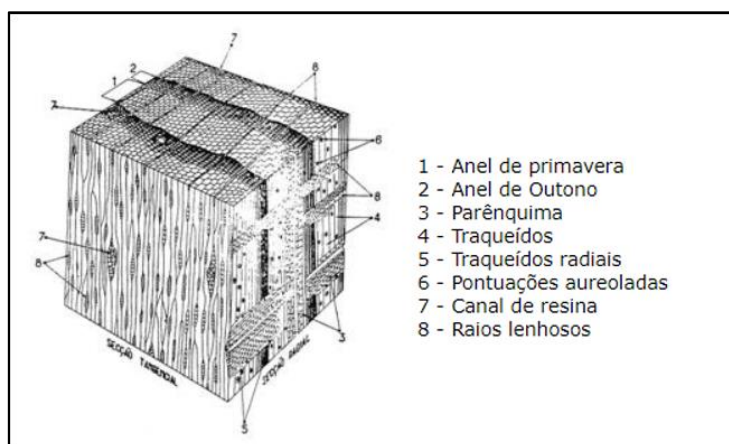
Martins (2010), divide as madeiras em dois grandes grupos de plantas, as gimnospermas também conhecidas como coníferas (*softwood*) e as angiospermas também chamadas de dicotiledôneas (*hardwood*). As características das duas categorias não estão ligadas com sua dureza e sim com a origem botânica delas.

### 2.3.1 Coníferas

As coníferas têm como principal característica o seu crescimento rápido, onde é possível corta-las após 30 anos. Porém o fato de crescer rápido faz com que a madeira fique com uma baixa densidade e com capacidades resistentes reduzidas. As coníferas devem receber um tratamento adequado para aumentar sua durabilidade, pois geralmente este fator é muito baixo. Todavia, pela sua característica principal de crescimento rápido as coníferas são as madeiras com maior disponibilidade no mercado e relativamente as mais baratas (MARTINS, 2010).

Os traqueoides longitudinais constituem cerca de 90% do lenho das coníferas onde eles têm a função de sustentação e condução da água. Algumas das espécies mais utilizadas na construção civil são as diversas espécies de Pinus. A Figura 2 mostra a composição do lenho das coníferas.

Figura 2 – Lenho das coníferas

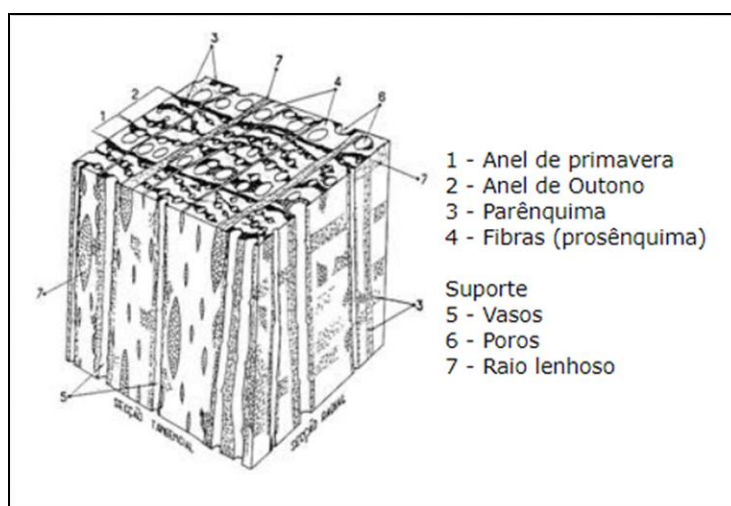


Fonte: Martins (2010).

### 2.3.2 Dicotiledôneas

Para Martins (2010), a principal característica das dicotiledôneas são madeiras de maior densidade e melhores capacidades de resistência, tornando assim o seu crescimento bastante lento, podendo passar dos 100 anos para sazonar. Por ser uma madeira de maior densidade as dicotiledôneas requerem um menor nível de tratamento. Sua composição é muito mais complexa que as coníferas, com as fibras dispostas longitudinalmente no caule, com seu diâmetro variável e dimensões reduzidas. São alguns exemplos de espécies lenhosas utilizadas na construção a família dos Eucaliptos, o Ipê, Angelim, entre outros. A Figura 3 representa o lenho das dicotiledôneas.

Figura 3 – Lenho das dicotiledôneas



Fonte: Martins (2010).

## 2.4 PROPRIEDADES FÍSICAS

### 2.4.1 Umidade

As árvores são seres vivos e por isso precisam de nutrientes, chamados de seiva elaborada, eles são transportados pela água que faz um ciclo da raiz até as folhas. Por isso, se diz que a madeira de uma árvore viva ou recém abatidas possuem um índice de umidade extremamente alto. Para estas condições chama-se, comumente, a madeira de saturada ou “verde” (CALIL, LAHR e DIAS, 2003).

Após ser abatida a madeira perde continuamente umidade ao logo do tempo, até o ponto onde se estabiliza e possui apenas as moléculas de água localizadas no interior das paredes celulares, também chamada de água de impregnação. Após a conclusão desta etapa, a madeira chega a um ponto chamado de ponto de saturação (PS), esta etapa ocorre de maneira rápida e, geralmente, a madeira tende a chegar à um teor de umidade de 20% a 30%. A ABNT NBR 7190:97 – Projeto de Estruturas de Madeira – adota como valor base a umidade em 25% no PS (CALIL, LAHR e DIAS, 2003).

A evaporação da água de impregnação continua ocorrendo, porém de maneira mais lenta, até chegar no ponto que chamamos de umidade de equilíbrio (UE). A velocidade para alcançar o UE varia com a espécie da madeira, a temperatura (T) e a umidade relativa do ar (URA). A ABNT NBR 7190:97 trabalha com o valor de UE = 12% para as condições de T = 20°C e URA = 65% (CALIL, LAHR e DIAS, 2003).

O processo de secagem mencionado acima é de grande importância por ele proporciona a diminuição da densidade da madeira, fazendo assim com que o preço de transporte diminua. Além disso, ajuda na prevenção de ataques de fungos e agentes biológicos, reduz a movimentação dimensional, melhorando seu desempenho nas condições de uso, possibilita um melhor desempenho com acabamentos do tipo tintas e vernizes e aumenta as propriedades de resistência e elasticidade da madeira (CALIL, LAHR e DIAS, 2003).

## **2.4.2 Densidade**

Conforme Calil, Lahr e Dias (2003), a densidade é uma das propriedades mais relevantes que difere as diversas espécies de madeira e revela aonde cada uma deve ser utilizada. A densidade nada mais é que a quantidade de massa que está contida em uma determinada unidade de volume.

A variação dimensional da madeira se dá pelas propriedades de retração e inchamento onde a retração está ligada a perda de umidade e o inchamento ao ganho de umidade. A estabilidade dimensional está diretamente ligada a presença de água no interior da madeira e pode ser referente a três direções diferentes: Axial, radial e tangencial. A direção axial é praticamente desprezível (0,1% a 0,9%), a radial é um pouco mais marcante (2,4% a 11,0%) e a máximas se dão na tangencial (3,5% a

15,0%), fazendo assim com que a madeira tenha uma variação volumétrica de até 27% (CALIL, LAHR e DIAS, 2003).

Segundo Calil, Lahr e Dias (2003), podemos determinar a porcentagem de inchamento e retração total conforme mostra a Tabela 1.

Tabela 1 - Variação dimensional da madeira

Variação dimensional de algumas espécies brasileiras			
Espécie	R (%)	T (%)	Relação T/R
Angelim Pedra	4,3	7	1,6
Cambará	3,6	8,7	2,4
Castanheira	4,7	9,4	2
Cedro	4	5,3	1,3
Cupiúba	4,3	7,1	1,7
Eucalipto Citríodora	6,5	9,6	1,5
Eucalipto Tereticornis	7,3	16,7	2,3
Freijó	6,3	11,7	1,9
Goiabão	8,9	18,8	2,1
Ipê	5,1	7,8	1,5
Jatobá	3,6	6,9	1,9
Louro Preto	4,2	8	1,9
Mandioqueira	4,7	9,3	2
Mogno	3	4,1	1,4
Sucupira	5,9	7,3	1,2
Tatajuba	4,1	5,9	1,4

Fonte: Calil, Lahr e Dias (2003).

### 2.4.3 Propriedades térmicas

As propriedades térmicas são de extrema importância na construção civil pois são elas que proporcionam o conforto térmico da residência. Para Moreschi (2012) a madeira possui quatro principais propriedades térmicas que devem ser levadas em consideração, são elas:

- a) condutividade térmica: representa a taxa de fluxo de calor através da matéria submetida a uma dada temperatura. Esta propriedade é mais relevante em utilização para isolamento térmico e também ajuda na industrialização da madeira, nos processos de aquecimento e secagem do material;

- b) calor específico: representa a capacidade de aquecimento de um material, onde depende da temperatura e do teor de umidade do material. O calor específico da madeira seca mostra que o material é adequado para a fabricação de utensílios como colheres de pau e palitos de fósforo, porém quando o material está úmido seu calor específico aumenta pela presença da água;
- c) transmissão térmica: representa a velocidade com que o material absorve o calor do ambiente. Para a madeira este fator é importante nas áreas de secagem, preservação e laminação;
- d) expansão térmica: representa a variação da dimensão com a troca de temperatura. A madeira possui um baixo índice de expansão térmica, esta vantagem se aplica na fabricação de compensados, chapas aglomeradas e também na propriedade de resistência ao fogo.

#### 2.4.4 Propriedades acústicas

A principal propriedade acústica que deve ser levada em consideração no uso da madeira como material de construção é a capacidade de absorção do som. Moreschi (2012) mostra que a madeira tem um grau de absorção favorável em relação a outros materiais, pois a mesma absorve mais as frequências baixas que as altas. A comparação com outros materiais da construção pode ser vista na Tabela 2.

Tabela 2 – Exemplos de absorção sonora de alguns materiais

Material	Grau de absorção (A)	
	120 Hz	2000 Hz
Janela aberta <sup>1</sup>	1	1
Telhas, cimento, água e vidro	0,1	0,02
Madeira	0,1	0,08
Chapa compensada	0,2	0,1
Chapas isolantes	0,12 - 0,30	0,20 - 0,75

Fonte: Calil, Lahr e Dias (2003).

(1) Referência para fins comparativos com os demais materiais - não há absorção de som em janela aberta: o som apenas sai do ambiente.

### 2.4.5 Resistividade

Moreschi (2012) divide o estudo das propriedades elétricas, ou resistividade, da madeira em três características mais importantes, sendo elas:

- a) Condutividade elétrica: representa a corrente elétrica que passa pelo material submetido a uma dada voltagem. Está diretamente ligada ao teor de umidade, onde até 8% de umidade o aumento da condutividade elétrica é pequeno, porém acima disto o aumento começa a se tornar exponencial. A madeira se considerada seca, ela é um bom isolante elétrico.
- b) Constante dielétrica: determina a quantidade de energia potencial armazenada em um dado volume de material não condutivo. Esta propriedade é utilizada como base para o desenvolvimento de aparelhos de medição de umidade na madeira.
- c) Fator de potência dielétrica: representa a quantidade de energia que foi absorvida, transformada em calor. Várias variáveis estão ligadas a este fator, como a frequência, o teor de umidade e a temperatura. A madeira tem seu potencial variando de 0,01 quando tem uma baixa densidade e está seca até 0,95 para madeiras mais densas e úmidas. O fator de potência dielétrica varia de 0 a 1.

### 2.4.6 Resistência ao fogo

A madeira é composta por elementos como carbono, oxigênio e hidrogênio, por isso é um material combustível. Em geral, a madeira possui um ponto de combustão na superfície quando atinge a temperatura de 300°C, quando existe a presença de chama, e na temperatura de 400°C quando a chama não está presente (CORREIA, 2009).

Correia (2009) ainda afirma que as arestas vivas e as fendas são outros fatores que interferem na temperatura de ignição, pois neles a oxidação se torna mais simples. Por isso quanto mais compacta for a madeira, mais lenta será a velocidade da combustão.

Diferente do que muitos imaginam a madeira é excelente quando se trata de resistência ao fogo. Correia (2009) diz que a estrutura de madeira, quando exposta ao fogo, tem melhor comportamento do que os materiais estruturais mais conhecidos, como o concreto e o aço.

As estruturas de madeira possuem uma camada superficial no exterior onde entram facilmente em combustão, porém logo criam uma camada carbonizada onde sua condutividade térmica corresponde a 1/6 da madeira maciça, fazendo assim com que o processo de combustão não afete o interior da sua secção (CORREIA, 2009).

Segundo Correia (2009), as estruturas de madeira mesmo quando submetidas a um incêndio com temperaturas que podem atingir 1000°C conservam uma boa resistência mecânica pelo fato de possuir a camada externa carbonizada. Diferente das estruturas metálicas onde a resistência mecânica diminui de maneira drástica acima de temperaturas da ordem de 300°C, pelo fato da plastificação do material, como mostra a Figura 4.

Figura 4 – Vigas de madeira e estruturas metálicas submetidas a um severo incêndio



Fonte: Correia (2009).



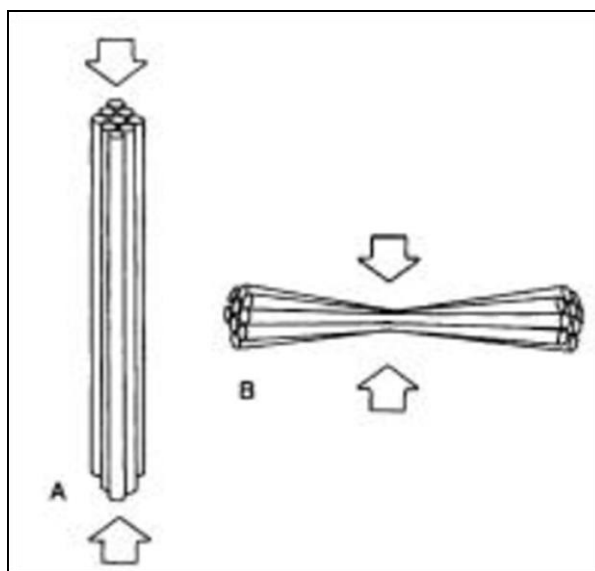
## 2.5 PROPRIEDADES MECÂNICAS

### 2.5.1 Compressão

Os esforços de compressão de uma peça de madeira, segundo Calil, Lahr e Dias (2003), podem ser associados a esforços perpendiculares as fibras ou a esforços paralelos as fibras, onde cada esforço possui um efeito diferente em relação ao material.

Quando uma peça de madeira é solicitada a esforço de compressão paralela as fibras, a solicitação está ocorrendo no eixo de maior resistência do material, fazendo assim com que a mesma tenha uma grande resistência. Porém, quando é solicitada a esforço de compressão perpendicular às fibras o esforço ocorre no eixo de menor resistência do material, fazendo com que a resistência seja até  $\frac{1}{4}$  do valor do eixo paralelo as fibras, como mostra na Figura 5, onde a peça (A) e a peça (B) estão solicitadas aos esforços de compressão paralelo e perpendicular as fibras, respectivamente (CALIL, LAHR e DIAS).

Figura 5 – Comportamento da madeira a compressão



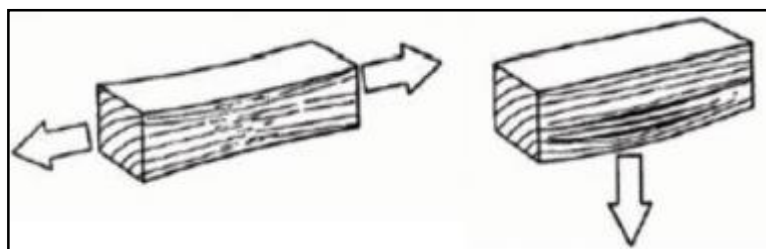
Fonte: Calil, Lahr e Dias apud Ritter (1990).

## 2.5.2 Tração

Os esforços de tração possuem a mesma característica dos esforços de compressão onde podem ocorrer em dois diferentes sentidos, são eles o esforço de tração paralelo as fibras e o esforço de tração perpendicular as fibras.

A ruptura por tração paralela as fibras podem ocorrer tanto por deslizamento entre as fibras quanto por ruptura das paredes do material. Em qualquer dos casos a madeira possui uma grande resistência por estar em seu eixo de maior resistência, logo sua taxa de deformação é baixa. O oposto ocorre quando a ruptura acontece na direção perpendicular as fibras, pois o esforço tende a separar as fibras, fazendo assim com que a resistência diminua drasticamente. A Figura 6 mostra as sollicitações paralela e perpendicular as fibras, respectivamente (CALIL, LAHR e DIAS, 2003).

Figura 6 – Comportamento da madeira a tração



Fonte: Calil, Lahr e Dias apud Ritter (1990).

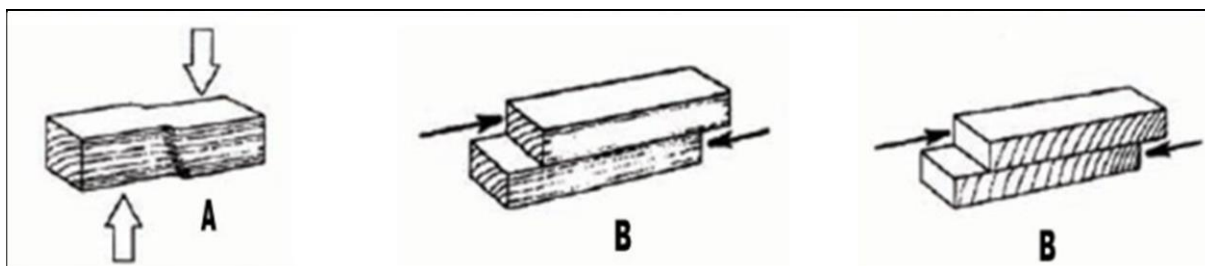
## 2.5.3 Cisalhamento

Como citado anteriormente, nos esforços de tração e compressão, a direção do plano de atuação das tensões influencia diretamente na resistência da madeira, mesmo ocorre com o cisalhamento. Calil, Lahr e Dias (2003) dizem que quando o plano de atuação das tensões é vertical, existe apenas a situação onde as tensões estão sendo aplicadas na direção perpendicular as fibras como mostra a Figura 7, situação A. A madeira possui uma ótima resistência neste caso, pelo fato da ruptura ter que cisalhar estes elementos. Antes da ruptura do material acontecer certamente haverá problemas de resistência à compressão normal.

Para o plano de ação sendo aplicado no sentido horizontal, duas situações distintas podem ocorrer. A primeira delas é quando a direção das tensões aplicadas é

o mesmo da direção das fibras, como mostra a Figura 7, na situação B, neste caso ocorre o cisalhamento horizontal, onde a resistência da madeira é bastante baixa. A segunda situação é quando as tensões são aplicadas na direção perpendicular as fibras, como mostra a Figura 7, na situação B, onde ocorre o cisalhamento por rolamento (CALIL, LAHR e DIAS, 2003).

Figura 7 – Cisalhamento na madeira



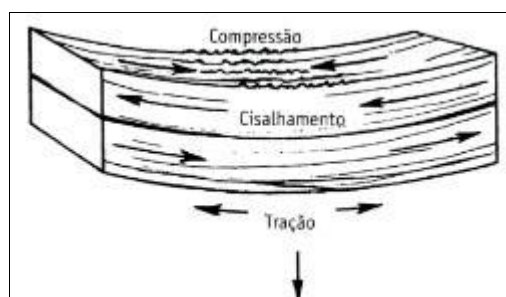
Fonte: Calil, Lahr e Dias apud Ritter (1990).

#### 2.5.4 Flexão simples

Para a situação de flexão simples Calil, Lahr e Dias (2003) dizem que a peça de madeira é solicitada por tensões de compressão e de tração paralela as fibras e tensões de cisalhamento horizontal, conforme mostra a Figura 8. Nas localidades dos apoios ocorre a compressão normal às fibras.

A ruptura por flexão nas peças em madeira ocorre normalmente pela formação de minúsculas falhas de compressão seguidas pelo desenvolvimento de enrugamentos de compressão macroscópico. Com isso ocorre o aumento da área comprimida e a diminuição da área tracionada (CALIL, LAHR e DIAS, 2003).

Figura 8 – Flexão na madeira



Fonte: Calil, Lahr e Dias apud Ritter (1990).

### 2.5.5 Torção

Segundo Moreschi (2012), a resistência à torção da madeira é possível de ser calculada, porém dificilmente é exigida. Calil, Lahr e Dias (2003) apontam que o risco de ruptura por torção normal às fibras é decorrente do estado múltiplo de tensões atuantes na seção e, por isso, a norma brasileira recomenda evitar este parâmetro.

## 2.6 PRINCIPAIS FALHAS

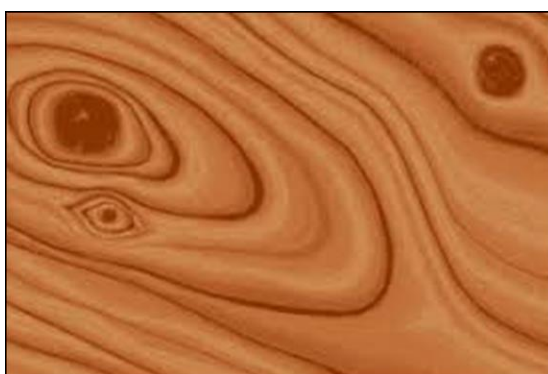
### 2.6.1 Agentes físicos

Por ser um material biológico, Pinheiro (2012) diz que é normal a presença de defeitos e imperfeições na madeira que vão desde a formação do tronco até a presença de nós ou fendas na seção do material.

#### 2.6.1.1 Nós

Calil, Lahr e Dias (2003) mostram que os nós são formados pelos galhos adjuntos ao tronco como mostra a Figura 9, e sua presença na seção da madeira reduz a sua resistência pelo fato de interromper a continuidade e sentido das fibras. A redução da resistência depende do seu tamanho, localização, forma e do tipo de solicitação analisada. Normalmente os nós exercem maiores reduções quando a solicitação é de tensões de tração.

Figura 9 – Presença de nó na madeira



Fonte: Pinheiro (2012).

### 2.6.1.2 Inclinação das fibras

A inclinação das fibras, segundo Calil, Lahr e Dias (2003), acontece quando as fibras não estão na direção paralela em relação a borda da peça. Com isso dependendo do ângulo de inclinação dessas fibras elas possuem uma grande influência na resistência da madeira. A norma brasileira permite uma angulação das fibras de até seis graus, ultrapassando isso uma análise diferente deve ser feita.

### 2.6.1.3 Formação do tronco

A árvore sempre procura o melhor caminho para seu desenvolvimento, Calil, Lahr e Dias (2003) dizem que isso usualmente faz com que ela cresça de maneira mais inclinada fazendo assim com que o alinhamento das fibras se altere influenciando diretamente na resistência do material.

### 2.6.1.4 Fendas

Segundo Martins (2010), as fendas são formadas no processo de secagem do material, onde ocorre a produção de tensões internas do material oriundas dos gradientes de umidade que geram retrações diferentes na massa lenhosa, fazendo assim com que ocorra a produção de tensões internas.

As fendas fazem com que a área de resistente sofra uma diminuição, fazendo assim com o a resistência à esforços cortantes diminua consideravelmente. Para os esforços de flexão a sua magnitude depende de onde as maiores tensões tangenciais são aplicadas, ou seja, depende diretamente do afastamento desta do plano neutro (MARTINS, 2010).

## 2.6.2 Agentes biológicos

### 2.6.2.1 Ataque de fungos

O ataque de fungos na madeira é uma falha que deve ser levada muito a sério pois a medida que vai se agravando a madeira vai apodrecendo fazendo com que a

resistência mecânica decaia até mesmo podendo chegar a zero. Esta falha pode ainda se agravar se o ambiente onde se encontra o material possua um alto índice de umidade (MARTINS, 2010).

#### 2.6.2.2 Ataque de insetos

Para Martins (2010), a gravidade dos ataques de insetos está diretamente ligada ao fato de os agentes xilófagos estarem ou não em atividade. Se ainda estiverem em atividade, a resistência mecânica da madeira pode diminuir drasticamente. Já se não estiverem mais em atividade, os furos ocasionados pela saída dos insetos não são de grande influência na resistência mecânica da madeira.

## 2.7 PRODUTOS DE MADEIRA

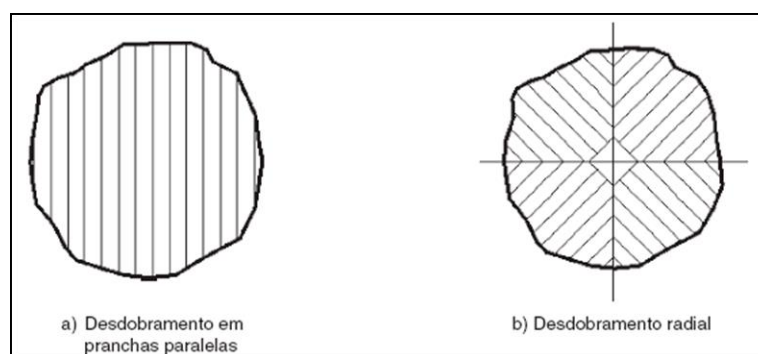
### 2.7.1 Madeira roliça

Para Pfeil (2003), a madeira roliça é utilizada normalmente em construções provisórias, com a função de escoras. O processo de secagem é de extrema importância para que não ocorra retrações transversais, resultando em rachaduras. Os roliços mais usados no Brasil são o pinho-do-paraná e o eucalipto.

### 2.7.2 Madeira serrada

Conforme Pfeil (2003), a madeira serrada é utilizada normalmente de modo estrutural, por isso as árvores devem ser abatidas quando atingem a maturidade, para que o cerne ocupe a maior parte do tronco. O corte pode ser executado de duas principais maneiras, em pranchas paralelas ou radial, como mostra a Figura 10.

Figura 10 – Esquema de corte das toras de madeira



Fonte: Pfeil (2003).

A melhor época para fazer o abatimento das árvores é em estações secas, para que o processo de secagem seja feito em menos tempo. O processo de secagem natural é feito em um galpão protegido da chuva e com alta taxa de ventilação onde as peças são empilhadas com o uso de separadores para que o ar circule entre todas as faces, este processo demanda bastante tempo podendo levar até 3 anos. Por isso foram desenvolvidos métodos de secagem artificiais, onde é feito a circulação de ar quente entre as peças empilhadas, com a temperatura controlada para que não haja a evaporação rápida demais, podendo prejudicar a durabilidade das peças (PFEIL, 2003).

### 2.7.3 Madeira compensada

Pfeil (2003) diz que a madeira compensada é um produto isotrópico onde é formada pela colagem de 3 ou mais lâminas alternando o sentido das fibras ortogonalmente. A espessura das lâminas pode variar de 1mm a 5mm. A secagem pode ser feita naturalmente onde as lâminas são armazenadas em galpões bem arejados. Para economia de tempo, pode-se utilizar a secagem artificial, onde a duração é muito rápida. Esta é realizada por prensas onde a temperatura varia de 80°C a 100°C.

Os compensados podem ser utilizados tanto na utilização a seco, na construção de armários, portas e divisórias, quanto sujeito a variações de umidade, em situações estruturais. As chapas de compensado são confeccionadas em tamanho padrão de 2,5m x 1,25m podendo variar sua espessura de 4mm a 30mm (PFEIL, 2003).

Segundo Pfeil (2003) a madeira compensada apresenta como ponto negativo um preço mais elevado em relação a madeira maciça, em compensação apresenta diversos fatores positivos:

- a) pode ser fabricado em folhas maiores, com o controle dos defeitos;
- b) reduz retração e inchamento, pelo fato da direção das fibras serem ortogonais de uma lâmina para a outra;
- c) maior resistência na direção normal às fibras;
- d) reduz trincas na cravação de pregos;
- e) permite o emprego de madeiras mais resistente nas capas externas e menos resistentes nas camadas interiores.

#### **2.7.4 Madeira laminada e colada**

Pfeil (2003), coloca que a madeira laminada e colada tem como principal característica a função estrutural. A mesma é composta por lâminas de madeira com fibras paralelas entre si. A fabricação das peças é dividida em cinco etapas:

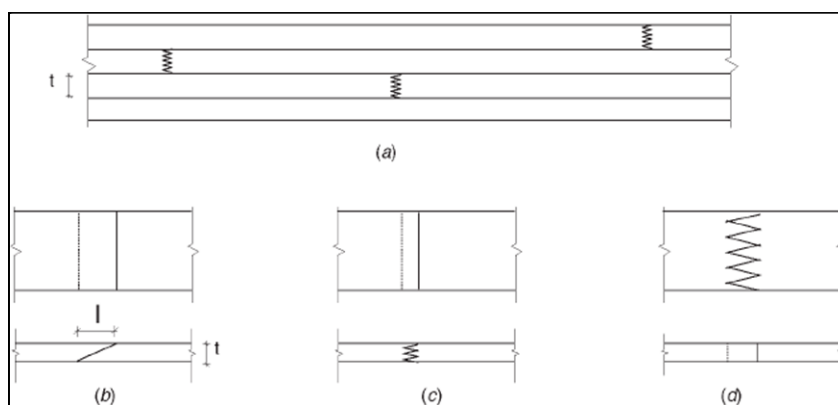
- a) Secagem das lâminas: O processo de secagem demora alguns dias, onde a madeira deve sair do processo com um grau de umidade máxima de 15% e a variação entre as lâminas não pode ser superior a 5%, para que não ocorra retração diferencial entre as lâminas resultando em tensões internas.
- b) Preparo das lâminas: O preparo das lâminas é feito aplainando as peças e as serrando nas dimensões desejadas.
- c) Execução de juntas e emendas: As emendas das lâminas normalmente são realizadas de forma desordenadas ao longo da peça como mostra a Figura 11. As juntas podem ser confeccionadas por cortes em chanfros (Figura 11), onde a sua eficiência depende da inclinação, quanto maior a inclinação melhor a eficiência. As juntas também poder ser realizadas com emendas denteadas, que por sua vez são mais eficientes que as chanfradas. Existe dois tipos de emendas denteadas, a horizontal e a vertical, como mostra a Figura 11.



- d) Colagem sob pressão: O tipo de cola a se usar depende da função que a peça irá exercer. Para a utilização a seco a cola que deve ser empregada é a caseína. Já para utilização em vigas sujeitas a umidade ou expostas ao tempo deve ser empregada colas sintéticas. Estas colagens são realizadas em uma máquina de prensa onde é exercida uma pressão variável de 0,7 a 1,5 MPa. A quantidade de cola utilizada é de 250g por metro quadrado de superfície colada.
- e) Acabamento e tratamento preservativo: Por fim é realizada a parte de acabamento da peça e seus tratamentos preservativos para que resulte em uma peça com alta durabilidade e resistência elevada.

Existem inúmeras vantagens entre a madeira laminada e colada em relação a madeira maciça, como a confecção de peças com grandes dimensões, melhor controle da umidade das lâminas, a seleção da qualidade das lâminas e a confecção de peças de eixo curvo, sendo que, a sua maior desvantagem é o seu custo elevado (PFEIL, 2003).

Figura 11 – Detalhes de emendas de lâminas



Fonte: Pfeil (2003).

(a) distribuição das emendas na direção longitudinal; (b) junta por corte em chanfro; (c) emenda denteada vertical; (d) emenda denteada horizontal.

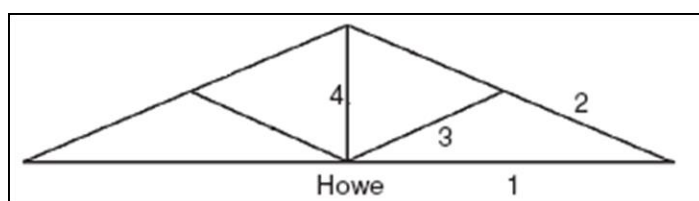
## 2.8 SISTEMAS ESTRUTURAIS EM MADEIRA

### 2.8.1 Treliças de cobertura

Conforme Pfeil (2003), as treliças de coberturas são um dos sistemas estruturais mais utilizado em madeira. Tem como função sustentar o peso próprio das telhas, o peso próprio do vigamento e as forças de ações do vento.

A treliça *Howe* é a mais utilizada na proposta da madeira como material estrutural. Os seus componentes podem ser visualizados na Figura 12, onde o montante e o banzo inferior estão solicitados à tração e o banzo superior e a diagonal estão solicitados à compressão (PFEIL, 2003).

Figura 12 – Treliça *Howe*



Fonte: Pfeil (2003).

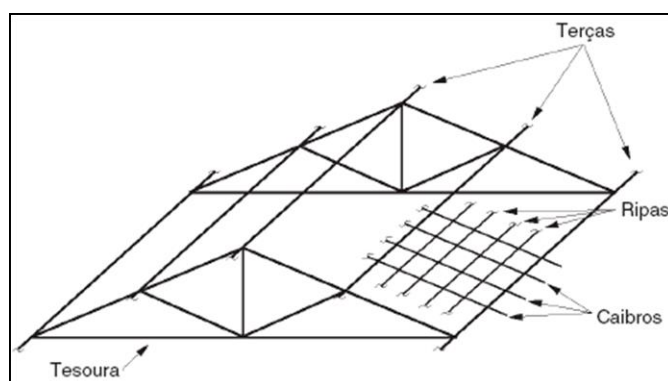
(1) banzo inferior; (2) banzo superior; (3) diagonal; (4) montante

Segundo Pfeil (2003), a inclinação do telhado depende da escolha da telha que será utilizada, podendo variar a inclinação mínima entre o plano do telhado com o plano horizontal em 25° para telhas cerâmicas e 2° para telhas metálicas.

Para o uso de telhas cerâmicas o vigamento é composto por três elementos, conforme mostra a Figura 13:

- Terças: tem a função de viga, vencendo o vão entre as treliças e se apoiando em seus nós;
- Caibros: são apoiados nas terças com espaçamento de 40cm a 60cm;
- Ripas: tem como objetivo o apoio das telhas, seu espaçamento varia conforme o comprimento da telha.

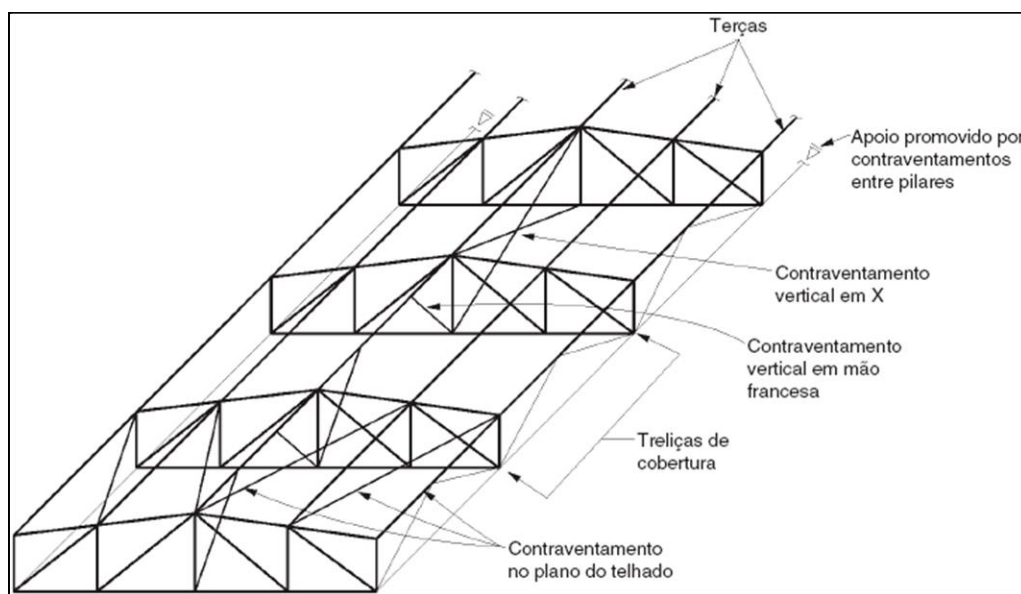
Figura 13 – Vigamento de apoio das terças



Fonte: Pfeil (2003).

O sistema de contraventamento promove a estabilidade vertical do sistema, onde os mais utilizados são o contraventamento no plano do telhado, o vertical em X ou em mão-francesa conforme Figura 14. A união dos dois sistemas promove um sistema tridimensional capaz de resistir às ações do vento originárias de qualquer direção horizontal. Além desta função, este sistema ajuda na redução do comprimento de flambagem fora dos planos verticais das treliças.

Figura 14 – Treliça de cobertura e sistemas de contraventamento

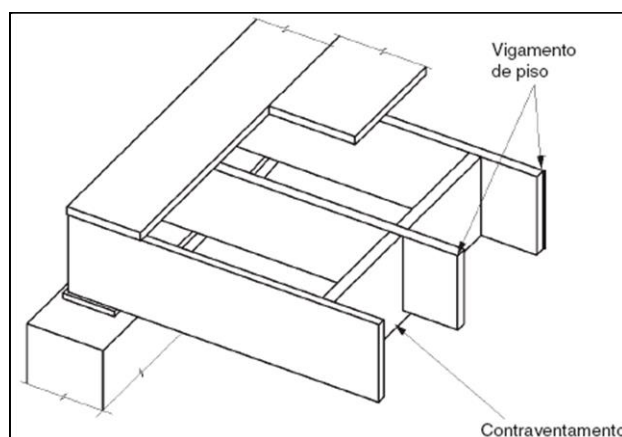


Fonte: Pfeil (2003).

## 2.8.2 Vigamento para pisos

Segundo Pfeil (2003), as vigas utilizadas para apoio do piso são biapoiadas de seção retangular ou I, com espaçamento normalmente de 50 cm e revestida por tábuas como mostra a Figura 15. O dimensionamento normalmente é feito levando em consideração uma carga estática uniformemente distribuída, porém deve-se levar em consideração a vibração excessiva resultante do caminhar das pessoas. Esse critério de estado limite de utilização pode ser encontrado de forma simplificada na ABNT NBR 7190:97. A utilização de contraventamento no piso ajuda na distribuição das cargas entre as vigas, resultando em uma melhora nesta questão da vibração.

Figura 15 – Vigamento para pisos de madeira



Fonte: Pfeil (2003).

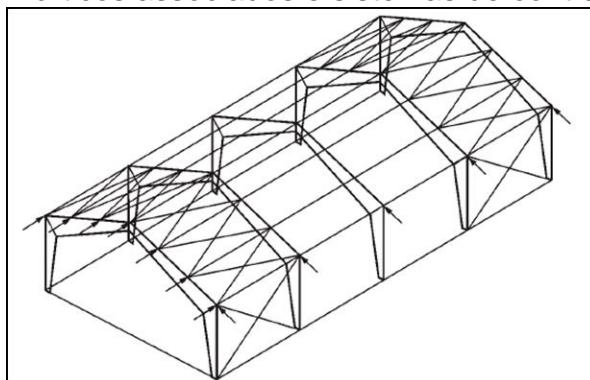
## 2.8.3 Pórticos

Os pórticos têm como sua maior característica a capacidade de vencer grandes vãos livres, podendo variar de 20m a 100m e, por isso, normalmente são utilizados para construções como galpões, estações rodoviárias e ginásios de esportes. Normalmente, são estruturas biarticuladas ou triarticuladas, onde por questão de agilidade na execução é utilizada na maioria das vezes as estruturas triarticuladas (PFEIL, 2003).

Ainda, conforme Pfeil (2003), há variados estilos de pórticos e todos são ligados pelas terças e pelos contraventamentos, como mostra a Figura 16. As cargas das ações do vento são transmitidas para os pilares pelos contraventamentos no plano do telhado e, subsequentemente, são transferidas para as fundações pelos

contraventamentos verticais, onde os mesmos ajudam na rigidez do conjunto na direção longitudinal.

Figura 16 – Pórticos associados e sistemas de contraventamento



Fonte: Pfeil (2003).

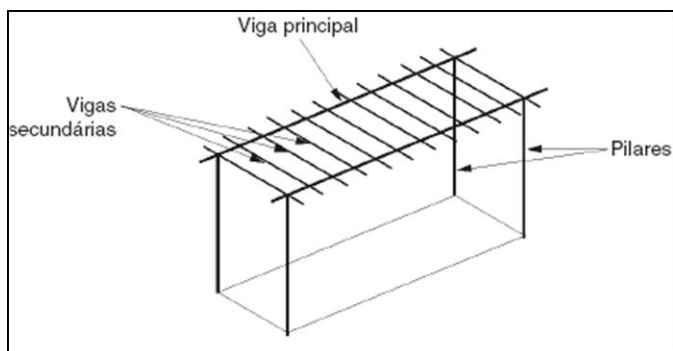
## 2.8.4 Pontes em madeira

Para Pfeil (2003), o principal aspecto do projeto de uma ponte é a sua durabilidade. Para pontes projetadas em madeira, antigamente, eram utilizadas coberturas para que a durabilidade da estrutura fosse alongada, porém, atualmente, com todos os sistemas de tratamento da madeira já é possível realizar pontes sem o uso de coberturas, fazendo o devido tratamento e sua impermeabilização. As pontes normalmente são realizadas por meio de treliças, pórticos ou arcos de diversas dimensões.

## 2.8.5 Estruturas aporticadas para edificações

As estruturas para edificações recebem esse nome de aporticadas, segundo Pfeil (2003), pelo fato de que são compostas por pisos construídos de grelhas planas e suas vigas principais são apoiadas nos pilares, com isso é formado um sistema de pórtico espacial. A transferência de carga ocorre das vigas secundárias para as vigas primárias e, em seguida, para os pilares, a Figura 17 ilustra o sistema. A ligação entre esses elementos, quando forem de seção simples pode ser feita pelo contato da viga com o pilar para edificações de até um pavimento, já quando forem de seções duplas, é permitido a continuidade da viga e do pilar e com isso a ligação é feita com o uso de conectores ou pinos metálicos.

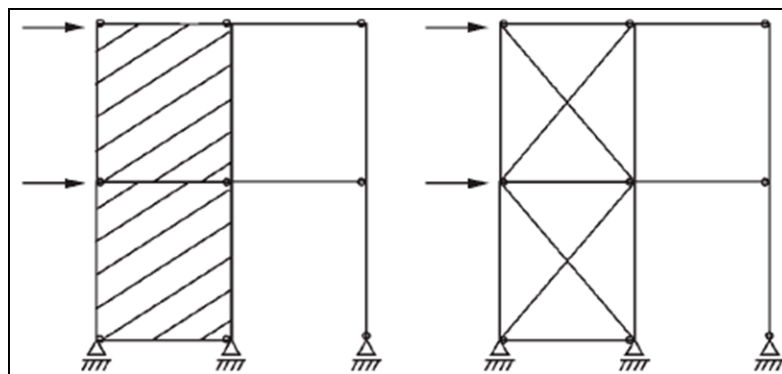
Figura 17– Sistema estrutural pórtico espacial



Fonte: Pfeil (2003).

Como as outras estruturas de madeira, Pfeil (2003) explica que a estabilidade da estrutura depende de sua rigidez. Com isso, se as ligações forem rígidas os esforços horizontais atuam diretamente sobre o pórtico, mas caso as ligações forem flexíveis estes esforços devem ser contidos por um sistema de contraventamento com paredes diafragmas ou treliçados em X, conforme mostra a Figura 18.

Figura 18 – Elementos de contraventamento vertical para edificações



Fonte: Pfeil (2003).

### 3 DIMENSIONAMENTO DE ESTRUTURAS DE MADEIRA CONFORME A ABNT NBR 7190:97

#### 3.1 CLASSE DE UMIDADE

As classes de umidade da madeira, conforme a ABNT NBR 7190:97, tem por finalidade o ajuste das propriedades de rigidez e resistência da madeira em função das condições do ambiente onde as estruturas serão utilizadas. Esta relação entre o ambiente onde as estruturas permanecerão e a umidade de equilíbrio da madeira será utilizada posteriormente no cálculo do coeficiente de modificação 2 e está disposta na Tabela 3.

Tabela 3 – Classes de umidade

Classes de umidade	Umidade relativa do ambiente $U_{amb}$	Umidade de equilíbrio da madeira $U_{eq}$
1	$U_{amb} \leq 65\%$	12%
2	$65\% < U_{amb} \leq 75\%$	15%
3	$75\% < U_{amb} \leq 85\%$	18%
4	$U_{amb} > 85\%$ durante longos períodos	$\geq 25\%$

Fonte: ABNT NBR 7190:97 – item 6.1.5

#### 3.2 CLASSE DE RESISTÊNCIA

As classes de resistência das madeiras têm por objetivo padronizar as suas características auxiliando na escolha do material para a elaboração do projeto estrutural de madeira.

A ABNT NBR 7190:97 apresenta as tabelas de classes de resistência considerando a umidade de equilíbrio da madeira  $U_{eq} = 12\%$ , que serão as classes utilizadas para os cálculos no presente trabalho. Onde a Tabela 4 representa as classes de resistência das coníferas e a Tabela 5 as classes de resistência das dicotiledôneas.

Tabela 4 – Classes de resistência das coníferas

Coníferas (Valores na condição padrão de referência $U = 12\%$ )					
Classes	$f_{c0k}$ MPa	$f_{vk}$ MPa	$E_{c0,m}$ MPa	$\rho_{bas,m}$ kg/m <sup>3</sup>	$\rho_{parente}$ kg/m <sup>3</sup>
C20	20	4	3500	400	500
C25	25	5	8500	450	550
C30	30	6	14500	500	600

Fonte: ABNT NBR 7190:97 – item 6.3.5

Tabela 5 – Classes de resistência das dicotiledôneas

Dicotiledôneas (Valores na condição padrão de referência $U = 12\%$ )					
Classes	$f_{c0k}$ MPa	$f_{vk}$ MPa	$E_{c0,m}$ MPa	$\rho_{bas,m}$ kg/m <sup>3</sup>	$\rho_{parente}$ kg/m <sup>3</sup>
C20	20	4	9500	500	650
C30	30	5	14500	650	800
C40	40	6	19500	750	950
C60	60	8	24500	800	1000

Fonte: ABNT NBR 7190:97 – item 6.3.5



### 3.3 COEFICIENTE DE MODIFICAÇÃO

A função do coeficiente de modificação, denominado como  $k_{mod}$ , é de ajustar os valores das propriedades da madeira em função dos coeficientes parciais dados pela classe de carregamento da estrutura, classe de umidade admitida e do eventual emprego de madeira de segunda categoria. O coeficiente de modificação é denominado pela ABNT NBR 7190:97 pelo produto dos coeficientes parciais:

$$k_{mod} = k_{mod,1} \times k_{mod,2} \times k_{mod,3} \quad (1)$$

#### 3.3.1 Classes de carregamento ( $k_{mod,1}$ )

O coeficiente  $k_{mod,1}$  leva em consideração a classe de carregamento e o tipo de material utilizado. Onde o carregamento é definido pelo conjunto de ações que tem probabilidade de ocorrer simultaneamente, com o intuito de determinar os efeitos mais desfavoráveis para a estrutura. A classe de carregamento de qualquer combinação de ações é definida pela duração acumulada para a ação variável tomada como ação principal nesta combinação que foi considerada com maiores prejuízos a estrutura. O coeficiente  $k_{mod,1}$  é definido pela Tabela 7, devendo ser escolhido com base nas informações da Tabela 6.

Tabela 6 – Classes de carregamentos

Classe de carregamento	Ação variável principal da combinação	
	Duração acumulada	Ordem de grandeza da duração acumulada da ação característica
Permanente	Permanente	Vida útil da construção
Longa duração	Longa duração	Mais de seis meses
Média duração	Média duração	Uma semana a seis meses
Curta duração	Curta duração	Menos de uma semana
Duração instantânea	Duração instantânea	Muito curta

Fonte: ABNT NBR 7190:97 – item 5.1.4

Tabela 7 – Valores de  $k_{mod,1}$ 

Classes de carregamento	Tipos de madeira	
	Madeira serrada Madeira laminada colada Madeira compensada	Madeira recomposta
Permanente	0,60	0,30
Longa duração	0,70	0,45
Média duração	0,80	0,65
Curta duração	0,90	0,90
Duração instantânea	1,10	1,10

Fonte: ABNT NBR 7190:97 – item 6.4.4

### 3.3.2 Classe de umidade ( $k_{mod,2}$ )

O coeficiente  $k_{mod,2}$  leva em consideração a classe de umidade descrita no item 3.1 deste trabalho e o material utilizado, conforme mostra a Tabela 8. No caso particular de madeira serrada submersa considera-se o valor de  $k_{mod,2} = 0,65$ .

Tabela 8 – Valores de  $k_{mod,2}$ 

Classes de umidade	Tipos de madeira	
	Madeira serrada Madeira laminada colada Madeira compensada	Madeira recomposta
(1) e (2)	1,0	1,0
(3) e (4)	0,8	0,9

Fonte: ABNT NBR 7190:97 – item 6.4.4

### 3.3.3 Classe de categoria ( $k_{mod,3}$ )

O coeficiente  $k_{mod,3}$  leva em consideração se a peça é de primeira ou segunda categoria. Para a condição de madeira de primeira categoria a peça não deve apresentar nenhum defeito, analisados por métodos visuais e mecânicos. Para as dicotiledôneas as peças de primeira categoria são consideradas com  $k_{mod,3} = 1,0$  e peças de segunda categoria  $k_{mod,3} = 0,8$ . Já para as coníferas o  $k_{mod,3} = 0,8$

considerando o risco de haver a presença de nós não identificados na análise de defeitos. Para madeiras laminadas e coladas o coeficiente altera dependendo da curvatura da peça, conforme mostra a Tabela 9.

Tabela 9 – Valores de  $k_{mod,3}$

Coníferas	Dicotiledôneas de 1ª categoria	Dicotiledôneas de 2ª categoria	Madeira laminada colada	
			Peças retas	Peças curvas
0,8	1	0,8	1	$1 - 2000 \left( \frac{t}{r} \right)$ (*)

(\*) t é a espessura das lâminas e r é o menor raio de curvatura das lâminas.

Fonte: Autor

### 3.4 COEFICIENTE DE MINORAÇÃO DA RESISTÊNCIA

Os cálculos a serem apresentados neste trabalho consideraram o Estado Limite Último (ELU), que segundo a ABNT NBR 7190:97, devem ser realizadas minorações de resistência utilizando os coeficientes de minoração decorrentes de tensões de compressão ( $\gamma_{wc} = 1,4$ ), decorrentes de tensões de tração ( $\gamma_{wt} = 1,8$ ) e decorrentes de tensões de cisalhamento ( $\gamma_{wc} = 1,8$ ).

### 3.5 RESISTÊNCIA DE CÁLCULO

Com os parâmetros de coeficiente de modificação (item 3.3), coeficiente de minoração da resistência (item 3.4) e a classe da madeira (item 3.2) determinados é possível calcular a resistência de cálculo, conforme fórmula da ABNT NBR 7190:97:

$$f_{wd} = k_{mod} \times \frac{f_{wk}}{\gamma_w} \quad (2)$$

Onde:

$f_{wd}$  = resistência de cálculo

$k_{mod}$  = coeficiente de modificação

$f_{wk}$  = resistência característica da madeira

$\gamma_w$  = coeficiente de minoração da resistência

### 3.6 DIMENSÕES MÍNIMAS

No presente trabalho, foram utilizados apenas peça simples, porém a norma também determina a seção mínima para peças de seções múltiplas. Ambas podem ser consideradas peças principais, ou seja, vigas ou barras longitudinais de treliça, ou podem ser consideradas secundárias. Segundo a ABNT NBR 7190:97, a seção e a espessura mínima permitida para todos os casos descritos acima estão descritos na Tabela 10.

Tabela 10 – Dimensões mínimas

Tipo de peça	Função estrutural da peça	Área mínima da seção (cm <sup>2</sup> )	Espessura mínima (cm)
Peça simples	Vigas e barras principais	50	5
	Peças secundárias	18	2,5
Peças de seções múltiplas	Peças principais	35	2,5
	Peças secundárias	18	1,8

Fonte: Autor

## 3.7 ESFORÇOS SOLICITANTES DE CÁLCULO – ESTADO LIMITE ÚLTIMO

### 3.7.1 Ações permanentes

Para as ações permanentes, todas as parcelas devem ser ponderadas pelo coeficiente  $\gamma_g$ , para que não ocorra majoração de certas partes e minoração de outras. As ações que são desfavoráveis a segurança recebem coeficientes de majoração, enquanto as ações favoráveis à segurança, recebem coeficientes de minoração, fazendo assim com que a estrutura se torne mais segura. As ações permanentes, segundo a ABNT NBR 7190:97, são divididas em três grupos, são eles: Ações permanentes de pequena variabilidade, ações permanentes de grande variabilidade e ações permanentes indiretos.

#### 3.7.1.1 Ações permanentes de pequena variabilidade

É considerado como de pequena variabilidade o peso próprio da estrutura, quando o peso da madeira classificada estruturalmente tenha peso específico com coeficiente de variação não maior que 10%. Para essa e outras ações permanentes de pequena variabilidade é necessário utilizar os coeficientes indicados pela Tabela 11.

Tabela 11 – Ações permanentes de pequena variabilidade

Combinações	Para efeitos	
	Desfavoráveis	Favoráveis
Normais	$\gamma_g = 1,3$	$\gamma_g = 1,0$
Especiais ou de construção	$\gamma_g = 1,2$	$\gamma_g = 1,0$
Excepcionais	$\gamma_g = 1,1$	$\gamma_g = 1,0$

Fonte: ABNT NBR 7190:97 – item 5.6.4

### 3.7.1.2 Ações permanentes de grande variabilidade

É considerado de grande variabilidade o peso próprio da estrutura, os elementos construtivos permanentes não estruturais e os equipamentos fixos, todos considerados globalmente, quando o peso próprio da estrutura não ultrapassa 75% de todas as ações permanentes. Para essas e outras ações permanentes de grande variabilidade são utilizados os coeficientes descritos na Tabela 12.

Tabela 12 – Ações permanentes de grande variabilidade

Combinações	Para efeitos	
	Desfavoráveis	Favoráveis
Normais	$\gamma_g = 1,4$	$\gamma_g = 0,9$
Especiais ou de construção	$\gamma_g = 1,3$	$\gamma_g = 0,9$
Excepcionais	$\gamma_g = 1,2$	$\gamma_g = 0,9$

Fonte: ABNT NBR 7190:97 – item 5.6.4

### 3.7.1.3 Ações permanentes indiretos

É considerado indireto os efeitos de recalque de apoio e retração dos materiais. Para esses e outras ações permanentes indiretas são considerados os coeficientes mostrados na Tabela 13.

Tabela 13 – Ações permanentes indiretos

Combinações	Para efeitos	
	Desfavoráveis	Favoráveis
Normais	$\gamma_\varepsilon = 1,2$	$\gamma_\varepsilon = 0$
Especiais ou de construção	$\gamma_\varepsilon = 1,2$	$\gamma_\varepsilon = 0$
Excepcionais	$\gamma_\varepsilon = 0$	$\gamma_\varepsilon = 0$

Fonte: ABNT NBR 7190:97 – item 5.6.4

### 3.7.2 Ações variáveis

As ações variáveis são consideradas de maneira diferente dos permanentes, quando são desfavoráveis à segurança da estrutura são considerados com o coeficiente de ponderação  $\gamma_Q$ , porém quando produzem efeitos favoráveis a segurança as parcelas não são consideradas nas combinações de ações.

Para as ações que possuem parcelas desfavoráveis e favoráveis, que não possam atuar fisicamente separadas, as mesmas devem ser consideradas como uma única ação.

Os coeficientes de ponderação das ações variáveis podem ser considerados segundo a Tabela 14.

Tabela 14 – Ações variáveis

Combinações	Ações variáveis em geral, incluídas as cargas acidentais de móveis	Efeitos da temperatura
Normais	$\gamma_Q = 1,4$	$\gamma_Q = 1,2$
Especiais ou de construção	$\gamma_Q = 1,2$	$\gamma_Q = 1,0$
Excepcionais	$\gamma_Q = 1,0$	$\gamma_Q = 0$

Fonte: ABNT NBR 7190:97 – item 5.6.5

### 3.7.3 Combinações de ações em estados limites últimos

Segundo a ABNT NBR 7190:97 existem três formas de realizar as combinações de ações em estado limite últimos. As combinações últimas especiais ou de construção, as combinações últimas excepcionais e a que está sendo utilizada neste trabalho que são as combinações últimas normais. Esta combinação pode ser determinada através da formula a seguir.

$$F_d = \sum_{i=1}^m \gamma_{g1} F_{Gi,k} + \gamma_Q \left[ F_{Q1,k} + \sum_{j=2}^n \Psi_{0j} F_{Qj,k} \right] \quad (3)$$

Onde:

$F_{Gi,k}$  = valor característico das ações permanentes

$F_{Q1,k}$  = valor característico da ação variável considerada como ação principal para a combinação em questão

$\Psi_{0j} F_{Qj,k}$  = valores reduzidos de combinação das demais ações variáveis

Os fatores de combinação e de utilização estão descritos na Tabela 15.

Tabela 15 – Fatores de combinação e de utilização

Ações em estruturas correntes	$\Psi_0$	$\Psi_1$	$\Psi_2$
Variações uniformes de temperatura em relação à média anual local	0,6	0,5	0,3
Pressão dinâmica do vento	0,5	0,2	0
Cargas acidentais dos edifícios	$\Psi_0$	$\Psi_1$	$\Psi_2$
Locais em que não há predominância de pesos de equipamentos fixos, nem de elevadas concentrações de pessoas	0,4	0,3	0,2
Locais em que há predominância de pesos de equipamentos fixos, ou de elevadas concentrações de pessoas	0,7	0,6	0,4
Bibliotecas, arquivos, oficinas e garagens	0,8	0,7	0,6
Cargas móveis e seus efeitos dinâmicos	$\Psi_0$	$\Psi_1$	$\Psi_2$
Pontes de pedestres	0,4	0,3	0,2
Pontes rodoviárias	0,6	0,4	0,2
Pontes ferroviárias (ferrovias não especializadas)	0,8	0,6	0,4

Fonte: ABNT NBR 7190:97 – item 5.4.6



### 3.8 ESFORÇO NORMAL DE TRAÇÃO RESISTENTE DE CÁLCULO

O esforço normal de tração resistente de cálculo deve ser comparado e verificado com o esforço normal de tração atuante de cálculo como mostra a fórmula a seguir:

$$N_d \leq N_{t,rd} \quad (4)$$

Onde:

$N_d$  = esforço normal de tração atuante de cálculo

$N_{t,rd}$  = esforço normal de tração resistente de cálculo

O esforço normal de tração resistente de cálculo depende da inclinação das fibras da madeira, sendo esse fator considerado na resistência característica da madeira. O esforço normal de tração resistente de cálculo é apresentado na expressão a seguir:

$$N_{t,rd} = k_{mod} \times \frac{f_{tak}}{\gamma_w} \times A \quad (5)$$

Onde:

$k_{mod}$  = coeficiente de modificação

$f_{tak}$  = resistência característica à tração

$\gamma_w$  = coeficiente de minoração da resistência

$A$  = área efetiva da seção transversal

### 3.8.1 Tensão característica à tração paralela as fibras

A tensão média à tração paralela as fibras é fornecida pela ABNT NBR 7190:97 em seu anexo E. Para fibras com inclinação de até 6° é permitido ignorar a influência da inclinação das tensões normais em relação as fibras da madeira, podendo utilizar a fórmula de resistência à tração no sentido paralelo às fibras:

$$f_{t0,d} = k_{mod} \times \frac{f_{t0,k}}{\gamma_w} \quad (6)$$

Onde:

$f_{t0,k}$  = Tensão média à tração paralela as fibras

### 3.8.2 Tensão característica à tração normal as fibras

A tensão média a tração normal as fibras assim como a paralela, é adquirida no anexo E da ABNT NBR 7190:97. Para as fibras com inclinação de 90° a fórmula para o esforço normal de tração resistente de cálculo para a utilização é a seguinte:

$$f_{t90,d} = k_{mod} \times \frac{f_{t90,k}}{\gamma_w} \quad (7)$$

Onde:

$f_{t90,k}$  = Tensão característica à tração normal as fibras

### 3.8.3 Tensão característica à tração inclinada em relação as fibras

Para inclinações maiores que 6° em relação as fibras da madeira é preciso considerar a redução de resistência, utilizando a fórmula empírica de Hankinson:

$$f_{t\alpha,k} = \frac{f_{t0,k} \times f_{t90,k}}{f_{t0,k} \times \sin^2(\alpha) + f_{t90,k} \times \cos^2(\alpha)} \quad (8)$$

Onde:

$f_{t\alpha,k}$  = tensão característica à tração inclinada em relação às fibras

$\alpha$  = ângulo de inclinação em relação as fibras

### 3.8.4 Índice de esbeltez

O cálculo de índice de esbeltez deve ser feito em ambas as direções da peça, tanto para a direção x, quanto para a direção y. Para elementos retangulares o índice de esbeltez é determinado pelas expressões apresentadas abaixo:

$$\lambda_x = \frac{L_0}{\sqrt{\frac{h^2}{12}}} \quad (9)$$

$$\lambda_y = \frac{L_0}{\sqrt{\frac{b^2}{12}}} \quad (10)$$

Onde:

$L_0$  = Comprimento da peça

$\lambda_{(y,x)}$  = índice de esbeltez

h = altura da peça

b = base da peça

Caso a verificação  $\lambda_{(x,y)} \leq 170$  não seja atendida, a peça deve ser redimensionada, aumentando a altura caso o  $\lambda_x$  não satisfaça a condição ou aumentando a base caso o  $\lambda_y$  não atenda a solicitação. Outra maneira ainda é diminuir o comprimento L do vão em questão.

### 3.9 ESFORÇO NORMAL DE COMPRESSÃO RESISTENTE DE CÁLCULO

O esforço normal de compressão resistente de cálculo deve ser comparado e verificado com o esforço normal de compressão atuante de cálculo, da mesma forma que o esforço normal de tração, como mostra a expressão a seguir:

$$N_d \leq N_{c,rd} \quad (11)$$

Onde:

$N_d$  = esforço normal de compressão atuante de cálculo

$N_{c,rd}$  = esforço normal de compressão resistente de cálculo

O esforço normal de compressão resistente de cálculo, assim como o de tração, depende da inclinação das fibras da madeira:

$$N_{c,rd} = k_{mod} \times \frac{f_{cak}}{\gamma_w} \times A \quad (12)$$

Onde:

$k_{mod}$  = coeficiente de modificação

$f_{cak}$  = resistência característica a compressão

$\gamma_w$  = coeficiente de minoração da resistência

#### 3.9.1 Tensão característica à compressão

A tensão característica a compressão pode ser considerada de três maneiras, assim como anteriormente:

- a) Paralela as fibras: onde o ângulo de inclinação das fibras não supera 6°, é considerado o valor tabelado pela norma:

$$f_{cak} = f_{c0,k} \quad (13)$$

- b) Normal as fibras: onde o ângulo de inclinação das fibras é de 90°, é considerado 25% da tensão paralela as fibras, considerando um coeficiente  $\alpha_n$  igual a 1 no quando a extensão da carga, medida na direção das fibras, for maior ou igual a 15cm. Quando a carga estiver com afastamento igual ou superior a 7,5cm da extremidade da peça, o coeficiente é determinado pela Tabela 16.

$$f_{c90,k} = 0,25 \times f_{c0,k} \times \alpha_n \quad (14)$$

Tabela 16 – Valores de  $\alpha_n$

Extensão da carga normal às fibras, medida paralelamente a estas cm	$\alpha_n$
1	2,00
2	1,70
3	1,55
4	1,40
5	1,30
7,5	1,15
10	1,10
15	1,00

Fonte: ABNT NBR 7190:97 – item 7.2.2

- c) Inclínada em relação as fibras: onde o ângulo de inclinação varia de 6° a 90°, é considerado a fórmula descrita no item 3.8.3.

### 3.9.2 Estabilidade

Para peças comprimidas deve ser levado em consideração sua esbeltez, pois é possível atingir seu estado limite por perda de estabilidade. Para este caso,

utilizamos o índice de esbeltez citado no item 3.8.4, porém este índice deve ser menor ou igual a 140, sendo este o limite.

### 3.9.2.1 Peças curtas

Para as peças curtas, definidas pelo índice de esbeltez  $\lambda$  menor que 40, que na situação de projeto são admitidas como solicitadas apenas à compressão simples, não é necessário considerar a redução de resistência devido ao processo de flambagem.

### 3.9.2.2 Peças medianamente esbeltas

Para as peças medianamente esbeltas, definidas pelo índice de esbeltez maior que 40 e menor ou igual a 80, a resistência é afetada pela ocorrência da flambagem, incluindo efeitos de imperfeições geométricas e da não linearidade do material. Com isso deve ser verificada a segurança em relação ao estado limite último de instabilidade pela teoria de validade comprovada experimentalmente.

Considera-se atendida a condição de segurança relativa ao estado limite último de instabilidade, se no ponto mais comprimido da seção transversal for respeitada a condição:

$$\frac{\sigma_{Nd}}{f_{cd}} + \frac{\sigma_{Md}}{f_{cd}} \leq 1 \quad (15)$$

Onde:

$\sigma_{Nd}$  = valor de cálculo da tensão de compressão devida à força normal de compressão

$\sigma_{Md}$  = valor de cálculo da tensão de compressão devida ao momento fletor:

$$\sigma_{Md} = \frac{M_d \times y}{I} \quad (16)$$

Onde:

$I$  = momento de inércia da seção transversal da peça relativo ao plano de flexão em que se está verificando a condição de segurança.

$M_d$  = Momento fletor de cálculo determinado a partir das expressões 18 e 19

$$M_d = N_d \times e_d \quad (17)$$

$$e_d = e_1 \times \left( \frac{F_E}{F_E - N_d} \right) \quad (18)$$

Onde:

$F_E$  = carga crítica

$$F_E = \frac{\pi^2 \times E_{co,ef} \times I}{L_0^2} \quad (19)$$

Onde:

$E_{co,ef}$  = Módulo de elasticidade à compressão paralelo às fibras efetivo

$$E_{co,ef} = k_{mod} \times E_{co,m} \quad (20)$$

Para a excentricidade deve ser considerado:

$$e_1 = e_i + e_a \quad (21)$$

Onde:

$e_i$  = excentricidade inicial

$e_a$  = excentricidade acidental mínima

Sendo:

$$e_i = \frac{M_{1d}}{N_d} \geq \frac{h}{30} \quad (22)$$

$$e_a = \frac{L_0}{300} \quad (23)$$

Onde:

$M_{1d}$  = momento de primeira ordem de cálculo

$h$  = altura na direção do momento

### 3.9.2.3 Peças esbeltas

Para as peças esbeltas, definidas pelo índice de esbeltez maior que 80 e menor ou igual a 140, não se permitindo valor maior que 140, a verificação pode ser feita pela expressão definida em 3.9.2.2, considerando o valor da excentricidade de cálculo conforme seguinte expressão:

$$e_d = e_{1,ef} \times \left( \frac{F_E}{F_E - N_d} \right) \quad (24)$$

Onde:

$e_{1,ef}$  = excentricidade efetiva de primeira ordem

$$e_{1,ef} = e_i + e_a + e_c \quad (25)$$

Onde:

$e_c$  = excentricidade suplementar de primeira ordem

$$e_c = (e_{ig} + e_a) \times (exp^k - 1) \quad (26)$$

$$e_{ig} = \frac{M_{1g,d}}{N_{1g,d}} \quad (27)$$

$$k = \frac{\phi \times k'}{F_E - k'} \quad (28)$$

$$k' = N_{gk} + (\psi_1 + \psi_2) \times N_{qk} \quad (29)$$

Onde:

$N_{gk}$  = força normal devido a carga permanente

$N_{qk}$  = força normal devido a carga variável



$\psi_1$  e  $\psi_2$  = fatores de combinação e de utilização, descrito na Tabela 15  
 $\phi$  = coeficiente de fluência, dado pela Tabela 17.

Tabela 17 – Coeficiente de fluência

Classes de carregamento	Classes de umidade	
	(1) e (2)	(3) e (4)
Permanente ou de longa duração	0,8	2,0
Média duração	0,3	1,0
Curta duração	0,1	0,5

Fonte: ABNT NBR 7190:97 – item 7.5.5

### 3.10 MOMENTO FLETOR RESISTENTE DE CÁLCULO

Para o presente trabalho será realizada somente a verificação para flexão simples reta, onde a segurança fica garantida quando for atendida as seguintes condições:

$$\sigma_{t2,d} \leq f_{td} \quad (30)$$

$$\sigma_{c1,d} \leq f_{cd} \quad (31)$$

Onde:

$f_{td}$  = resistência à tração definida em 3.8

$f_{cd}$  = resistência à compressão definida em 3.9

$\sigma_{t2,d}$  = Tensão atuante de cálculo nas bordas mais tracionadas

$\sigma_{c1,d}$  = Tensão atuante de cálculo nas bordas mais comprimidas

$$\sigma_{t2,d} = \frac{M_d}{W_t} \quad (32)$$

$$\sigma_{c1,d} = \frac{M_d}{W_c} \quad (33)$$

Onde:

$W_t$  = módulo de resistência à tração

$W_c$  = módulo de resistência à compressão

$$W_t = \frac{I}{y_{t2}} \quad (34)$$

$$W_c = \frac{I}{y_{c1}} \quad (35)$$

Onde:

$y_{t2}$  = distância mais afastada da linha neutra até a fibra mais tracionada

$y_{c1}$  = distância mais afastada da linha neutra até a fibra mais comprimida

Assim, o momento fletor resistente de cálculo é obtido pelo menor momento determinado segundo as expressões abaixo:

$$M_d = \frac{f_{t,d}}{y_{t2}} \times I \quad (36)$$

$$M_d = \frac{f_{c,d}}{y_{c1}} \times I \quad (37)$$

### 3.10.1 Instabilidade lateral

Para vigas fletidas, além das verificações citadas anteriormente, deve ser verificada a instabilidade lateral. Para isso as condições abaixo devem ser satisfeitas:

- a) os apoios de extremidade da viga devem impedir a rotação de suas seções extremas em torno do eixo longitudinal da peça;
- b) a peça deve conter um conjunto de elementos de travamento ao longo do comprimento L da viga, afastados entre si de uma distância não maior que  $L/1$ , que também impedem a rotação dessas seções transversais em torno do eixo longitudinal da peça;

$$\frac{L_1}{b} \leq \frac{E_{c0,ef}}{\beta_M \times f_{c0,d}} \quad (38)$$

Onde:

$$\beta_M = \frac{1}{0,26\pi} \times \frac{\beta_E}{\gamma_F} \times \frac{\left(\frac{h}{b}\right)^{3/2}}{\left(\frac{h}{b} - 0,63\right)^{1/2}} \quad (39)$$

$$\beta_E = 4 \quad (40)$$

$$\gamma_F = 1,4 \quad (41)$$

Onde:

$\beta_E$  = coeficiente de correção

$\gamma_F$  = coeficiente parcial de segurança correspondente à ponderação para os estados limites últimos

h = altura da viga

b = base da viga

Para peças que satisfaçam as condições a seguir é dispensável a verificação da segurança em relação ao estado limite último de estabilidade lateral:

$$\sigma_{c1,d} \leq f_{cd} \quad (42)$$

$$\sigma_{c1,d} = \frac{E_{c0,ef}}{\frac{L_1}{b} \times \beta_M} \quad (43)$$

### 3.11 ESFORÇO CISALHANTE RESISTENTE DE CÁLCULO

Como citado anteriormente, o presente trabalho se restringe a seções retangulares e com isso a verificação de segurança de cisalhamento longitudinal se dá quando a condição a seguir é satisfeita:

$$V_d \leq \frac{2}{3} \times f_{v0,d} \times A \quad (44)$$

Onde:

A: área da seção transversal

$f_{v0,d}$  = resistência ao cisalhamento na direção das fibras

## 4. O PROGRAMA COMPUTACIONAL DE DIMENSIONAMENTO DE ELEMENTOS ESTRUTURAIS DE MADEIRA

O IDE (ambiente de desenvolvimento integrado) utilizado para o desenvolvimento do programa computacional de dimensionamento de elementos estruturais de madeira será o NetBeans 8.2. A IDE é excelente para a criação de interfaces gráficas simples como a do presente trabalho pois possui ferramentas simples e eficientes para auxiliar na criação do programa computacional. Segundo o próprio site da IDE, o editor suporta diversas linguagens, como JavaScript, HTML5, PHP, C/C++ e, também, a linguagem Java, que será a utilizada no presente trabalho.

A linguagem Java foi desenvolvida para ser mais simples que as demais e mais parecida com o cotidiano. Com sua principal característica sendo a orientação ao objeto, a linguagem Java trabalha com o código escrito dentro de uma classe e tudo que está dentro dele é um objeto. As aplicações realizadas nesta linguagem podem ser executadas em qualquer plataforma que possua a Java Virtual Machine (JVM) instalada. Atualmente, os principais sistemas operacionais já possuem a JVM instalada, incluindo o sistema foco do trabalho que é o Windows.

Os cálculos realizados pelo programa computacional serão todos com base na ABNT NBR 7190:97, assim como todas as verificações que a norma exige para cada solicitação. O retorno do programa será as possíveis seções comerciais para a dada solicitação que foi descrita pelo usuário. Tornando assim o programa uma ferramenta útil não somente para fins acadêmicos, mas também para auxílio de profissionais da área de dimensionamento de elementos estruturais de madeira.

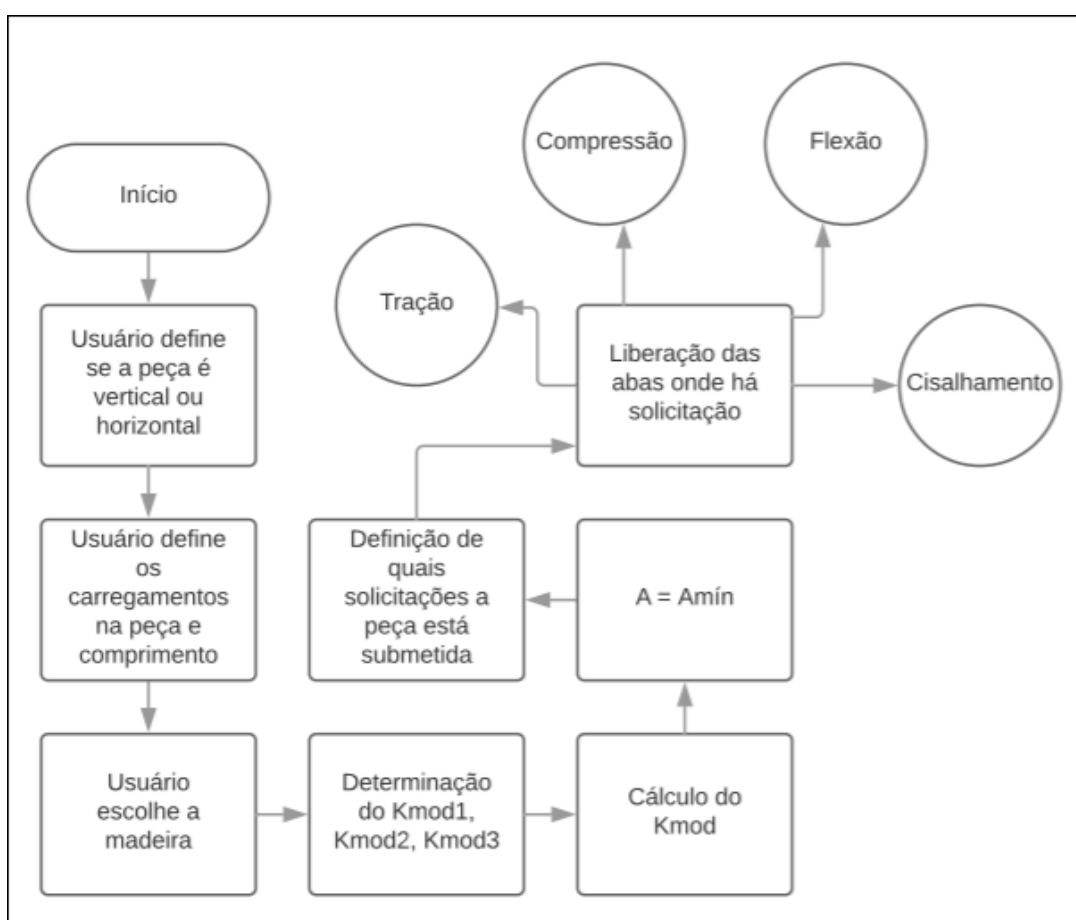
### 4.1 FLUXOGRAMAS

Para o início do desenvolvimento do programa foram realizados alguns fluxogramas auxiliares, com a utilização da ferramenta LucidCharts, disponível no site: *lucidchart.com*. Estes fluxogramas são fundamentais para guiar o programador no desenvolvimento do código.

### 4.1.1 Fluxograma geral

O fluxograma geral mostra o que será realizado inicialmente no programa computacional. A partir da definição de parâmetros de projeto pelo usuário, o programa faz uma primeira verificação adotando uma área mínima, conforme a ABNT NBR 7190:97. A partir disso, o programa retorna quais as solicitações são exercidas na peça, passando para os fluxogramas posteriores de cada solicitação.

Figura 19 – Fluxograma geral

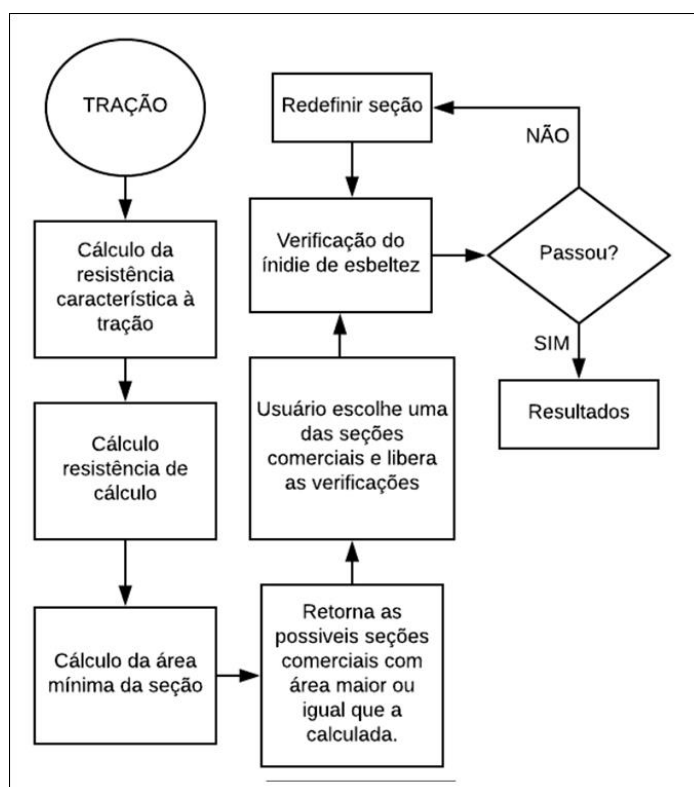


Fonte: Autor.

### 4.1.2 Fluxograma de tração

Para o dimensionamento à tração da peça será realizada as verificações de segurança, posteriormente, calculando a área mínima da seção para a peça a ser dimensionada, sendo assim realizado a verificação da área mínima, onde, caso a área calculada seja inferior a  $50\text{cm}^2$ , a mesma é atribuída como área mínima, conforme indicado na norma ABNT NBR 7190:97. Com a área determinada, o programa retorna as seções comerciais com área maior ou igual a dimensionada e, com isso, libera a verificação do índice de esbeltez, onde em caso de a verificação não passar o usuário deve escolher outra seção da lista para refazer a verificação.

Figura 20 – Fluxograma de tração

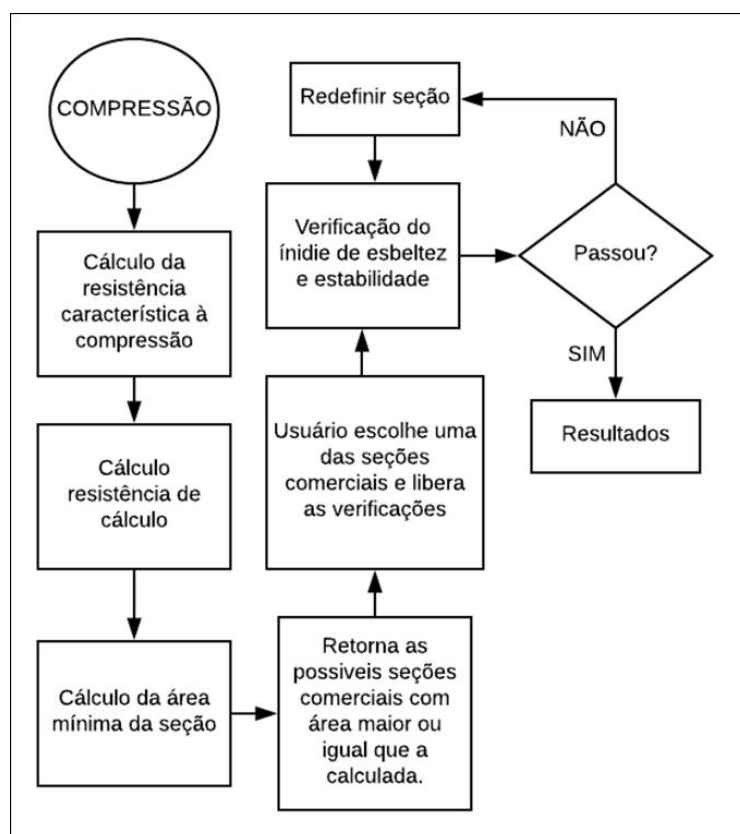


Fonte: Autor.

### 4.1.3 Fluxograma de compressão

Para as barras requeridas à compressão o programa faz um caminho semelhante ao de tração, porém com as formulações diferentes e com a verificação de instabilidade adicionada.

Figura 21 – Fluxograma de compressão



Fonte: Autor.

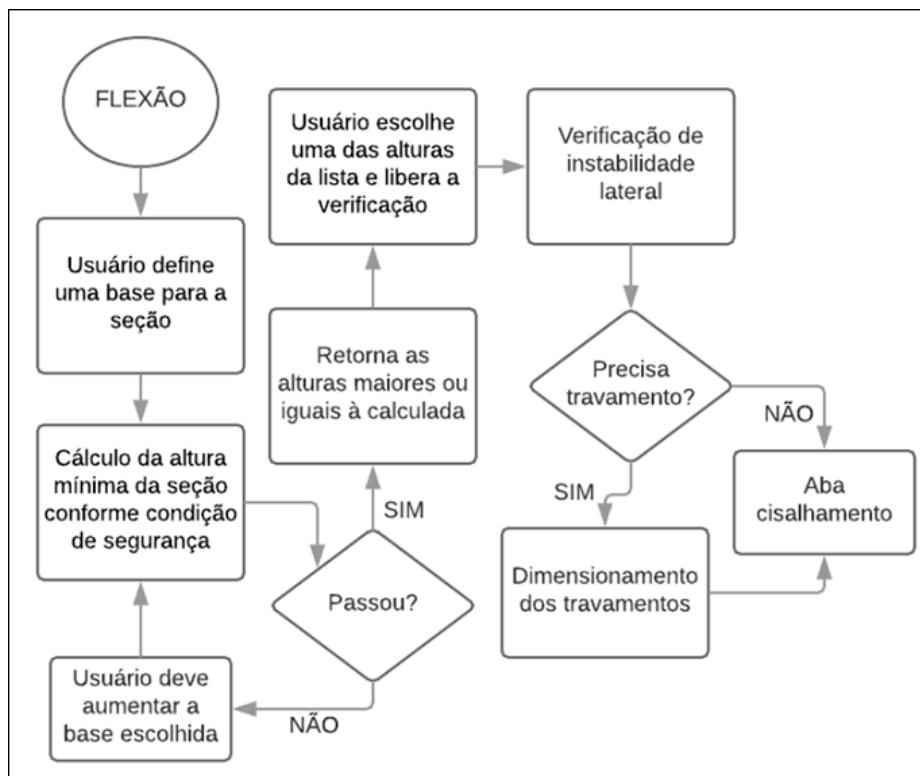
### 4.1.4 Fluxograma de flexão

Para as peças solicitadas a flexão, primeiramente, o usuário deve informar uma base para a seção, assim é liberado o cálculo da altura mínima, conforme condição de segurança, caso o programa retornar uma altura acima das comerciais cadastradas no mesmo, retorna que o usuário deve aumentar a base e refazer o cálculo. Caso contrário, o programa retorna a lista de alturas igual ou maior que a calculada. Desta forma, o usuário escolhe uma altura e libera a verificação de instabilidade lateral, onde o programa retorna se precisa ou não de travamentos ao



longo da viga. Caso precise de travamentos, o programa calcula quantos travamentos são necessários e o espaçamento entre eles, caso contrário, libera a aba de cisalhamento.

Figura 22 – Fluxograma de flexão

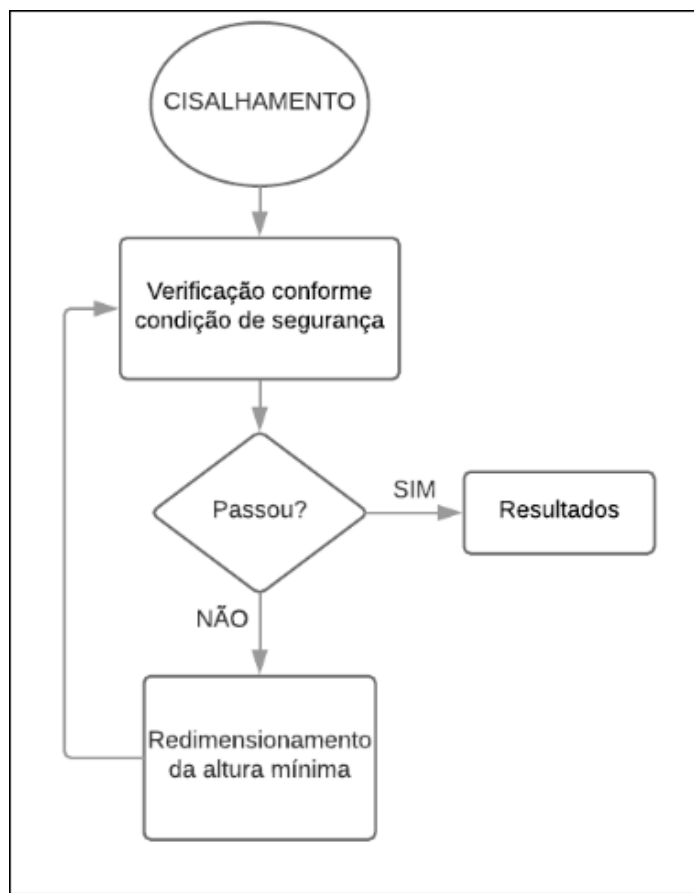


Fonte: Autor.

#### 4.1.5 Fluxograma de cisalhamento

Para o cisalhamento é realizada a verificação conforme condição de segurança, usando os dados calculados na aba de flexão. Caso a verificação não seja atendida, o programa faz o redimensionamento da altura mínima da seção.

Figura 23 – Fluxograma de cisalhamento

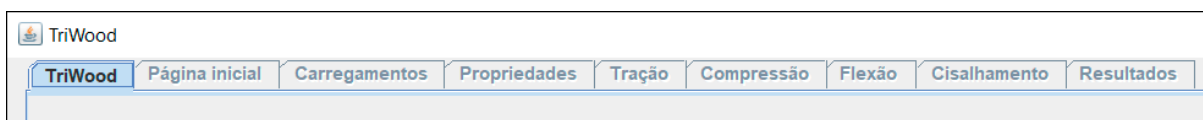


Fonte: Autor.

## 4.2 INTERFACE

A interface do programa computacional foi desenvolvida em um JFrame onde o mesmo foi dividido em nove abas, como mostra a Figura 24, cada uma delas com uma função específica para o dimensionamento da peça desejada. Para facilitar o uso do programa, as abas vão sendo liberadas conforme as informações vão sendo adicionadas. Caso o usuário deseje alterar qualquer informação colocada no programa basta voltar para a aba desejada e refazer o processo de dimensionamento. O detalhamento de cada aba será determinado a seguir.

Figura 24 - Interface

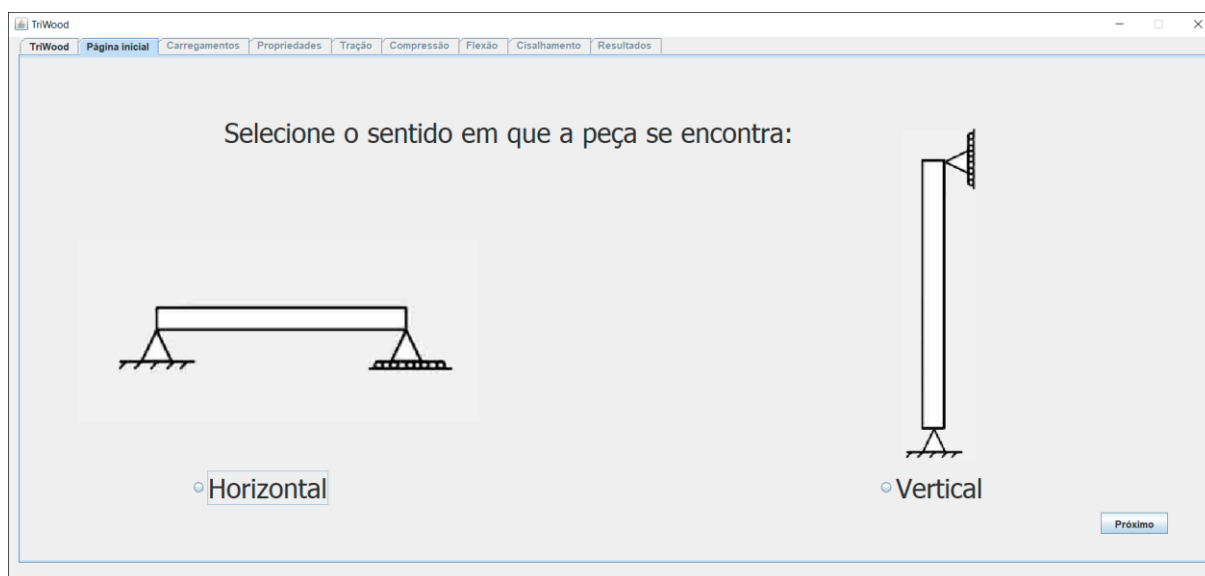


Fonte: Autor.

### 4.3 ABA PÁGINA INICIAL

Nesta aba, o usuário define qual o sentido em que a peça a ser dimensionada se encontra, podendo ser no sentido vertical ou horizontal, como mostra a Figura 25.

Figura 25 – Página inicial



Fonte: Autor.

### 4.4 ABA CARREGAMENTOS

Esta aba é dividida em duas possibilidades, que dependem do sentido escolhido pelo usuário na aba anterior. Cada uma delas possuem atributos diferentes como mostrado nos itens a seguir.

#### 4.4.1 Carregamento horizontal

Caso o usuário escolha pelo sentido horizontal, será realizado o dimensionamento à flexão e posteriormente ao cisalhamento. Esta aba está dividida em quatro parcelas. Primeiramente, o usuário deve informar o comprimento da barra, como mostra a Figura 26, depois o carregamento permanente e sua variabilidade, conforme Figura 27 e, por fim, o tipo carregamento variável e sua intensidade mostrado na Figura 28. Desta forma é habilitado o botão para calcular as combinações das ações, onde as duas combinações de momento solicitante de cálculo e o cortante

solicitante de cálculo são apresentadas para o usuário como mostra a Figura 29. Enfim, o usuário pode avançar para a próxima aba.

Figura 26 – Comprimento da peça horizontal

Determine o comprimento da peça:

Comprimento (L) =  cm

Fonte: Autor.

Figura 27 – Carga permanente horizontal

Determine a carga permanente:

Grande variabilidade

Pequena variabilidade

Carga permanente (g) =  kN/m

Fonte: Autor.

Figura 28 – Carga variável horizontal

Determine a carga variável:

Selecione o tipo de carga:  ▼

Carga variável (q) =  kN/m

Fonte: Autor.

Figura 29 – Combinações horizontal

**Combinações:**

Msd1 =  kN.cm

Msd2 =  kN.cm

Vsd1 =  kN

Vsd2 =  kN

Fonte: Autor.

#### 4.4.2 Carregamento vertical

Caso o usuário escolha pelo sentido vertical, será realizado o dimensionamento à tração caso as combinações calculadas forem positivas ou à compressão se os valores forem negativos. Assim, primeiramente, o usuário deve informar o comprimento da barra como já mostrado anteriormente na Figura 26, a seguir o carregamento permanente e sua variabilidade como mostrado na Figura 30 e, por fim, o tipo de carregamento variável e sua intensidade como na Figura 31. Desta forma é habilitado o botão para calcular as combinações das ações, onde as duas combinações de carregamento solicitante de cálculo são apresentadas para o usuário como pode ser visualizado na Figura 32. Podendo assim, avançar para a próxima aba.

Figura 30 – Carga permanente vertical

**Determine a carga permanente:**

Grande variabilidade  Pequena variabilidade

Carga permanente (Ngk) =  kN \*Para carregamentos de compressão usar o sinal negativo.

Fonte: Autor.

Figura 31 – Carga variável vertical

Determine a carga variável:

Selecione o tipo de carga:  ▼

Carga variável (N<sub>qk</sub>) =  kN \*Para carregamentos de compressão usar o sinal negativo.

Fonte: Autor.

Figura 32 – Combinações vertical

Combinações:

N<sub>sd1</sub> =  kN

N<sub>sd2</sub> =  kN

Fonte: Autor.

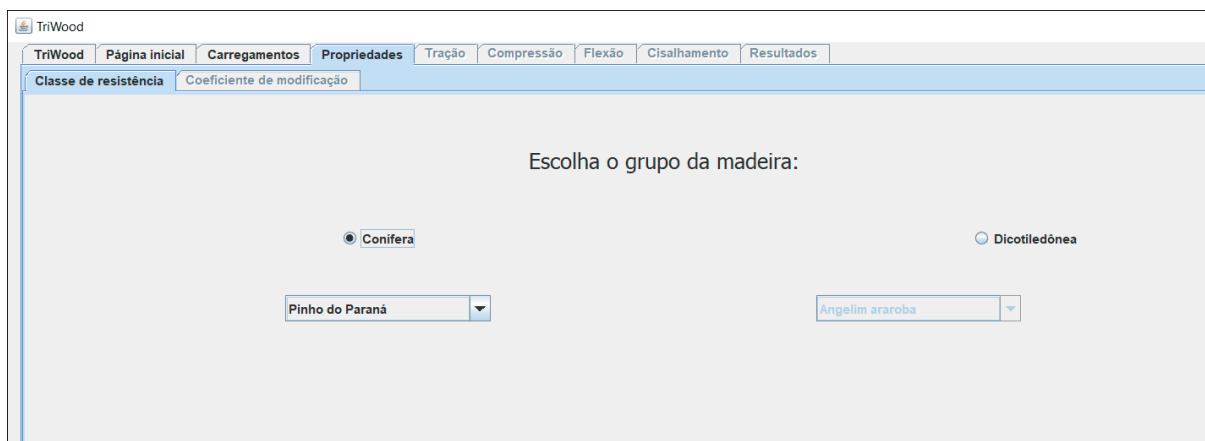
## 4.5 ABA PROPRIEDADES

Nesta etapa é determinada as propriedades da madeira que será utilizada no dimensionamento posterior, dividido em duas abas.

### 4.5.1 Classe de resistência

Nesta aba, o usuário define qual grupo o material se encaixa e posteriormente qual a madeira que deseja utilizar no dimensionamento. Assim o programa computacional registra as informações equivalentes ao anexo E da ABNT NBR 7190:97 da madeira escolhida para uso no dimensionamento.

Figura 33 – Classe de resistência

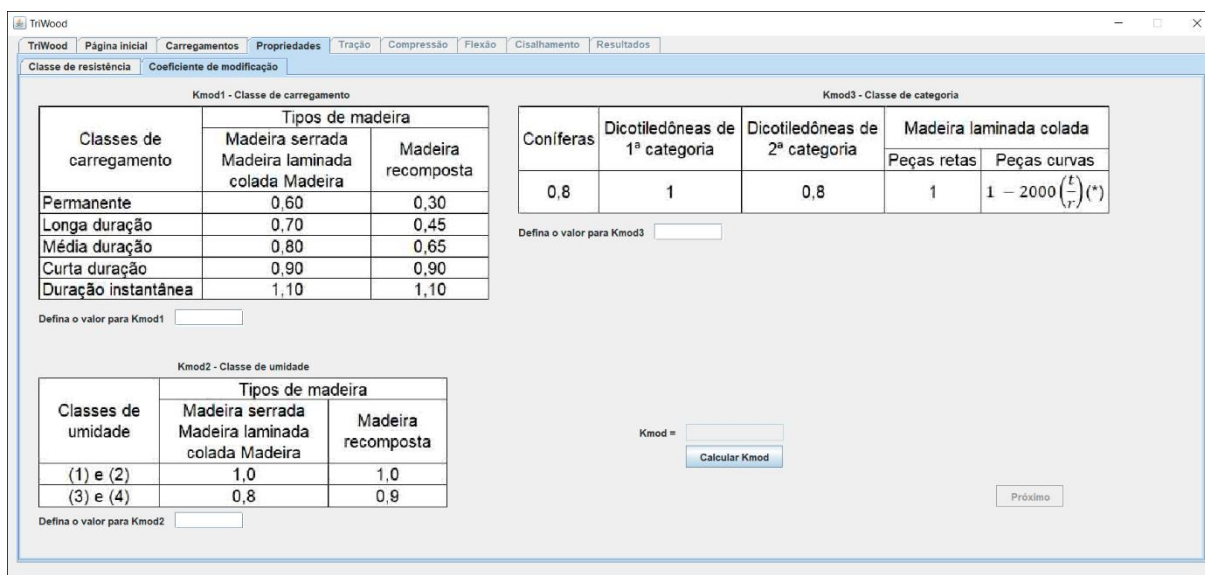


Fonte: Autor.

#### 4.5.2 Coeficiente de modificação da resistência

O coeficiente de modificação é dividido em três etapas, onde o usuário define através das tabelas, o valor para o  $k_{mod,1}$ , que é dado pela classe de carregamento e o tipo de material empregado, o  $k_{mod,2}$ , pela classe de umidade e o  $k_{mod,3}$ , pela classe de categoria. Desta forma, o programa utiliza a fórmula citada no item 3.3 deste trabalho, para o cálculo do coeficiente de modificação.

Figura 34 – Coeficiente de modificação



Fonte: Autor.

## 4.6 ABA TRAÇÃO

Caso o usuário tenha escolhido o sentido da peça sendo vertical e as combinações tenham resultados com sinal positivo, esta aba será liberada para ser realizado o dimensionamento à tração e a verificação do índice de esbeltez da peça. A aba é dividida em 3 partes, entrada de dados, dimensionamento e verificação.

O quadro de dados é composto pelo esforço normal de tração atuante de cálculo ( $N_d$ ), que é definido pelo maior valor calculado nas combinações citadas no item 4.4.2 deste trabalho, pelo coeficiente de modificação ( $k_{mod}$ ), calculado conforme item 4.5.2, e o coeficiente de minoração, conforme item 3.4 deste trabalho.

Figura 35 – Dados tração

<b>Dados:</b>		
<b>Nd =</b>	<input type="text"/>	<b>kN</b>
<b>Kmod =</b>	<input type="text"/>	
<b>Ywt =</b>	<input type="text" value="1,8"/>	

Fonte: Autor.

O dimensionamento inicia com o usuário determinando o ângulo de inclinação das fibras em relação a solicitação imposta. Podendo escolher por paralela ( $0^\circ$  a  $6^\circ$ ) calculando assim a resistência à tração na direção das fibras, conforme item 3.8.1 deste trabalho, ou normal ( $90^\circ$ ) calculando então a resistência à tração perpendicular (normal) às fibras, conforme item 3.8.2 deste trabalho, ou outro ângulo a sua escolha calculando então a resistência à tração inclinada em relação às fibras, conforme mostra o item 3.8.3 deste trabalho.

Após o cálculo da resistência à tração é liberado o botão “Dimensionar” no programa, onde o mesmo utiliza a fórmula descrita no item 3.8 deste trabalho para encontrar a área mínima para a situação até então especificada.



Figura 36 – Dimensionamento à tração

Selecione o ângulo de inclinação da tensão característica em relação as fibras:

Paralela (0° a 6°)     
 Normal (90°)     
 Outro =  °

$f_{t0,d} =$    $\text{kN/cm}^2$      
 $f_{t90,d} =$    $\text{kN/cm}^2$      
 $f_{ta,d} =$    $\text{kN/cm}^2$

Área mínima da seção:   $\text{cm}^2$

Fonte: Autor.

Com a área mínima calculada, o programa utiliza a base de dados de 64 diferentes seções comerciais para alimentar a lista de seções com área maior ou igual à calculada anteriormente. Assim, o usuário pode escolher qual seção deseja utilizar e, posteriormente, realizar a verificação do índice de esbeltez.

A verificação do índice de esbeltez é realizada conforme o item 3.8.4 deste trabalho. Este índice deve ser menor ou igual a 170. Caso a verificação proceda, o programa retorna “OK!” para o usuário e libera o botão “resultados”, caso contrário, retorna “Redefinir seção” e o usuário deve escolher outra seção para realizar o teste.

Figura 37 – Verificação tração

**Verificação**

Selecione a opção de seção que deseja utilizar:

Esbeltez

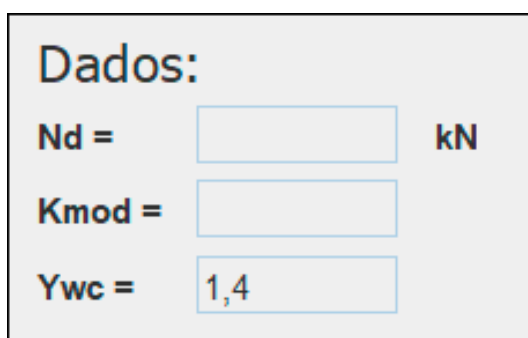
Fonte: Autor.

## 4.7 ABA COMPRESSÃO

Caso o usuário tenha escolhido o sentido da peça sendo vertical e as combinações resultarem em números negativos, esta aba será liberada para ser realizado o dimensionamento à compressão, a verificação do índice de esbeltez e estabilidade da peça. Como a aba de tração esta também é dividida nas mesmas 3 partes, entrada de dados, dimensionamento e verificação.

O quadro de dados é composto pelo esforço normal de compressão atuante de cálculo ( $N_d$ ), que é definido pelo maior valor em módulo calculado nas combinações citadas no item 4.4.2 deste trabalho, pelo coeficiente de modificação ( $k_{mod}$ ) calculado conforme item 4.5.2 e o coeficiente de minoração, conforme item 3.4 deste trabalho.

Figura 38 – Dados compressão



Dados:

$N_d =$   kN

$K_{mod} =$

$Y_{wc} =$

Fonte: Autor.

O dimensionamento inicia com o usuário determinando o ângulo de inclinação das fibras em relação a solicitação imposta. Podendo escolher por paralela ( $0^\circ$  a  $6^\circ$ ) calculando assim a resistência à compressão na direção das fibras, normal ( $90^\circ$ ) calculando então a resistência à compressão perpendicular (normal) às fibras ou outro ângulo a sua escolha calculando então a resistência à tração inclinada em relação às fibras. Onde para ângulos acima de  $6^\circ$  é necessário informar a extensão da carga, conforme item 3.9.1 deste trabalho.

Após o cálculo da resistência à compressão é liberado o botão “Dimensionar” no programa, onde o mesmo utiliza a fórmula descrita no item 3.9 deste trabalho para encontrar a área mínima para a situação até então especificada.

Figura 39 – Dimensionamento à compressão

Selecione o ângulo de inclinação da tensão característica em relação as fibras:

Paralela (0° a 6°)
  Normal (90°)
  Outro =  °

Extensão da carga

Extensão da carga

$f_{c0,d} =$    $\text{kN/cm}^2$ 
 $f_{c90,d} =$    $\text{kN/cm}^2$ 
 $f_{c\alpha,d} =$    $\text{kN/cm}^2$

Dimensionar

Área mínima da seção:   $\text{cm}^2$

Fonte: Autor.

Com a área mínima calculada, o programa utiliza a base de dados de 64 diferentes seções comerciais para alimentar a lista de seções com área maior ou igual à calculada anteriormente. Onde, com isso o usuário pode escolher qual seção deseja utilizar e posteriormente realizar as verificações do índice de esbeltez e estabilidade.

A verificação do índice de esbeltez é realizada conforme o item 3.8.4 deste trabalho e a verificação da estabilidade é feita como citado no item 3.9.2 deste trabalho. Para ambas verificações, caso a mesma proceda, o programa retorna “OK!” para o usuário e libera o botão “resultados”, caso contrário, retorna “Redefinir seção” e o usuário deve escolher outra seção para realizar o teste.

Figura 40 – Verificações compressão

Verificação

Selecione a opção de seção que deseja utilizar:

Esbeltez

Estabilidade

Verificar

Fonte: Autor.

## 4.8 ABA FLEXÃO

Para o caso de o usuário ter escolhido o sentido da peça sendo horizontal, o programa libera esta aba inicialmente para ser realizado o dimensionamento a flexão, a verificação de instabilidade lateral e se preciso o dimensionamento dos travamentos necessários ao longo da peça. A aba é dividida em quatro etapas, sendo ela a parte

de dados já obtidos, a de dimensionamento da altura mínima, a de verificação da instabilidade lateral e por fim a de dimensionamento dos travamentos.

Os dados que são apresentados para o usuário são compostos pelo momento fletor resistente de cálculo ( $M_d$ ), onde é apresentado o maior valor dentre as combinações calculadas anteriormente, seguindo o item 4.4.1 deste trabalho, o coeficiente de modificação ( $k_{mod}$ ) já calculado anteriormente, como citado no item 4.5.2, a resistência à tração na direção das fibras calculada conforme item 3.8.1 e a resistência à compressão na direção das fibras calculada de acordo com o item 3.9.1 do presente trabalho.

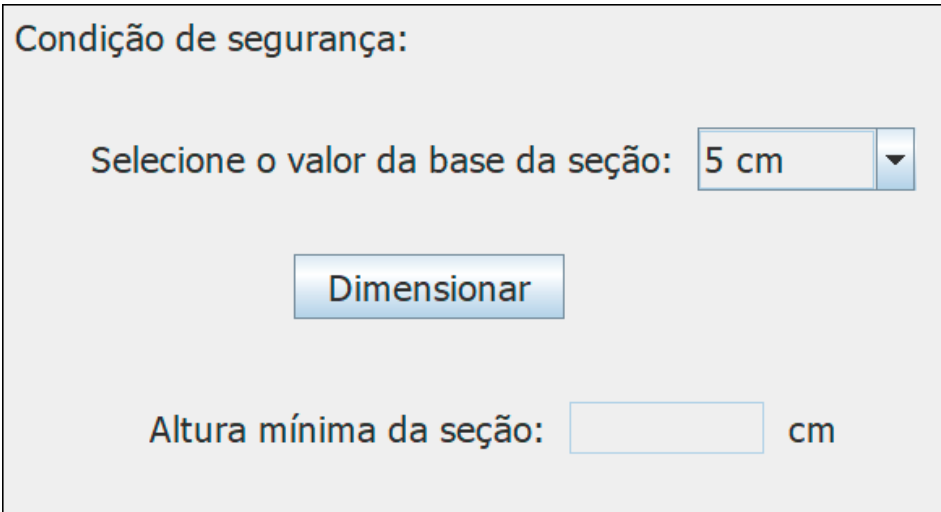
Figura 41 – Dados flexão

<b>Dados:</b>		
<b><math>M_d =</math></b>	<input type="text"/>	<b>kN.cm</b>
<b><math>K_{mod} =</math></b>	<input type="text"/>	
<b><math>f_{t,d} =</math></b>	<input type="text"/>	<b>kN/cm<sup>2</sup></b>
<b><math>f_{c,d} =</math></b>	<input type="text"/>	<b>kN/cm<sup>2</sup></b>

Fonte: Autor.

Para o dimensionamento o usuário deve escolher a base que deseja utilizar onde o programa dá as opções de 5 cm até 30 cm com diferença de 2,5 cm entre elas. Com a base selecionada é possível fazer o dimensionamento da altura mínima para a seção conforme o item 3.10 do presente trabalho. O usuário pode modificar a base escolhida e redimensionar a seção a qualquer momento.

Figura 42 – Dimensionamento à flexão



Condição de segurança:

Selecione o valor da base da seção: 5 cm

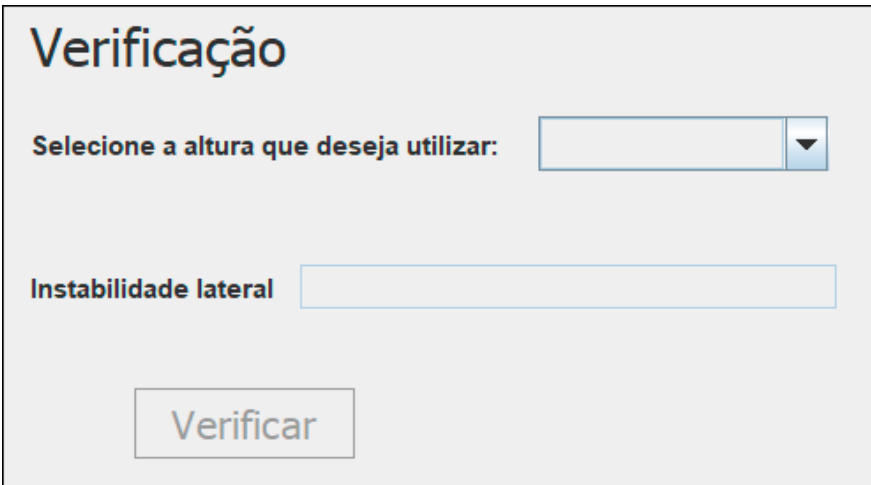
Dimensionar

Altura mínima da seção:  cm

Fonte: Autor.

Com a altura mínima calculada pelo programa a parte de verificação é liberada e, então, o usuário pode escolher entre as alturas maiores ou iguais a mínima que o programa disponibiliza em uma lista para o usuário, com esta altura escolhida é possível realizar a verificação da instabilidade lateral conforme item 3.10.1 deste trabalho, onde a mesma retorna para o usuário se será ou não necessário o uso de travamentos ao longo da peça. Caso seja necessários travamentos na peça, é liberado a parte do dimensionamento dos mesmos, caso contrário, é liberado diretamente a aba de cisalhamento.

Figura 43 – Verificação flexão



Verificação

Selecione a altura que deseja utilizar:

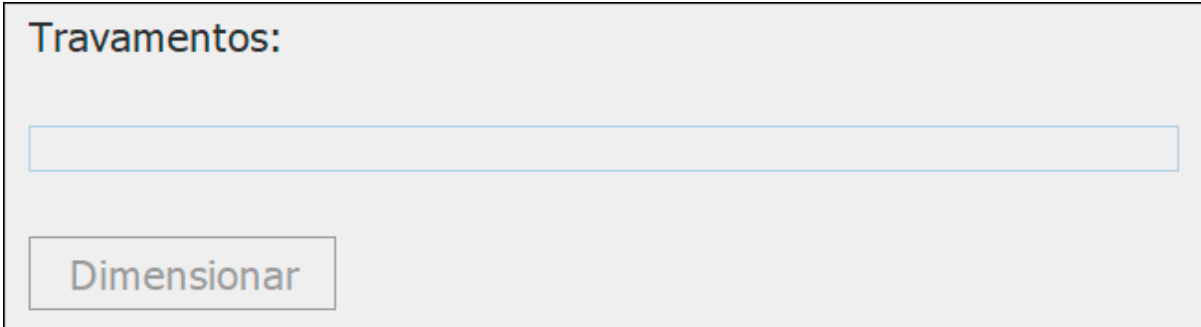
Instabilidade lateral

Verificar

Fonte: Autor.

A etapa do dimensionamento dos travamentos é liberada caso a verificação de instabilidade lateral retornar que é precisa o uso de travamentos na peça. Desta forma o programa calcula quantos travamentos são necessários ao longo da peça e o espaçamento entre eles. Após dimensionado os travamentos é liberado a aba de cisalhamento.

Figura 44 – Dimensionamento de travamentos



Travamentos:

Fonte: Autor.

#### 4.9 ABA CISALHAMENTO

Para a aba de cisalhamento foi realizado um processo diferente aos anteriores, onde primeiramente o programa faz uma verificação da condição de segurança levando em consideração a seção dimensionada na aba de flexão e caso a verificação falhe o programa faz um redimensionamento da altura da peça. A aba é dividida em três partes, os dados, a verificação da condição de segurança e o redimensionamento.

Os dados apresentados nesta aba são compostos pelo esforço cisalhante resistente de cálculo ( $V_d$ ), onde é apresentado o maior valor dentre as combinações calculadas anteriormente seguindo o item 4.4.1 deste trabalho, a base e a altura escolhidas anteriormente como mostrado no item 4.8 do presente trabalho, a resistência ao cisalhamento na direção das fibras da madeira selecionada no item 4.5.1 citado acima e o coeficiente de modificação ( $k_{mod}$ ) já calculado anteriormente como citado no item 4.5.2.

Figura 45 – Dados cisalhamento

**Dados:**

Vd =  kN

base =  cm

Altura =  cm

fv0 =  kN/cm<sup>2</sup>

Kmod =

Fonte: Autor.

A verificação de segurança é realizada conforme o item 3.11 deste trabalho, utilizando os dados obtidos através do dimensionamento e verificação de flexão, assim o programa retorna, em caso de obedecida a verificação, que a está com a "Verificação OK!" e libera a aba de resultados, caso contrário retorna "Falha na verificação. Redimensionar seção" e libera o botão de redimensionamento ao cisalhamento.

Figura 46 – Verificação cisalhamento

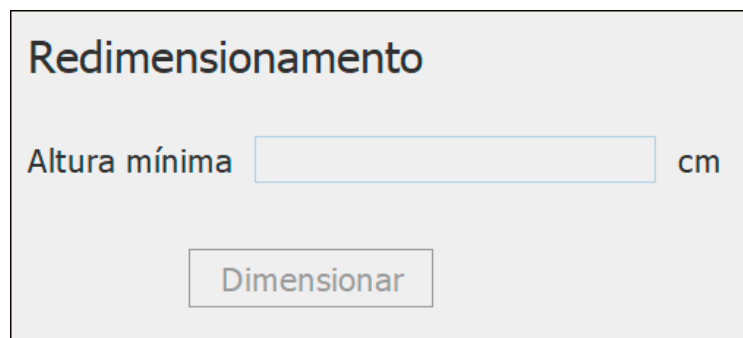
**Verificação**

Condição de segurança:

Fonte: Autor.

Caso a verificação falhe, a parte de redimensionamento é liberada e o programa recalcula a altura mínima que deve ser utilizada pela formulação citada no item 3.11 e desta forma o usuário pode alterar a altura selecionada para refazer a verificação.

Figura 47 – Dimensionamento ao cisalhamento



Redimensionamento

Altura mínima  cm

Dimensionar

Fonte: Autor.

#### 4.10 ABA RESULTADOS

Na aba dos resultados se concentra todas as informações que foram processadas pelo programa. As propriedades físicas e mecânicas da peça são impressas, os carregamentos e combinações realizados são evidenciados novamente, os dados do dimensionamento realizado são mostrados e as verificações realizadas são expostas, tudo isso de forma mais detalhada que no decorrer do programa, para que o usuário possa acompanhar a realização dos cálculos que foram feitos. A Figura 49 mostra um exemplo dos resultados.



Figura 48 – Resultados

**Relatório final de cálculos**

---- Propriedades da peça ----

Material escolhido: Pinho do Paraná  
fc0 = 40,9 MPa  
ft0 = 93,1 MPa  
ft90 = 1,6 MPa  
fv = 8,8 MPa  
Ec0 = 15225 MPa

Kmod = 0.8 \* 0.8 \* 0.8  
Kmod = 0,512

---- Carregamentos e combinações ----

Carregamentos:  
gk = 1.0 kN/m (carga permanente)  
Mg = 128,801 kN.cm

Vg = 1,605 kN  
qk = 1.0 kN/m (carga variável em função da temperatura)  
Mq = 128,801 kN/m  
Vq = 1,605 kN/m

Combinações:  
Msd = (yg \* Mg) + (yq \* Mq)  
Msd1 = (0.9 \* 128,801) + (0.0 \* 128,801)  
Msd1 = 115,92kN.cm  
Msd2 = (1.4 \* 128,801) + (1.4 \* 128,801)  
Msd1 = 360,64kN.cm

Vsd = (va \* Va) + (va \* Va)

Fonte: Autor.

## 5 CONSIDERAÇÕES FINAIS

O programa computacional foi finalizado com sucesso e agora inicia o processo para disponibilização do mesmo para o público, para que todos estudantes e engenheiros possam desfrutar do programa desenvolvido neste trabalho.

A quantidade de trabalhos acadêmicos desenvolvidos na área de dimensionamento de elementos estruturais em madeira ainda é bem baixa, por isso o presente trabalho deve acrescentar um bom estudo para os interessados na área.

Como o desenvolvimento de forma geral foi seguindo a ABNT NBR 7190:97, o programa pode ser aproveitado tanto para o uso didático no meio acadêmico, quanto no auxílio do dimensionamento de elementos estruturais em madeira de forma profissional.

Todos os objetivos previstos inicialmente foram alcançados e o programa abrange tanto a parte de dimensionamento quanto a de verificação, para que o usuário que está utilizando o programa possa ter um poder de escolha maior, foi desenvolvido o dimensionamento no primeiro momento e, a partir disso disponibilizado para que o usuário possa escolher uma seção de sua preferência, igual ou maior à dimensionada, para dar continuidade às verificações.

Esta é a versão inicial do *TriWood* e por isso podem haver alguns problemas que não foram identificados na fase de testes, desta forma com o passar do tempo serão feitas atualizações com o intuito de melhorar o programa, não somente nas funções atuais, mas também podendo ser adicionadas novas ferramentas.

## REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023: Informação e documentação - Referências - Elaboração**. Rio de Janeiro. 2002

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 7190: Projetos de estruturas de madeira**. Rio de Janeiro. 1997

CALIL, Carlito Junior; LAHR, Francisco Antonio Rocco; DIAS, Antonio Alves. **Dimensionamento de elementos estruturais de madeira**. 1. ed., 1ª reimpressão. Barueri, SP: Manole, 2003. 152p.

CORREIA, Emanuel André Soares. **Análise e dimensionamento de estruturas de madeira**. 2009. 143f. Dissertação (Mestre em Engenharia Civil) - Faculdade de Engenharia da Universidade do Porto, Portugal, 2009.

MARTINS, Tomás Francisco Ribeiro Mendes. **Dimensionamento de Estruturas em Madeira: Coberturas e Pavimentos**. 2010. 151f. Dissertação (Mestre em Engenharia Civil) – Instituto Superior Técnico da Universidade Técnica de Lisboa, Lisboa, 2010.

MORESHI, João Carlos. **Propriedades da madeira**. Curitiba, 2012.

PFEIL, Walter; PFEIL, Michèle. **Estruturas de madeira: Dimensionamento segundo a NBR 7190/97 e critérios das Normas Norte-americanas NDS e Européia EUROCODE 5**. 6. ed. Rio de Janeiro: Geethik, 2003.

PINHEIRO, Vítor Hugo Martins. **Projeto em estruturas de madeira**. 2012. 142f. Dissertação (Mestre em Engenharia Civil) - Faculdade de Engenharia da Universidade do Porto, Portugal, 2012.

RUSSO, Ricardo. **Construções em madeira e mudanças climáticas**. Disponível em: <<http://madeiraeconstrucao.com.br/construcoes-em-madeira-e-mudancas-climaticas-2/>>. Acesso em: 09 maio 2016.

RUSSO, Ricardo. **Uso da madeira passa pela informação**. Disponível em: <<http://madeiraeconstrucao.com.br/uso-da-madeira-passa-pela-informacao/>>. Acesso em: 09 maio 2016.

SILVA, Juliano Lima da. **Desenvolvimento de software em java para cálculo de estruturas de madeira**. 2014. 96f. Trabalho de conclusão de curso (Graduação em Engenharia Civil) - Faculdade de Engenharia e Arquitetura da Universidade de Passo Fundo, Passo Fundo, 2014.

ZERBINI, Fabíola. **Cenário da Madeira FSC no Brasil 2012 – 2013**. 1. ed. São Paulo: Pancrom, 2014. 80p.

## APÊNDICE A – CÓDIGO-FONTE DO PROGRAMA COMPUTACIONAL

```
package view;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import javax.swing.JOptionPane;

import java.text.DecimalFormat;

import java.text.NumberFormat;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.ImageIcon;

import javax.swing.JFileChooser;

import javax.swing.text.NumberFormatter;

import model.Medidas;

public class telaInicial extends javax.swing.JFrame {
```

```
ArrayList<Medidas> varMedidas = new ArrayList<>();  
  
double kmod;  
  
double kmod1;  
  
double kmod2;  
  
double kmod3;  
  
double comprimentoH;  
  
double comprimentoV;  
  
double Ng;  
  
double Nq;  
  
double Nsd;  
  
double Mq;  
  
double Mg;  
  
double Msd;  
  
double Vq;  
  
double Vg;  
  
double Vsd;  
  
double base = 5;  
  
double altura = 5;  
  
double area = 50;  
  
String material;
```

```
double I;  
  
double esb;  
  
boolean IgBloqueioKmod = true;  
  
//Verticais  
  
double YgV1;  
  
double YqV1;  
  
double YgV2;  
  
double YqV2;  
  
double psi0V;  
  
double psi1V;  
  
double psi2V;  
  
double Nsd1;  
  
double Nsd2;  
  
boolean IgBloqueioVer = true;  
  
//Horizontal  
  
double q;  
  
double g;  
  
double YgH1;  
  
double YqH1;  
  
double YgH2;
```

```
double YqH2;  
  
double psi0H;  
  
double psi1H;  
  
double psi2H;  
  
double Msd1;  
  
double Msd2;  
  
double Vsd1;  
  
double Vsd2;  
  
boolean lgBloqueioHor = true;
```

```
//Tração
```

```
double fwkp;  
  
double fwkn;  
  
double fwko;  
  
double fwt = 1.8;  
  
double areaT = 50;  
  
boolean lgBloqueioTra = true;
```

```
//Compressão
```

```
double fc0k;  
  
double fwc = 1.4;
```

```
double an;  
  
double areaC = 50;  
  
double Ec0m;  
  
double anguloT;  
  
double Fe;  
  
double ei;  
  
double ea;  
  
double e1;  
  
double ed;  
  
double Mdt;  
  
double Tmd;  
  
double Tnd;  
  
double verifica;  
  
double y;  
  
double k;  
  
double exp;  
  
double ec;  
  
boolean lgBloqueioComp = true;  
  
  
//Flexão  
  
double baseF;  
  
double alturaF = 5;
```



```
double alturaR;
```

```
double lf;
```

```
double L1;
```

```
double Ttd = 0;
```

```
double Tcd = 0;
```

```
double[] alturas = {5, 10, 15, 20, 25, 30};
```

```
boolean vt;
```

```
double ftd;
```

```
double fcd;
```

```
double Bm;
```

```
//Cisalhamento
```

```
double fv0;
```

```
double alturaCl;
```

```
public telaInicial() {  
  
    initComponents();  
  
  
  
    //Listas  
  
    listaConiferas.setEnabled(false);  
  
    listaDicotiledoneas.setEnabled(false);  
  
    listaExtensao90.setEnabled(false);  
  
    listaExtensaoOutro.setEnabled(false);  
  
  
  
    listaConiferas.removeAllItems();  
  
    listaDicotiledoneas.removeAllItems();  
  
    listaExtensao90.removeAllItems();  
  
    listaExtensaoOutro.removeAllItems();  
  
    listaSecoesT.removeAllItems();  
  
    listaSecoesC.removeAllItems();  
  
    listaCargaVariavelV.removeAllItems();  
  
    listaCargaVariavelH.removeAllItems();  
  
    listaBaseF.removeAllItems();  
  
    listaAlturaF.removeAllItems();  
  
  
  
    listaConiferas.addItem("Pinho do Paraná");
```

listaConiferas.addItem("Pinus caribea");

listaConiferas.addItem("Pinus bahamensis");

listaConiferas.addItem("Pinus hondurensis");

listaConiferas.addItem("Pinus elliottii");

listaConiferas.addItem("Pinus oocarpa");

listaConiferas.addItem("Pinus taeda");

listaDicotiledoneas.addItem("Angelim araroba");

listaDicotiledoneas.addItem("Angelim ferro");

listaDicotiledoneas.addItem("Angelim pedra");

listaDicotiledoneas.addItem("Angelim pedra verdadeiro");

listaDicotiledoneas.addItem("Branquilha");

listaDicotiledoneas.addItem("Cafearana");

listaDicotiledoneas.addItem("Canafístula");

listaDicotiledoneas.addItem("Casca grossa");

listaDicotiledoneas.addItem("Castelo");

listaDicotiledoneas.addItem("Cedro amargo");

listaDicotiledoneas.addItem("Cedro doce");

listaDicotiledoneas.addItem("Champagne");

listaDicotiledoneas.addItem("Cupiúba");

listaDicotiledoneas.addItem("Catiúba");

listaDicotiledoneas.addItem("Eucalipto Alba");

listaDicotiledoneas.addItem("Eucalipto Camaldulensis");

listaDicotiledoneas.addItem("Eucalipto Citriodora");

listaDicotiledoneas.addItem("Eucalipto Cloeziana");

listaDicotiledoneas.addItem("Eucalipto Dunnii");

listaDicotiledoneas.addItem("Eucalipto Grandis");

listaDicotiledoneas.addItem("Eucalipto Maculata");

listaDicotiledoneas.addItem("Eucalipto Maidene");

listaDicotiledoneas.addItem("Eucalipto Microcorys");

listaDicotiledoneas.addItem("Eucalipto Paniculata");

listaDicotiledoneas.addItem("Eucalipto Propinqua");

listaDicotiledoneas.addItem("Eucalipto Punctata");

listaDicotiledoneas.addItem("Eucalipto Saligna");

listaDicotiledoneas.addItem("Eucalipto Tereticornis");

listaDicotiledoneas.addItem("Eucalipto Triantha");

listaDicotiledoneas.addItem("Eucalipto Umbra");

listaDicotiledoneas.addItem("Eucalipto Urophylla");

listaDicotiledoneas.addItem("Garapa Roraima");

listaDicotiledoneas.addItem("Guaíçara");

listaDicotiledoneas.addItem("Guarucaia");

listaDicotiledoneas.addItem("Ipê");

listaDicotiledoneas.addItem("Jatobá");

listaDicotiledoneas.addItem("Louro preto");

listaDicotiledoneas.addItem("Maçaranduba");

listaDicotiledoneas.addItem("Mandioqueira");

listaDicotiledoneas.addItem("Oiticica amarela");

listaDicotiledoneas.addItem("Quarubarana");

listaDicotiledoneas.addItem("Sucupira");

listaDicotiledoneas.addItem("Tatajuba");

listaExtensao90.addItem("1 cm");

listaExtensao90.addItem("2 cm");

listaExtensao90.addItem("3 cm");

listaExtensao90.addItem("4 cm");

listaExtensao90.addItem("5 cm");

listaExtensao90.addItem("7,5 cm");

listaExtensao90.addItem("10 cm");

listaExtensao90.addItem("15 cm");

listaExtensaoOutro.addItem("1 cm");

listaExtensaoOutro.addItem("2 cm");

listaExtensaoOutro.addItem("3 cm");

listaExtensaoOutro.addItem("4 cm");

listaExtensaoOutro.addItem("5 cm");

listaExtensaoOutro.addItem("7,5 cm");

listaExtensaoOutro.addItem("10 cm");

listaExtensaoOutro.addItem("15 cm");

listaCargaVariavelV.addItem("Variações uniformes de temperatura em relação à média anual local");

listaCargaVariavelV.addItem("Pressão dinâmica do vento");

listaCargaVariavelV.addItem("Carga acidental em edifícios sem elevadas concentrações de pessoas");

listaCargaVariavelV.addItem("Carga acidental em edifícios com elevadas concentrações de pessoas");

listaCargaVariavelV.addItem("Bibliotecas, arquivos, oficinas e garagens");

listaCargaVariavelH.addItem("Variações uniformes de temperatura em relação à média anual local");

listaCargaVariavelH.addItem("Pressão dinâmica do vento");

listaCargaVariavelH.addItem("Carga acidental em edifícios sem elevadas concentrações de pessoas");

listaCargaVariavelH.addItem("Carga acidental em edifícios com elevadas concentrações de pessoas");

listaCargaVariavelH.addItem("Bibliotecas, arquivos, oficinas e garagens");

listaBaseF.addItem("5 cm");

listaBaseF.addItem("7.5 cm");

```
listaBaseF.addItem("10 cm");  
  
listaBaseF.addItem("12.5 cm");  
  
listaBaseF.addItem("15 cm");  
  
listaBaseF.addItem("17.5 cm");  
  
listaBaseF.addItem("20 cm");  
  
listaBaseF.addItem("22.5 cm");  
  
listaBaseF.addItem("25 cm");  
  
listaBaseF.addItem("27.5 cm");  
  
listaBaseF.addItem("30 cm");
```

```
//Botão próximo
```

```
bpKmod.setEnabled(false);  
  
bpVertical.setEnabled(false);  
  
bpHorizontal.setEnabled(false);  
  
bpFlexao.setEnabled(false);  
  
bpClasseResistencia.setEnabled(false);
```

```
//BotõesCalcular
```

```
bDimensionarT.setEnabled(false);  
  
bDimensionarC.setEnabled(false);  
  
bTravamento.setEnabled(false);
```

```
bRedimCI.setEnabled(false);
```

```
//Botões Verificação
```

```
bVerificarT.setEnabled(false);
```

```
bVerificarC.setEnabled(false);
```

```
bVerificaF.setEnabled(false);
```

```
//Botão resultados
```

```
brTracao.setEnabled(false);
```

```
brCompressao.setEnabled(false);
```

```
brCisalhamento.setEnabled(false);
```

```
//Painéis
```

```
// painelAbas.setEnabled(false);
```

```
painelAbas.setEnabledAt(1, false);
```

```
painelAbas.setEnabledAt(2, false);
```

```
painelAbas.setEnabledAt(3, false);
```

```
painelAbas.setEnabledAt(4, false);
```

```
painelAbas.setEnabledAt(5, false);
```

```
painelAbas.setEnabledAt(6, false);
```

```
painelAbas.setEnabledAt(7, false);
```



```
painelAbas.setEnabledAt(8, false);  
painelPropriedades.setEnabledAt(1, false);
```

```
//Campos não editáveis
```

```
campoYwt.setEditable(false);  
campoKmod.setEditable(false);  
campoKmodT.setEditable(false);  
campoNdT.setEditable(false);  
campoYwt.setEditable(false);  
campoAreaMinT.setEditable(false);  
campoAreaMinC.setEditable(false);  
campoKmodC.setEditable(false);  
campoNdC.setEditable(false);  
campoYwc.setEditable(false);  
campoMdF.setEditable(false);  
campoVdCl.setEditable(false);  
campoEsbeltezT.setEditable(false);  
campoEsbeltezC.setEditable(false);  
campoEstabilidadeC.setEditable(false);  
campoNsd1.setEditable(false);
```

```
campoNsd2.setEditable(false);  
campoMsd1.setEditable(false);  
campoMsd2.setEditable(false);  
campoVsd1.setEditable(false);  
campoVsd2.setEditable(false);  
campoFcdF.setEditable(false);  
campoFtdF.setEditable(false);  
campoKmodF.setEditable(false);  
campoAlturaMinF.setEditable(false);  
campoInstabilidadeL.setEditable(false);  
campoTravamento.setEditable(false);  
campoft0d.setEnabled(false);  
campoft90d.setEnabled(false);  
campoftad.setEnabled(false);  
campoOutroT.setEnabled(false);  
campofc0d.setEnabled(false);  
campofc90d.setEnabled(false);  
campofcad.setEnabled(false);  
campoOutroC.setEnabled(false);  
campoBaseCl.setEditable(false);  
campoAlturaCl.setEditable(false);  
campoFv0Cl.setEditable(false);
```

```
campoKmodCI.setEditable(false);  
campoCondCI.setEditable(false);  
campoRedimCI.setEditable(false);  
areaTxtResultados.setEditable(false);
```

```
//RB
```

```
rbGrandeV.setSelected(true);  
rbGrandeH.setSelected(true);
```

```
// ArrayList seções
```

```
Medidas medidasTemp1 = new Medidas();  
Medidas medidasTemp2 = new Medidas();  
Medidas medidasTemp3 = new Medidas();  
Medidas medidasTemp4 = new Medidas();  
Medidas medidasTemp5 = new Medidas();  
Medidas medidasTemp6 = new Medidas();  
Medidas medidasTemp7 = new Medidas();  
Medidas medidasTemp8 = new Medidas();  
Medidas medidasTemp9 = new Medidas();
```

```
Medidas medidasTemp10 = new Medidas();  
Medidas medidasTemp11 = new Medidas();  
Medidas medidasTemp12 = new Medidas();  
Medidas medidasTemp13 = new Medidas();  
Medidas medidasTemp14 = new Medidas();  
Medidas medidasTemp15 = new Medidas();  
Medidas medidasTemp16 = new Medidas();  
Medidas medidasTemp17 = new Medidas();  
Medidas medidasTemp18 = new Medidas();  
Medidas medidasTemp19 = new Medidas();  
Medidas medidasTemp20 = new Medidas();  
Medidas medidasTemp21 = new Medidas();  
Medidas medidasTemp22 = new Medidas();  
Medidas medidasTemp23 = new Medidas();  
Medidas medidasTemp24 = new Medidas();  
Medidas medidasTemp25 = new Medidas();  
Medidas medidasTemp26 = new Medidas();  
Medidas medidasTemp27 = new Medidas();  
Medidas medidasTemp28 = new Medidas();  
Medidas medidasTemp29 = new Medidas();  
Medidas medidasTemp30 = new Medidas();  
Medidas medidasTemp31 = new Medidas();
```

```
Medidas medidasTemp32 = new Medidas();  
Medidas medidasTemp33 = new Medidas();  
Medidas medidasTemp34 = new Medidas();  
Medidas medidasTemp35 = new Medidas();  
Medidas medidasTemp36 = new Medidas();  
Medidas medidasTemp37 = new Medidas();  
Medidas medidasTemp38 = new Medidas();  
Medidas medidasTemp39 = new Medidas();  
Medidas medidasTemp40 = new Medidas();  
Medidas medidasTemp41 = new Medidas();  
Medidas medidasTemp42 = new Medidas();  
Medidas medidasTemp43 = new Medidas();  
Medidas medidasTemp44 = new Medidas();  
Medidas medidasTemp45 = new Medidas();  
Medidas medidasTemp46 = new Medidas();  
Medidas medidasTemp47 = new Medidas();  
Medidas medidasTemp48 = new Medidas();  
Medidas medidasTemp49 = new Medidas();  
Medidas medidasTemp50 = new Medidas();  
Medidas medidasTemp51 = new Medidas();  
Medidas medidasTemp52 = new Medidas();  
Medidas medidasTemp53 = new Medidas();
```

```
Medidas medidasTemp54 = new Medidas();  
Medidas medidasTemp55 = new Medidas();  
Medidas medidasTemp56 = new Medidas();  
Medidas medidasTemp57 = new Medidas();  
Medidas medidasTemp58 = new Medidas();  
Medidas medidasTemp59 = new Medidas();  
Medidas medidasTemp60 = new Medidas();  
Medidas medidasTemp61 = new Medidas();  
Medidas medidasTemp62 = new Medidas();  
Medidas medidasTemp63 = new Medidas();  
Medidas medidasTemp64 = new Medidas();
```

```
medidasTemp1.setBase(5);  
medidasTemp1.setAltura(10);  
medidasTemp1.setArea(50);  
  
//adicionar no ArrayList  
varMedidas.add(medidasTemp1);  
  
medidasTemp2.setBase(10);  
medidasTemp2.setAltura(5);
```

```
medidasTemp2.setArea(50);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp2);  
  
medidasTemp3.setBase(12.5);  
  
medidasTemp3.setAltura(5);  
  
medidasTemp3.setArea(62.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp3);  
  
medidasTemp4.setBase(5);  
  
medidasTemp4.setAltura(15);  
  
medidasTemp4.setArea(75);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp4);  
  
medidasTemp5.setBase(7.5);  
  
medidasTemp5.setAltura(10);  
  
medidasTemp5.setArea(75);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp5);  
  
medidasTemp6.setBase(15);  
  
medidasTemp6.setAltura(5);  
  
medidasTemp6.setArea(75);  
  
//adicionar no ArrayList
```

```
varMedidas.add(medidasTemp6);

medidasTemp7.setBase(17.5);

medidasTemp7.setAltura(5);

medidasTemp7.setArea(87.5);

//adicionar no ArrayList

varMedidas.add(medidasTemp7);

medidasTemp8.setBase(5);

medidasTemp8.setAltura(20);

medidasTemp8.setArea(100);

//adicionar no ArrayList

varMedidas.add(medidasTemp8);

medidasTemp9.setBase(10);

medidasTemp9.setAltura(10);

medidasTemp9.setArea(100);

//adicionar no ArrayList

varMedidas.add(medidasTemp9);

medidasTemp10.setBase(20);

medidasTemp10.setAltura(5);

medidasTemp10.setArea(100);

//adicionar no ArrayList

varMedidas.add(medidasTemp10);

medidasTemp11.setBase(7.5);
```



```
medidasTemp11.setAltura(15);  
  
medidasTemp11.setArea(112.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp11);  
  
medidasTemp12.setBase(22.5);  
  
medidasTemp12.setAltura(5);  
  
medidasTemp12.setArea(112.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp12);  
  
medidasTemp13.setBase(5);  
  
medidasTemp13.setAltura(25);  
  
medidasTemp13.setArea(125);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp13);  
  
medidasTemp14.setBase(12.5);  
  
medidasTemp14.setAltura(10);  
  
medidasTemp14.setArea(125);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp14);  
  
medidasTemp15.setBase(25);  
  
medidasTemp15.setAltura(5);  
  
medidasTemp15.setArea(125);
```

```
//adicionar no ArrayList

varMedidas.add(medidasTemp15);

medidasTemp16.setBase(27.5);

medidasTemp16.setAltura(5);

medidasTemp16.setArea(137.5);

//adicionar no ArrayList

varMedidas.add(medidasTemp16);

medidasTemp17.setBase(5);

medidasTemp17.setAltura(30);

medidasTemp17.setArea(150);

//adicionar no ArrayList

varMedidas.add(medidasTemp17);

medidasTemp18.setBase(7.5);

medidasTemp18.setAltura(20);

medidasTemp18.setArea(150);

//adicionar no ArrayList

varMedidas.add(medidasTemp18);

medidasTemp19.setBase(10);

medidasTemp19.setAltura(15);

medidasTemp19.setArea(150);

//adicionar no ArrayList

varMedidas.add(medidasTemp19);
```

```
medidasTemp20.setBase(15);  
  
medidasTemp20.setAltura(10);  
  
medidasTemp20.setArea(150);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp20);  
  
medidasTemp21.setBase(30);  
  
medidasTemp21.setAltura(5);  
  
medidasTemp21.setArea(150);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp21);  
  
medidasTemp22.setBase(17.5);  
  
medidasTemp22.setAltura(10);  
  
medidasTemp22.setArea(175);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp22);  
  
medidasTemp23.setBase(7.5);  
  
medidasTemp23.setAltura(25);  
  
medidasTemp23.setArea(187.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp23);  
  
medidasTemp24.setBase(12.5);  
  
medidasTemp24.setAltura(15);
```

```
medidasTemp24.setArea(187.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp24);  
  
medidasTemp25.setBase(10);  
  
medidasTemp25.setAltura(20);  
  
medidasTemp25.setArea(200);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp25);  
  
medidasTemp26.setBase(20);  
  
medidasTemp26.setAltura(10);  
  
medidasTemp26.setArea(200);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp26);  
  
medidasTemp27.setBase(7.5);  
  
medidasTemp27.setAltura(30);  
  
medidasTemp27.setArea(225);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp27);  
  
medidasTemp28.setBase(15);  
  
medidasTemp28.setAltura(15);  
  
medidasTemp28.setArea(225);  
  
//adicionar no ArrayList
```

```
varMedidas.add(medidasTemp28);

medidasTemp29.setBase(22.5);

medidasTemp29.setAltura(10);

medidasTemp29.setArea(225);

//adicionar no ArrayList

varMedidas.add(medidasTemp29);

medidasTemp30.setBase(10);

medidasTemp30.setAltura(25);

medidasTemp30.setArea(250);

//adicionar no ArrayList

varMedidas.add(medidasTemp30);

medidasTemp31.setBase(12.5);

medidasTemp31.setAltura(20);

medidasTemp31.setArea(250);

//adicionar no ArrayList

varMedidas.add(medidasTemp31);

medidasTemp32.setBase(25);

medidasTemp32.setAltura(10);

medidasTemp32.setArea(250);

//adicionar no ArrayList

varMedidas.add(medidasTemp32);

medidasTemp33.setBase(17.5);
```

```
medidasTemp33.setAltura(15);  
  
medidasTemp33.setArea(262.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp33);  
  
medidasTemp34.setBase(27.5);  
  
medidasTemp34.setAltura(10);  
  
medidasTemp34.setArea(275);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp34);  
  
medidasTemp35.setBase(10);  
  
medidasTemp35.setAltura(30);  
  
medidasTemp35.setArea(300);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp35);  
  
medidasTemp36.setBase(15);  
  
medidasTemp36.setAltura(20);  
  
medidasTemp36.setArea(300);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp36);  
  
medidasTemp37.setBase(20);  
  
medidasTemp37.setAltura(15);  
  
medidasTemp37.setArea(300);
```

```
//adicionar no ArrayList

varMedidas.add(medidasTemp37);

medidasTemp38.setBase(30);

medidasTemp38.setAltura(10);

medidasTemp38.setArea(300);

//adicionar no ArrayList

varMedidas.add(medidasTemp38);

medidasTemp39.setBase(12.5);

medidasTemp39.setAltura(25);

medidasTemp39.setArea(312.5);

//adicionar no ArrayList

varMedidas.add(medidasTemp39);

medidasTemp40.setBase(22.5);

medidasTemp40.setAltura(15);

medidasTemp40.setArea(337.5);

//adicionar no ArrayList

varMedidas.add(medidasTemp40);

medidasTemp41.setBase(17.5);

medidasTemp41.setAltura(20);

medidasTemp41.setArea(350);

//adicionar no ArrayList

varMedidas.add(medidasTemp41);
```

```
medidasTemp42.setBase(12.5);  
  
medidasTemp42.setAltura(30);  
  
medidasTemp42.setArea(375);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp42);  
  
medidasTemp43.setBase(15);  
  
medidasTemp43.setAltura(25);  
  
medidasTemp43.setArea(375);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp43);  
  
medidasTemp44.setBase(25);  
  
medidasTemp44.setAltura(15);  
  
medidasTemp44.setArea(375);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp44);  
  
medidasTemp45.setBase(20);  
  
medidasTemp45.setAltura(20);  
  
medidasTemp45.setArea(400);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp45);  
  
medidasTemp46.setBase(27.5);  
  
medidasTemp46.setAltura(15);
```



```
medidasTemp46.setArea(412.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp46);  
  
medidasTemp47.setBase(17.5);  
  
medidasTemp47.setAltura(25);  
  
medidasTemp47.setArea(437.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp47);  
  
medidasTemp48.setBase(15);  
  
medidasTemp48.setAltura(30);  
  
medidasTemp48.setArea(450);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp48);  
  
medidasTemp49.setBase(22.5);  
  
medidasTemp49.setAltura(20);  
  
medidasTemp49.setArea(450);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp49);  
  
medidasTemp50.setBase(30);  
  
medidasTemp50.setAltura(15);  
  
medidasTemp50.setArea(450);  
  
//adicionar no ArrayList
```

```
varMedidas.add(medidasTemp50);

medidasTemp51.setBase(20);

medidasTemp51.setAltura(25);

medidasTemp51.setArea(500);

//adicionar no ArrayList

varMedidas.add(medidasTemp51);

medidasTemp52.setBase(25);

medidasTemp52.setAltura(20);

medidasTemp52.setArea(500);

//adicionar no ArrayList

varMedidas.add(medidasTemp52);

medidasTemp53.setBase(17.5);

medidasTemp53.setAltura(30);

medidasTemp53.setArea(525);

//adicionar no ArrayList

varMedidas.add(medidasTemp53);

medidasTemp54.setBase(27.5);

medidasTemp54.setAltura(20);

medidasTemp54.setArea(550);

//adicionar no ArrayList

varMedidas.add(medidasTemp54);

medidasTemp55.setBase(22.5);
```

```
medidasTemp55.setAltura(25);  
  
medidasTemp55.setArea(562.5);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp55);  
  
medidasTemp56.setBase(20);  
  
medidasTemp56.setAltura(30);  
  
medidasTemp56.setArea(600);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp56);  
  
medidasTemp57.setBase(30);  
  
medidasTemp57.setAltura(20);  
  
medidasTemp57.setArea(600);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp57);  
  
medidasTemp58.setBase(25);  
  
medidasTemp58.setAltura(25);  
  
medidasTemp58.setArea(625);  
  
//adicionar no ArrayList  
  
varMedidas.add(medidasTemp58);  
  
medidasTemp59.setBase(22.5);  
  
medidasTemp59.setAltura(30);  
  
medidasTemp59.setArea(675);
```

```
//adicionar no ArrayList

varMedidas.add(medidasTemp59);

medidasTemp60.setBase(27.5);

medidasTemp60.setAltura(25);

medidasTemp60.setArea(687.5);

//adicionar no ArrayList

varMedidas.add(medidasTemp60);

medidasTemp61.setBase(25);

medidasTemp61.setAltura(30);

medidasTemp61.setArea(750);

//adicionar no ArrayList

varMedidas.add(medidasTemp61);

medidasTemp62.setBase(30);

medidasTemp62.setAltura(25);

medidasTemp62.setArea(750);

//adicionar no ArrayList

varMedidas.add(medidasTemp62);

medidasTemp63.setBase(27.5);

medidasTemp63.setAltura(30);

medidasTemp63.setArea(825);

//adicionar no ArrayList

varMedidas.add(medidasTemp63);
```

```
medidasTemp64.setBase(30);

medidasTemp64.setAltura(30);

medidasTemp64.setArea(900);

//adicionar no ArrayList

varMedidas.add(medidasTemp64);

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

    sentidoPeca = new javax.swing.ButtonGroup();

    grupoMadeira = new javax.swing.ButtonGroup();

    classeResistencia = new javax.swing.ButtonGroup();

    inclinacaoTracao = new javax.swing.ButtonGroup();

    inclinacaoCompressao = new javax.swing.ButtonGroup();

    cargaTeC = new javax.swing.ButtonGroup();

    variabilidadeV = new javax.swing.ButtonGroup();

    variabilidadeH = new javax.swing.ButtonGroup();

    painelAbas = new javax.swing.JTabbedPane();

    painelTriWood = new javax.swing.JPanel();
```

```
txtBemVindo = new javax.swing.JLabel();

btnIniciar = new javax.swing.JButton();

painelPaginaInicial = new javax.swing.JPanel();

btnPaginaPrincipal = new javax.swing.JButton();

txtSelecionarSentido = new javax.swing.JLabel();

rbHorizontal = new javax.swing.JRadioButton();

rbVertical = new javax.swing.JRadioButton();

imgViga = new javax.swing.JLabel();

imgPilar = new javax.swing.JLabel();

painelCarregamentos = new javax.swing.JTabbedPane();

painelHorizontal = new javax.swing.JPanel();

imgVigac = new javax.swing.JLabel();

txtComprimento = new javax.swing.JLabel();

txtUnidadeComprimento = new javax.swing.JLabel();

btnHorizontal = new javax.swing.JButton();

campoComprimentoH = new javax.swing.JFormattedTextField();

jSeparator2 = new javax.swing.JSeparator();

jSeparator6 = new javax.swing.JSeparator();

jLabel4 = new javax.swing.JLabel();

txtDCargaPerm1 = new javax.swing.JLabel();

rbGrandeH = new javax.swing.JRadioButton();

rbPequenaH = new javax.swing.JRadioButton();
```

```
txtDCargaVariH = new javax.swing.JLabel();

txtTipoCargaH = new javax.swing.JLabel();

listaCargaVariavelH = new javax.swing.JComboBox<>();

txtCargagH = new javax.swing.JLabel();

campoCargaQQH = new javax.swing.JFormattedTextField();

txtUnidadeCargagH = new javax.swing.JLabel();

txtCombH = new javax.swing.JLabel();

txtMsd1 = new javax.swing.JLabel();

campoMsd1 = new javax.swing.JTextField();

txtUnMsd1 = new javax.swing.JLabel();

txtMsd2 = new javax.swing.JLabel();

campoMsd2 = new javax.swing.JTextField();

txtUnMsd2 = new javax.swing.JLabel();

bCalculaCombH = new javax.swing.JButton();

jSeparator8 = new javax.swing.JSeparator();

jSeparator9 = new javax.swing.JSeparator();

txtCargaqH = new javax.swing.JLabel();

campoCargaGGH = new javax.swing.JFormattedTextField();

txtUnidadeCargaqH = new javax.swing.JLabel();

txtVsd1 = new javax.swing.JLabel();

campoVsd1 = new javax.swing.JTextField();

txtUnVsd1 = new javax.swing.JLabel();
```

```
txtUnVsd2 = new javax.swing.JLabel();

campoVsd2 = new javax.swing.JTextField();

txtVsd2 = new javax.swing.JLabel();

painelVertical = new javax.swing.JPanel();

txtComprimentoV = new javax.swing.JLabel();

txtUnidadeComprimentoV = new javax.swing.JLabel();

campoComprimentoV = new javax.swing.JFormattedTextField();

txtCargaNgV = new javax.swing.JLabel();

campoCargaNgV = new javax.swing.JFormattedTextField();

txtUnidadeCargaNgV = new javax.swing.JLabel();

bpVertical = new javax.swing.JButton();

txtCargaNqV = new javax.swing.JLabel();

campoCargaNqV = new javax.swing.JFormattedTextField();

txtUnidadeCargaNqV = new javax.swing.JLabel();

imgPilarCarreg = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

jSeparator3 = new javax.swing.JSeparator();

jLabel1 = new javax.swing.JLabel();

txtDCargaPerm = new javax.swing.JLabel();

rbPequenaV = new javax.swing.JRadioButton();

rbGrandeV = new javax.swing.JRadioButton();

jSeparator4 = new javax.swing.JSeparator();
```



```
txtDCargaVari = new javax.swing.JLabel();

listaCargaVariavelV = new javax.swing.JComboBox<>();

txtTipoCarga = new javax.swing.JLabel();

jSeparator5 = new javax.swing.JSeparator();

txtComb = new javax.swing.JLabel();

txtNsd1 = new javax.swing.JLabel();

txtNsd2 = new javax.swing.JLabel();

campoNsd1 = new javax.swing.JTextField();

campoNsd2 = new javax.swing.JTextField();

txtUnNsd1 = new javax.swing.JLabel();

txtUnNsd2 = new javax.swing.JLabel();

bCalculaCombV = new javax.swing.JButton();

jLabel16 = new javax.swing.JLabel();

jLabel17 = new javax.swing.JLabel();

painelPropriedades = new javax.swing.JTabbedPane();

painelClasseResistencia = new javax.swing.JPanel();

txtClasseResistencia = new javax.swing.JLabel();

rbConifera = new javax.swing.JRadioButton();

rbDicotiledonea = new javax.swing.JRadioButton();

bpClasseResistencia = new javax.swing.JButton();

listaConiferas = new javax.swing.JComboBox<>();

listaDicotiledoneas = new javax.swing.JComboBox<>();
```

```
painelCoeficienteModificacao = new javax.swing.JPanel();

tabkmod1 = new javax.swing.JLabel();

tabkmod2 = new javax.swing.JLabel();

tabkmod3 = new javax.swing.JLabel();

txtkmod1 = new javax.swing.JLabel();

txtkmod2 = new javax.swing.JLabel();

txtkmod3 = new javax.swing.JLabel();

txtdefkmod1 = new javax.swing.JLabel();

campoKmod1 = new javax.swing.JTextField();

txtdefkmod3 = new javax.swing.JLabel();

campoKmod3 = new javax.swing.JTextField();

txtdefkmod2 = new javax.swing.JLabel();

campoKmod2 = new javax.swing.JTextField();

bCalcularKmod = new javax.swing.JButton();

campoKmod = new javax.swing.JTextField();

txtKmod = new javax.swing.JLabel();

bpKmod = new javax.swing.JButton();

painelTracao = new javax.swing.JPanel();

txtSelecioneInclinacao = new javax.swing.JLabel();

campoOutroT = new javax.swing.JTextField();

rb0T = new javax.swing.JRadioButton();

rb90T = new javax.swing.JRadioButton();
```

```
rbOutroT = new javax.swing.JRadioButton();

txtgrausT = new javax.swing.JLabel();

txfft0d = new javax.swing.JLabel();

campoft0d = new javax.swing.JTextField();

txfft90dT = new javax.swing.JLabel();

campoft90d = new javax.swing.JTextField();

campoftad = new javax.swing.JTextField();

txftadT = new javax.swing.JLabel();

txtNdT = new javax.swing.JLabel();

campoNdT = new javax.swing.JTextField();

txtkNT = new javax.swing.JLabel();

txtKmodT = new javax.swing.JLabel();

campoKmodT = new javax.swing.JTextField();

txtYwt = new javax.swing.JLabel();

campoYwt = new javax.swing.JTextField();

txtDadosT = new javax.swing.JLabel();

bDimensionarT = new javax.swing.JButton();

txtUnT1 = new javax.swing.JLabel();

txtUnT2 = new javax.swing.JLabel();

txtUnT3 = new javax.swing.JLabel();

txtAreaMinT = new javax.swing.JLabel();

campoAreaMinT = new javax.swing.JTextField();
```

```
txtUnidadeAreaT = new javax.swing.JLabel();

brTracao = new javax.swing.JButton();

listaSecoesT = new javax.swing.JComboBox<>();

txtSecaoT = new javax.swing.JLabel();

bVerificarT = new javax.swing.JButton();

txtEsbeltezT = new javax.swing.JLabel();

campoEsbeltezT = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jSeparator7 = new javax.swing.JSeparator();

jSeparator10 = new javax.swing.JSeparator();

jSeparator11 = new javax.swing.JSeparator();

jLabel6 = new javax.swing.JLabel();

painelCompressao = new javax.swing.JPanel();

txtDadosC = new javax.swing.JLabel();

txtNdC = new javax.swing.JLabel();

txtKmodC = new javax.swing.JLabel();

txtYwc = new javax.swing.JLabel();

campoYwc = new javax.swing.JTextField();

campoKmodC = new javax.swing.JTextField();

campoNdC = new javax.swing.JTextField();

txtkNC = new javax.swing.JLabel();

txtSelecioneInclinacaoC = new javax.swing.JLabel();
```

```
rb0C = new javax.swing.JRadioButton();

bDimensionarC = new javax.swing.JButton();

txafc0d = new javax.swing.JLabel();

campofc0d = new javax.swing.JTextField();

txtUnC1 = new javax.swing.JLabel();

txafc90d = new javax.swing.JLabel();

rb90C = new javax.swing.JRadioButton();

rbOutroC = new javax.swing.JRadioButton();

campoOutroC = new javax.swing.JTextField();

txtgrausC = new javax.swing.JLabel();

campofc90d = new javax.swing.JTextField();

txtUnC2 = new javax.swing.JLabel();

txafcad = new javax.swing.JLabel();

campofcad = new javax.swing.JTextField();

txtUnC3 = new javax.swing.JLabel();

listaExtensao90 = new javax.swing.JComboBox<>();

listaExtensaoOutro = new javax.swing.JComboBox<>();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

txtAreaMinC = new javax.swing.JLabel();

campoAreaMinC = new javax.swing.JTextField();

txtUnidadeAreaC = new javax.swing.JLabel();
```

```
txtSecaoC = new javax.swing.JLabel();

listaSecoesC = new javax.swing.JComboBox<>();

txtEsbeltezC = new javax.swing.JLabel();

campoEsbeltezC = new javax.swing.JTextField();

bVerificarC = new javax.swing.JButton();

txtEstabilidadeC = new javax.swing.JLabel();

campoEstabilidadeC = new javax.swing.JTextField();

jSeparator12 = new javax.swing.JSeparator();

jSeparator13 = new javax.swing.JSeparator();

jLabel7 = new javax.swing.JLabel();

jSeparator14 = new javax.swing.JSeparator();

jLabel8 = new javax.swing.JLabel();

brCompressao = new javax.swing.JButton();

painelFlexao = new javax.swing.JPanel();

txtDadosF = new javax.swing.JLabel();

campoMdF = new javax.swing.JTextField();

txtMdF = new javax.swing.JLabel();

txtkNmF = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jSeparator15 = new javax.swing.JSeparator();

jSeparator16 = new javax.swing.JSeparator();

txtKmodF = new javax.swing.JLabel();
```

```
txtFcdF = new javax.swing.JLabel();

campoFcdF = new javax.swing.JTextField();

campoKmodF = new javax.swing.JTextField();

txtFtdF = new javax.swing.JLabel();

campoFtdF = new javax.swing.JTextField();

txtUniFcdF = new javax.swing.JLabel();

txtUniFtdF = new javax.swing.JLabel();

txtBaseF = new javax.swing.JLabel();

listaBaseF = new javax.swing.JComboBox<>();

bDimensionarFlexao = new javax.swing.JButton();

txtAlturaMinF = new javax.swing.JLabel();

campoAlturaMinF = new javax.swing.JTextField();

txtUniAlturaF = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

txtTravamento = new javax.swing.JLabel();

campoInstabilidadeL = new javax.swing.JTextField();

txtAlturaF = new javax.swing.JLabel();

jSeparator17 = new javax.swing.JSeparator();

listaAlturaF = new javax.swing.JComboBox<>();

jLabel11 = new javax.swing.JLabel();

bVerificaF = new javax.swing.JButton();

jSeparator18 = new javax.swing.JSeparator();
```

```
jLabel12 = new javax.swing.JLabel();

campoTravamento = new javax.swing.JTextField();

bTravamento = new javax.swing.JButton();

bpFlexao = new javax.swing.JButton();

painelCisalhamento = new javax.swing.JPanel();

txtVdCI = new javax.swing.JLabel();

txtDadosCI = new javax.swing.JLabel();

campoVdCI = new javax.swing.JTextField();

txtUniVdCI = new javax.swing.JLabel();

txtBaseCI = new javax.swing.JLabel();

campoBaseCI = new javax.swing.JTextField();

txtUniBase = new javax.swing.JLabel();

txtAlturaCI = new javax.swing.JLabel();

campoAlturaCI = new javax.swing.JTextField();

txtUniAlt = new javax.swing.JLabel();

txtfv0CI = new javax.swing.JLabel();

campoFv0CI = new javax.swing.JTextField();

txtUniFv0 = new javax.swing.JLabel();

txtKmodCI = new javax.swing.JLabel();

campoKmodCI = new javax.swing.JTextField();

jLabel13 = new javax.swing.JLabel();

jSeparator19 = new javax.swing.JSeparator();
```



```
jSeparator20 = new javax.swing.JSeparator();

jLabel14 = new javax.swing.JLabel();

txtCondCI = new javax.swing.JLabel();

campoCondCI = new javax.swing.JTextField();

bVerificaCI = new javax.swing.JButton();

jSeparator21 = new javax.swing.JSeparator();

txtRedimCI = new javax.swing.JLabel();

campoRedimCI = new javax.swing.JTextField();

txtUniRedimCI = new javax.swing.JLabel();

bRedimCI = new javax.swing.JButton();

bCisalhamento = new javax.swing.JButton();

painelResultados = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

areaTxtResultados = new javax.swing.JTextArea();

jLabel15 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("TriWood");

setResizable(false);

txtBemVindo.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N

txtBemVindo.setText("Bem vindo ao TriWood");
```



```

        .addGap(497, 497, 497)

        .addComponent(bIniciar,
javax.swing.GroupLayout.PREFERRED_SIZE, 296,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(painelTriWoodLayout.createSequentialGroup())

        .addGap(461, 461, 461)

        .addComponent(txtBemVindo)))

    .addContainerGap(565, Short.MAX_VALUE)

);

painelTriWoodLayout.setVerticalGroup(

painelTriWoodLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(painelTriWoodLayout.createSequentialGroup())

        .addContainerGap(245, Short.MAX_VALUE)

        .addComponent(txtBemVindo,
javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(105, 105, 105)

        .addComponent(bIniciar, javax.swing.GroupLayout.PREFERRED_SIZE,
69, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(107, 107, 107))

);

painelAbas.addTab("TriWood", painelTriWood);

```

```
bpPaginaPrincipal.setText("Próximo");

bpPaginaPrincipal.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bpPaginaPrincipalActionPerformed(evt);

    }

});

txtSelecionarSentido.setFont(new java.awt.Font("Tahoma", 0, 32)); // NOI18N

txtSelecionarSentido.setText("Selecione o sentido em que a peça se
encontra:");

sentidoPeca.add(rbHorizontal);

rbHorizontal.setFont(new java.awt.Font("Tahoma", 0, 32)); // NOI18N

rbHorizontal.setText("Horizontal");

rbHorizontal.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusLost(java.awt.event.FocusEvent evt) {

        rbHorizontalFocusLost(evt);

    }

});

rbHorizontal.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mousePressed(java.awt.event.MouseEvent evt) {

        rbHorizontalMousePressed(evt);

    }

});
```

```
});
```

```
sentidoPeca.add(rbVertical);
```

```
rbVertical.setFont(new java.awt.Font("Tahoma", 0, 32)); // NOI18N
```

```
rbVertical.setText("Vertical");
```

```
rbVertical.addFocusListener(new java.awt.event.FocusAdapter() {
```

```
    public void focusLost(java.awt.event.FocusEvent evt) {
```

```
        rbVerticalFocusLost(evt);
```

```
    }
```

```
});
```

```
    imgViga.setIcon(new  
    javax.swing.ImageIcon(getClass().getResource("/images/Viga.jpg"))); // NOI18N
```

```
    imgPilar.setIcon(new  
    javax.swing.ImageIcon(getClass().getResource("/images/Pilar.png"))); // NOI18N
```

```
    javax.swing.GroupLayout painelPaginaInicialLayout = new  
    javax.swing.GroupLayout(painelPaginaInicial);
```

```
    painelPaginaInicial.setLayout(painelPaginaInicialLayout);
```

```
    painelPaginaInicialLayout.setHorizontalGroup(
```

```
    painelPaginaInicialLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(painelPaginaInicialLayout.createSequentialGroup()
```

```
.addGroup(painelPaginaInicialLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING)
```

```
    .addGroup(painelPaginaInicialLayout.createSequentialGroup())
```

```
.addGroup(painelPaginaInicialLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING)
```

```
    .addGroup(painelPaginaInicialLayout.createSequentialGroup())
```

```
        .addGap(203, 203, 203)
```

```
        .addComponent(rbHorizontal))
```

```
    .addGroup(painelPaginaInicialLayout.createSequentialGroup())
```

```
        .addGap(69, 69, 69)
```

```
        .addComponent(imgViga)))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 461,  
Short.MAX_VALUE))
```

```
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
painelPaginaInicialLayout.createSequentialGroup())
```

```
        .addGap(0, 229, Short.MAX_VALUE)
```

```
        .addComponent(txtSelecionarSentido,  
javax.swing.GroupLayout.PREFERRED_SIZE, 712,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addGap(67, 67, 67)))
```

```
.addGroup(painelPaginaInicialLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING)
```

```
    .addGroup(painelPaginaInicialLayout.createSequentialGroup())
```

```
        .addComponent(rbVertical)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 123,
Short.MAX_VALUE)
```

```
    .addComponent(bpPaginaPrincipal)
```

```
    .addGap(45, 45, 45))
```

```
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelPaginaInicialLayout.createSequentialGroup())
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(imgPilar)
```

```
    .addGap(273, 273, 273)))
```

```
);
```

```
    painelPaginaInicialLayout.setVerticalGroup(
```

```
    painelPaginaInicialLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(painelPaginaInicialLayout.createSequentialGroup())
```

```
            .addGap(65, 65, 65)
```

```
                .addComponent(txtSelecionarSentido,
javax.swing.GroupLayout.PREFERRED_SIZE, 48,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 101,
Short.MAX_VALUE)
```

```
    .addComponent(imgViga)
```

```
    .addGap(164, 164, 164))
```

```
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelPaginaInicialLayout.createSequentialGroup())
```

```

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(imgPilar)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(painelPaginaInicialLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelPaginaInicialLayout.createSequentialGroup())

.addGroup(painelPaginaInicialLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)

        .addComponent(rbVertical)

        .addComponent(rbHorizontal))

.addGap(63, 63, 63))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelPaginaInicialLayout.createSequentialGroup()

        .addComponent(bpPaginaPrincipal)

        .addGap(31, 31, 31))))

);

painelAbas.addTab("Página inicial", painelPaginaInicial);

imgVigac.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/images/Viga carga.png")); //
NOI18N

```



```
txtComprimento.setText("Comprimento (L) =");

txtUnidadeComprimento.setText("cm");

bpHorizontal.setText("Próximo");

bpHorizontal.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bpHorizontalActionPerformed(evt);

    }

});

campoComprimentoH.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0"))));

campoComprimentoH.addPropertyChangeListener(new
java.beans.PropertyChangeListener() {

    public void propertyChange(java.beans.PropertyChangeEvent evt) {

        campoComprimentoHPropertyChange(evt);

    }

});

campoComprimentoH.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        campoComprimentoHKeyReleased(evt);

    }

});
```

```
});
```

```
jSeparator2.setOrientation(javax.swing.SwingConstants.VERTICAL);
```

```
jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
jLabel4.setText("Determine o comprimento da peça:");
```

```
txtDCargaPerm1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtDCargaPerm1.setText("Determine a carga permanente:");
```

```
variabilidadeH.add(rbGrandeH);
```

```
rbGrandeH.setText("Grande variabilidade");
```

```
rbGrandeH.addFocusListener(new java.awt.event.FocusAdapter() {
```

```
    public void focusGained(java.awt.event.FocusEvent evt) {
```

```
        rbGrandeHFocusGained(evt);
```

```
    }
```

```
});
```

```
variabilidadeH.add(rbPequenaH);
```

```
rbPequenaH.setText("Pequena variabilidade");
```

```
rbPequenaH.addFocusListener(new java.awt.event.FocusAdapter() {
```

```
    public void focusGained(java.awt.event.FocusEvent evt) {
```

```

        rbPequenaHFocusGained(evt);

    }

});

txtDCargaVariH.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

txtDCargaVariH.setText("Determine a carga variável:");

txtTipoCargaH.setText("Selecione o tipo de carga:");

listaCargaVariavelH.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));

listaCargaVariavelH.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        listaCargaVariavelHFocusGained(evt);

    }

});

txtCargagH.setText("Carga variável (q) =");

campoCargaQQH.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0.000"))));

campoCargaQQH.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

```

```
        campoCargaQQHFocusGained(evt);
    }
});

campoCargaQQH.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        campoCargaQQHKeyTyped(evt);
    }
});

txtUnidadeCargagH.setText("kN/m");

txtCombH.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
txtCombH.setText("Combinações:");

txtMsd1.setText("Msd1 =");

txtUnMsd1.setText("kN.cm");

txtMsd2.setText("Msd2 =");

txtUnMsd2.setText("kN.cm");
```

```
bCalculaCombH.setText("Calcular");

bCalculaCombH.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bCalculaCombHActionPerformed(evt);

    }

});

jSeparator8.setOrientation(javax.swing.SwingConstants.VERTICAL);

txtCargaqH.setText("Carga permanente (g) =");

campoCargaGGH.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0.000"))));

campoCargaGGH.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        campoCargaGGHFocusGained(evt);

    }

});

campoCargaGGH.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        campoCargaGGHKeyTyped(evt);

    }

});
```

```
});
```

```
txtUnidadeCargaqH.setText("kN/m");
```

```
txtVsd1.setText("Vsd1 =");
```

```
txtUnVsd1.setText("kN");
```

```
txtUnVsd2.setText("kN");
```

```
txtVsd2.setText("Vsd2 =");
```

```
javax.swing.GroupLayout painelHorizontalLayout = new  
javax.swing.GroupLayout(painelHorizontal);
```

```
painelHorizontal.setLayout(painelHorizontalLayout);
```

```
painelHorizontalLayout.setHorizontalGroup(
```

```
painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(painelHorizontalLayout.createSequentialGroup()
```

```
        .addGap()
```

```
    .addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(painelHorizontalLayout.createSequentialGroup()
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
        .addComponent(txtComprimento)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(campoComprimentoH,
        javax.swing.GroupLayout.PREFERRED_SIZE, 84,
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(txtUnidadeComprimento)
```

```
    .addGap(63, 63, 63))
```

```
    .addComponent(jLabel4,
        javax.swing.GroupLayout.Alignment.TRAILING))
```

```
    .addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 1,
    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(rbPequenaH)
```

```
    .addComponent(rbGrandeH))))
```

```
.addGap(6, 6, 6)
```

```
.addComponent(imgVigac)

.addGap(352, 419, Short.MAX_VALUE))

.addGroup(painelHorizontalLayout.createSequentialGroup())

.addComponent(txtDCargaPerm1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(bCalculaCombH)

.addGap(90, 90, 90)))

.addComponent(jSeparator6)

.addGroup(painelHorizontalLayout.createSequentialGroup())

.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

.addGroup(painelHorizontalLayout.createSequentialGroup())

.addContainerGap()

.addComponent(txtCargaqH)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(campoCargaGGH,
javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(txtUnidadeCargaqH)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```



```
.addComponent(jSeparator2,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
  
.addComponent(jSeparator9))  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING)  
  
.addComponent(txtDCargaVariH)  
  
.addGroup(painelHorizontalLayout.createSequentialGroup())  
  
.addComponent(txtTipoCargaH)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addComponent(listaCargaVariavelH,  
javax.swing.GroupLayout.PREFERRED_SIZE, 435,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
  
.addGroup(painelHorizontalLayout.createSequentialGroup())  
  
.addComponent(txtCargagH)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addComponent(campoCargaQQH,  
javax.swing.GroupLayout.PREFERRED_SIZE, 84,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addComponent(txtUnidadeCargagH)))  
  
.addGap(32, 32, 32)
```

```
.addComponent(jSeparator8,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING)
```

```
.addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
.addGap(266, 266, 266)
```

```
.addComponent(bpHorizontal))
```

```
.addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING)
```

```
.addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
.addGap(9, 9, 9)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING)
```

```
.addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING)
```

```
.addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
.addComponent(txtMsd2)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
        .addComponent(campoMsd2,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(painelHorizontalLayout.createSequentialGroup())

        .addComponent(txtMsd1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoMsd1,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

        .addComponent(txtUnMsd2)

        .addComponent(txtUnMsd1)))

.addGroup(painelHorizontalLayout.createSequentialGroup())

.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

.addGroup(painelHorizontalLayout.createSequentialGroup())

        .addComponent(txtVsd2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoVsd2,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```

.addGroup(painelHorizontalLayout.createSequentialGroup()
            .addComponent(txtVsd1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(campoVsd1,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)
            .addComponent(txtUnVsd2)
            .addComponent(txtUnVsd1))))
        .addComponent(txtCombH))))
    .addGap(46, 46, 46)
);

painelHorizontalLayout.setVerticalGroup(

painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
        .addGroup(painelHorizontalLayout.createSequentialGroup()
            .addGap(6, 6, 6)
            .addComponent(imgVigac)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali

```

```
gnment.LEADING)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING, false)
```

```
    .addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
        .addComponent(jSeparator6,
javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)
```

```
    .addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING, false)
```

```
    .addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
        .addComponent(jLabel4)
```

```
        .addGap(18, 18, 18)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
```

```
    .addComponent(txtComprimento)
```

```
    .addComponent(txtUnidadeComprimento)
```

```
        .addComponent(campoComprimentoH,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
        .addComponent(jSeparator9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addGap(8, 8, 8))
```

```
        .addGroup(painelHorizontalLayout.createSequentialGroup())
```

```
        .addComponent(txtDCargaVariH)
```

```
        .addGap(18, 18, 18)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
```

```
        .addComponent(txtTipoCargaH)
```

```
        .addComponent(listaCargaVariavelH,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
        .addGap(18, 18, 18)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
```

```
        .addComponent(txtCargagH)
```

```
        .addComponent(txtUnidadeCargagH)
```

```
        .addComponent(campoCargaQQH,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
```

```
        .addComponent(txtDCargaPerm1)
```

```
        .addGap(18, 18, 18)
```

```
.addComponent(rbGrandeH)

.addGap(18, 18, 18)

.addComponent(rbPequenaH)

.addGap(18, 18, 18)

.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)

        .addComponent(txtCargaqH)

            .addComponent(campoCargaGGH,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(txtUnidadeCargaqH))

.addGap(0, 0, Short.MAX_VALUE))

.addGroup(painelHorizontalLayout.createSequentialGroup())

        .addComponent(txtCombH)

.addGap(24, 24, 24)

.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)

        .addComponent(txtMsd1)

            .addComponent(campoMsd1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(txtUnMsd1))

.addGap(18, 18, 18)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.BASELINE)
```

```
    .addComponent(txtMsd2)
```

```
    .addComponent(campoMsd2,  
    javax.swing.GroupLayout.PREFERRED_SIZE,  
    javax.swing.GroupLayout.DEFAULT_SIZE,  
    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(txtUnMsd2))
```

```
    .addGap(2, 2, 2)
```

```
    .addComponent(bCalculaCombH)
```

```
    .addGap(1, 1, 1)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.BASELINE)
```

```
    .addComponent(txtVsd1)
```

```
    .addComponent(campoVsd1,  
    javax.swing.GroupLayout.PREFERRED_SIZE,  
    javax.swing.GroupLayout.DEFAULT_SIZE,  
    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(txtUnVsd1))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(painelHorizontalLayout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.BASELINE)
```

```
    .addComponent(txtVsd2)
```

```
    .addComponent(campoVsd2,  
    javax.swing.GroupLayout.PREFERRED_SIZE,  
    javax.swing.GroupLayout.DEFAULT_SIZE,  
    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(txtUnVsd2))
```



```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(bpHorizontal)
```

```
    .addGap(28, 28, 28)))
```

```
        .addComponent(jSeparator2,
javax.swing.GroupLayout.PREFERRED_SIZE, 322,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
        .addComponent(jSeparator8,
javax.swing.GroupLayout.PREFERRED_SIZE, 322,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
```

```
);
```

```
painelCarregamentos.addTab("Horizontal", painelHorizontal);
```

```
txtComprimentoV.setText("Comprimento (L) =");
```

```
txtUnidadeComprimentoV.setText("cm");
```

```
    campoComprimentoV.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0"))));
```

```
    campoComprimentoV.addPropertyChangeListener(new
java.beans.PropertyChangeListener() {
```

```
        public void propertyChange(java.beans.PropertyChangeEvent evt) {
```

```
            campoComprimentoVPropertyChange(evt);
```

```

    }
});

campoComprimentoV.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyReleased(java.awt.event.KeyEvent evt) {

        campoComprimentoVKeyReleased(evt);

    }

});

txtCargaNgV.setText("Carga permanente (Ngk) =");

campoCargaNgV.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0.000"))));

campoCargaNgV.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        campoCargaNgVFocusGained(evt);

    }

});

campoCargaNgV.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        campoCargaNgVKeyTyped(evt);

    }

});

```

```
txtUnidadeCargaNgV.setText("kN");

bpVertical.setText("Próximo");

bpVertical.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bpVerticalActionPerformed(evt);

    }

});

txtCargaNqV.setText("Carga variável (Nqk) =");

campoCargaNqV.setFormatterFactory(new
javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0.000"))));

campoCargaNqV.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        campoCargaNqVFocusGained(evt);

    }

});

campoCargaNqV.addKeyListener(new java.awt.event.KeyAdapter() {

    public void keyTyped(java.awt.event.KeyEvent evt) {

        campoCargaNqVKeyTyped(evt);

    }

});
```

```
    }  
});  
  
txtUnidadeCargaNqV.setText("kN");  
  
imgPilarCarreg.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/images/PilarCarreg.jpg"))); //  
NOI18N  
  
jSeparator3.setOrientation(javax.swing.SwingConstants.VERTICAL);  
  
jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel1.setText("Determine o comprimento da peça:");  
  
txtDCargaPerm.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
txtDCargaPerm.setText("Determine a carga permanente.");  
  
variabilidadeV.add(rbPequenaV);  
rbPequenaV.setText("Pequena variabilidade");  
rbPequenaV.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusGained(java.awt.event.FocusEvent evt) {  
        rbPequenaVFocusGained(evt);  
    }  
})
```

```
});
```

```
variabilidadeV.add(rbGrandeV);
```

```
rbGrandeV.setText("Grande variabilidade");
```

```
rbGrandeV.addFocusListener(new java.awt.event.FocusAdapter() {
```

```
    public void focusGained(java.awt.event.FocusEvent evt) {
```

```
        rbGrandeVFocusGained(evt);
```

```
    }
```

```
});
```

```
txtDCargaVari.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtDCargaVari.setText("Determine a carga variável:");
```

```
listaCargaVariavelV.setModel(new javax.swing.DefaultComboBoxModel<>(new  
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
```

```
listaCargaVariavelV.addFocusListener(new java.awt.event.FocusAdapter() {
```

```
    public void focusGained(java.awt.event.FocusEvent evt) {
```

```
        listaCargaVariavelVFocusGained(evt);
```

```
    }
```

```
});
```

```
txtTipoCarga.setText("Selecione o tipo de carga:");
```

```
txtComb.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

txtComb.setText("Combinações:");

txtNsd1.setText("Nsd1 =");

txtNsd2.setText("Nsd2 =");

txtUnNsd1.setText("kN");

txtUnNsd2.setText("kN");

bCalculaCombV.setText("Calcular");

bCalculaCombV.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bCalculaCombVActionPerformed(evt);

    }

});

jLabel16.setText("*Para carregamentos de compressão usar o sinal negativo.");

jLabel17.setText("*Para carregamentos de compressão usar o sinal negativo.");
```

```

    javax.swing.GroupLayout painelVerticalLayout = new
    javax.swing.GroupLayout(painelVertical);

    painelVertical.setLayout(painelVerticalLayout);

    painelVerticalLayout.setHorizontalGroup(

painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(painelVerticalLayout.createSequentialGroup())

            .addGap(64, 64, 64)

            .addComponent(imgPilarCarreg)

            .addGap(16, 16, 16)

            .addComponent(jSeparator3,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                painelVerticalLayout.createSequentialGroup())

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(bpVertical)

                .addGap(40, 40, 40))

            .addGroup(painelVerticalLayout.createSequentialGroup())

        .addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```
.addGroup(painelVerticalLayout.createSequentialGroup())

    .addGap(27, 27, 27)

.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(painelVerticalLayout.createSequentialGroup())

        .addComponent(txtNsd2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addComponent(campoNsd2,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGroup(painelVerticalLayout.createSequentialGroup())

        .addComponent(txtNsd1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addComponent(campoNsd1,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(txtUnNsd2)

    .addGroup(painelVerticalLayout.createSequentialGroup())

        .addComponent(txtUnNsd1)

        .addGap(18, 18, 18)
```



```
.addComponent(bCalculaCombV))))  
  
.addGroup(painelVerticalLayout.createSequentialGroup()  
  
.addGap(18, 18, 18)  
  
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.LEADING)  
  
.addGroup(painelVerticalLayout.createSequentialGroup()  
  
.addComponent(rbGrandeV)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
  
.addComponent(rbPequenaV))  
  
.addComponent(txtDCargaPerm)  
  
.addGroup(painelVerticalLayout.createSequentialGroup()  
  
.addComponent(txtCargaNgV)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addComponent(campoCargaNgV,  
javax.swing.GroupLayout.PREFERRED_SIZE, 84,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addComponent(txtUnidadeCargaNgV)  
  
.addGap(18, 18, 18)  
  
.addComponent(jLabel16))  
  
.addComponent(jSeparator1,  
javax.swing.GroupLayout.PREFERRED_SIZE, 1001,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jSeparator4,
javax.swing.GroupLayout.PREFERRED_SIZE, 995,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(paineVerticalLayout.createSequentialGroup())

        .addComponent(txtTipoCarga)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(listaCargaVariavelV,
javax.swing.GroupLayout.PREFERRED_SIZE, 435,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addComponent(txtDCargaVari)

        .addGroup(paineVerticalLayout.createSequentialGroup())

        .addComponent(txtCargaNqV)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoCargaNqV,
javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUnidadeCargaNqV)

        .addGap(18, 18, 18)

        .addComponent(jLabel17))

        .addGroup(paineVerticalLayout.createSequentialGroup())

        .addComponent(txtComprimentoV)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoComprimentoV,
```

```

javax.swing.GroupLayout.PREFERRED_SIZE, 84,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUnidadeComprimentoV))

        .addComponent(jLabel1)

        .addComponent(jSeparator5,
javax.swing.GroupLayout.PREFERRED_SIZE, 995,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtComb))))

        .addContainerGap(64, Short.MAX_VALUE))))

);

painelVerticalLayout.setVerticalGroup(

painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(painelVerticalLayout.createSequentialGroup())

        .addGap(37, 37, 37)

        .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(txtComprimentoV)

        .addComponent(txtUnidadeComprimentoV)

        .addComponent(campoComprimentoV,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGap(18, 18, 18)
```

```
.addComponent(jSeparator1,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addComponent(txtDCargaPerm)
```

```
.addGap(18, 18, 18)
```

```
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.BASELINE)
```

```
.addComponent(rbPequenaV)
```

```
.addComponent(rbGrandeV))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.BASELINE)
```

```
.addComponent(txtCargaNgV)
```

```
.addComponent(campoCargaNgV,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(txtUnidadeCargaNgV)
```

```
.addComponent(jLabel16))
```

```
.addGap(18, 18, 18)
```

```
.addComponent(jSeparator4,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
    .addGap(18, 18, 18)  
  
    .addComponent(txtDCargaVari)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
  
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.BASELINE)  
  
    .addComponent(txtTipoCarga)  
  
        .addComponent(listaCargaVariavelV,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
  
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.BASELINE)  
  
    .addComponent(txtCargaNqV)  
  
    .addComponent(txtUnidadeCargaNqV)  
  
        .addComponent(campoCargaNqV,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
    .addComponent(jLabel17))  
  
.addGap(18, 18, 18)  
  
    .addComponent(jSeparator5,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(txtComb)
```

```
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(painelVerticalLayout.createSequentialGroup())
```

```
        .addGap(24, 24, 24)
```

```
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(txtNsd1)
```

```
        .addComponent(campoNsd1,  
            javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE,  
            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(txtUnNsd1))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(txtNsd2)
```

```
        .addComponent(campoNsd2,  
            javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE,  
            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(txtUnNsd2)))
```

```
    .addGroup(painelVerticalLayout.createSequentialGroup())
```

```
        .addGap(42, 42, 42)
```

```

        .addComponent(bCalculaCombV)))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(bpVertical)

        .addGap(48, 48, 48))

    .addGroup(painelVerticalLayout.createSequentialGroup())

    .addGroup(painelVerticalLayout.createParallelGroup(javax.swing.GroupLayout.Align
    ment.LEADING)

        .addGroup(painelVerticalLayout.createSequentialGroup())

            .addContainerGap()

            .addComponent(imgPilarCarreg))

            .addComponent(jSeparator3,
    javax.swing.GroupLayout.PREFERRED_SIZE, 565,
    javax.swing.GroupLayout.PREFERRED_SIZE))

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE))

    );

    painelCarregamentos.addTab("Vertical", painelVertical);

    painelAbas.addTab("Carregamentos", painelCarregamentos);

    txtClasseResistencia.setFont(new java.awt.Font("Tahoma", 0, 20)); // NOI18N

    txtClasseResistencia.setText("Escolha o grupo da madeira:");

```

```
grupoMadeira.add(rbConifera);

rbConifera.setText("Conífera");

rbConifera.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mousePressed(java.awt.event.MouseEvent evt) {

        rbConiferaMousePressed(evt);

    }

});

rbConifera.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        rbConiferaActionPerformed(evt);

    }

});

grupoMadeira.add(rbDicotiledonea);

rbDicotiledonea.setText("Dicotiledônea");

rbDicotiledonea.addFocusListener(new java.awt.event.FocusAdapter() {

    public void focusGained(java.awt.event.FocusEvent evt) {

        rbDicotiledoneaFocusGained(evt);

    }

});

rbDicotiledonea.addActionListener(new java.awt.event.ActionListener() {
```



```

public void actionPerformed(java.awt.event.ActionEvent evt) {
    rbDicotiledoneaActionPerformed(evt);
}
});

```

```

bpClasseResistencia.setText("Próximo");

bpClasseResistencia.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bpClasseResistenciaActionPerformed(evt);
    }
});

```

```

listaConiferas.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));

```

```

listaConiferas.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        listaConiferasFocusGained(evt);
    }
});

```

```

listaDicotiledoneas.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));

```

```

listaDicotiledoneas.addFocusListener(new java.awt.event.FocusAdapter() {

```

```

public void focusGained(java.awt.event.FocusEvent evt) {

    listaDicotiledoneasFocusGained(evt);

}

});

    javax.swing.GroupLayout painelClasseResistenciaLayout = new
javax.swing.GroupLayout(painelClasseResistencia);

    painelClasseResistencia.setLayout(painelClasseResistenciaLayout);

    painelClasseResistenciaLayout.setHorizontalGroup(

painelClasseResistenciaLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

        .addGroup(painelClasseResistenciaLayout.createSequentialGroup()

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(bpClasseResistencia)

            .addGap(45, 45, 45))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelClasseResistenciaLayout.createSequentialGroup()

            .addGroup(painelClasseResistenciaLayout.createParallelGroup(javax.s
wing.GroupLayout.Alignment.LEADING)

                .addGroup(painelClasseResistenciaLayout.createSequentialGroup()

                    .addGap(298, 298, 298)

                    .addComponent(rbConifera))

                .addGroup(painelClasseResistenciaLayout.createSequentialGroup()

```

```

        .addGap(247, 247, 247)

        .addComponent(listaConiferas,
javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(painelClasseResistenciaLayout.createSequentialGroup())

        .addGap(477, 477, 477)

        .addComponent(txtClasseResistencia)))

    .addGroup(painelClasseResistenciaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(painelClasseResistenciaLayout.createSequentialGroup()

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 138,
Short.MAX_VALUE)

                .addComponent(rbDicotiledonea)

                    .addGap(408, 408, 408))

            .addGroup(painelClasseResistenciaLayout.createSequentialGroup()

                .addGap(16, 16, 16)

                    .addComponent(listaDicotiledoneas,
javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE)

                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))))

    );

    painelClasseResistenciaLayout.setVerticalGroup(

painelClasseResistenciaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```
.addGroup(painelClasseResistenciaLayout.createSequentialGroup())

    .addGap(50, 50, 50)

    .addComponent(txtClasseResistencia)

    .addGap(49, 49, 49)

.addGroup(painelClasseResistenciaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(rbConifera)

    .addComponent(rbDicotiledonea))

    .addGap(42, 42, 42)

.addGroup(painelClasseResistenciaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(listaConiferas,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(listaDicotiledoneas,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 322,
    Short.MAX_VALUE)

    .addComponent(bpClasseResistencia)

    .addGap(28, 28, 28))

);

painelPropriedades.addTab("Classe de resist\u00eancia", painelClasseResistencia);
```

```
tabkmod1.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/images/kmod1.jpg"))); // NOI18N
```

```
tabkmod2.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/images/kmod2.jpg"))); // NOI18N
```

```
tabkmod3.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/images/kmod3.jpg"))); // NOI18N
```

```
txtkmod1.setText("Kmod1 - Classe de carregamento ");
```

```
txtkmod2.setText("Kmod2 - Classe de umidade ");
```

```
txtkmod3.setText("Kmod3 - Classe de categoria");
```

```
txtdefkmod1.setText("Defina o valor para Kmod1");
```

```
campoKmod1.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusGained(java.awt.event.FocusEvent evt) {  
        campoKmod1FocusGained(evt);  
    }  
});
```

```
txtdefkmod3.setText("Defina o valor para Kmod3");
```

```
campoKmod3.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mousePressed(java.awt.event.MouseEvent evt) {  
        campoKmod3MousePressed(evt);  
    }  
});
```

```
txtdefkmod2.setText("Defina o valor para Kmod2");
```

```
campoKmod2.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusGained(java.awt.event.FocusEvent evt) {  
        campoKmod2FocusGained(evt);  
    }  
});
```

```
bCalcularKmod.setText("Calcular Kmod");
```

```
bCalcularKmod.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        bCalcularKmodActionPerformed(evt);  
    }  
});
```

```
txtKmod.setText("Kmod =");
```

```
bpKmod.setText("Próximo");
```

```
bpKmod.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bpKmodActionPerformed(evt);
    }
});
```

```
javax.swing.GroupLayout painelCoeficienteModificacaoLayout = new
javax.swing.GroupLayout(painelCoeficienteModificacao);
```

```
painelCoeficienteModificacao.setLayout(painelCoeficienteModificacaoLayout);
```

```
painelCoeficienteModificacaoLayout.setHorizontalGroup(
```

```
painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup()
```

```
        .addGap(205, 205, 205)
```

```
        .addComponent(txtkmod1)
```

```
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(txtkmod3)
```

```
    .addGap(290, 290, 290))
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
.addGap(22, 22, 22)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
.addComponent(tabkmod1)
```

```
.addGap(35, 35, 35)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addComponent(tabkmod3)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
.addComponent(txtdefkmod3)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addComponent(campoKmod3,  
javax.swing.GroupLayout.PREFERRED_SIZE, 82,  
javax.swing.GroupLayout.PREFERRED_SIZE))))
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
.addComponent(txtdefkmod1)
```



```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
        .addComponent(campoKmod1,  
javax.swing.GroupLayout.PREFERRED_SIZE, 82,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
        .addComponent(txtdefkmod2)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
        .addComponent(campoKmod2,  
javax.swing.GroupLayout.PREFERRED_SIZE, 82,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
        .addComponent(tabkmod2)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.Gro  
upLayout.Alignment.LEADING)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
        .addGap(229, 229, 229)
```

```
        .addComponent(txtKmod)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.Gro  
upLayout.Alignment.LEADING)
```

```
        .addComponent(campoKmod,  
javax.swing.GroupLayout.PREFERRED_SIZE, 117,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

        .addComponent(bCalculaKmod)))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelCoeficienteModificacaoLayout.createSequentialGroup())

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(bpKmod)

        .addGap(54, 54, 54))))))

.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())

        .addGap(187, 187, 187)

        .addComponent(txtkmod2)))

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

);

painelCoeficienteModificacaoLayout.setVerticalGroup(

painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.A
lignment.LEADING)

        .addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())

        .addGap(12, 12, 12)

.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.BASELINE)

        .addComponent(txtkmod1)

        .addComponent(txtkmod3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(tabkmod1)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
    .addComponent(tabkmod3)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(txtdefkmod3)
```

```
        .addComponent(campoKmod3,  
            javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE,  
            javax.swing.GroupLayout.PREFERRED_SIZE))))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(txtdefkmod1)
```

```
        .addComponent(campoKmod1,  
            javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE,  
            javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(txtkmod2)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(campoKmod,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(txtKmod))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(bCalculaKmod)
```

```
    .addGap(53, 53, 53))
```

```
.addGroup(painelCoeficienteModificacaoLayout.createSequentialGroup())
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
    .addComponent(bpKmod)
```

```
    .addComponent(tabkmod2))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
```

```
.addGroup(painelCoeficienteModificacaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(txtdefkmod2)
```

```

        .addComponent(campoKmod2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addGap(34, 34, 34))

```

```

    );

```

```

    painelPropriedades.addTab("Coeficiente de modificação",
painelCoeficienteModificacao);

```

```

    painelAbas.addTab("Propriedades", painelPropriedades);

```

```

    txtSelecioneInclinacao.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

```

```

    txtSelecioneInclinacao.setText("Selecione o ângulo de inclinação da tensão
característica em relação as fibras:");

```

```

    inclinacaoTracao.add(rb0T);

```

```

    rb0T.setText("Paralela (0° a 6°)");

```

```

    rb0T.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

            rb0TActionPerformed(evt);

```

```

        }

```

```

    });

```

```

    inclinacaoTracao.add(rb90T);

```

```
rb90T.setText("Normal (90°)");

rb90T.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        rb90TActionPerformed(evt);

    }

});

inclinacaoTracao.add(rbOutroT);

rbOutroT.setText("Outro =");

rbOutroT.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        rbOutroTActionPerformed(evt);

    }

});

txtgrausT.setText("0");

txtft0d.setText("ft0,d =");

txtft90dT.setText("ft90,d =");

txtftadT.setText("fta,d =");
```

```
txtNdT.setText("Nd =");
```

```
txtkNT.setText("kN");
```

```
txtKmodT.setText("Kmod =");
```

```
txtYwt.setText("Ywt =");
```

```
campoYwt.setText("1,8");
```

```
txtDadosT.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtDadosT.setText("Dados:");
```

```
bDimensionarT.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
bDimensionarT.setText("Dimensionar");
```

```
bDimensionarT.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        bDimensionarTActionPerformed(evt);
```

```
    }
```

```
});
```

```
txtUnT1.setText("kN/cm²");
```

```
txtUnT2.setText("kN/cm²");
```

```
txtUnT3.setText("kN/cm²");
```

```
txtAreaMinT.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtAreaMinT.setText("Área mínima da seção:");
```

```
campoAreaMinT.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtUnidadeAreaT.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtUnidadeAreaT.setText("cm²");
```

```
brTracao.setText("Ver resultados");
```

```
brTracao.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        brTracaoActionPerformed(evt);
```

```
    }
```

```
});
```

```
listaSecoesT.setModel(new javax.swing.DefaultComboBoxModel<>(new String[]  
{ "Item 1", "Item 2", "Item 3", "Item 4" }));
```



```
listaSecoesT.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusGained(java.awt.event.FocusEvent evt) {  
        listaSecoesTFocusGained(evt);  
    }  
});  
  
txtSecaoT.setText("Selecione a opção de seção que deseja utilizar:");  
  
bVerificarT.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
bVerificarT.setText("Verificar");  
bVerificarT.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        bVerificarTActionPerformed(evt);  
    }  
});  
  
txtEsbeltezT.setText("Esbeltez");  
  
jLabel5.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N  
jLabel5.setText("Dimensionamento");  
  
jSeparator10.setOrientation(javax.swing.SwingConstants.VERTICAL);
```

```

jLabel6.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel6.setText("Verificação");

    javax.swing.GroupLayout painelTracaoLayout = new
javax.swing.GroupLayout(painelTracao);

    painelTracao.setLayout(painelTracaoLayout);

    painelTracaoLayout.setHorizontalGroup(

painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jSeparator7)

        .addComponent(jSeparator11)

        .addGroup(painelTracaoLayout.createSequentialGroup()
            .addComponent(jSeparator11)
            .addGap(10, 10, 10)
            .addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel5)
                .addGroup(painelTracaoLayout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel6)
                        .addComponent(jLabel7)
                        .addComponent(jLabel8)
                        .addComponent(jLabel9)
                        .addComponent(jLabel10)
                        .addComponent(jLabel11)
                        .addComponent(jLabel12)
                        .addComponent(jLabel13)
                        .addComponent(jLabel14)
                        .addComponent(jLabel15)
                        .addComponent(jLabel16)
                        .addComponent(jLabel17)
                        .addComponent(jLabel18)
                        .addComponent(jLabel19)
                        .addComponent(jLabel20)
                        .addComponent(jLabel21)
                        .addComponent(jLabel22)
                        .addComponent(jLabel23)
                        .addComponent(jLabel24)
                        .addComponent(jLabel25)
                        .addComponent(jLabel26)
                        .addComponent(jLabel27)
                        .addComponent(jLabel28)
                        .addComponent(jLabel29)
                        .addComponent(jLabel30)
                        .addComponent(jLabel31)
                        .addComponent(jLabel32)
                        .addComponent(jLabel33)
                        .addComponent(jLabel34)
                        .addComponent(jLabel35)
                        .addComponent(jLabel36)
                        .addComponent(jLabel37)
                        .addComponent(jLabel38)
                        .addComponent(jLabel39)
                        .addComponent(jLabel40)
                        .addComponent(jLabel41)
                        .addComponent(jLabel42)
                        .addComponent(jLabel43)
                        .addComponent(jLabel44)
                        .addComponent(jLabel45)
                        .addComponent(jLabel46)
                        .addComponent(jLabel47)
                        .addComponent(jLabel48)
                        .addComponent(jLabel49)
                        .addComponent(jLabel50)
                        .addComponent(jLabel51)
                        .addComponent(jLabel52)
                        .addComponent(jLabel53)
                        .addComponent(jLabel54)
                        .addComponent(jLabel55)
                        .addComponent(jLabel56)
                        .addComponent(jLabel57)
                        .addComponent(jLabel58)
                        .addComponent(jLabel59)
                        .addComponent(jLabel60)
                        .addComponent(jLabel61)
                        .addComponent(jLabel62)
                        .addComponent(jLabel63)
                        .addComponent(jLabel64)
                        .addComponent(jLabel65)
                        .addComponent(jLabel66)
                        .addComponent(jLabel67)
                        .addComponent(jLabel68)
                        .addComponent(jLabel69)
                        .addComponent(jLabel70)
                        .addComponent(jLabel71)
                        .addComponent(jLabel72)
                        .addComponent(jLabel73)
                        .addComponent(jLabel74)
                        .addComponent(jLabel75)
                        .addComponent(jLabel76)
                        .addComponent(jLabel77)
                        .addComponent(jLabel78)
                        .addComponent(jLabel79)
                        .addComponent(jLabel80)
                        .addComponent(jLabel81)
                        .addComponent(jLabel82)
                        .addComponent(jLabel83)
                        .addComponent(jLabel84)
                        .addComponent(jLabel85)
                        .addComponent(jLabel86)
                        .addComponent(jLabel87)
                        .addComponent(jLabel88)
                        .addComponent(jLabel89)
                        .addComponent(jLabel90)
                        .addComponent(jLabel91)
                        .addComponent(jLabel92)
                        .addComponent(jLabel93)
                        .addComponent(jLabel94)
                        .addComponent(jLabel95)
                        .addComponent(jLabel96)
                        .addComponent(jLabel97)
                        .addComponent(jLabel98)
                        .addComponent(jLabel99)
                    )
                )
            )
        )
    )

```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(txtDadosT)
```

```
    .addGroup(painelTracaoLayout.createSequentialGroup())
```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
    .addComponent(txtYwt,
    javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(javax.swing.GroupLayout.Alignment.LEADING,
    painelTracaoLayout.createSequentialGroup())
```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(txtKmodT)
```

```
    .addComponent(txtNdT))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
    .addComponent(campoYwt)
```

```
    .addComponent(campoNdT)
```

```
    .addComponent(campoKmodT,
    javax.swing.GroupLayout.PREFERRED_SIZE, 70,
    javax.swing.GroupLayout.PREFERRED_SIZE))))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(txtkNT)))
```

```
.addGap(79, 79, 79)

        .addComponent(jSeparator10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

        .addGroup(painelTracaoLayout.createSequentialGroup())

        .addGap(12, 12, 12)

        .addComponent(txtSelecioneInclinacao))

        .addGroup(painelTracaoLayout.createSequentialGroup())

.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

        .addGroup(painelTracaoLayout.createSequentialGroup())

        .addGap(82, 82, 82)

        .addComponent(rb0T))

        .addGroup(painelTracaoLayout.createSequentialGroup())

        .addGap(53, 53, 53)

        .addComponent(txtft0d)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoft0d,
javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addComponent(txtUnT1)))

.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(painelTracaoLayout.createSequentialGroup())

.addGap(190, 190, 190)

.addComponent(rb90T)

.addGap(211, 211, 211)

.addComponent(rbOutroT)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(campoOutroT,
javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(txtgrausT))

.addGroup(painelTracaoLayout.createSequentialGroup())

.addGap(135, 135, 135)

.addComponent(txtft90dT)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(campoft90d,
javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(23, 23, 23)

.addComponent(txtUnT2)
```

```

        .addGap(120, 120, 120)

        .addComponent(txtftadT)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoftad,
javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUnT3))))

        .addGroup(painelTracaoLayout.createSequentialGroup())

        .addGap(327, 327, 327)

        .addComponent(txtAreaMinT)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoAreaMinT,
javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUnidadeAreaT))

        .addGroup(painelTracaoLayout.createSequentialGroup())

        .addGap(431, 431, 431)

        .addComponent(bDimensionarT))))

.addComponent(jLabel6)

.addGroup(painelTracaoLayout.createSequentialGroup())

        .addComponent(txtSecaoT)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(listaSecoesT,
javax.swing.GroupLayout.PREFERRED_SIZE, 482,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.TRAILING)

        .addComponent(bVerificarT)

        .addGroup(painelTracaoLayout.createSequentialGroup())

        .addComponent(txtEsbeltezT)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoEsbeltezT,
javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))))

        .addContainerGap(217, Short.MAX_VALUE))))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelTracaoLayout.createSequentialGroup()

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(brTracao)

        .addGap(59, 59, 59))

);

painelTracaoLayout.setVerticalGroup(

painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,

```

```
painelTracaoLayout.createSequentialGroup()  
  
    .addContainerGap()  
  
    .addComponent(jLabel5)  
  
    .addGap(6, 6, 6)  
  
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.LEADING)  
  
    .addGroup(painelTracaoLayout.createSequentialGroup()  
  
        .addComponent(jSeparator10,  
javax.swing.GroupLayout.PREFERRED_SIZE, 278,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))  
  
    .addGroup(painelTracaoLayout.createSequentialGroup()  
  
        .addComponent(jSeparator7,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.LEADING)  
  
    .addGroup(painelTracaoLayout.createSequentialGroup()  
  
        .addComponent(txtSelecaoInclinacao)  
  
        .addGap(18, 18, 18)  
  
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.TRAILING)
```



```
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
painelTracaoLayout.createSequentialGroup())
```

```
        .addComponent(rb0T)
```

```
        .addGap(18, 18, 18)
```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
```

```
        .addComponent(txtft0d)
```

```
        .addComponent(campoft0d,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(txtUnT1)))
```

```
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
painelTracaoLayout.createSequentialGroup())
```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
```

```
        .addComponent(rb90T)
```

```
        .addComponent(campoOutroT,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(rbOutroT)
```

```
        .addComponent(txtgrausT))
```

```
        .addGap(18, 18, 18)
```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
```

```
.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
```

ment.BASELINE)

.addComponent(txtft90dT)

.addComponent(campoft90d,  
javax.swing.GroupLayout.PREFERRED\_SIZE,  
javax.swing.GroupLayout.DEFAULT\_SIZE,  
javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(txtUnT2))

.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtftadT)

.addComponent(campoftad,  
javax.swing.GroupLayout.PREFERRED\_SIZE,  
javax.swing.GroupLayout.DEFAULT\_SIZE,  
javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(painelTracaoLayout.createSequentialGroup())

.addComponent(txtUnT3)

.addGap(2, 2, 2))))))

.addGroup(painelTracaoLayout.createSequentialGroup())

.addComponent(txtDadosT)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelTracaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtNdT)

```

        .addComponent(campoNdT,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtkNT))

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelTracaoLayout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)

        .addComponent(campoKmodT,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtKmodT))

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelTracaoLayout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)

        .addComponent(txtYwt)

        .addComponent(campoYwt,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE))))

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(bDimensionarT)

        .addGap(32, 32, 32)

.addGroup(painelTracaoLayout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)

```

```
.addComponent(txtAreaMinT)

        .addComponent(campoAreaMinT,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUnidadeAreaT)

        .addGap(33, 33, 33)))

        .addComponent(jSeparator1,
javafx.swing.GroupLayout.PREFERRED_SIZE, 10,
javafx.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel6)

        .addGap(18, 18, 18)

.addGroup(painelTracaoLayout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)

        .addComponent(listaSecoesT,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtSecaoT))

        .addGap(34, 34, 34)

.addGroup(painelTracaoLayout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)

        .addComponent(txtEsbeltezT)

        .addComponent(campoEsbeltezT,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGap(29, 29, 29)

.addComponent(bVerificarT)

.addGap(18, 18, 18)

.addComponent(brTracao)

.addGap(40, 40, 40)

);

painelAbas.addTab("Tração", painelTracao);

txtDadosC.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

txtDadosC.setText("Dados:");

txtNdC.setText("Nd =");

txtKmodC.setText("Kmod =");

txtYwc.setText("Ywc =");

campoYwc.setText("1,4");

txtkNC.setText("kN");
```

```
txtSelecioneInclinacaoC.setFont(new java.awt.Font("Tahoma", 0, 18)); //
NOI18N
```

```
txtSelecioneInclinacaoC.setText("Selecione o ângulo de inclinação da tensão
característica em relação as fibras:");
```

```
inclinacaoCompressao.add(rb0C);
```

```
rb0C.setText("Paralela (0° a 6°)");
```

```
rb0C.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        rb0CActionPerformed(evt);
    }
});
```

```
bDimensionarC.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
bDimensionarC.setText("Dimensionar");
```

```
bDimensionarC.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bDimensionarCActionPerformed(evt);
    }
});
```

```
txtfc0d.setText("fc0,d =");
```

```
txtUnC1.setText("kN/cm²");
```

```
txafc90d.setText("fc90,d =");
```

```
inclinacaoCompressao.add(rb90C);
```

```
rb90C.setText("Normal (90°)");
```

```
rb90C.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        rb90CActionPerformed(evt);  
    }  
});
```

```
inclinacaoCompressao.add(rbOutroC);
```

```
rbOutroC.setText("Outro =");
```

```
rbOutroC.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        rbOutroCActionPerformed(evt);  
    }  
});
```

```
txtgrausC.setText("0");
```

```
txtUnC2.setText("kN/cm2");
```

```
txtfcad.setText("fca,d =");
```

```
txtUnC3.setText("kN/cm2");
```

```
listaExtensao90.setModel(new javax.swing.DefaultComboBoxModel<>(new  
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
```

```
listaExtensao90.addFocusListener(new java.awt.event.FocusAdapter() {  
  
    public void focusGained(java.awt.event.FocusEvent evt) {  
  
        listaExtensao90FocusGained(evt);  
  
    }  
  
});
```

```
listaExtensaoOutro.setModel(new javax.swing.DefaultComboBoxModel<>(new  
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
```

```
listaExtensaoOutro.addFocusListener(new java.awt.event.FocusAdapter() {  
  
    public void focusGained(java.awt.event.FocusEvent evt) {  
  
        listaExtensaoOutroFocusGained(evt);  
  
    }  
  
});
```

```
jLabel2.setText("Extensão da carga");
```



```
jLabel3.setText("Extensão da carga");
```

```
txtAreaMinC.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtAreaMinC.setText("Área mínima da seção:");
```

```
campoAreaMinC.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtUnidadeAreaC.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtUnidadeAreaC.setText("cm²");
```

```
txtSecaoC.setText("Selecione a opção de seção que deseja utilizar:");
```

```
listaSecoesC.setModel(new javax.swing.DefaultComboBoxModel<>(new  
String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
```

```
listaSecoesC.addFocusListener(new java.awt.event.FocusAdapter() {
```

```
    public void focusGained(java.awt.event.FocusEvent evt) {
```

```
        listaSecoesCFocusGained(evt);
```

```
    }
```

```
});
```

```
txtEsbeltezC.setText("Esbeltez");
```

```
bVerificarC.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

bVerificarC.setText("Verificar");

bVerificarC.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bVerificarCActionPerformed(evt);

    }

});

txtEstabilidadeC.setText("Estabilidade");

jSeparator12.setOrientation(javax.swing.SwingConstants.VERTICAL);

jLabel7.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel7.setText("Dimensionamento");

jLabel8.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel8.setText("Verificação");

brCompressao.setText("Ver resultados");

brCompressao.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        brCompressaoActionPerformed(evt);

    }

});
```

```
    }  
  });  
  
  javax.swing.GroupLayout painelCompressaoLayout = new  
  javax.swing.GroupLayout(painelCompressao);  
  
  painelCompressao.setLayout(painelCompressaoLayout);  
  
  painelCompressaoLayout.setHorizontalGroup(  
  
  painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
  .addGroup(painelCompressaoLayout.createSequentialGroup()  
  
  .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
  .addComponent(jSeparator13)  
  
  .addComponent(jSeparator14)  
  
  .addGroup(painelCompressaoLayout.createSequentialGroup()  
  
  .addContainerGap()  
  
  .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
  .addGroup(painelCompressaoLayout.createSequentialGroup()  
  
  .addComponent(jLabel7)  
  
  .addGap(178, 178, 178))  
  
  .addGroup(painelCompressaoLayout.createSequentialGroup()  
  
  .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
  .addGroup(painelCompressaoLayout.createSequentialGroup()  
  
  .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
  
  .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

Alignment.LEADING)

```
.addGroup(painelCompressaoLayout.createSequentialGroup())
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING)
```

```
.addComponent(txtDadosC)
```

```
.addGroup(painelCompressaoLayout.createSequentialGroup())
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.TRAILING)
```

```
.addComponent(txtYwc,  
javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(javax.swing.GroupLayout.Alignment.LEADING,  
painelCompressaoLayout.createSequentialGroup())
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING)
```

```
.addComponent(txtKmodC)
```

```
.addComponent(txtNdC))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING, false)
```

```
.addComponent(campoYwc,  
javax.swing.GroupLayout.DEFAULT_SIZE, 70, Short.MAX_VALUE)
```

```
.addComponent(campoKmodC,  
javax.swing.GroupLayout.DEFAULT_SIZE, 70, Short.MAX_VALUE)
```

```
.addComponent(campoNdC)))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtkNC)))

        .addGap(81, 81, 81)

        .addComponent(jSeparator12,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(painelCompressaoLayout.createSequentialGroup())

        .addGap(80, 80, 80)

        .addComponent(rb0C)

        .addGap(8, 8, 8)

.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(painelCompressaoLayout.createSequentialGroup())

        .addGap(151, 151, 151)

.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.TRAILING)

.addGroup(painelCompressaoLayout.createSequentialGroup())

        .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```

        .addComponent(listaExtensao90,
javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGroup(painelCompressaoLayout.createSequentialGroup())

        .addComponent(txtfc90d)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

        .addComponent(bDimensionarC)

    .addGroup(painelCompressaoLayout.createSequentialGroup())

        .addComponent(campofc90d,
javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUnC2))))))

    .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

    .addGroup(painelCompressaoLayout.createSequentialGroup())

        .addGap(95, 95, 95)

        .addComponent(jLabel3)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(listaExtensaoOutro,
javax.swing.GroupLayout.PREFERRED_SIZE, 118,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelCompressaoLayout.createSequentialGroup())

        .addGap(119, 119, 119)

        .addComponent(txtfcad)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campofcad,
javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUnC3))))

.addGroup(painelCompressaoLayout.createSequentialGroup())

        .addGap(205, 205, 205)

        .addComponent(rb90C)

        .addGap(213, 213, 213)

        .addComponent(rbOutroC)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoOutroC,
javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtgrausC))))

```

```
.addGroup(painelCompressaoLayout.createSequentialGroup())  
  
        .addGap(18, 18, 18)  
  
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.TRAILING)  
  
.addGroup(painelCompressaoLayout.createSequentialGroup())  
  
        .addComponent(txtAreaMinC)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
        .addComponent(campoAreaMinC,  
javax.swing.GroupLayout.PREFERRED_SIZE, 96,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
        .addComponent(txtUnidadeAreaC))  
  
        .addComponent(txtSelecioneInclinacaoC)))  
  
.addGroup(painelCompressaoLayout.createSequentialGroup())  
  
        .addGap(44, 44, 44)  
  
        .addComponent(txtfc0d)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
  
        .addComponent(campofc0d,  
javax.swing.GroupLayout.PREFERRED_SIZE, 111,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```



```
        .addComponent(txtUnC1))))
    .addComponent(jLabel8)
    .addGroup(painelCompressaoLayout.createSequentialGroup()
        .addComponent(txtSecaoC)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(listaSecoesC,
            javax.swing.GroupLayout.PREFERRED_SIZE, 469,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 206, Short.MAX_VALUE))))
    .addContainerGap()
    .addGroup(painelCompressaoLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.TRAILING)
            .addGroup(painelCompressaoLayout.createSequentialGroup()
                .addComponent(txtEsbeltezC)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(campoEsbeltezC,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 113,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
                    painelCompressaoLayout.createSequentialGroup()
                        .addComponent(txtEstabilidadeC)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```

        .addComponent(campoEstabilidadeC,
javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGap(34, 34, 34)

        .addComponent(bVerificarC)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelCompressaoLayout.createSequentialGroup())

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(brCompressao)

        .addGap(46, 46, 46))

);

painelCompressaoLayout.setVerticalGroup(

painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)

        .addGroup(painelCompressaoLayout.createSequentialGroup())

        .addContainerGap()

        .addComponent(jLabel7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)

        .addGroup(painelCompressaoLayout.createSequentialGroup())

        .addComponent(jSeparator12,

```

```
javax.swing.GroupLayout.PREFERRED_SIZE, 307,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
```

```
    .addGroup(painelCompressaoLayout.createSequentialGroup())
```

```
        .addComponent(jSeparator13,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
```

```
    .addComponent(txtDadosC)
```

```
    .addComponent(txtSelecaoInclinacaoC))
```

```
.addGap(4, 4, 4)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
```

```
    .addGroup(painelCompressaoLayout.createSequentialGroup())
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
```

```
    .addComponent(txtNdC)
```

```
        .addComponent(campoNdC,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(txtkNC))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
```

```
        .addComponent(campoKmodC,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(txtKmodC))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
```

```
        .addComponent(txtYwc)
```

```
        .addComponent(campoYwc,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
        .addGroup(painelCompressaoLayout.createSequentialGroup())
```

```
        .addGap(14, 14, 14)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
```

```
        .addGroup(painelCompressaoLayout.createSequentialGroup())
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
```

```
        .addComponent(campoOutroC,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(rbIntroC)
```

```
.addComponent(txtgrausC))
```

```
.addComponent(rb90C))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.LEADING)
```

```
.addComponent(jLabel2,  
javax.swing.GroupLayout.Alignment.TRAILING)
```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.B  
ASELINE)
```

```
.addComponent(listaExtensao90,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(listaExtensaoIntro,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(jLabel3)))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.  
Alignment.BASELINE)
```

```
.addComponent(txtfc90d)
```

```
.addComponent(campoFc90d,  
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUnC2)

        .addComponent(txtfcad)

        .addComponent(campofcad,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUnC3)))

    .addGroup(painelCompressaoLayout.createSequentialGroup())

        .addComponent(rb0C)

        .addGap(59, 59, 59)

    .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)

        .addComponent(txtfc0d)

        .addComponent(campofc0d,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUnC1))))))

    .addGap(35, 35, 35)

    .addComponent(bDimensionarC)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)

```

```
.addComponent(txtAreaMinC)

        .addComponent(campoAreaMinC,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUnidadeAreaC))

        .addGap(25, 25, 25)))

        .addComponent(jSeparator14,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jLabel8)

        .addGroup(painelCompressaoLayout.createParallelGroup(javafx.swing.GroupLayout.
Alignment.LEADING)

        .addGroup(painelCompressaoLayout.createSequentialGroup())

        .addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(painelCompressaoLayout.createParallelGroup(javafx.swing.GroupLayout.
Alignment.BASELINE)

        .addComponent(listaSecoesC,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtSecaoC))

        .addGap(43, 43, 43)

        .addGroup(painelCompressaoLayout.createParallelGroup(javafx.swing.GroupLayout.
```

Alignment.BASELINE)

```
        .addComponent(txtEsbeltezC)

        .addComponent(campoEsbeltezC,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelCompressaoLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)

        .addComponent(txtEstabilidadeC)

        .addComponent(campoEstabilidadeC,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(16, 16, 16)

        .addComponent(brCompressao)

        .addContainerGap(26, Short.MAX_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelCompressaoLayout.createSequentialGroup())

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(bVerificarC)

        .addGap(79, 79, 79)))

);

painelAbas.addTab("Compressão", painelCompressao);
```



```
txtDadosF.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
txtDadosF.setText("Dados:");  
  
txtMdF.setText("Md = ");  
  
txtkNmF.setText("kN.cm");  
  
jLabel9.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N  
jLabel9.setText("Dimensionamento");  
  
jSeparator16.setOrientation(javax.swing.SwingConstants.VERTICAL);  
  
txtKmodF.setText("Kmod =");  
  
txtFcdF.setText("fc,d = ");  
  
txtFtdF.setText("ft,d = ");  
  
txtUniFcdF.setText("kN/cm2");  
  
txtUniFtdF.setText("kN/cm2");
```

```
txtBaseF.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtBaseF.setText("Selecione o valor da base da seção:");
```

```
listaBaseF.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
listaBaseF.setMaximumRowCount(11);
```

```
listaBaseF.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {  
"5 cm", "7.5 cm", "10 cm", "12.5 cm", "15 cm", "17.5 cm", "20 cm", "22.5 cm", "25  
cm", "27.5 cm", "30 cm", " " }));
```

```
listaBaseF.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        listaBaseFActionPerformed(evt);
```

```
    }
```

```
});
```

```
bDimensionarFlexao.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
bDimensionarFlexao.setText("Dimensionar");
```

```
bDimensionarFlexao.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        bDimensionarFlexaoActionPerformed(evt);
```

```
    }
```

```
});
```

```
txtAlturaMinF.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtAlturaMinF.setText("Altura mínima da seção:");
```

```
campoAlturaMinF.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtUniAlturaF.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtUniAlturaF.setText("cm");
```

```
jLabel10.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
jLabel10.setText("Condição de segurança:");
```

```
txtTravamento.setText("Instabilidade lateral");
```

```
txtAlturaF.setText("Selecione a altura que deseja utilizar:");
```

```
listaAlturaF.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {  
"Item 1", "Item 2", "Item 3", "Item 4" }));
```

```
listaAlturaF.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        listaAlturaFActionPerformed(evt);
```

```
    }
```

```
});
```

```
jLabel11.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

jLabel11.setText("Verificação");

bVerificaF.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

bVerificaF.setText("Verificar");

bVerificaF.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bVerificaFActionPerformed(evt);

    }

});

jSeparator18.setOrientation(javax.swing.SwingConstants.VERTICAL);

jLabel12.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel12.setText("Travamentos:");

bTravamento.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

bTravamento.setText("Dimensionar");

bTravamento.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bTravamentoActionPerformed(evt);

    }

});
```

```
});
```

```
bpFlexao.setText("Próximo");
```

```
bpFlexao.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        bpFlexaoActionPerformed(evt);
```

```
    }
```

```
});
```

```
    javax.swing.GroupLayout painelFlexaoLayout = new
    javax.swing.GroupLayout(painelFlexao);
```

```
    painelFlexao.setLayout(painelFlexaoLayout);
```

```
    painelFlexaoLayout.setHorizontalGroup(
```

```
    painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jSeparator15,
        javax.swing.GroupLayout.Alignment.TRAILING)
```

```
        .addComponent(jSeparator17)
```

```
        .addGroup(painelFlexaoLayout.createSequentialGroup()
```

```
            .addComponent(jSeparator16)
```

```
            .addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                .addComponent(jLabel11)
```

```
.addGroup(painelFlexaoLayout.createSequentialGroup())

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)

    .addGroup(painelFlexaoLayout.createSequentialGroup())

        .addComponent(jLabel9)

        .addGap(55, 55, 55))

    .addGroup(painelFlexaoLayout.createSequentialGroup())

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

    .addComponent(txtDadosF)

    .addGroup(painelFlexaoLayout.createSequentialGroup())

        .addComponent(txtMdF)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addComponent(campoMdF,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addComponent(txtkNmF))

    .addGroup(painelFlexaoLayout.createSequentialGroup())

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

    .addComponent(txtFcdF)

    .addGroup(painelFlexaoLayout.createSequentialGroup())
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
```

```
    .addComponent(txtKmodF)
```

```
    .addComponent(txtFtdF))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
```

```
    .addComponent(campoFtdF,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
```

```
    .addComponent(campoFcdF)
```

```
    .addComponent(campoKmodF,
javax.swing.GroupLayout.DEFAULT_SIZE, 70, Short.MAX_VALUE))))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
```

```
    .addComponent(txtUniFcdF)
```

```
    .addComponent(txtUniFtdF))))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(jSeparator16,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(8, 8, 8)))

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

        .addGroup(painelFlexaoLayout.createSequentialGroup())

            .addGap(67, 67, 67)

            .addComponent(txtAlturaMinF)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoAlturaMinF,
javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUniAlturaF))

        .addGroup(painelFlexaoLayout.createSequentialGroup())

            .addGap(38, 38, 38)

            .addComponent(txtBaseF)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(listaBaseF,
javax.swing.GroupLayout.PREFERRED_SIZE, 108,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addComponent(jLabel10)

        .addGroup(painelFlexaoLayout.createSequentialGroup())

            .addGap(138, 138, 138)
```



```
.addComponent(bDimensionarFlexao)))

.addGap(66, 66, 66)

.addComponent(jSeparator18,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

.addComponent(jLabel12)

.addComponent(bTravamento)

.addComponent(campoTravamento,
javax.swing.GroupLayout.PREFERRED_SIZE, 500,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.TRAILING)

.addGroup(painelFlexaoLayout.createSequentialGroup())

.addComponent(txtTravamento)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(campoInstabilidadeL,
javax.swing.GroupLayout.PREFERRED_SIZE, 244,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING,
painelFlexaoLayout.createSequentialGroup())

.addComponent(txtAlturaF)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```

        .addComponent(listaAlturaF,
javafx.swing.GroupLayout.PREFERRED_SIZE, 132,
javafx.swing.GroupLayout.PREFERRED_SIZE))))

        .addGap(114, 114, Short.MAX_VALUE))

        .addGroup(painelFlexaoLayout.createSequentialGroup())

        .addGap(59, 59, 59)

        .addComponent(bVerificaF)

        .addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(bpFlexao)

        .addGap(83, 83, 83))

);

painelFlexaoLayout.setVerticalGroup(

painelFlexaoLayout.createParallelGroup(javafx.swing.GroupLayout.Alignment.LEADING)

        .addGroup(painelFlexaoLayout.createSequentialGroup())

        .addContainerGap()

        .addComponent(jLabel9)

        .addGap(6, 6, 6)

        .addGroup(painelFlexaoLayout.createParallelGroup(javafx.swing.GroupLayout.Alignment.LEADING)

        .addGroup(painelFlexaoLayout.createSequentialGroup())

        .addComponent(jSeparator15,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,

```

```
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(painelFlexaoLayout.createSequentialGroup())
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
    .addGroup(painelFlexaoLayout.createSequentialGroup())
```

```
        .addComponent(txtDadosF)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(txtMdF)
```

```
        .addComponent(campoMdF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(txtkNmF))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(campoKmodF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(txtKmodF))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)

        .addComponent(txtFtdF)

        .addComponent(campoFtdF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUniFtdF))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)

        .addComponent(txtFcdF)

        .addComponent(campoFcdF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUniFcdF))

.addGap(59, 59, 59))

.addGroup(painelFlexaoLayout.createSequentialGroup())

        .addComponent(jLabel10)

        .addGap(34, 34, 34)

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
```

```
.addComponent(txtBaseF)

        .addComponent(listaBaseF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(bDimensionarFlexao)

        .addGap(41, 41, 41)))

.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)

        .addComponent(txtAlturaMinF)

        .addComponent(campoAlturaMinF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUniAlturaF)))

.addGroup(painelFlexaoLayout.createSequentialGroup()

        .addComponent(jLabel12)

        .addGap(28, 28, 28)

        .addComponent(campoTravamento,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(28, 28, 28)

        .addComponent(bTravamento))))

        .addComponent(jSeparator16,
javax.swing.GroupLayout.PREFERRED_SIZE, 278,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(jSeparator18,  
javafx.swing.GroupLayout.PREFERRED_SIZE, 278,  
javafx.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED,  
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
.addComponent(jSeparator17,  
javafx.swing.GroupLayout.PREFERRED_SIZE,  
javafx.swing.GroupLayout.DEFAULT_SIZE,  
javafx.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addComponent(jLabel11)
```

```
.addGap(18, 18, 18)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javafx.swing.GroupLayout.Align  
ment.BASELINE)
```

```
.addComponent(txtAlturaF)
```

```
.addComponent(listaAlturaF,  
javafx.swing.GroupLayout.PREFERRED_SIZE,  
javafx.swing.GroupLayout.DEFAULT_SIZE,  
javafx.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGap(42, 42, 42)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javafx.swing.GroupLayout.Align  
ment.BASELINE)
```

```
.addComponent(campoInstabilidadeL,  
javafx.swing.GroupLayout.PREFERRED_SIZE,  
javafx.swing.GroupLayout.DEFAULT_SIZE,  
javafx.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(txtTravamento))
```

```
.addGap(36, 36, 36)
```

```
.addGroup(painelFlexaoLayout.createParallelGroup(javax.swing.GroupLayout.Align  
ment.LEADING)
```

```
    .addComponent(bVerificaF)
```

```
    .addComponent(bpFlexao))
```

```
    .addGap(69, 69, 69))
```

```
);
```

```
painelAbas.addTab("Flexão", painelFlexao);
```

```
txtVdCl.setText("Vd =");
```

```
txtDadosCl.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtDadosCl.setText("Dados:");
```

```
txtUniVdCl.setText("kN");
```

```
txtBaseCl.setText("base =");
```

```
txtUniBase.setText("cm");
```

```
txtAlturaCl.setText("Altura =");
```

```
txtUniAlt.setText("cm");
```

```
txtfv0Cl.setText("fv0 =");
```

```
txtUniFv0.setText("kN/cm2");
```

```
txtKmodCl.setText("Kmod =");
```

```
jLabel13.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
```

```
jLabel13.setText("Redimensionamento");
```

```
jSeparator20.setOrientation(javax.swing.SwingConstants.VERTICAL);
```

```
jLabel14.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
```

```
jLabel14.setText("Verificação");
```

```
txtCondCl.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
txtCondCl.setText("Condição de segurança:");
```

```
campoCondCl.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
bVerificaCl.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```



```
bVerificaCI.setText("Verificar");

bVerificaCI.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bVerificaCIActionPerformed(evt);

    }

});

txtRedimCI.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

txtRedimCI.setText("Altura mínima");

campoRedimCI.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

txtUniRedimCI.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

txtUniRedimCI.setText("cm");

bRedimCI.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

bRedimCI.setText("Dimensionar");

bRedimCI.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        bRedimCIActionPerformed(evt);

    }

});
```

```

brCisalhamento.setText("Ver resultados");

brCisalhamento.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        brCisalhamentoActionPerformed(evt);

    }

});

    javax.swing.GroupLayout painelCisalhamentoLayout = new
javax.swing.GroupLayout(painelCisalhamento);

    painelCisalhamento.setLayout(painelCisalhamentoLayout);

    painelCisalhamentoLayout.setHorizontalGroup(

painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

        .addGroup(painelCisalhamentoLayout.createSequentialGroup()

            .addComponent(jSeparator21)

            .addGap())

        .addGroup(painelCisalhamentoLayout.createSequentialGroup()

            .addGap())

        .addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)

            .addGroup(painelCisalhamentoLayout.createSequentialGroup()

```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.LEADING)
```

```
    .addGroup(painelCisalhamentoLayout.createSequentialGroup())
```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.LEADING)
```

```
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
painelCisalhamentoLayout.createSequentialGroup())
```

```
        .addComponent(txtKmodCI)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(campoKmodCI,  
javax.swing.GroupLayout.PREFERRED_SIZE, 70,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addGap(22, 22, 22))
```

```
.addGroup(painelCisalhamentoLayout.createSequentialGroup())
```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.LEADING)
```

```
    .addComponent(txtDadosCI)
```

```
.addGroup(painelCisalhamentoLayout.createSequentialGroup())
```

```
    .addComponent(txtAlturaCI)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(campoAlturaCI,  
javax.swing.GroupLayout.PREFERRED_SIZE, 70,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtUniAlt)))

        .addGap(0, 0, Short.MAX_VALUE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelCisalhamentoLayout.createSequentialGroup())

.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.TRAILING)

.addGroup(painelCisalhamentoLayout.createSequentialGroup())

        .addComponent(txtVdCI)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(campoVdCI,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(painelCisalhamentoLayout.createSequentialGroup())

        .addComponent(txtBaseCI)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(campoBaseCI,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
```

```
.addComponent(txtUniBase)

.addComponent(txtUniVdCI)))

.addGap(84, 84, 84))

.addGroup(painelCisalhamentoLayout.createSequentialGroup()

.addComponent(txtfv0CI)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(campoFv0CI,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(txtUniFv0)

.addGap(59, 59, 59)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jSeparator20,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)

.addGroup(painelCisalhamentoLayout.createSequentialGroup()

.addGap(20, 20, 20)

.addComponent(jLabel14))

.addGroup(painelCisalhamentoLayout.createSequentialGroup()
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jSeparator19,
javax.swing.GroupLayout.PREFERRED_SIZE, 1126,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(painelCisalhamentoLayout.createSequentialGroup())

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(txtCondCI)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoCondCI,
javax.swing.GroupLayout.PREFERRED_SIZE, 375,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(bVerificaCI)))

        .addContainerGap(13, Short.MAX_VALUE))

        .addGroup(painelCisalhamentoLayout.createSequentialGroup())

        .addComponent(jLabel13)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addGroup(painelCisalhamentoLayout.createSequentialGroup())

        .addComponent(txtRedimCI)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(campoRedimCI,
javax.swing.GroupLayout.PREFERRED_SIZE, 219,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtUniRedimCI)
        .addGap(0, 0, Short.MAX_VALUE))))
.addGroup(painelCisalhamentoLayout.createSequentialGroup()
        .addGap(100, 100, 100)
        .addComponent(bRedimCI)
        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
painelCisalhamentoLayout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(brCisalhamento)
        .addGap(34, 34, 34))
);

painelCisalhamentoLayout.setVerticalGroup(

painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

        .addGroup(painelCisalhamentoLayout.createSequentialGroup()

.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)

        .addGroup(painelCisalhamentoLayout.createSequentialGroup()

        .addGap(8, 8, 8)

        .addComponent(jLabel14)

```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
    .addComponent(jSeparator19,  
        javax.swing.GroupLayout.PREFERRED_SIZE,  
        javax.swing.GroupLayout.DEFAULT_SIZE,  
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.LEADING)
```

```
    .addGroup(painelCisalhamentoLayout.createSequentialGroup())
```

```
        .addGap(15, 15, 15)
```

```
        .addComponent(txtDadosCI)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.BASELINE)
```

```
    .addComponent(txtVdCI)
```

```
        .addComponent(campoVdCI,  
            javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE,  
            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(txtUniVdCI))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.BASELINE)
```

```
    .addComponent(txtBaseCI)
```

```
        .addComponent(campoBaseCI,  
            javax.swing.GroupLayout.PREFERRED_SIZE,  
            javax.swing.GroupLayout.DEFAULT_SIZE,
```



```
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUniBase))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)

        .addComponent(txtAlturaCI)

        .addComponent(campoAlturaCI,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUniAlt))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)

        .addComponent(txtfv0CI)

        .addComponent(campoFv0CI,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtUniFv0))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)

        .addComponent(txtKmodCI)

        .addComponent(campoKmodCI,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
    .addGroup(painelCisalhamentoLayout.createSequentialGroup())
```

```
        .addGap(36, 36, 36)
```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
```

```
    .addComponent(txtCondCI)
```

```
        .addComponent(campoCondCI,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(bVerificaCI))))))
```

```
        .addComponent(jSeparator20,
javax.swing.GroupLayout.PREFERRED_SIZE, 326,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
    .addComponent(jSeparator21,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addGap(18, 18, 18)
```

```
    .addComponent(jLabel13)
```

```
        .addGap(29, 29, 29)
```

```
.addGroup(painelCisalhamentoLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
```

```
    .addComponent(txtRedimCI)
```

```
        .addComponent(campoRedimCI,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
        .addComponent(txtUniRedimCI)  
  
        .addGap(36, 36, 36)  
  
        .addComponent(bRedimCI)  
  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 36,  
Short.MAX_VALUE)  
  
        .addComponent(brCisalhamento)  
  
        .addGap(24, 24, 24))  
  
);  
  
painelAbas.addTab("Cisalhamento", painelCisalhamento);  
  
areaTxtResultados.setColumns(20);  
areaTxtResultados.setRows(5);  
jScrollPane1.setViewportView(areaTxtResultados);  
  
jLabel15.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel15.setText("Relatório final de cálculos");  
  
javax.swing.GroupLayout painelResultadosLayout = new  
javax.swing.GroupLayout(painelResultados);  
  
painelResultados.setLayout(painelResultadosLayout);
```

```
painelResultadosLayout.setHorizontalGroup(
```

```
painelResultadosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(painelResultadosLayout.createSequentialGroup())
```

```
        .addGap(27, 27, 27)
```

```
    .addGroup(painelResultadosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jLabel15)
```

```
        .addComponent(jScrollPane1,  
            javax.swing.GroupLayout.PREFERRED_SIZE, 1010,  
            javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
        .addContainerGap(351, Short.MAX_VALUE))
```

```
);
```

```
painelResultadosLayout.setVerticalGroup(
```

```
painelResultadosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(painelResultadosLayout.createSequentialGroup())
```

```
        .addGap(23, 23, 23)
```

```
        .addComponent(jLabel15)
```

```
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
        .addComponent(jScrollPane1,  
            javax.swing.GroupLayout.PREFERRED_SIZE, 496,  
            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addContainerGap(43, Short.MAX_VALUE))
```

```
);
```

```
painelAbas.addTab("Resultados", painelResultados);
```

```
    javax.swing.GroupLayout layout = new  
    javax.swing.GroupLayout(getContentPane());
```

```
    getContentPane().setLayout(layout);
```

```
    layout.setHorizontalGroup(  
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
                layout.createSequentialGroup()
```

```
                    .addGap()
```

```
                    .addComponent(painelAbas)
```

```
                    .addGap())
```

```
            .addGap()
```

```
);
```

```
    layout.setVerticalGroup(  
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(layout.createSequentialGroup()
```

```
                .addComponent(painelAbas,  
                    javax.swing.GroupLayout.PREFERRED_SIZE, 627,  
                    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                    .addGap(26, 26, 26))
```

```
            .addGap(26, 26, 26))
```

```
);
```

```
pack();

setLocationRelativeTo(null);

} // </editor-fold>

private void loadListaT(String filename){

}

private void bVerificarCActionPerformed(java.awt.event.ActionEvent evt) {

    lgBloqueioComp = false;

    // setar base e altura conforme item selecionado.

    if(listaSecoesC.getSelectedItemAt().equals("Seção: 5.0cm x 10.0cm. Com área =
50.0cm². ")) {

        base = 5;

        altura = 10;

        area = (base * altura);

    } else if(listaSecoesC.getSelectedItemAt().equals("Seção: 10.0cm x 5.0cm. Com
área = 50.0cm². ")) {

        base = 10;
```

```
altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 5.0cm x 15.0cm. Com
área = 75.0cm².)){

base = 5;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 15.0cm x 5.0cm. Com
área = 75.0cm².)){

base = 15;

altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 5.0cm x 20.0cm. Com
área = 100.0cm².)){

base = 5;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 10.0cm x 10.0cm. Com
área = 100.0cm².)){

base = 10;

altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 20.0cm x 5.0cm. Com
área = 100.0cm².)){

base = 20;
```

```
altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 5.0cm x 25.0cm. Com
área = 125.0cm².)){

base = 5;

altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 25.0cm x 5.0cm. Com
área = 125.0cm².)){

base = 25;

altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 5.0cm x 30.0cm. Com
área = 150.0cm².)){

base = 5;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 10.0cm x 15.0cm. Com
área = 150.0cm².)){

base = 10;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 15.0cm x 10.0cm. Com
área = 150.0cm².)){

base = 15;
```



```
altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 30.0cm x 5.0cm. Com
área = 150.0cm².)){

base = 30;

altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 10.0cm x 20.0cm. Com
área = 200.0cm².)){

base = 10;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 20.0cm x 10.0cm. Com
área = 200.0cm².)){

base = 20;

altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 15.0cm x 15.0cm. Com
área = 225.0cm².)){

base = 15;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 10.0cm x 25.0cm. Com
área = 250.0cm².)){

base = 10;
```

```
altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 25.0cm x 10.0cm. Com
área = 250.0cm².)){

base = 25;

altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 10.0cm x 30.0cm. Com
área = 300.0cm².)){

base = 10;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 15.0cm x 20.0cm. Com
área = 300.0cm².)){

base = 15;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 20.0cm x 15.0cm. Com
área = 300.0cm².)){

base = 20;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 30.0cm x 10.0cm. Com
área = 300.0cm².)){

base = 30;
```

```
altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 15.0cm x 25.0cm. Com
área = 375.0cm².)){

base = 15;

altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 25.0cm x 15.0cm. Com
área = 375.0cm².)){

base = 25;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 20.0cm x 20.0cm. Com
área = 400.0cm².)){

base = 20;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 15.0cm x 30.0cm. Com
área = 450.0cm².)){

base = 15;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 30.0cm x 15.0cm. Com
área = 450.0cm².)){

base = 30;
```

```
altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 20.0cm x 25.0cm. Com
área = 500.0cm².)){

base = 20;

altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 25.0cm x 20.0cm. Com
área = 500.0cm².)){

base = 25;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 20.0cm x 30.0cm. Com
área = 600.0cm².)){

base = 20;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 30.0cm x 20.0cm. Com
área = 600.0cm².)){

base = 30;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 25.0cm x 25.0cm. Com
área = 625.0cm².)){

base = 25;
```

```
altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 25.0cm x 30.0cm. Com
área = 750.0cm².)){

base = 25;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 30.0cm x 25.0cm. Com
área = 750.0cm².)){

base = 30;

altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 30.0cm x 30.0cm. Com
área = 900.0cm².)){

base = 30;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 12.5cm x 5.0cm. Com
área = 62.5cm².)){

base = 12.5;

altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 7.5cm x 10.0cm. Com
área = 75.0cm².)){

base = 7.5;
```

```
altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 17.5cm x 5.0cm. Com
área = 87.5cm².)){

base = 17.5;

altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 7.5cm x 15.0cm. Com
área = 112.5cm².)){

base = 7.5;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 22.5cm x 5.0cm. Com
área = 112.5cm².)){

base = 22.5;

altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 12.5cm x 10.0cm. Com
área = 125.0cm².)){

base = 12.5;

altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 27.5cm x 5.0cm. Com
área = 137.5cm².)){

base = 27.5;
```

```
altura = 5;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 7.5cm x 20.0cm. Com
área = 150.0cm².)){

base = 7.5;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 17.5cm x 10.0cm. Com
área = 175.0cm².)){

base = 17.5;

altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 7.5cm x 25.0cm. Com
área = 187.5cm².)){

base = 7.5;

altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 12.5cm x 15.0cm. Com
área = 187.5cm².)){

base = 12.5;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedltem().equals("Seção: 7.5cm x 30.0cm. Com
área = 225.0cm².)){

base = 7.5;
```

```
altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 22.5cm x 10.0cm. Com
área = 225.0cm².)){

base = 22.5;

altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 12.5cm x 20.0cm. Com
área = 250.0cm².)){

base = 12.5;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 17.5cm x 15.0cm. Com
área = 262.5cm².)){

base = 17.5;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 27.5cm x 10.0cm. Com
área = 275.0cm².)){

base = 27.5;

altura = 10;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 12.5cm x 25.0cm. Com
área = 312.5cm².)){

base = 12.5;
```



```
altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 22.5cm x 15.0cm. Com
área = 337.5cm².)){

base = 22.5;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 17.5cm x 20.0cm. Com
área = 350.0cm².)){

base = 17.5;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 12.5cm x 30.0cm. Com
área = 375.0cm².)){

base = 12.5;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 27.5cm x 15.0cm. Com
área = 412.5cm².)){

base = 27.5;

altura = 15;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 17.5cm x 25.0cm. Com
área = 437.5cm².)){

base = 17.5;
```

```
altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 22.5cm x 20.0cm. Com
área = 450.0cm².)){

base = 22.5;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 17.5cm x 30.0cm. Com
área = 525.0cm².)){

base = 17.5;

altura = 30;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 27.5cm x 20.0cm. Com
área = 550.0cm².)){

base = 27.5;

altura = 20;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 22.5cm x 25.0cm. Com
área = 562.5cm².)){

base = 22.5;

altura = 25;

area = (base * altura);

}else if(listaSecoesC.getSelectedItemAt().equals("Seção: 22.5cm x 30.0cm. Com
área = 675.0cm².)){

base = 22.5;
```

```
        altura = 30;

        area = (base * altura);

        }else if(listaSecoesC.getSelectedItemId().equals("Seção: 27.5cm x 25.0cm. Com
        área = 687.5cm².)){

            base = 27.5;

            altura = 25;

            area = (base * altura);

        }else if(listaSecoesC.getSelectedItemId().equals("Seção: 27.5cm x 30.0cm. Com
        área = 825.0cm².)){

            base = 27.5;

            altura = 30;

            area = (base * altura);

        }

        double lx = (base*Math.pow(altura, 3))/12;

        double ly = (altura*Math.pow(base, 3))/12;

        if(lx > ly){

            l = ly;

            double aux = altura;

            altura = base;

            base = aux;
```

```
}else{  
  
    I = lx;  
  
}  
  
// Verificação da esbeltez  
  
String sL0 = campoComprimentoV.getText().replaceAll( ",", "." );  
  
double l0 = Double.parseDouble(sL0);  
  
  
double r = Math.sqrt(I/area);  
  
esb = l0/r;  
  
double lamb = 140;  
  
if (esb <= lamb){  
  
    campoEsbeltezC.setText("OK!");  
  
}else{  
  
    campoEsbeltezC.setText("Redefinir seção.");  
  
}  
  
  
//Verificação da estabilidade  
  
if(esb <= 40){
```

```
campoEstabilidadeC.setText("OK!");
```

```
}else if(esb > 40 && esb <= 80){
```

```
    ei = altura/30;
```

```
    ea = l0/300;
```

```
    e1 = ei + ea;
```

```
    double Ec0ef = kmod * Ec0m;
```

```
    Fe = ((Math.pow(Math.PI, 2)*Ec0ef*I)/(Math.pow(l0, 2)));
```

```
    ed = (e1*(Fe/(Fe-Nsd)));
```

```
    Mdt = Nsd*ed;
```

```
    if(base < altura){
```

```
        y = base/2;
```

```
    }else{
```

```
        y = altura/2;
```

```
    }
```

```
    Tmd = (Mdt * y)/l;
```

```
    Tnd = Nsd/area;
```

```
    if(rb0C.isSelected()){
```

```
        String sfcd = campofc0d.getText().replaceAll(" , ", ".");
```

```
double fcd = Double.parseDouble(sfcd);

verifica = (Tnd/fcd)+(Tmd/fcd);

if(verifica <= 1){

    campoEstabilidadeC.setText("OK!");

}else{

    campoEstabilidadeC.setText("Redefinir seção.");

}

}else if(rb90C.isSelected()){

String sfcd = campofc90d.getText().replaceAll(" ", ".");

double fcd = Double.parseDouble(sfcd);

verifica = (Tnd/fcd)+(Tmd/fcd);

if(verifica <= 1){

    campoEstabilidadeC.setText("OK!");

}else{

    campoEstabilidadeC.setText("Redefinir seção.");

}
```



```
int carregamento = 0;

int umidade = 0;

if(kmod1 == 0.6 || kmod1 == 0.7 || kmod1 == 0.3 || kmod1 == 0.45){
    carregamento = 1;
}
else if(kmod1 == 0.8 || kmod1 == 0.65){
    carregamento = 2;
}
else if(kmod1 == 0.9 || kmod1 == 1.1){
    carregamento = 3;
}

if(kmod2 == 1){
    umidade = 1;
}
else if(kmod2 == 0.8 || kmod2 == 0.9){
    umidade = 2;
}

if(carregamento == 1 && umidade == 1){
    flu = 0.8;
}
else if(carregamento == 1 && umidade == 2){
    flu = 2;
}
```



```

}else if(carregamento == 2 && umidade == 1){
    flu = 0.3;
}else if(carregamento == 2 && umidade == 2){
    flu = 1;
}else if(carregamento == 3 && umidade == 1){
    flu = 0.1;
}else if(carregamento == 3 && umidade == 2){
    flu = 0.5;
}

exp = Math.E;

double Ec0ef = kmod * Ec0m;

Fe = ((Math.pow(Math.PI, 2)*Ec0ef*I)/(I0*I0));

double kl = (Ng) + ((psi1V + psi2V) * (Nq));

k = (flu * kl)/(Fe - kl);

if(altura > base){
    ei = altura/30;
}else{
    ei = base/30;
}

ea = I0/300;

ec = (ei+ea)*((Math.pow(exp, k))-1);

```

```
e1 = ei + ea + ec;

ed = (e1*(Fe/(Fe-Nsd)));

Mdt = Nsd*ed;

double y = altura/2;

Tmd = (Mdt * y)/l;

if(Tmd < 0){

    Tmd = Tmd * (-1);

}

Tnd = Nsd/area;

if(rb0C.isSelected()){

    String sfcd = campofc0d.getText().replaceAll(" ", ".");

    double fcd = Double.parseDouble(sfcd);

    verifica = (Tnd/fcd)+(Tmd/fcd);

    if(verifica <= 1){

        campoEstabilidadeC.setText("OK!");

    }else{

        campoEstabilidadeC.setText("Redefinir seção.");

    }

}
```



```
        campoEstabilidadeC.setText("Redefinir seção.");
    }

}

}else{

        campoEstabilidadeC.setText("Redefinir seção.");

    }

    if(campoEsbeltezC.getText().equals("OK!") &&
campoEstabilidadeC.getText().equals("OK!")){

        brCompressao.setEnabled(true);

    }else{

        brCompressao.setEnabled(false);

    }

}

private void rbOutroCActionPerformed(java.awt.event.ActionEvent evt) {

    if(rbOutroC.isSelected()){

        listaExtensaoOutro.setEnabled(true);

        listaExtensao90.setEnabled(false);

        campoOutroC.setEnabled(true);

    }

}
```

```
bDimensionarC.setEnabled(true);

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

campoAreaMinC.setText("");

campofc0d.setText("");

campofc90d.setText("");

listaSecoesC.setEnabled(false);

listaSecoesC.removeAllItems();

bVerificarC.setEnabled(false);

brCompressao.setEnabled(false);

painelAbas.setEnabledAt(8, false);

}

}

private void rb90CActionPerformed(java.awt.event.ActionEvent evt) {

    if(rb90C.isSelected()){

        listaExtensao90.setEnabled(true);

        listaExtensaoOutro.setEnabled(false);

        campoOutroC.setEnabled(false);

        bDimensionarC.setEnabled(true);

        campoEsbeltezC.setText("");

        campoEstabilidadeC.setText("");
```

```
campoAreaMinC.setText("");

campofc0d.setText("");

campofcad.setText("");

campoOutroC.setText("");

listaSecoesC.setEnabled(false);

listaSecoesC.removeAllItems();

bVerificarC.setEnabled(false);

brCompressao.setEnabled(false);

painelAbas.setEnabledAt(8, false);

}

}

private void bDimensionarCActionPerformed(java.awt.event.ActionEvent evt) {

    if(rb0C.isSelected()){

        anguloT = 0;

        campoEsbeltezC.setText("");

        campoEstabilidadeC.setText("");

        double fc0d = kmod * (0.7*fc0k/fwc);
```

```
DecimalFormat df = new DecimalFormat("0.####");

String sfc0d = df.format(fc0d);

campoFc0d.setText(sfc0d);

String sndC = campoNdC.getText().replaceAll( ",", "." );

double ndC = Double.parseDouble(sndC);

areaC = (ndC/fc0d);

if(areaC < 50){
    areaC = 50;
}

String sareaC = df.format(areaC);

campoAreaMinC.setText(sareaC);

String sAreaMin = campoAreaMinC.getText().replaceAll( ",", "." );

double areaMin = Double.parseDouble(sAreaMin);

// Achar seções

listaSecoesC.removeAllItems();
```

```
ArrayList<Medidas> maioresQueMin = new ArrayList<>();

if(areaMin > 900){

    listaSecoesC.addItem("Área mínima acima das seções utilizadas, redefina
o carregamento.");

    bVerificarC.setEnabled(false);

}else{

    //Percorrer ArrayList e grava valores acima da área minima

    for(Medidas total : varMedidas) {

        if(total.getArea() >= areaMin){

            maioresQueMin.add(total);

        }

    }

    // Percorre novo vetor e adiciona as opções

    for(Medidas maiores : maioresQueMin){

        listaSecoesC.addItem("Seção: " + maiores.getBase() + "cm x " +
maiores.getAltura() + "cm. Com área = " + maiores.getArea() + "cm².");

    }

}
```



```
        bVerificarC.setEnabled(true);

    }

}

}else if(rb90C.isSelected()){

    anguloT = 90;

    campoEsbeltezC.setText("");

    campoEstabilidadeC.setText("");

    bVerificarC.setEnabled(true);

    if(listaExtensao90.getSelectedItem().equals("1 cm")){

        an = 2;

    }else if(listaExtensao90.getSelectedItem().equals("2 cm")){

        an = 1.7;

    }else if(listaExtensao90.getSelectedItem().equals("3 cm")){

        an = 1.55;

    }else if(listaExtensao90.getSelectedItem().equals("4 cm")){

        an = 1.4;

    }else if(listaExtensao90.getSelectedItem().equals("5 cm")){

        an = 1.3;

    }else if(listaExtensao90.getSelectedItem().equals("7,5 cm")){
```

```
        an = 1.15;

    }else if(listaExtensao90.getSelectedItem().equals("10 cm")){

        an = 1.1;

    }else if(listaExtensao90.getSelectedItem().equals("15 cm")){

        an = 1;

    }

    double fc90d = kmod * (0.7*(0.25*fc0k*an)/fwc);

    DecimalFormat df = new DecimalFormat("0.####");

    String sfc90d = df.format(fc90d);

    campofc90d.setText(sfc90d);

    String sndC = campoNdC.getText().replaceAll( ",", "." );

    double ndC = Double.parseDouble(sndC);

    areaC = (ndC/fc90d);

    if(areaC < 50){

        areaC = 50;

    }

    String sareaC = df.format(areaC);
```

```
campoAreaMinC.setText(sareaC);

String sAreaMin = campoAreaMinC.getText().replaceAll( ",", "." );

double areaMin = Double.parseDouble(sAreaMin);

// Achar seções

listaSecoesC.removeAllItems();

ArrayList<Medidas> maioresQueMin = new ArrayList<>();

if(areaMin > 900){

    listaSecoesC.addItem("Área mínima acima das seções utilizadas, redefine
o carregamento.");

    bVerificarC.setEnabled(false);

}else{

    //Percorrer ArrayList e grava valores acima da área minima

    for(Medidas total : varMedidas) {

        if(total.getArea() >= areaMin){

            maioresQueMin.add(total);

        }

    }

}
```

```
}

// Percorre novo vetor e adiciona as opções

for(Medidas maiores : maioresQueMin){

    listaSecoesC.addItem("Seção: " + maiores.getBase() + "cm x " +
maiores.getAltura() + "cm. Com área = " + maiores.getArea() + "cm².");

}

bVerificarC.setEnabled(true);

}

}else if(rbOutroC.isSelected()){

    campoEsbeltezC.setText("");
    campoEstabilidadeC.setText("");
    bVerificarC.setEnabled(true);

    if(listaExtensaoOutro.getSelectedItem().equals("1 cm")){

        an = 2;

    }else if(listaExtensaoOutro.getSelectedItem().equals("2 cm")){

        an = 1.7;

    }else if(listaExtensaoOutro.getSelectedItem().equals("3 cm")){
```

```

        an = 1.55;

    }else if(listaExtensaoOutro.getSelectedItem().equals("4 cm")){

        an = 1.4;

    }else if(listaExtensaoOutro.getSelectedItem().equals("5 cm")){

        an = 1.3;

    }else if(listaExtensaoOutro.getSelectedItem().equals("7,5 cm")){

        an = 1.15;

    }else if(listaExtensaoOutro.getSelectedItem().equals("10 cm")){

        an = 1.1;

    }else if(listaExtensaoOutro.getSelectedItem().equals("15 cm")){

        an = 1;

    }

    if(campoOutroC.getText().equals("")){

        JOptionPane.showMessageDialog(rootPane, "Determine o ângulo de
inclinação.");

        bVerificarC.setEnabled(false);

    }else{

        String sangulo = campoOutroC.getText().replaceAll(" ", ".");

        double a = Double.parseDouble(sangulo);

        anguloT = a;

        double angulo = Math.toRadians(a);

        double fc0d = kmod * (0.7*fc0k/fwc);

```

```
double fc90d = kmod * (0.7*(0.25*fc0k*an)/fwc);
```

```
double fcad = (fc0d * fc90d)/((fc0d * (Math.sin(angulo) *
Math.sin(angulo)))+(fc90d * (Math.cos(angulo) * Math.cos(angulo))));
```

```
DecimalFormat df = new DecimalFormat("0.####");
```

```
String sfcad = df.format(fcad);
```

```
campoofcad.setText(sfcad);
```

```
String sndC = campoNdC.getText().replaceAll( ", " , "." );
```

```
double ndC = Double.parseDouble(sndC);
```

```
areaC = (ndC/fcad);
```

```
if(areaC < 50){
```

```
    areaC = 50;
```

```
}
```

```
String sareaC = df.format(areaC);
```

```
campoAreaMinC.setText(sareaC);
```

```
}
```

```
String sAreaMin = campoAreaMinC.getText().replaceAll( ", " , "." );
```

```
double areaMin = Double.parseDouble(sAreaMin);
```

```
// Achar seções

listaSecoesC.removeAllItems();

ArrayList<Medidas> maioresQueMin = new ArrayList<>();

if(areaMin > 900){

    listaSecoesC.addItem("Área mínima acima das seções utilizadas, redefina
o carregamento.");

    bVerificarC.setEnabled(false);

}else{

    //Percorrer ArrayList e grava valores acima da área minima

    for(Medidas total : varMedidas) {

        if(total.getArea() >= areaMin){

            maioresQueMin.add(total);

        }

    }

    // Percorre novo vetor e adiciona as opções

    for(Medidas maiores : maioresQueMin){
```

```
        listaSecoesC.addItem("Seção: " + maiores.getBase() + "cm x " +  
maiores.getAltura() + "cm. Com área = " + maiores.getArea() + "cm².");
```

```
    }
```

```
        bVerificarC.setEnabled(true);
```

```
    }
```

```
}
```

```
listaSecoesC.setEnabled(true);
```

```
}
```

```
private void rb0CActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if(rb0C.isSelected()){
```

```
        campoOutroC.setEnabled(false);
```

```
        listaExtensao90.setEnabled(false);
```

```
        listaExtensaoOutro.setEnabled(false);
```

```
        bDimensionarC.setEnabled(true);
```

```
        campoEsbeltezC.setText("");
```

```
        campoEstabilidadeC.setText("");
```

```
        campoAreaMinC.setText("");
```

```
        campofc90d.setText("");
```

```
        campofcad.setText("");
```



```
        campofc0d.setText("");

        campoOutroC.setText("");

        listaSecoesC.setEnabled(false);

        listaSecoesC.removeAllItems();

        bVerificarC.setEnabled(false);

        brCompressao.setEnabled(false);

        painelAbas.setEnabledAt(8, false);

    }
}

private void bpKmodActionPerformed(java.awt.event.ActionEvent evt) {

    if (lgBloqueioKmod){

        JOptionPane.showMessageDialog(null, "Recalcule o Kmod");

        return;

    }

    DecimalFormat df = new DecimalFormat("0.####");

    if(rbHorizontal.isSelected()){

        painelAbas.setEnabledAt(4, false);

        painelAbas.setEnabledAt(5, false);

        painelAbas.setEnabledAt(6, true);
```

```
painelAbas.setEnabledAt(7, false);

painelAbas.setSelectedIndex(6);

//Calcula fcd

fcd = kmod * (0.7*fc0k/fwc);

String sfcd = df.format(fcd);

campoFcdF.setText(sfcd);

//Calcula ftd

ftd = kmod * (0.7*fwkp/fwt);

String sftd = df.format(ftd);

campoFtdF.setText(sftd);

}else if(rbVertical.isSelected()){

    if(Nsd > 0){

        painelAbas.setEnabledAt(4, true);

        painelAbas.setEnabledAt(5, false);

        painelAbas.setEnabledAt(6, false);

        painelAbas.setEnabledAt(7, false);
```

```
        painelAbas.setSelectedIndex(4);

        bVerificarT.setEnabled(false);

        brTracao.setEnabled(false);
    }else if(Nsd < 0){

        Nsd = Nsd*-1;

        painelAbas.setEnabledAt(4, false);

        painelAbas.setEnabledAt(5, true);

        painelAbas.setEnabledAt(6, false);

        painelAbas.setEnabledAt(7, false);

        painelAbas.setSelectedIndex(5);

        bVerificarC.setEnabled(false);

        brCompressao.setEnabled(false);
    }

}

}

private void bCalcularKmodActionPerformed(java.awt.event.ActionEvent evt) {

    lgBloqueioKmod = false;

    String skmod;

    kmod1 = 0;
```

```
kmod2 = 0;

kmod3 = 0;

int v1 = 0;

int v2 = 0;

int v3 = 0;

try{

    String skmod1;

    skmod1 = campoKmod1.getText().replaceAll( ",", "." );

    kmod1 = Double.parseDouble(skmod1);

}catch(Exception e){

    JOptionPane.showMessageDialog(rootPane, ("Defina um valor para
Kmod1"));

}

try{

    String skmod2;

    skmod2 = campoKmod2.getText().replaceAll( ",", "." );

    kmod2 = Double.parseDouble(skmod2);

}catch(Exception e){

    JOptionPane.showMessageDialog(rootPane, ("Defina um valor para
Kmod2"));

}
```

```
try{

    String skmod3;

    skmod3 = campoKmod3.getText().replaceAll( ",", "." );

    kmod3 = Double.parseDouble(skmod3);

}catch(Exception e){

    JOptionPane.showMessageDialog(rootPane, ("Defina um valor para
Kmod3"));

}

if(kmod1 != 0){

    if(kmod1 != 0.6 && kmod1 != 0.7 && kmod1 != 0.8 && kmod1 != 0.9 &&
kmod1 != 1.1 && kmod1 != 0.3 && kmod1 != 0.45 && kmod1 != 0.65){

        JOptionPane.showMessageDialog(rootPane, ("Valor definido para Kmod1
inválido, redefina o valor"));

    }else{

        v1 = 1;

    }

}

if(kmod2 != 0){

    if(kmod2 != 1 && kmod2 != 0.8 && kmod2 != 0.9){

        JOptionPane.showMessageDialog(rootPane, ("Valor definido para Kmod2
inválido, redefina o valor"));

    }

}
```

```
    }else{  
        v2 = 1;  
    }  
}  
  
if(kmod3 != 0){  
    if(kmod3 != 1 && kmod3 != 0.8){  
        JOptionPane.showMessageDialog(rootPane, ("Valor definido para Kmod3  
inválido, redefine el valor"));  
    }else{  
        v3 = 1;  
    }  
}  
  
if(v1 == 1 && v2 == 1 && v3 == 1){  
    kmod = kmod1 * kmod2 * kmod3;  
  
    DecimalFormat df = new DecimalFormat("0.##");  
    skmod = df.format(kmod);  
    campoKmod.setText(skmod);  
    bpKmod.setEnabled(true);  
    campoKmodT.setText(skmod);  
    campoKmodC.setText(skmod);  
    campoKmodF.setText(skmod);
```

```
    }  
}  
  
private void bpClasseResistenciaActionPerformed(java.awt.event.ActionEvent evt)  
{  
  
    if(!rbConifera.isSelected() && !rbDicotiledonea.isSelected()){  
  
        JOptionPane.showMessageDialog(rootPane, "Selecione um grupo de  
madeira.");  
  
    }else{  
  
        if(rbDicotiledonea.isSelected()){  
  
            if(listaDicotiledoneas.getSelectedItem().equals("Angelim araroba")){  
  
                fwkp = 6.92;  
  
                fwkn = 0.31;  
  
                fc0k = 5.05;  
  
                Ec0m = 1287.6;  
  
                fv0 = 0.71;  
  
                material = "Angelim araroba";  
  
            }else if(listaDicotiledoneas.getSelectedItem().equals("Angelim ferro")){  
  
                fwkp = 11.78;  
  
                fwkn = 0.37;
```

```
fc0k = 7.95;

Ec0m = 2082.7;

fv0 = 1.18;

material = "Angelim ferro";

}else if(listaDicotiledoneas.getSelectedItem().equals("Angelim pedra")){

fwkp = 7.55;

fwkn = 0.35;

fc0k = 5.98;

Ec0m = 1291.2;

fv0 = 0.88;

material = "Angelim pedra";

}else if(listaDicotiledoneas.getSelectedItem().equals("Angelim pedra
verdadeiro")){

fwkp = 10.49;

fwkn = 0.48;

fc0k = 7.67;

Ec0m = 1669.4;

fv0 = 1.13;

material = "Angelim pedra verdadeiro";

}else if(listaDicotiledoneas.getSelectedItem().equals("Branquilho")){

fwkp = 8.79;

fwkn = 0.32;

fc0k = 4.81;
```



```
Ec0m = 1348.1;

fv0 = 0.98;

material = "Branquilha";

}else if(listaDicotiledoneas.getSelectedItem().equals("Cafearana")){

fwkp = 7.97;

fwkn = 0.3;

fc0k = 5.91;

Ec0m = 1409.8;

fv0 = 0.59;

material = "Cafearana";

}else if(listaDicotiledoneas.getSelectedItem().equals("Canafístula")){

fwkp = 8.49;

fwkn = 0.62;

fc0k = 5.2;

Ec0m = 1461.3;

fv0 = 1.11;

material = "Canafístula";

}else if(listaDicotiledoneas.getSelectedItem().equals("Casca grossa")){

fwkp = 12.02;

fwkn = 0.41;

fc0k = 5.6;

Ec0m = 1622.4;
```

```
fv0 = 0.82;

material = "Casca grossa";

}else if(listaDicotiledoneas.getSelectedItem().equals("Castelo")){

    fwkp = 9.95;

    fwkn = 0.75;

    fc0k = 5.48;

    Ec0m = 1110.5;

    fv0 = 1.28;

    material = "Castelo";

}else if(listaDicotiledoneas.getSelectedItem().equals("Cedro amargo")){

    fwkp = 5.81;

    fwkn = 0.3;

    fc0k = 3.9;

    Ec0m = 983.9;

    fv0 = 0.61;

    material = "Cedro amargo";

}else if(listaDicotiledoneas.getSelectedItem().equals("Cedro doce")){

    fwkp = 7.14;

    fwkn = 0.3;

    fc0k = 3.15;

    Ec0m = 805.8;

    fv0 = 0.56;
```

```
material = "Cedro doce";

}else if(listaDicotiledoneas.getSelectedItem().equals("Champagne")){

    fwkp = 13.35;

    fwkn = 0.29;

    fc0k = 9.32;

    Ec0m = 2300.2;

    fv0 = 1.07;

    material = "Champagne";

}else if(listaDicotiledoneas.getSelectedItem().equals("Cupiúba")){

    fwkp = 6.21;

    fwkn = 0.33;

    fc0k = 5.44;

    Ec0m = 1362.7;

    fv0 = 1.04;

    material = "Cupiúba";

}else if(listaDicotiledoneas.getSelectedItem().equals("Catiúba")){

    fwkp = 8.62;

    fwkn = 0.33;

    fc0k = 8.38;

    Ec0m = 1942.6;

    fv0 = 1.11;

    material = "Catiúba";
```

```
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto Alba")){  
  
    fwkp = 6.94;  
  
    fwkn = 0.46;  
  
    fc0k = 4.73;  
  
    Ec0m = 1340.9;  
  
    fv0 = 0.95;  
  
    material = "Eucalipto Alba";  
  
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto  
Camaldulensis")){  
  
    fwkp = 7.81;  
  
    fwkn = 0.46;  
  
    fc0k = 4.8;  
  
    Ec0m = 1328.6;  
  
    fv0 = 0.90;  
  
    material = "Eucalipto Camaldulensis";  
  
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto  
Citriodora")){  
  
    fwkp = 12.36;  
  
    fwkn = 0.39;  
  
    fc0k = 6.2;  
  
    Ec0m = 1842.1;  
  
    fv0 = 1.07;  
  
    material = "Eucalipto Citriodora";
```

```
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto  
Cloeziana")){  
  
        fwkp = 9.08;  
  
        fwkn = 0.4;  
  
        fc0k = 5.18;  
  
        Ec0m = 1396.3;  
  
        fv0 = 1.05;  
  
        material = "Eucalipto Cloeziana";  
  
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto Dunnii")){  
  
        fwkp = 13.92;  
  
        fwkn = 0.69;  
  
        fc0k = 4.89;  
  
        Ec0m = 1802.9;  
  
        fv0 = 0.98;  
  
        material = "Eucalipto Dunnii";  
  
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto Grandis")){  
  
        fwkp = 7.02;  
  
        fwkn = 0.26;  
  
        fc0k = 4.03;  
  
        Ec0m = 1281.3;  
  
        fv0 = 0.70;  
  
        material = "Eucalipto Grandis";  
  
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto
```

```
Maculata")){  
  
    fwkp = 11.56;  
  
    fwkn = 0.41;  
  
    fc0k = 6.35;  
  
    Ec0m = 1809.9;  
  
    fv0 = 1.06;  
  
    material = "Eucalipto Maculata";  
  
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto Maidene")){  
  
    fwkp = 8.37;  
  
    fwkn = 0.48;  
  
    fc0k = 4.83;  
  
    Ec0m = 1443.1;  
  
    fv0 = 1.03;  
  
    material = "Eucalipto Maidene";  
  
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto  
Microcorys")){  
  
    fwkp = 11.86;  
  
    fwkn = 0.45;  
  
    fc0k = 5.49;  
  
    Ec0m = 1678.2;  
  
    fv0 = 1.03;  
  
    material = "Eucalipto Microcorys";  
  
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto
```

```
Paniculata")){  
  
    fwkp = 14.74;  
  
    fwkn = 0.47;  
  
    fc0k = 7.27;  
  
    Ec0m = 1988.1;  
  
    fv0 = 1.24;  
  
    material = "Eucalipto Paniculata";  
  
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto  
Propinqua")){  
  
    fwkp = 8.91;  
  
    fwkn = 0.47;  
  
    fc0k = 5.16;  
  
    Ec0m = 1556.1;  
  
    fv0 = 0.97;  
  
    material = "Eucalipto Propinqua";  
  
    }else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto  
Punctata")){  
  
    fwkp = 12.56;  
  
    fwkn = 0.6;  
  
    fc0k = 7.85;  
  
    Ec0m = 1936.0;  
  
    fv0 = 1.29;  
  
    material = "Eucalipto Punctata";
```

```
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto Saligna")){  
  
    fwkp = 9.55;  
  
    fwkn = 0.4;  
  
    fc0k = 4.68;  
  
    Ec0m = 1493.3;  
  
    fv0 = 0.82;  
  
    material = "Eucalipto Saligna";  
  
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto  
Tereticornis")){  
  
    fwkp = 11.59;  
  
    fwkn = 0.46;  
  
    fc0k = 5.77;  
  
    Ec0m = 1719.8;  
  
    fv0 = 0.97;  
  
    material = "Eucalipto Tereticornis";  
  
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto Triantha")){  
  
    fwkp = 10.09;  
  
    fwkn = 0.27;  
  
    fc0k = 5.39;  
  
    Ec0m = 1461.7;  
  
    fv0 = 0.92;  
  
    material = "Eucalipto Triantha";  
  
}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto Umbra")){
```



```
fwkp = 9.04;

fwkn = 0.3;

fc0k = 4.27;

Ec0m = 1457.7;

fv0 = 0.94;

material = "Eucalipto Umbra";

}else if(listaDicotiledoneas.getSelectedItem().equals("Eucalipto
Urophylla")){

fwkp = 8.51;

fwkn = 0.41;

fc0k = 4.6;

Ec0m = 1316.6;

fv0 = 0.83;

material = "Eucalipto Urophylla";

}else if(listaDicotiledoneas.getSelectedItem().equals("Garapa Roraima")){

fwkp = 10.80;

fwkn = 0.69;

fc0k = 7.84;

Ec0m = 1835.9;

fv0 = 1.19;

material = "Garapa Roraima";

}else if(listaDicotiledoneas.getSelectedItem().equals("Guaíçara")){

fwkp = 11.56;
```

```
fwkn = 0.42;

fc0k = 7.14;

Ec0m = 1462.4;

fv0 = 1.25;

material = "Guaíçara";

}else if(listaDicotiledoneas.getSelectedItem().equals("Guarucaia")){

fwkp = 7.09;

fwkn = 0.55;

fc0k = 6.24;

Ec0m = 1721.2;

fv0 = 1.55;

material = "Guarucaia";

}else if(listaDicotiledoneas.getSelectedItem().equals("Ipê")){

fwkp = 9.68;

fwkn = 0.31;

fc0k = 7.6;

Ec0m = 1801.1;

fv0 = 1.31;

material = "Ipê";

}else if(listaDicotiledoneas.getSelectedItem().equals("Jatobá")){

fwkp = 15.75;

fwkn = 0.32;
```

```
fc0k = 9.33;

Ec0m = 2360.7;

fv0 = 1.57;

material = "Jatobá";

}else if(listaDicotiledoneas.getSelectedItem().equals("Louro preto")){

fwkp = 11.19;

fwkn = 0.33;

fc0k = 5.65;

Ec0m = 1418.5;

fv0 = 0.90;

material = "Louro preto";

}else if(listaDicotiledoneas.getSelectedItem().equals("Maçaranduba")){

fwkp = 13.85;

fwkn = 0.54;

fc0k = 8.29;

Ec0m = 2273.3;

fv0 = 1.49;

material = "Maçaranduba";

}else if(listaDicotiledoneas.getSelectedItem().equals("Mandioqueira")){

fwkp = 8.91;

fwkn =0.27 ;

fc0k = 7.14;
```

```
Ec0m = 1897.1;

fv0 = 1.06;

material = "Mandioqueira";

}else if(listaDicotiledoneas.getSelectedItem().equals("Oiticica amarela")){

    fwkp = 8.25;

    fwkn = 0.39;

    fc0k = 6.99;

    Ec0m = 1471.9;

    fv0 = 1.06;

    material = "Oiticica amarela";

}else if(listaDicotiledoneas.getSelectedItem().equals("Quarubarana")){

    fwkp = 5.81;

    fwkn = 0.26;

    fc0k = 3.78;

    Ec0m = 906.7;

    fv0 = 0.58;

    material = "Quarubarana";

}else if(listaDicotiledoneas.getSelectedItem().equals("Sucupira")){

    fwkp = 12.34;

    fwkn = 0.34;

    fc0k = 9.52;

    Ec0m = 2172.4;
```

```
fv0 = 1.18;

material = "Sucupira";

}else if(listaDicotiledoneas.getSelectedItemAt().equals("Tatajuba")){

    fwkp = 7.88;

    fwkn = 0.39;

    fc0k = 7.95;

    Ec0m = 1958.3;

    fv0 = 1.22;

    material = "Tatajuba";

}

}else if(rbConifera.isSelected()){

    if(listaConiferas.getSelectedItemAt().equals("Pinho do Paraná")){

        fwkp = 9.31;

        fwkn = 0.16;

        fc0k = 4.09;

        Ec0m = 1522.5;

        fv0 = 0.88;

        material = "Pinho do Paraná";

    }else if(listaConiferas.getSelectedItemAt().equals("Pinus caribea")){

        fwkp = 6.48;
```

```
fwkn = 0.32;

fc0k = 3.54;

Ec0m = 843.1;

fv0 = 0.78;

material = "Pinus caribea";

}else if(listaConiferas.getSelectedItem().equals("Pinus bahamensis")){

fwkp = 5.27;

fwkn = 0.24;

fc0k = 3.26;

Ec0m = 711;

fv0 = 0.68;

material = "Pinus bahamensis";

}else if(listaConiferas.getSelectedItem().equals("Pinus hondurensis")){

fwkp = 5.03;

fwkn = 0.26;

fc0k = 4.23;

Ec0m = 986.8;

fv0 = 0.78;

material = "Pinus hondurensis";

}else if(listaConiferas.getSelectedItem().equals("Pinus elliottii")){

fwkp =6.6 ;

fwkn = 0.25;
```

```
fc0k = 4.04;

Ec0m = 1188.9;

fv0 = 0.74;

material = "Pinus elliottii";

}else if(listaConiferas.getSelectedItem().equals("Pinus oocarpa")){

    fwkp = 6.09;

    fwkn = 0.25;

    fc0k = 4.36;

    Ec0m = 1090.4;

    fv0 = 0.80;

    material = "Pinus oocarpa";

}else if(listaConiferas.getSelectedItem().equals("Pinus taeda")){

    fwkp = 8.28;

    fwkn = 0.28;

    fc0k = 4.44;

    Ec0m = 1330.4;

    fv0 = 0.77;

    material = "Pinus taeda";

}

}
```

```
painelPropriedades.setSelectedIndex(1);  
  
painelPropriedades.setEnabledAt(1, true);  
  
}  
  
}  
  
private void rbDicotiledoneaActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if(rbDicotiledonea.isSelected()){  
  
        listaConiferas.setEnabled(false);  
  
        listaDicotiledoneas.setEnabled(true);  
  
        bpClasseResistencia.setEnabled(true);  
  
    }  
  
}  
  
private void rbConiferaActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if(rbConifera.isSelected()){  
  
        listaConiferas.setEnabled(true);  
  
        listaDicotiledoneas.setEnabled(false);  
  
        bpClasseResistencia.setEnabled(true);  
  
    }  
  
}
```



```
private void bpPaginaPrincipalActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if(rbHorizontal.isSelected()){  
  
        painelAbas.setSelectedIndex(2);  
  
        painelCarregamentos.setSelectedIndex(0);  
  
        painelCarregamentos.setEnabledAt(1, false);  
  
        painelCarregamentos.setEnabledAt(0, true);  
  
        painelAbas.setEnabledAt(2, true);  
  
    }else if(rbVertical.isSelected()){  
  
        painelAbas.setSelectedIndex(2);  
  
        painelCarregamentos.setSelectedIndex(1);  
  
        painelCarregamentos.setEnabledAt(0, false);  
  
        painelCarregamentos.setEnabledAt(1, true);  
  
        painelAbas.setEnabledAt(2, true);  
  
    }else{  
  
        JOptionPane.showMessageDialog(rootPane, "Selecione uma das opções.");  
  
    }  
  
}
```

```
private void bIniciarActionPerformed(java.awt.event.ActionEvent evt) {  
  
    painelAbas.setEnabled(true);  
  
}
```

```
painelAbas.setSelectedIndex(1);

painelAbas.setEnabledAt(1, true);

}

private void bIniciarStateChanged(javax.swing.event.ChangeEvent evt) {

    // TODO add your handling code here:

}

private void campoCargaNqVKeyTyped(java.awt.event.KeyEvent evt) {

    String value = campoCargaNqV.getText();

    if (value.length() == 0)

        return;

    if (value.substring(value.length() - 1, value.length()).matches("[a-zA-Z]+")){

        value = value.substring(0, value.length() - 1);

    }

    campoCargaNqV.setText(value);

}

private void bpVerticalActionPerformed(java.awt.event.ActionEvent evt) {
```

```
if (lgBloqueioVer){  
    JOptionPane.showMessageDialog(null, "Recalcule as combinações");  
    return;  
}  
  
if(campoComprimentoV.getText().equals("")){  
    JOptionPane.showMessageDialog(rootPane, "Determine um valor para a  
altura.");  
}  
else if(campoCargaNgV.getText().equals("")){  
    JOptionPane.showMessageDialog(rootPane, "Determine um valor para a  
carga permanente.");  
}  
else{  
  
    //Apagar dados de Tração e Compressão  
  
    DecimalFormat df = new DecimalFormat("0.####");  
  
    String scompV = campoComprimentoV.getText().replaceAll(" ", ".");  
    comprimentoV = Double.parseDouble(scompV);  
  
    if(Nsd > 0){  
        String scargaT = df.format(Nsd);  
        campoNdT.setText(scargaT);  
        painelAbas.setSelectedIndex(3);  
    }  
}
```

```
        painelPropiedades.setSelectedIndex(0);

        painelAbas.setEnabledAt(3, true);
    }else if(Nsd < 0){

        double Nsdn = Nsd * -1;

        String scargaC = df.format(Nsdn);

        campoNdC.setText(scargaC);

        painelAbas.setSelectedIndex(3);

        painelPropiedades.setSelectedIndex(0);

        painelAbas.setEnabledAt(3, true);

    }

}

}

private void campoCargaNgVKeyTyped(java.awt.event.KeyEvent evt) {

    String value = campoCargaNgV.getText();

    if (value.length() == 0)

        return;

    if (value.substring(value.length() - 1, value.length()).matches("[a-zA-Z]+")){

        value = value.substring(0, value.length() - 1);
```

```
}
```

```
campoCargaNgV.setText(value);
```

```
}
```

```
private void campoComprimentoVKeyReleased(java.awt.event.KeyEvent evt) {
```

```
    int x;
```

```
    try {
```

```
        x = Integer.parseInt(campoComprimentoV.getText());
```

```
    } catch (NumberFormatException nfe) {
```

```
        campoComprimentoV.setText(campoComprimentoV.getText().substring(0,  
campoComprimentoV.getText().length() - 1));
```

```
    }
```

```
}
```

```
private void bCalculaCombVActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    lgBloqueioVer = false;
```

```
    try{
```

```
        DecimalFormat df = new DecimalFormat("0.####");
```

```
        String sNg = campoCargaNgV.getText().replaceAll( " , " , "." );
```

```
        Ng = Double.parseDouble(sNg);
```

```
String sNq = campoCargaNqV.getText().replaceAll( " , " , "." );
```

```
Nq = Double.parseDouble(sNq);
```

```
//Fator de cobinação
```

```
if(listaCargaVariavelV.getSelectedItemId().equals("Variações uniformes de temperatura em relação à média anual local")){
```

```
    psi0V = 0.6;
```

```
    psi1V = 0.5;
```

```
    psi2V = 0.3;
```

```
}else if(listaCargaVariavelV.getSelectedItemId().equals("Pressão dinâmica do vento")){
```

```
    psi0V = 0.6;
```

```
    psi1V = 0.5;
```

```
    psi2V = 0.3;
```

```
}else if(listaCargaVariavelV.getSelectedItemId().equals("Carga acidental em edifícios sem elevadas concentrações de pessoas")){
```

```
    psi0V = 0.6;
```

```
    psi1V = 0.5;
```

```
    psi2V = 0.3;
```

```
}else if(listaCargaVariavelV.getSelectedItemId().equals("Carga acidental em edifícios com elevadas concentrações de pessoas")){
```

```
    psi0V = 0.6;
```

```
    psi1V = 0.5;
```

```
    psi2V = 0.3;
```

```
}else if(listaCargaVariavelV.getSelectedltem().equals("Bibliotecas, arquivos,  
oficinas e garagens")){
```

```
    psi0V = 0.6;
```

```
    psi1V = 0.5;
```

```
    psi2V = 0.3;
```

```
}
```

```
//Combinação 1
```

```
// Carga Permanente
```

```
if(rbGrandeV.isSelected() && Ng < 0 ){
```

```
    YgV1 = 1.4;
```

```
}else if (rbPequenaV.isSelected() && Ng < 0 ){
```

```
    YgV1 = 1.3;
```

```
}else if(rbGrandeV.isSelected() && Ng > 0 ){
```

```
    YgV1= 0.9;
```

```
}else if(rbPequenaV.isSelected() && Ng > 0 ){
```

```
    YgV1 = 1;
```

```
}
```

```
// Carga variável
```

```
if(Nq < 0){
    YqV1 = 1.4;
}
else{
    YqV1 = 0;
}

// Nsd1

if(listaCargaVariavelV.getSelectedItem().equals("Pressão dinâmica do
vento")){

    Nsd1 = (YgV1*Ng) + (0.75*YqV1*Nq);

    String sNsd1 = df.format(Nsd1);

    campoNsd1.setText(sNsd1);

}
else{

    Nsd1 = (YgV1*Ng) + (YqV1*Nq);

    String sNsd1 = df.format(Nsd1);

    campoNsd1.setText(sNsd1);

}

//Combinação 2

// Carga Permanente

if(rbGrandeV.isSelected() && Ng > 0 ){
```



```

    YgV2 = 1.4;

}else if (rbPequenaV.isSelected() && Ng > 0 ){

    YgV2 = 1.3;

}else if (rbGrandeV.isSelected() && Ng < 0 ){

    YgV2 = 0.9;

}else if (rbPequenaV.isSelected() && Ng < 0 ){

    YgV2 = 1;

}

```

// Carga variável

```

if(Nq > 0){

    YqV2 = 1.4;

}else{

    YqV2 = 0;

}

```

// Nsd2

```

if(listaCargaVariavelV.getSelectedItem().equals("Pressão dinâmica do
vento")){

```

```

    Nsd2 = (YgV2*Ng) + (0.75*YqV2*Nq);

```

```

    String sNsd2 = df.format(Nsd2);

```

```

    campoNsd2.setText(sNsd2);

```

```
}else{

    Nsd2 = (YgV2*Ng) + (YqV2*Nq);

    String sNsd2 = df.format(Nsd2);

    campoNsd2.setText(sNsd2);

}

//Nsd

if(Nsd1 > 0 && Nsd2 > 0){

    if(Nsd1 > Nsd2){

        Nsd = Nsd1;

    }else{

        Nsd = Nsd2;

    }

}

}else if(Nsd1 < 0 && Nsd2 < 0){

    if(Nsd1 < Nsd2){

        Nsd = Nsd1;

    }else{

        Nsd = Nsd2;

    }

}

}else{

    JOptionPane.showMessageDialog(rootPane, "O programa só aceita
    combinações com o mesmo sentido, redefinir carregamentos.");

    bpVertical.setEnabled(false);
```

```
        campoCargaNgV.setText("");

        campoCargaNqV.setText("");

        campoNsd1.setText("");

        campoNsd2.setText("");

    }

    bpVertical.setEnabled(true);

}catch(Exception e){

    if(campoComprimentoV.getText().equals("")){

        JOptionPane.showMessageDialog(rootPane, "Insira o valor do
comprimento.");

    }else if(campoCargaNgV.getText().equals("")){

        JOptionPane.showMessageDialog(rootPane, "Insira o valor da carga
permanente");

    }else if(campoCargaNqV.getText().equals("")){

        JOptionPane.showMessageDialog(rootPane, "Insira o valor da carga
variável, caso não houver digite 0.");

    }

}

}
```

```
}

private void campoComprimentoHKeyReleased(java.awt.event.KeyEvent evt) {

    int x;

    try {

        x = Integer.parseInt(campoComprimentoH.getText());

    } catch (NumberFormatException nfe) {

        campoComprimentoH.setText(campoComprimentoH.getText().substring(0,
        campoComprimentoH.getText().length() - 1));

    }

}

private void bpHorizontalActionPerformed(java.awt.event.ActionEvent evt) {

    if (lgBloqueioHor){

        JOptionPane.showMessageDialog(null, "Recalcule as combinações");

        return;

    }

    DecimalFormat df = new DecimalFormat("0.####");

    String sMsd = df.format(Msd);

    campoMdF.setText(sMsd);

    String sVsd = df.format(Vsd);
```

```
campoVdCI.setText(sVsd);

paineIAbas.setSelectedIndex(3);

paineIPropriedades.setSelectedIndex(0);

paineIAbas.setEnabledAt(3, true);

}

private void campoCargaQQHKeyTyped(java.awt.event.KeyEvent evt) {

    String value = campoCargaQQH.getText();

    if (value.length() == 0)

        return;

    if (value.substring(value.length() - 1, value.length()).matches("[a-zA-Z]+")){

        value = value.substring(0, value.length() - 1);

    }

    campoCargaQQH.setText(value);

}
```

```
private void bCalculaCombHActionPerformed(java.awt.event.ActionEvent evt) {  
  
    lgBloqueioHor = false;  
  
    try{  
  
        DecimalFormat df = new DecimalFormat("0.####");  
  
        String sg = campoCargaGGH.getText().replaceAll( ",", "." );  
  
        g = Double.parseDouble(sg);  
  
        String sq = campoCargaQQH.getText().replaceAll( ",", "." );  
  
        q = Double.parseDouble(sq);  
  
        String sComprimentoH = campoComprimentoH.getText().replaceAll( ",", "." );  
  
        comprimentoH = Double.parseDouble(sComprimentoH);  
  
        Mg = (g/100 * Math.pow((comprimentoH), 2))/8;  
  
        Mq = (q/100 * Math.pow((comprimentoH), 2))/8;  
  
        Vg = (g/100 * comprimentoH)/2;  
  
        Vq = (q/100 * comprimentoH)/2;
```

```
//Fator de combinação

if(listaCargaVariavelV.getSelectedltem().equals("Variações uniformes de
temperatura em relação à média anual local")){

    psi0V = 0.6;

    psi1V = 0.5;

    psi2V = 0.3;

}else if(listaCargaVariavelV.getSelectedltem().equals("Pressão dinâmica do
vento")){

    psi0V = 0.6;

    psi1V = 0.5;

    psi2V = 0.3;

}else if(listaCargaVariavelV.getSelectedltem().equals("Carga acidental em
edifícios sem elevadas concentrações de pessoas")){

    psi0V = 0.6;

    psi1V = 0.5;

    psi2V = 0.3;

}else if(listaCargaVariavelV.getSelectedltem().equals("Carga acidental em
edifícios com elevadas concentrações de pessoas")){

    psi0V = 0.6;

    psi1V = 0.5;

    psi2V = 0.3;

}else if(listaCargaVariavelV.getSelectedltem().equals("Bibliotecas, arquivos,
oficinas e garagens")){

    psi0V = 0.6;
```

```
psi1V = 0.5;

psi2V = 0.3;

}

//Combinação 1

// Carga Permanente

if(rbGrandeH.isSelected() && g < 0 ){

    YgH1 = 1.4;

}else if (rbPequenaH.isSelected() && g < 0 ){

    YgH1 = 1.3;

}else if(rbGrandeH.isSelected() && g > 0 ){

    YgH1 = 0.9;

}else if(rbPequenaH.isSelected() && g > 0 ){

    YgH1 = 1;

}

// Carga variável

if(q < 0){

    YqH1 = 1.4;
```



```
}else{
```

```
    YqH1 = 0;
```

```
}
```

```
// Msd1
```

```
Msd1 = (YgH1*Mg) + (YqH1*Mq);
```

```
String sMsd1 = df.format(Msd1);
```

```
campoMsd1.setText(sMsd1);
```

```
//Vsd1
```

```
Vsd1 = (YgH1*Vg) + (YqH1*Vq);
```

```
String sVsd1 = df.format(Vsd1);
```

```
campoVsd1.setText(sVsd1);
```

```
//Combinação 2
```

```
// Carga Permanente
```

```
if(rbGrandeH.isSelected() && g > 0){
```

```
    YgH2 = 1.4;
```

```
}else if (rbPequenaH.isSelected() && g > 0){
```

```
    YgH2 = 1.3;
```

```
}else if(rbGrandeH.isSelected() && g < 0 ){  
    YgH2 = 0.9;  
}else if(rbPequenaH.isSelected() && g < 0 ){  
    YgH2 = 1;  
}  
  
// Carga variável  
  
if(q > 0){  
    YqH2 = 1.4;  
}else{  
    YqH2 = 0;  
}  
  
// Msd2  
  
Msd2 = (YgH2*Mg) + (YqH2*Mq);  
String sMsd2 = df.format(Msd2);  
campoMsd2.setText(sMsd2);  
  
//Vsd1  
  
Vsd2 = (YgH2*Vg) + (YqH2*Vq);  
String sVsd2 = df.format(Vsd2);
```

```
campoVsd2.setText(sVsd2);

//Msd

if(Msd1 > 0 && Msd2 > 0){

    if(Msd1 > Msd2){

        Msd = Msd1;

    }else{

        Msd = Msd2;

    }

}

}else if(Msd1 < 0 && Msd2 < 0){

    if(Msd1 < Msd2){

        Msd = Msd1;

    }else{

        Msd = Msd2;

    }

}

}else{

    JOptionPane.showMessageDialog(rootPane, "Configurar Flexão");

}

//Vsd

if(Vsd1 > 0 && Vsd2 > 0){
```

```
        if(Vsd1 > Vsd2){
            Vsd = Vsd1;
        }else{
            Vsd = Vsd2;
        }
    }else if(Vsd1 < 0 && Vsd2 < 0){
        if(Vsd1 < Vsd2){
            Vsd = Vsd1;
        }else{
            Vsd = Vsd2;
        }
    }else{
        JOptionPane.showMessageDialog(rootPane, "Configurar Cisalhamento");
    }

    bpHorizontal.setEnabled(true);

}catch(Exception e){
    if(campoComprimentoH.getText().equals("")){
        JOptionPane.showMessageDialog(rootPane, "Insira o valor do comprimento.");
    }else if(campoCargaGGH.getText().equals("")){
        JOptionPane.showMessageDialog(rootPane, "Insira o valor da carga
```

```
permanente");

        }else if(campoCargaQQH.getText().equals("")){

            JOptionPane.showMessageDialog(rootPane, "Insira o valor da carga
variável, caso não houver digite 0.");

        }

    }

}
```

```
private void campoCargaGGHKeyTyped(java.awt.event.KeyEvent evt) {

    String value = campoCargaGGH.getText();

    if (value.length() == 0)

        return;

    if (value.substring(value.length() - 1, value.length()).matches("[a-zA-Z]+")){

        value = value.substring(0, value.length() - 1);

    }

    campoCargaGGH.setText(value);

}
```

```
}  
  
private void bVerificarTActionPerformed(java.awt.event.ActionEvent evt) {  
  
    lgBloqueioTra = false;  
  
    // setar base e altura conforme item selecionado.  
  
    if(listaSecoesT.getSelectedItem().equals("Seção: 5.0cm x 10.0cm. Com área =  
50.0cm². ")) {  
  
        base = 5;  
  
        altura = 10;  
  
    }else if(listaSecoesT.getSelectedItem().equals("Seção: 10.0cm x 5.0cm. Com  
área = 50.0cm². ")) {  
  
        base = 10;  
  
        altura = 5;  
  
    }else if(listaSecoesT.getSelectedItem().equals("Seção: 5.0cm x 15.0cm. Com  
área = 75.0cm². ")) {  
  
        base = 5;  
  
        altura = 15;  
  
    }else if(listaSecoesT.getSelectedItem().equals("Seção: 15.0cm x 5.0cm. Com  
área = 75.0cm². ")) {  
  
        base = 15;  
  
        altura = 5;  
  
    }else if(listaSecoesT.getSelectedItem().equals("Seção: 5.0cm x 20.0cm. Com  
área = 100.0cm². ")) {  
  
        base = 5;  
  
        altura = 20;
```

```
    }else if(listaSecoesT.getSelectedltem().equals("Seção: 10.0cm x 10.0cm. Com
área = 100.0cm².)){

        base = 10;

        altura = 10;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 20.0cm x 5.0cm. Com
área = 100.0cm².)){

        base = 20;

        altura = 5;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 5.0cm x 25.0cm. Com
área = 125.0cm².)){

        base = 5;

        altura = 25;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 25.0cm x 5.0cm. Com
área = 125.0cm².)){

        base = 25;

        altura = 5;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 5.0cm x 30.0cm. Com
área = 150.0cm².)){

        base = 5;

        altura = 30;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 10.0cm x 15.0cm. Com
área = 150.0cm².)){

        base = 10;

        altura = 15;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 15.0cm x 10.0cm. Com
área = 150.0cm².)){
```

```
base = 15;

altura = 10;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 30.0cm x 5.0cm. Com
área = 150.0cm².)){

base = 30;

altura = 5;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 10.0cm x 20.0cm. Com
área = 200.0cm².)){

base = 10;

altura = 20;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 20.0cm x 10.0cm. Com
área = 200.0cm².)){

base = 20;

altura = 10;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 15.0cm x 15.0cm. Com
área = 225.0cm².)){

base = 15;

altura = 15;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 10.0cm x 25.0cm. Com
área = 250.0cm².)){

base = 10;

altura = 25;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 25.0cm x 10.0cm. Com
área = 250.0cm².)){

base = 25;
```



```
altura = 10;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 10.0cm x 30.0cm. Com
área = 300.0cm².)){

base = 10;

altura = 30;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 15.0cm x 20.0cm. Com
área = 300.0cm².)){

base = 15;

altura = 20;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 20.0cm x 15.0cm. Com
área = 300.0cm².)){

base = 20;

altura = 15;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 30.0cm x 10.0cm. Com
área = 300.0cm².)){

base = 30;

altura = 10;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 15.0cm x 25.0cm. Com
área = 375.0cm².)){

base = 15;

altura = 25;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 25.0cm x 15.0cm. Com
área = 375.0cm².)){

base = 25;

altura = 15;
```

```
    }else if(listaSecoesT.getSelectedltem().equals("Seção: 20.0cm x 20.0cm. Com
área = 400.0cm².)){

        base = 20;

        altura = 20;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 15.0cm x 30.0cm. Com
área = 450.0cm².)){

        base = 15;

        altura = 30;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 30.0cm x 15.0cm. Com
área = 450.0cm².)){

        base = 30;

        altura = 15;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 20.0cm x 25.0cm. Com
área = 500.0cm².)){

        base = 20;

        altura = 25;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 25.0cm x 20.0cm. Com
área = 500.0cm².)){

        base = 25;

        altura = 20;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 20.0cm x 30.0cm. Com
área = 600.0cm².)){

        base = 20;

        altura = 30;

    }else if(listaSecoesT.getSelectedltem().equals("Seção: 30.0cm x 20.0cm. Com
área = 600.0cm².)){
```

```
base = 30;

altura = 20;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 25.0cm x 25.0cm. Com
área = 625.0cm².)){

base = 25;

altura = 25;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 25.0cm x 30.0cm. Com
área = 750.0cm².)){

base = 25;

altura = 30;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 30.0cm x 25.0cm. Com
área = 750.0cm².)){

base = 30;

altura = 25;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 30.0cm x 30.0cm. Com
área = 900.0cm².)){

base = 30;

altura = 30;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 12.5cm x 5.0cm. Com
área = 62.5cm².)){

base = 12.5;

altura = 5;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 7.5cm x 10.0cm. Com
área = 75.0cm².)){

base = 7.5;
```

```
altura = 10;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 17.5cm x 5.0cm. Com
área = 87.5cm².)){

    base = 17.5;

    altura = 5;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 7.5cm x 15.0cm. Com
área = 112.5cm².)){

    base = 7.5;

    altura = 15;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 22.5cm x 5.0cm. Com
área = 112.5cm².)){

    base = 22.5;

    altura = 5;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 12.5cm x 10.0cm. Com
área = 125.0cm².)){

    base = 12.5;

    altura = 10;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 27.5cm x 5.0cm. Com
área = 137.5cm².)){

    base = 27.5;

    altura = 5;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 7.5cm x 20.0cm. Com
área = 150.0cm².)){

    base = 7.5;

    altura = 20;
```

```
}else if(listaSecoesT.getSelectedltem().equals("Seção: 17.5cm x 10.0cm. Com  
área = 175.0cm².)){
```

```
    base = 17.5;
```

```
    altura = 10;
```

```
}else if(listaSecoesT.getSelectedltem().equals("Seção: 7.5cm x 25.0cm. Com  
área = 187.5cm².)){
```

```
    base = 7.5;
```

```
    altura = 25;
```

```
}else if(listaSecoesT.getSelectedltem().equals("Seção: 12.5cm x 15.0cm. Com  
área = 187.5cm².)){
```

```
    base = 12.5;
```

```
    altura = 15;
```

```
}else if(listaSecoesT.getSelectedltem().equals("Seção: 7.5cm x 30.0cm. Com  
área = 225.0cm².)){
```

```
    base = 7.5;
```

```
    altura = 30;
```

```
}else if(listaSecoesT.getSelectedltem().equals("Seção: 22.5cm x 10.0cm. Com  
área = 225.0cm².)){
```

```
    base = 22.5;
```

```
    altura = 10;
```

```
}else if(listaSecoesT.getSelectedltem().equals("Seção: 12.5cm x 20.0cm. Com  
área = 250.0cm².)){
```

```
    base = 12.5;
```

```
    altura = 20;
```

```
}else if(listaSecoesT.getSelectedltem().equals("Seção: 17.5cm x 15.0cm. Com  
área = 262.5cm².)){
```

```
base = 17.5;

altura = 15;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 27.5cm x 10.0cm. Com
área = 275.0cm².)){

base = 27.5;

altura = 10;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 12.5cm x 25.0cm. Com
área = 312.5cm².)){

base = 12.5;

altura = 25;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 22.5cm x 15.0cm. Com
área = 337.5cm².)){

base = 22.5;

altura = 15;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 17.5cm x 20.0cm. Com
área = 350.0cm².)){

base = 17.5;

altura = 20;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 12.5cm x 30.0cm. Com
área = 375.0cm².)){

base = 12.5;

altura = 30;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 27.5cm x 15.0cm. Com
área = 412.5cm².)){

base = 27.5;
```

```
altura = 15;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 17.5cm x 25.0cm. Com
área = 437.5cm².)){

    base = 17.5;

    altura = 25;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 22.5cm x 20.0cm. Com
área = 450.0cm².)){

    base = 22.5;

    altura = 20;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 17.5cm x 30.0cm. Com
área = 525.0cm².)){

    base = 17.5;

    altura = 30;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 27.5cm x 20.0cm. Com
área = 550.0cm².)){

    base = 27.5;

    altura = 20;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 22.5cm x 25.0cm. Com
área = 562.5cm².)){

    base = 22.5;

    altura = 25;

}else if(listaSecoesT.getSelectedltem().equals("Seção: 22.5cm x 30.0cm. Com
área = 675.0cm².)){

    base = 22.5;

    altura = 30;
```

```
    }else if(listaSecoesT.getSelectedltem().equals("Seção: 27.5cm x 25.0cm. Com  
    área = 687.5cm².)){
```

```
        base = 27.5;
```

```
        altura = 25;
```

```
    }else if(listaSecoesT.getSelectedltem().equals("Seção: 27.5cm x 30.0cm. Com  
    área = 825.0cm².)){
```

```
        base = 27.5;
```

```
        altura = 30;
```

```
    }
```

```
double lx = (base*Math.pow(altura, 3))/12;
```

```
double ly = (altura*Math.pow(base, 3))/12;
```

```
if(lx > ly){
```

```
    l = ly;
```

```
    double aux = altura;
```

```
    altura = base;
```

```
    base = aux;
```

```
    }else{
```

```
        l = lx;
```

```
    }
```

```
String sL0 = campoComprimentoV.getText().replaceAll( ",", "." );
```



```
double l0 = Double.parseDouble(sL0);

area = (base * altura);

double r = Math.sqrt(l/area);

esb = l0/r;

double lamb = 170;

if (esb <= lamb){

    campoEsbeltezT.setText("OK!");

    brTracao.setEnabled(true);

}else {

    campoEsbeltezT.setText("Redefinir seção.");

    brTracao.setEnabled(false);

}

}

private void brTracaoActionPerformed(java.awt.event.ActionEvent evt) {

    if (lgBloqueioTra){

        JOptionPane.showMessageDialog(null, "Seção alterada, favor refazer a
verificação");

        return;

    }

    DecimalFormat df2 = new DecimalFormat("0.##");
```

```
DecimalFormat df3 = new DecimalFormat("0.###");
```

```
String sKmod = df3.format(kmod);
```

```
String sNsd1 = df2.format(Nsd1);
```

```
String sNsd2 = df2.format(Nsd2);
```

```
String sAngulo = df2.format(anguloT);
```

```
String sfv0 = df2.format(fv0*10);
```

```
String sl = df3.format(l);
```

```
String sesb = df3.format(esb);
```

```
String sfc0 = df2.format(fc0k*10);
```

```
String sft0 = df2.format(fwkp*10);
```

```
String sft90 = df2.format(fwkn*10);
```

```
String sEc0 = df2.format(Ec0m*10);
```

```
String ang;
```

```
String rft;
```

```
if(rb0T.isSelected()){
```

```
    ang = "0";
```

```
    rft = campoft0d.getText();
```

```
}else if(rb90T.isSelected()){
```

```
    ang = "90";
```

```
    rft = campoft90d.getText();
```

```
}else{  
    ang = "a";  
    rft = campoftad.getText();  
}
```

```
areaTxtResultados.setText("");
```

```
areaTxtResultados.setText
```

```
(" ---- Propriedades da peça ----"
```

```
+ "\n\n Material escolhido: " + material
```

```
+ "\n fc0 = " + sfc0 + " MPa"
```

```
+ "\n ft0 = " + sft0 + " MPa"
```

```
+ "\n ft90 = " + sft90 + " MPa"
```

```
+ "\n fv = " + sfv0 + " MPa"
```

```
+ "\n Ec0 = " + sEc0 + " MPa"
```

```
+ "\n\n Kmod = " + kmod1 + " * " + kmod2 + " * " + kmod3
```

+ "\n Kmod = " + sKmod

+ "\n\n ---- Carregamentos e combinações ----"

+ "\n\n Carregamentos:"

+ "\n Ngk = " + Ng + " kN"

+ "\n Nqk = " + Nq + " kN"

+ "\n\n Combinações:"

+ "\n Nsd = ( $\gamma_g * Ng$ ) + ( $\gamma_q * Nq$ )"

+ "\n Nsd1 = (" + YgV1 + " \* " + Ng + ") + (" + YqV1 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd1 + "kN"

+ "\n Nsd2 = (" + YgV2 + " \* " + Ng + ") + (" + YqV2 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd2 + "kN"

+ "\n\n ---- Dados do dimensionamento ----"

+ "\n\n Nd = " + campoNdT.getText() + " kN"

+ "\n Kmod = " + sKmod

+ "\n Ywt = 1.8"

+ "\n Ângulo de inclinação = " + sAngulo + "°"

+ "\n ft" + ang + ",d = " + rft + " kN/cm²"

+ "\n\n Dados geométricos:"

+ "\n Comprimento = " + comprimentoV + "cm"

+ "\n Base = " + base + "cm"

+ "\n Altura = " + altura + "cm"

+ "\n Área = " + base \* altura + "cm<sup>2</sup>"

+ "\n\n Momento de inércia:"

+ "\n  $I = b * h^3 / 12$ "

+ "\n  $I =$  + base + " \* " + altura + "<sup>3</sup> / 12"

+ "\n  $I =$  + sl + " cm<sup>4</sup>"

+ "\n\n ---- Verificações ----"

+ "\n\n Verificação de esbeltez:"

+ "\n  $\lambda \leq 170$ "

+ "\n " + sesb + "  $\leq 170$ "

+ "\n Verificação OK!"

```
);
```

```
painelAbas.setSelectedIndex(8);
```

```
painelAbas.setEnabledAt(8, true);
```

```
}
```

```
private void bDimensionarTActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if(rb0T.isSelected()){
```

```
        anguloT = 0;
```

```
        campoEsbeltezT.setText("");
```

```
        bVerificarT.setEnabled(true);
```

```
        double ft0dt = kmod * (0.7*fwkp/fwt);
```

```
DecimalFormat df = new DecimalFormat("0.####");
```

```
String sft0dt = df.format(ft0dt);
```

```
campoft0d.setText(sft0dt);
```

```
String sndT = campoNdT.getText().replaceAll( ",", "." );
```

```
double ndT = Double.parseDouble(sndT);
```

```
areaT = (ndT/ft0dt);
```

```
if(areaT < 50){
```

```
    areaT = 50;
```

```
}
```

```
String sareaT = df.format(areaT);
```

```
campoAreaMinT.setText(sareaT);
```

```
String sAreaMin = campoAreaMinT.getText().replaceAll( ",", "." );
```

```
double areaMin = Double.parseDouble(sAreaMin);
```

```
// Achar seções
```

```
listaSecoesT.removeAllItems();

ArrayList<Medidas> maioresQueMin = new ArrayList<>();

if(areaMin > 900){

    listaSecoesT.addItem("Área mínima acima das seções utilizadas, redefina
o carregamento.");

    bVerificarT.setEnabled(false);

}else{

    //Percorrer ArrayList e grava valores acima da área minima

    for(Medidas total : varMedidas) {

        if(total.getArea() >= areaMin){

            maioresQueMin.add(total);

        }

    }

    // Percorre novo vetor e adiciona as opções

    for(Medidas maiores : maioresQueMin){

        listaSecoesT.addItem("Seção: " + maiores.getBase() + "cm x " +
maiores.getAltura() + "cm. Com área = " + maiores.getArea() + "cm².");

    }

}
```



```
bVerificarT.setEnabled(true);

}

}else if(rb90T.isSelected()){

    anguloT = 90;

    campoEsbeltezT.setText("");

    bVerificarT.setEnabled(true);

    double ft90dt = kmod * (0.7*fwkn/fwt);

    DecimalFormat df = new DecimalFormat("0.####");

    String sft90dt = df.format(ft90dt);

    campoft90d.setText(sft90dt);

    String sndT = campoNdT.getText().replaceAll( ",","." );

    double ndT = Double.parseDouble(sndT);

    areaT = (ndT/ft90dt);

    if(areaT < 50){
```

```
        areaT = 50;
    }

    String sareaT = df.format(areaT);
    campoAreaMinT.setText(sareaT);

    String sAreaMin = campoAreaMinT.getText().replaceAll( ",", "." );
    double areaMin = Double.parseDouble(sAreaMin);

    // Achar seções

    listaSecoesT.removeAllItems();

    ArrayList<Medidas> maioresQueMin = new ArrayList<>();

    if(areaMin > 900){

        listaSecoesT.addItem("Área mínima acima das seções utilizadas, redefine
o carregamento.");

        bVerificarT.setEnabled(false);
    }else{

        //Percorrer ArrayList e grava valores acima da área minima

        for(Medidas total : varMedidas) {
```

```
        if(total.getArea() >= areaMin){  
            maioresQueMin.add(total);  
        }  
    }  
  
    // Percorre novo vetor e adiciona as opções  
    for(Medidas maiores : maioresQueMin){  
        listaSecoesT.addItem("Seção: " + maiores.getBase() + "cm x " +  
maiores.getAltura() + "cm. Com área = " + maiores.getArea() + "cm².");  
    }  
  
    bVerificarT.setEnabled(true);  
  
    }  
}else if(rbOutroT.isSelected()){  
    campoEsbeltezT.setText("");  
  
    if(campoOutroT.getText().equals("")){  
        JOptionPane.showMessageDialog(rootPane, "Determine o ângulo de  
inclinação.");  
        listaSecoesT.setEnabled(false);  
    }else{
```

```

String sangulo = campoOutroT.getText().replaceAll( ",", "." );

double a = Double.parseDouble(sangulo);

anguloT = a;

double angulo = Math.toRadians(a);

double ft0dt = kmod * (0.7*fwkp/fwt);

double ft90dt = kmod * (0.7*fwkn/fwt);

double ftadt = (ft0dt * ft90dt)/((ft0dt * (Math.sin(angulo) *
Math.sin(angulo)))+(ft90dt * (Math.cos(angulo) * Math.cos(angulo))));

DecimalFormat df = new DecimalFormat("0.####");

String sftadt = df.format(ftadt);

campoftad.setText(sftadt);

String sndT = campoNdT.getText().replaceAll( ",", "." );

double ndT = Double.parseDouble(sndT);

areaT = (ndT/ftadt);

if(areaT < 50){
    areaT = 50;
}

String sareaT = df.format(areaT);

```

```
campoAreaMinT.setText(sareaT);

String sAreaMin = campoAreaMinT.getText().replaceAll( ",", "." );

double areaMin = Double.parseDouble(sAreaMin);

// Achar seções

listaSecoesT.removeAllItems();

ArrayList<Medidas> maioresQueMin = new ArrayList<>();

if(areaMin > 900){

    listaSecoesT.addItem("Área mínima acima das seções utilizadas,
redefine o carregamento.");

    bVerificarT.setEnabled(false);

}else{

    //Percorrer ArrayList e grava valores acima da área minima

    for(Medidas total : varMedidas) {

        if(total.getArea() >= areaMin){

            maioresQueMin.add(total);

        }

    }

}
```

```
    }

    // Percorre novo vetor e adiciona as opções

    for(Medidas maiores : maioresQueMin){

        listaSecoesT.addItem("Seção: " + maiores.getBase() + "cm x " +
maiores.getAltura() + "cm. Com área = " + maiores.getArea() + "cm².");

    }

    bVerificarT.setEnabled(true);

}

}

}

listaSecoesT.setEnabled(true);

}

private void rbOutroTActionPerformed(java.awt.event.ActionEvent evt) {

    if(rbOutroT.isSelected()){

        campoOutroT.setEnabled(true);

        bDimensionarT.setEnabled(true);

        campoEsbeltezT.setText("");

    }

}
```

```
    campoAreaMinT.setText("");

    campoft90d.setText("");

    campoft0d.setText("");

    campoOutroT.setText("");

    listaSecoesT.setEnabled(false);

    listaSecoesT.removeAllItems();

    bVerificarT.setEnabled(false);

    brTracao.setEnabled(false);

    painelAbas.setEnabledAt(8, false);

}

}

private void rb90TActionPerformed(java.awt.event.ActionEvent evt) {

    if(rb90T.isSelected()){

        campoOutroT.setEnabled(false);

        bDimensionarT.setEnabled(true);

        campoEsbeltezT.setText("");

        campoAreaMinT.setText("");

        campoft0d.setText("");

        campoftad.setText("");

        campoOutroT.setText("");

        listaSecoesT.setEnabled(false);
```

```
        listaSecoesT.removeAllItems();

        bVerificarT.setEnabled(false);

        brTracao.setEnabled(false);

        painelAbas.setEnabledAt(8, false);
    }
}

private void rb0TActionPerformed(java.awt.event.ActionEvent evt) {

    if(rb0T.isSelected()){

        campoOutroT.setEnabled(false);

        bDimensionarT.setEnabled(true);

        campoEsbeltezT.setText("");

        campoAreaMinT.setText("");

        campoft90d.setText("");

        campoftad.setText("");

        campoOutroT.setText("");

        listaSecoesT.setEnabled(false);

        listaSecoesT.removeAllItems();

        bVerificarT.setEnabled(false);

        brTracao.setEnabled(false);

        painelAbas.setEnabledAt(8, false);
    }
}
```



```
}  
  
private void bDimensionarFlexaoActionPerformed(java.awt.event.ActionEvent evt)  
{  
  
    listaAlturaF.removeAllItems();  
  
    alturaF = 0;  
  
    campoInstabilidadeL.setText("");  
  
    painelAbas.setEnabledAt(8, false);  
  
    areaTxtResultados.setText("");  
  
  
    DecimalFormat df = new DecimalFormat("0.####");  
  
    String sMd = campoMdF.getText().replaceAll( ",", "." );  
  
    double Md = Double.parseDouble(sMd);  
  
  
    String sftd = campoFtdF.getText().replaceAll( ",", "." );  
  
    ftd = Double.parseDouble(sftd);  
  
  
    String sfcd = campoFcdF.getText().replaceAll( ",", "." );  
  
    fcd = Double.parseDouble(sfcd);
```

```
if(listaBaseF.getSelectedltem().equals("5 cm")){  
    baseF = 5;  
}else if(listaBaseF.getSelectedltem().equals("7.5 cm")){  
    baseF = 7.5;  
}else if(listaBaseF.getSelectedltem().equals("10 cm")){  
    baseF = 10;  
}else if(listaBaseF.getSelectedltem().equals("12.5 cm")){  
    baseF = 12.5;  
}else if(listaBaseF.getSelectedltem().equals("15 cm")){  
    baseF = 15;  
}else if(listaBaseF.getSelectedltem().equals("17.5 cm")){  
    baseF = 17.5;  
}else if(listaBaseF.getSelectedltem().equals("20 cm")){  
    baseF = 20;  
}else if(listaBaseF.getSelectedltem().equals("22.5 cm")){  
    baseF = 22.5;  
}else if(listaBaseF.getSelectedltem().equals("25 cm")){  
    baseF = 25;  
}else if(listaBaseF.getSelectedltem().equals("27.5 cm")){  
    baseF = 27.5;  
}else if(listaBaseF.getSelectedltem().equals("30 cm")){  
    baseF = 30;
```

```
}

// Se ftd for menor
if(fcd > ftd){
    try{

        //Altura mínima
        alturaF = Math.sqrt((6*Md)/(baseF*ftd));

        String sAltF = df.format(alturaF);

        if (alturaF <= 30){
            campoAlturaMinF.setText(sAltF);

            //Adicionar alturas na lista
            for(int i = 0; i < 6; i++){
                if(alturas[i] > alturaF){
                    listaAlturaF.addItem(alturas[i] + " cm");
                }
            }

            bVerificaF.setEnabled(true);

            listaAlturaF.setEnabled(true);
```



```
// Se fcd for menor

}else if(fcd < ftd){

    try{

        //Altura mínima

        alturaF = Math.sqrt((6*Md)/(baseF*fcd));

        String sAltF = df.format(alturaF);

        if (alturaF <= 30){

            campoAlturaMinF.setText(sAltF);

            //Adicionar alturas na lista

            for(int i = 0; i < 6; i++){

                if(alturas[i] > alturaF){

                    listaAlturaF.addItem(alturas[i] + " cm");

                }

            }

            bVerificaF.setEnabled(true);

            listaAlturaF.setEnabled(true);
```

```
}else{

    if(baseF == 30){

        JOptionPane.showMessageDialog(rootPane,("Dimensionamento com
seções acima das comerciais. Rever carregamentos"));

        painelAbas.setSelectedIndex(2);

        painelCarregamentos.setSelectedIndex(0);

        painelAbas.setEnabledAt(3, false);

        painelAbas.setEnabledAt(6, false);

    }else{

        campoInstabilidadeL.setText("");

        campoAlturaMinF.setText("");

        JOptionPane.showMessageDialog(rootPane,("Dimensionamento com
altura acima das seções comerciais. Aumentar a base da seção."));

    }

}

}catch(Exception e){

}

}
```

```
}
```

```
private void bVerificaFActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    //Deletando anterior
```

```
    campoTravamento.setText("");
```

```
    bTravamento.setEnabled(false);
```

```
    painelAbas.setEnabledAt(8, false);
```

```
    areaTxtResultados.setText("");
```

```
    double b1;
```

```
    double b2;
```

```
    double b3;
```

```
    double Ec0ef = Ec0m * kmod;
```

```
    String sfcd = campoFcdF.getText().replaceAll( " , " , "." );
```

```
    double fcd = Double.parseDouble(sfcd);
```

```
    if(listaAlturaF.getSelectedItem().equals("5.0 cm")){
```

```
        alturaR = 5;
```

```
    }else if(listaAlturaF.getSelectedItem().equals("10.0 cm")){
```

```

    alturaR = 10;

}else if(listaAlturaF.getSelectedItem().equals("15.0 cm")){

    alturaR = 15;

}else if(listaAlturaF.getSelectedItem().equals("20.0 cm")){

    alturaR = 20;

}else if(listaAlturaF.getSelectedItem().equals("25.0 cm")){

    alturaR = 25;

}else if(listaAlturaF.getSelectedItem().equals("30.0 cm")){

    alturaR = 30;

}

// Dimensionamento de travamentos

b1 = (1/(0.26*Math.PI));

b2 = 4/1.4;

b3 = ((Math.pow((alturaR/baseF),(3/2))/(Math.pow(((alturaR/baseF)-
0.63),(1/2))))));

Bm = (b1*b2*b3);

L1 = ((baseF * Ec0ef)/(Bm * fcd));

if(comprimentoH <= L1){

    campoInstabilidadeL.setText("A barra não precisa de travamentos");

    vt = false;

```



```
        campoTravamento.setText("");

        bpFlexao.setEnabled(true);

    }else{

        campoInstabilidadeL.setText("A barra precisa de travamentos.");

        vt = true;

        bTravamento.setEnabled(true);

    }

}

private void listaBaseFActionPerformed(java.awt.event.ActionEvent evt) {

    campoAlturaMinF.setText("");

    campoInstabilidadeL.setText("");

    campoTravamento.setText("");

    listaAlturaF.removeAllItems();

    listaAlturaF.setEnabled(false);

    bVerificaF.setEnabled(false);

    bTravamento.setEnabled(false);

    bpFlexao.setEnabled(false);

    painelAbas.setEnabledAt(7, false);
```

```
painelAbas.setEnabledAt(8, false);

areaTxtResultados.setText("");

}

private void bTravamentoActionPerformed(java.awt.event.ActionEvent evt) {

    DecimalFormat df = new DecimalFormat("0.##");

    int x = (int)(comprimentoH / L1);

    double y = (comprimentoH/(x+1));

    String sY = df.format(y);

    campoTravamento.setText("É preciso colocar " + x + " travamentos ao longo da
viga com espaçamento de " + sY + "cm entre eles.");

    bpFlexao.setEnabled(true);

}

private void listaAlturaFActionPerformed(java.awt.event.ActionEvent evt) {

    campoTravamento.setText("");
```

```
campoInstabilidadeL.setText("");

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

painelAbas.setEnabledAt(7, false);

painelAbas.setEnabledAt(8, false);

areaTxtResultados.setText("");

}

private void bpFlexaoActionPerformed(java.awt.event.ActionEvent evt) {

    campoCondCl.setText("");

    campoAlturaCl.setText("");

    bRedimCl.setEnabled(false);

    DecimalFormat df = new DecimalFormat("0.##");

    String sAltura = df.format(alturaR);

    campoAlturaCl.setText(sAltura);

    String sBase = df.format(baseF);
```

```
campoBaseCl.setText(sBase);
```

```
String sFv0 = df.format(fv0);
```

```
campoFv0Cl.setText(sFv0);
```

```
String sKmod = df.format(kmod);
```

```
campoKmodCl.setText(sKmod);
```

```
painelAbas.setEnabledAt(7, true);
```

```
painelAbas.setSelectedIndex(7);
```

```
}
```

```
private void bVerificaClActionPerformed(java.awt.event.ActionEvent evt) {
```

```
double v1 = ((3/2)*(Vsd/(baseF*alturaR)));
```

```
double v2 = kmod * ((0.54 * fv0)/1.8);
```

```
if(v1 <= v2){
```

```
    campoCondCl.setText("Verificação OK!");
```

```
    brCisalhamento.setEnabled(true);
```

```
}else{
```

```
        campoCondCI.setText("Falha na verificação. Redimensionar seção");  
        bRedimCI.setEnabled(true);  
    }  
}
```

```
private void bRedimCIActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    DecimalFormat df = new DecimalFormat("0.##");
```

```
    if(alturaR <= 25){
```

```
        alturaCI = alturaR;
```

```
        while(alturaCI < 30){
```

```
            double v1 = ((3/2)*(Vsd/(baseF*alturaCI)));
```

```
            double v2 = kmod * ((0.54 * fv0)/1.8);
```

```
            if(v1 <= v2){
```

```
                String sAltura = df.format(alturaCI);
```



```
        painelAbas.setSelectedIndex(6);

    }else{

        JOptionPane.showMessageDialog(rootPane,("Dimensionamento com
seções acima das comerciais. Rever carregamentos"));

    }

}

}
```

```
private void brCompressaoActionPerformed(java.awt.event.ActionEvent evt) {

    if (lgBloqueioComp){

        JOptionPane.showMessageDialog(null, "Seção alterada, favor refazer a
verificação");

        return;

    }

    DecimalFormat df2 = new DecimalFormat("0.##");

    DecimalFormat df3 = new DecimalFormat("0.###");

    String sKmod = df3.format(kmod);

    String sNsd1 = df2.format(Nsd1);

    String sNsd2 = df2.format(Nsd2);

    String sAngulo = df2.format(anguloT);

    String sfv0 = df2.format(fv0*10);

    String sl = df3.format(l);
```

```
String sesb = df3.format(esb);

String sfc0 = df2.format(fc0k*10);

String sft0 = df2.format(fwkp*10);

String sft90 = df2.format(fwkn*10);

String sEc0 = df2.format(Ec0m*10);

String ang;

String rft;

String sEc0ef = df2.format(Ec0m*kmod);

String sFe = df3.format(Fe);

String sei = df3.format(ei);

String sea = df3.format(ea);

String se1 = df3.format(e1);

String sed = df3.format(ed);

String sMd = df3.format(Mdt);

String sTmd = df3.format(Tmd);

String sTnd = df3.format(Tnd);

String sverifica = df3.format(verifica);

String sexpK = df3.format(Math.pow(exp, k));

String sec = df3.format(ec);

if(rb0C.isSelected()){
```



```
    ang = "0";  
  
    rft = campofc0d.getText();  
  
}else if(rb90C.isSelected()){  
  
    ang = "90";  
  
    rft = campofc90d.getText();  
  
}else{  
  
    ang = "a";  
  
    rft = campofcad.getText();  
  
}
```

```
areaTxtResultados.setText("");
```

```
if(esb <= 40){  
  
    areaTxtResultados.setText  
  
        (" ---- Propriedades da peça ----"  
  
        + "\n\n Material escolhido: " + material  
  
        + "\n fc0 = " + sfc0 + " MPa"  
  
        + "\n ft0 = " + sft0 + " MPa"
```

+ "\n ft90 = " + sft90 + " MPa"

+ "\n fv = " + sfv0 + " MPa"

+ "\n Ec0 = " + sEc0 + " MPa"

+ "\n\n Kmod = " + kmod1 + " \* " + kmod2 + " \* " + kmod3

+ "\n Kmod = " + sKmod

+ "\n\n ---- Carregamentos e combinações ----"

+ "\n\n Carregamentos:"

+ "\n Ngk = " + Ng + " kN"

+ "\n Nqk = " + Nq + " kN"

+ "\n\n Combinações:"

+ "\n Nsd = ( $\gamma_g * Ng$ ) + ( $\gamma_q * Nq$ )"

+ "\n Nsd1 = (" + YgV1 + " \* " + Ng + ") + (" + YqV1 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd1 + "kN"

+ "\n Nsd2 = (" + YgV2 + " \* " + Ng + ") + (" + YqV2 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd2 + "kN"

+ "\n\n ---- Dados do dimensionamento ----"

+ "\n\n Nd = " + campoNdC.getText() + " kN"

+ "\n Kmod = " + sKmod

+ "\n Ywc = 1.4"

+ "\n Ângulo de inclinação = " + sAngulo + "°"

+ "\n fc" + ang + ",d = " + rft + " kN/cm<sup>2</sup>"

+ "\n\n Dados geométricos:"

+ "\n Comprimento = " + comprimentoV + "cm"

+ "\n Base = " + base + "cm"

+ "\n Altura = " + altura + "cm"

+ "\n Área = " + base \* altura + "cm<sup>2</sup>"

+ "\n\n Momento de inércia:"

+ "\n I = b \* h<sup>3</sup> / 12"

+ "\n I = " + base + " \* " + altura + "<sup>3</sup> / 12"

+ "\n I = " + sl + " cm<sup>4</sup>"

+ "\n\n ---- Verificações ----"

+ "\n\n Verificação de esbeltez:"

+ "\n  $\lambda \leq 170$ "

+ "\n " + sesb + "  $\leq 170$ "

+ "\n Verificação OK!"

```

+ "\n\n Verificação de estabilidade:"

+ "\n  $\lambda \leq 40$ "

+ "\n " + sesb + "  $\leq 40$ "

+ "\n Verificação OK!"

);

}else if (esb > 40 && esb <= 80){

    areaTxtResultados.setText

        (" ---- Propriedades da peça ----"

+ "\n\n Material escolhido: " + material

+ "\n fc0 = " + sfc0 + " MPa"

+ "\n ft0 = " + sft0 + " MPa"

+ "\n ft90 = " + sft90 + " MPa"

+ "\n fv = " + sfv0 + " MPa"

+ "\n Ec0 = " + sEc0 + " MPa"

+ "\n\n Kmod = " + kmod1 + " * " + kmod2 + " * " + kmod3

+ "\n Kmod = " + sKmod

+ "\n\n ---- Carregamentos e combinações ----"

```

+ "\n\n Carregamentos:"

+ "\n Ngk = " + Ng + " kN"

+ "\n Nqk = " + Nq + " kN"

+ "\n\n Combinações:"

+ "\n Nsd = ( $\gamma_g * Ng$ ) + ( $\gamma_q * Nq$ )"

+ "\n Nsd1 = (" + YgV1 + " \* " + Ng + ") + (" + YqV1 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd1 + "kN"

+ "\n Nsd2 = (" + YgV2 + " \* " + Ng + ") + (" + YqV2 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd2 + "kN"

+ "\n\n ---- Dados do dimensionamento ----"

+ "\n\n Nd = " + campoNdC.getText() + " kN"

+ "\n Kmod = " + sKmod

+ "\n Ywc = 1.4"

+ "\n Ângulo de inclinação = " + sAngulo + "°"

+ "\n fc" + ang + ",d = " + rft + " kN/cm<sup>2</sup>"

+ "\n\n Dados geométricos:"

+ "\n Comprimento = " + comprimentoV + "cm"

+ "\n Base = " + base + "cm"

+ "\n Altura = " + altura + "cm"

+ "\n Área = " + base \* altura + "cm<sup>2</sup>"

+ "\n\n Momento de inércia:"

+ "\n I = b \* h<sup>3</sup> / 12"

+ "\n I = " + base + " \* " + altura + "<sup>3</sup> / 12"

+ "\n I = " + sl + " cm<sup>4</sup>"

+ "\n\n ---- Verificações ----"

+ "\n\n Verificação de esbeltez:"

+ "\n  $\lambda \leq 170$ "

+ "\n " + sesb + "  $\leq 170$ "

+ "\n Verificação OK!"

+ "\n\n Verificação de estabilidade:"

+ "\n  $40 < \lambda \leq 80$ "

+ "\n  $40 < " + sesb + " \leq 80$ "

+ "\n\n  $E_{c0,ef} = k_{mod} * E_{c0,m}$ "

+ "\n  $E_{c0,ef} = " + sK_{mod} + " * " + sE_{c0}$

+ "\n  $E_{c0,ef} = " + sE_{c0ef}$

+ "\n\n  $F_e = (\pi^2 * E_{c0,ef} * I) / (L_0^2)$ "

+ "\n  $F_e = (\pi^2 * " + sE_{c0ef} + " * " + sl + ") / (" + comprimentoV + ")^2"$

+ "\n  $F_e = " + sF_e$

```

+ "\n\n ei = h/30"

+ "\n ei = " + altura + "/30"

+ "\n ei = " + sei

+ "\n\n ea = L0/300"

+ "\n ea = " + comprimentoV + "/300"

+ "\n ea = " + sea

+ "\n\n e1 = ei + ea"

+ "\n e1 = " + sei + " + " + sea

+ "\n e1 = " + se1

+ "\n\n ed = e1 * (Fe/(Fe - Nd))"

+ "\n ed = " + se1 + " * (" + sFe + "/" + sFe + " - " + campoNdC.getText() +
"))"

+ "\n ed = " + sed

+ "\n\n Md = Nd * ed"

+ "\n Md = " + campoNdC.getText() + " * " + sed

+ "\n Md = " + sMd

+ "\n\n  $\sigma_{Md} = (Md * y)/I$ "

+ "\n  $\sigma_{Md} = (" + sMd + " * " + y + ")/" + sl$ "

+ "\n  $\sigma_{Md} = " + sTmd$ "

+ "\n\n  $\sigma_{Nd} = Nd/A$ "

+ "\n  $\sigma_{Nd} = " + campoNdC.getText() + " / " + area$ "

+ "\n  $\sigma_{Nd} = " + sTnd$ 

```

```

+ "\n\n Verificação:"
+ "\n ( $\sigma_{Nd}/f_{cd}$ ) + ( $\sigma_{Md}/f_{cd}$ )  $\leq$  1"
+ "\n (" + sTnd + "/" + rft + ") + (" + sTmd + "/" + rft + ")  $\leq$  1"
+ "\n " + sverifica + "  $\leq$  1"
+ "\n Verificação OK!"
);
}else if (esb > 80 && esb <= 140){
    areaTxtResultados.setText
    (" ---- Propriedades da peça ----"

+ "\n\n Material escolhido: " + material
+ "\n fc0 = " + sfc0 + " MPa"
+ "\n ft0 = " + sft0 + " MPa"
+ "\n ft90 = " + sft90 + " MPa"
+ "\n fv = " + sfv0 + " MPa"
+ "\n Ec0 = " + sEc0 + " MPa"

+ "\n\n Kmod = " + kmod1 + " * " + kmod2 + " * " + kmod3
+ "\n Kmod = " + sKmod

+ "\n\n ---- Carregamentos e combinações ----"

```



+ "\n\n Carregamentos:"

+ "\n Ngk = " + Ng + " kN"

+ "\n Nqk = " + Nq + " kN"

+ "\n\n Combinações:"

+ "\n Nsd = ( $\gamma_g * Ng$ ) + ( $\gamma_q * Nq$ )"

+ "\n Nsd1 = (" + YgV1 + " \* " + Ng + ") + (" + YqV1 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd1 + "kN"

+ "\n Nsd2 = (" + YgV2 + " \* " + Ng + ") + (" + YqV2 + " \* " + Nq + ")"

+ "\n Nsd1 = " + sNsd2 + "kN"

+ "\n\n ---- Dados do dimensionamento ----"

+ "\n\n Nd = " + campoNdC.getText() + " kN"

+ "\n Kmod = " + sKmod

+ "\n Ywc = 1.4"

+ "\n Ângulo de inclinação = " + sAngulo + "°"

+ "\n fc" + ang + ",d = " + rft + " kN/cm<sup>2</sup>"

+ "\n\n Dados geométricos:"

+ "\n Comprimento = " + comprimentoV + "cm"

+ "\n Base = " + base + "cm"

+ "\n Altura = " + altura + "cm"

+ "\n Área = " + base \* altura + "cm<sup>2</sup>"

+ "\n\n Momento de inércia:"

+ "\n I = b \* h<sup>3</sup> / 12"

+ "\n I = " + base + " \* " + altura + "<sup>3</sup> / 12"

+ "\n I = " + sl + " cm<sup>4</sup>"

+ "\n\n ---- Verificações ----"

+ "\n\n Verificação de esbeltez:"

+ "\n  $\lambda \leq 170$ "

+ "\n " + sesb + "  $\leq 170$ "

+ "\n Verificação OK!"

+ "\n\n Verificação de estabilidade:"

+ "\n  $40 < \lambda \leq 80$ "

+ "\n  $40 < " + sesb + " \leq 80$ "

+ "\n\n  $E_{c0,ef} = k_{mod} * E_{c0,m}$ "

+ "\n  $E_{c0,ef} = " + sK_{mod} + " * " + sE_{c0}$

+ "\n  $E_{c0,ef} = " + sE_{c0ef}$

+ "\n\n  $F_e = (\pi^2 * E_{c0,ef} * I) / (L_0^2)$ "

+ "\n  $F_e = (\pi^2 * " + sE_{c0ef} + " * " + sl + ") / (" + comprimentoV + ")^2"$

+ "\n  $F_e = " + sF_e$

```

+ "\n\n ei = h/30"

+ "\n ei = " + altura + "/30"

+ "\n ei = " + sei

+ "\n\n ea = L0/300"

+ "\n ea = " + comprimentoV + "/300"

+ "\n ea = " + sea

+ "\n\n ec = ((h/30) + ea) * (expK - 1)"

+ "\n ec = ((" + altura + "/30) + " + sea + ") * (" + sexpK + " - 1)"

+ "\n ec = " + sec

+ "\n\n e1 = ei + ea"

+ "\n e1 = " + sei + " + " + sea

+ "\n e1 = " + se1

+ "\n\n ed = e1 * (Fe/(Fe - Nd))"

+ "\n ed = " + se1 + " * (" + sFe + "/" + sFe + " - " + campoNdC.getText() +
"))"

+ "\n ed = " + sed

+ "\n\n Md = Nd * ed"

+ "\n Md = " + campoNdC.getText() + " * " + sed

+ "\n Md = " + sMd

+ "\n\n  $\sigma_{Md} = (Md * y)/I$ "

+ "\n  $\sigma_{Md} = (" + sMd + " * " + y + ")/" + sl$ "

+ "\n  $\sigma_{Md} = " + sTmd$ "

+ "\n\n  $\sigma_{Nd} = Nd/A$ "

```

```
+ "\n σNd = " + campoNdC.getText() + " / " + area
+ "\n σNd = " + sTnd

+ "\n\n Verificação:"
+ "\n (σNd/fcd) + (σMd/fcd) ≤ 1"
+ "\n (" + sTnd + "/" + rft + ") + (" + sTmd + "/" + rft + ") ≤ 1"
+ "\n " + sverifica + " ≤ 1"
+ "\n Verificação OK!"

);
}
```

```
painelAbas.setSelectedIndex(8);
```

```
painelAbas.setEnabledAt(8, true);
```

```
}
```

```
private void brCisalhamentoActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    DecimalFormat df2 = new DecimalFormat("0.##");
```

```
    DecimalFormat df3 = new DecimalFormat("0.###");
```

```
    String sfd;
```

```
    double fd;
```

```
    double Td;
```

```
    double parclnst;
```

```
    double parc1Cl;
```

```
    double parc2Cl;
```

```
    parc1Cl = ((3*Vsd)/(2*baseF*alturaR));
```

```
    parc2Cl = (kmod*((0.54*fv0)/1.8));
```

```
    parclnst = (baseF*Ec0m*kmod)/(Bm*fcd);
```

```
    int x = (int)(comprimentoH / L1);
```

```
    double yT = (comprimentoH/(x+1));
```

```
        /**
```

```
        double lfx = (baseF*Math.pow(alturaR, 3))/12;
```

```
        double lfy = (alturaR*Math.pow(baseF, 3))/12;
```

```
if(lfx > lfy){

    lf = lfy;

    double aux = alturaR;

    alturaR = baseF;

    baseF = aux;

}

}

**/

lf = (baseF*Math.pow(alturaR, 3))/12;

y = alturaR/2;

Td = Msd/(lf/y);

if(fcd > ftd){

    sfd = "t";

    fd = ftd;

}

}

else{

    sfd = "c";

    fd = fcd;
```

```
}  
  
String sKmod = df3.format(kmod);  
  
String sMsd1 = df2.format(Msd1);  
  
String sMsd2 = df2.format(Msd2);  
  
String sVsd1 = df2.format(Vsd1);  
  
String sVsd2 = df2.format(Vsd2);  
  
String sfv0 = df2.format(fv0*10);  
  
String ssfv0 = df2.format(fv0);  
  
String sEc0ef = df2.format(Ec0m*kmod);  
  
String sfc0 = df2.format(fc0k*10);  
  
String sft0 = df2.format(fwkp*10);  
  
String sft90 = df2.format(fwkn*10);  
  
String sEc0 = df2.format(Ec0m*10);  
  
String sBm = df2.format(Bm);  
  
String sMg = df3.format(Mg);  
  
String sMq = df3.format(Mq);  
  
String sVg = df3.format(Vg);  
  
String sVq = df3.format(Vq);  
  
String sTd = df3.format(Td);  
  
String sFd = df3.format(fd);  
  
String sVsd = df3.format(Vsd);
```

```
String sYT = df2.format(yT);  
  
String sL1 = df2.format(L1);  
  
String sParc1CI = df3.format(parc1CI);  
  
String sParc2CI = df3.format(parc2CI);  
  
String sParcInst = df2.format(parcInst);  
  
String funcao;
```

```
    if(listaCargaVariavelH.getSelectedItem().equals("Variações uniformes de  
temperatura em relação à média anual local")){
```

```
        funcao = "da temperatura";
```

```
    }else if(listaCargaVariavelH.getSelectedItem().equals("Pressão dinâmica do  
vento")){
```

```
        funcao = "do vento";
```

```
    }else if(listaCargaVariavelH.getSelectedItem().equals("Carga acidental em  
edifícios sem elevadas concentrações de pessoas")){
```

```
        funcao = "da carga acidental sem concentração de pessoas";
```

```
    }else if(listaCargaVariavelH.getSelectedItem().equals("Carga acidental em  
edifícios com elevadas concentrações de pessoas")){
```

```
        funcao = "da carga acidental com concentração de pessoas";
```

```
    }else{
```

```
        funcao = "do uso e ocupação";
```

```
    }
```



```
areaTxtResultados.setText("");

if(vt == false){

    areaTxtResultados.setText

        (" ---- Propriedades da peça ----"

+ "\n\n Material escolhido: " + material

+ "\n fc0 = " + sfc0 + " MPa"

+ "\n ft0 = " + sft0 + " MPa"

+ "\n ft90 = " + sft90 + " MPa"

+ "\n fv = " + sfv0 + " MPa"

+ "\n Ec0 = " + sEc0 + " MPa"

+ "\n\n Kmod = " + kmod1 + " * " + kmod2 + " * " + kmod3

+ "\n Kmod = " + sKmod

+ "\n\n ---- Carregamentos e combinações ----"

+ "\n\n Carregamentos:"
```

+ "\n gk = " + g + " kN/m (carga permanente)"

+ "\n Mg = " + sMg + " kN.cm"

+ "\n\n Vg = " + sVg + " kN)"

+ "\n qk = " + q + " kN/m (carga variável em função " + funcao + ")"

+ "\n Mq = " + sMq + " kN/m"

+ "\n Vq = " + sVq + " kN/m"

+ "\n\n Combinações:"

+ "\n Msd = ( $\gamma_g * Mg$ ) + ( $\gamma_q * Mq$ )"

+ "\n Msd1 = (" + YgH1 + " \* " + sMg + ") + (" + YqH1 + " \* " + sMq + ")"

+ "\n Msd1 = " + sMsd1 + "kN.cm"

+ "\n Msd2 = (" + YgH2 + " \* " + sMg + ") + (" + YqH2 + " \* " + sMq + ")"

+ "\n Msd1 = " + sMsd2 + "kN.cm"

+ "\n\n Vsd = ( $\gamma_g * Vg$ ) + ( $\gamma_q * Vq$ )"

+ "\n Vsd1 = (" + YgH1 + " \* " + sVg + ") + (" + YqH1 + " \* " + sVq + ")"

+ "\n Vsd1 = " + sVsd1 + "kN"

+ "\n Vsd2 = (" + YgH2 + " \* " + sVg + ") + (" + YqH2 + " \* " + sVq + ")"

+ "\n Vsd1 = " + sVsd2 + "kN"

+ "\n\n ---- Dados do dimensionamento a flexão ----"

+ "\n\n Md = " + campoMdF.getText() + " kN.cm"

+ "\n Kmod = " + sKmod

+ "\n ft,d = " + campoFtdF.getText() + " kN.cm<sup>2</sup>"

+ "\n fc,d = " + campoFcdF.getText() + " kN.cm<sup>2</sup>"

+ "\n\n Dados geométricos:"

+ "\n Comprimento = " + comprimentoH + "cm"

+ "\n Base = " + baseF + "cm"

+ "\n Altura = " + alturaR + "cm"

+ "\n Área = " + baseF \* alturaR + "cm<sup>2</sup>"

+ "\n\n ---- Verificações ----"

+ "\n\n Verificação da condição de segurança:"

+ "\n  $\sigma$  + sfd + ",d ≤ f" + sfd + "d"

+ "\n " + sTd + " ≤ " + sFd

+ "\n Verificação OK!"

+ "\n\n Verificação da instabilidade lateral:"

+ "\n  $L1 \leq (b * Eco,ef) / (\beta_m * fc_{0,d})$ "

```

+ fcd
+ "\n " + comprimentoH + " ≤ (" + baseF + " * " + sEc0ef + ")/(" + sBm + " * "

+ "\n " + comprimentoH + " ≤ " + sParcInst

+ "\n A peça não precisa de travamentos"

+ "\n\n Verificação de cisalhamento:"

+ "\n ((3 * Vd)/(2 * b * h) ≤ kmod * ((0.54 * fv0)/1.8)"

+ "\n ((3 * " + sVsd + ")/(2 * " + baseF + " * " + alturaR + ") ≤ " + sKmod + " *
((0.54 * " + ssv0 + ")/1.8)"

+ "\n " + sParc1Cl + " ≤ " + sParc2Cl

+ "\n Verificação OK!"

);

}else{

areaTxtResultados.setText

(" ---- Propriedades da peça ----"

+ "\n\n Material escolhido: " + material

+ "\n fc0 = " + sfc0 + " MPa"

+ "\n ft0 = " + sft0 + " MPa"

+ "\n ft90 = " + sft90 + " MPa"

+ "\n fv = " + sfv0 + " MPa"

+ "\n Ec0 = " + sEc0 + " MPa"

```

+ "\n\n Kmod = " + kmod1 + " \* " + kmod2 + " \* " + kmod3

+ "\n Kmod = " + sKmod

+ "\n\n ---- Carregamentos e combinações ----"

+ "\n\n Carregamentos:"

+ "\n gk = " + g + " kN/m (carga permanente)"

+ "\n Mg = " + sMg + " kN.cm"

+ "\n\n Vg = " + sVg + " kN)"

+ "\n qk = " + q + " kN/m (carga variável em função " + funcao + ")"

+ "\n Mq = " + sMq + " kN/m"

+ "\n Vq = " + sVq + " kN/m"

+ "\n\n Combinações:"

+ "\n Msd = ( $\gamma_g * Mg$ ) + ( $\gamma_q * Mq$ )"

+ "\n Msd1 = (" + YgH1 + " \* " + sMg + ") + (" + YqH1 + " \* " + sMq + ")"

+ "\n Msd1 = " + sMsd1 + "kN.cm"

+ "\n Msd2 = (" + YgH2 + " \* " + sMg + ") + (" + YqH2 + " \* " + sMq + ")"

+ "\n Msd1 = " + sMsd2 + "kN.cm"

+ "\n\n Vsd = ( $\gamma_g * V_g$ ) + ( $\gamma_q * V_q$ )"

+ "\n Vsd1 = (" +  $Y_{gH1}$  + " \* " +  $sV_g$  + ") + (" +  $Y_{qH1}$  + " \* " +  $sV_q$  + ")"

+ "\n Vsd1 = " +  $sV_{sd1}$  + "kN"

+ "\n Vsd2 = (" +  $Y_{gH2}$  + " \* " +  $sV_g$  + ") + (" +  $Y_{qH2}$  + " \* " +  $sV_q$  + ")"

+ "\n Vsd1 = " +  $sV_{sd2}$  + "kN"

+ "\n\n ---- Dados do dimensionamento a flexão ----"

+ "\n\n Md = " + `campoMdF.getText()` + " kN.cm"

+ "\n Kmod = " +  $sK_{mod}$

+ "\n  $f_{t,d}$  = " + `campoFtdF.getText()` + " kN.cm<sup>2</sup>"

+ "\n  $f_{c,d}$  = " + `campoFcdF.getText()` + " kN.cm<sup>2</sup>"

+ "\n\n Dados geométricos:"

+ "\n Comprimento = " +  $comprimentoH$  + "cm"

+ "\n Base = " +  $baseF$  + "cm"

+ "\n Altura = " +  $alturaR$  + "cm"

+ "\n Área = " +  $baseF * alturaR$  + "cm<sup>2</sup>"

+ "\n\n ---- Verificações ----"

+ "\n\n Verificação da condição de segurança:"

+ "\n  $\sigma + s_{fd} + \sigma_{d} \leq f + s_{fd} + \sigma_{d}$ "

+ "\n " +  $s_{Td} + \sigma \leq \sigma + s_{Fd}$

+ "\n Verificação OK!"

+ "\n\n Verificação da instabilidade lateral:"

+ "\n  $L1 \leq (b * E_{co,ef}) / (\beta_m * f_{c0,d})$ "

+ fcd

+ "\n " + comprimentoH + "  $\leq (\sigma + baseF + \sigma * \sigma + s_{Ec0ef} + \sigma) / (\sigma + s_{Bm} + \sigma * \sigma)$ "

+ "\n " + comprimentoH + "  $\leq \sigma + s_{Parclnst}$

+ "\n A peça precisa de travamentos"

+ "\n\n Dimensionamento dos travamentos:"

+ "\n Quantidade de travamentos =  $(L0/L1) - 1$  "

+ "\n Quantidade de travamentos =  $(\sigma + comprimentoH + \sigma) / (\sigma + s_{L1} + \sigma) - 1$ "

mais)"

+ "\n Quantidade de travamentos = " + x + " (sempre arredondar para

+1 "

+ "\n Espaçamento entre os travamentos =  $L0 / \text{Quantidade de travamentos}$

+1 "

+ "\n Espaçamento entre os travamentos = " + comprimentoH + " / " + x + "

+ "\n Espaçamento entre os travamentos = " +  $s_{YT} + \text{"cm"}$

+ "\n É preciso colocar " + x + " travamentos ao longo da viga com espaçamento de " +  $s_{YT} + \text{"cm"}$  entre eles."

```

+ "\n\n Verificação de cisalhamento:"

+ "\n ((3 * Vd)/(2 * b * h) ≤ kmod * ((0.54 * fv0)/1.8)"

+ "\n ((3 * " + sVsd + ")/(2 * " + baseF + " * " + alturaR + ") ≤ " + sKmod + " *
((0.54 * " + ssfv0 + ")/1.8)"

+ "\n " + sParc1Cl + " ≤ " + sParc2Cl

+ "\n Verificação OK!"

);

}

```

```

painelAbas.setSelectedIndex(8);

painelAbas.setEnabledAt(8, true);

```

```

}

```

```

private void rbPequenaHFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioHor = true;

```



```
painelAbas.setEnabledAt(3, false);  
painelAbas.setEnabledAt(6, false);  
painelAbas.setEnabledAt(7, false);  
painelAbas.setEnabledAt(8, false);
```

```
//Flexão
```

```
campoAlturaMinF.setText("");  
campoInstabilidadeL.setText("");  
campoTravamento.setText("");  
listaAlturaF.removeAllItems();  
bVerificaF.setEnabled(false);  
bTravamento.setEnabled(false);  
bpFlexao.setEnabled(false);
```

```
//Cisalhamento
```

```
campoCondCI.setText("");  
campoRedimCI.setText("");  
bRedimCI.setEnabled(false);  
brCisalhamento.setEnabled(false);
```

```
//Resultados

areaTxtResultados.setText("");

}

private void rbGrandeHFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioHor = true;

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);
```

```
bpFlexao.setEnabled(false);
```

```
//Cisalhamento
```

```
campoCondCI.setText("");
```

```
campoRedimCI.setText("");
```

```
bRedimCI.setEnabled(false);
```

```
brCisalhamento.setEnabled(false);
```

```
//Resultados
```

```
areaTxtResultados.setText("");
```

```
}
```

```
private void listaCargaVariavelHFocusGained(java.awt.event.FocusEvent evt) {
```

```
    lgBloqueioHor = true;
```

```
    painelAbas.setEnabledAt(3, false);
```

```
    painelAbas.setEnabledAt(6, false);
```

```
    painelAbas.setEnabledAt(7, false);
```

```
    painelAbas.setEnabledAt(8, false);
```

```
//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

brCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void campoCargaQQHFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioHor = true;
```

```
painelAbas.setEnabledAt(3, false);  
painelAbas.setEnabledAt(6, false);  
painelAbas.setEnabledAt(7, false);  
painelAbas.setEnabledAt(8, false);
```

```
//Flexão
```

```
campoAlturaMinF.setText("");  
campoInstabilidadeL.setText("");  
campoTravamento.setText("");  
listaAlturaF.removeAllItems();  
bVerificaF.setEnabled(false);  
bTravamento.setEnabled(false);  
bpFlexao.setEnabled(false);
```

```
//Cisalhamento
```

```
campoCondCI.setText("");  
campoRedimCI.setText("");  
bRedimCI.setEnabled(false);  
brCisalhamento.setEnabled(false);
```

```
//Resultados

areaTxtResultados.setText("");

}

private void campoCargaGGHFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioHor = true;

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

    //Flexão

    campoAlturaMinF.setText("");

    campoInstabilidadeL.setText("");

    campoTravamento.setText("");

    listaAlturaF.removeAllItems();

    bVerificaF.setEnabled(false);

    bTravamento.setEnabled(false);

    bpFlexao.setEnabled(false);
```

```
//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

bCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void rbGrandeVFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioVer = true;

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(8, false);

//tração
```

```
campoft0d.setText("");  
  
campoft90d.setText("");  
  
campoftad.setText("");  
  
campoAreaMinT.setText("");  
  
campoEsbeltezT.setText("");  
  
listaSecoesT.removeAllItems();  
  
  
//compressão  
  
campofc0d.setText("");  
  
campofc90d.setText("");  
  
campofcad.setText("");  
  
campoAreaMinC.setText("");  
  
campoEsbeltezC.setText("");  
  
campoEstabilidadeC.setText("");  
  
listaSecoesC.removeAllItems();  
  
  
//Resultados  
  
areaTxtResultados.setText("");  
  
}  
  
private void rbPequenaVFocusGained(java.awt.event.FocusEvent evt) {  
  
    lgBloqueioVer = true;
```



```
painelAbas.setEnabledAt(3, false);  
painelAbas.setEnabledAt(4, false);  
painelAbas.setEnabledAt(5, false);  
painelAbas.setEnabledAt(8, false);
```

```
//tração
```

```
campoft0d.setText("");  
campoft90d.setText("");  
campoftad.setText("");  
campoAreaMinT.setText("");  
campoEsbeltezT.setText("");  
listaSecoesT.removeAllItems();
```

```
//compressão
```

```
campofc0d.setText("");  
campofc90d.setText("");  
campofcad.setText("");  
campoAreaMinC.setText("");  
campoEsbeltezC.setText("");
```

```
campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Resultados

areaTxtResultados.setText("");

}

private void campoCargaNgVFFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioVer = true;

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(8, false);

    //tração

    campoft0d.setText("");

    campoft90d.setText("");

    campoftad.setText("");

    campoAreaMinT.setText("");
```

```
campoEsbeltezT.setText("");

listaSecoesT.removeAllItems();

//compressão

campofc0d.setText("");

campofc90d.setText("");

campofcad.setText("");

campoAreaMinC.setText("");

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Resultados

areaTxtResultados.setText("");

}

private void listaCargaVariavelVFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioVer = true;

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(4, false);
```

```
painelAbas.setEnabledAt(5, false);
```

```
painelAbas.setEnabledAt(8, false);
```

```
//tração
```

```
campoft0d.setText("");
```

```
campoft90d.setText("");
```

```
campoftad.setText("");
```

```
campoAreaMinT.setText("");
```

```
campoEsbeltezT.setText("");
```

```
listaSecoesT.removeAllItems();
```

```
//compressão
```

```
campofc0d.setText("");
```

```
campofc90d.setText("");
```

```
campofcad.setText("");
```

```
campoAreaMinC.setText("");
```

```
campoEsbeltezC.setText("");
```

```
campoEstabilidadeC.setText("");
```

```
listaSecoesC.removeAllItems();
```

```
//Resultados
```

```
        areaTxtResultados.setText("");
    }

private void campoCargaNqVFocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioVer = true;

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(8, false);

    //tração

    campoft0d.setText("");

    campoft90d.setText("");

    campoftad.setText("");

    campoAreaMinT.setText("");

    campoEsbeltezT.setText("");

    listaSecoesT.removeAllItems();

    //compressão
```

```
campofc0d.setText("");

campofc90d.setText("");

campofcad.setText("");

campoAreaMinC.setText("");

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Resultados

areaTxtResultados.setText("");

}

private void
campoComprimentoHPropertyChange(java.beans.PropertyChangeEvent evt) {

    lgBloqueioHor = true;

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);
```

```
//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

brCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void
campoComprimentoVPropertyChange(java.beans.PropertyChangeEvent evt) {

    lgBloqueioVer = true;
```

```
painelAbas.setEnabledAt(3, false);  
painelAbas.setEnabledAt(4, false);  
painelAbas.setEnabledAt(5, false);  
painelAbas.setEnabledAt(8, false);
```

```
//tração
```

```
campoft0d.setText("");  
campoft90d.setText("");  
campoftad.setText("");  
campoAreaMinT.setText("");  
campoEsbeltezT.setText("");  
listaSecoesT.removeAllItems();
```

```
//compressão
```

```
campofc0d.setText("");  
campofc90d.setText("");  
campofcad.setText("");  
campoAreaMinC.setText("");  
campoEsbeltezC.setText("");  
campoEstabilidadeC.setText("");
```



```
listaSecoesC.removeAllItems();

//Resultados

areaTxtResultados.setText("");
}

private void rbHorizontalFocusLost(java.awt.event.FocusEvent evt) {

    painelAbas.setEnabledAt(2, false);

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

    //Vertical

    campoComprimentoV.setText("");

    campoCargaNgV.setText("");

    campoCargaNqV.setText("");

    campoNsd1.setText("");
```

```
campoNsd2.setText("");
```

```
bpVertical.setEnabled(false);
```

```
//Horizontal
```

```
campoComprimentoH.setText("");
```

```
campoCargaGGH.setText("");
```

```
campoCargaQQH.setText("");
```

```
campoMsd1.setText("");
```

```
campoMsd2.setText("");
```

```
bpHorizontal.setEnabled(false);
```

```
//tração
```

```
campoft0d.setText("");
```

```
campoft90d.setText("");
```

```
campoftad.setText("");
```

```
campoAreaMinT.setText("");
```

```
campoEsbeltezT.setText("");
```

```
listaSecoesT.removeAllItems();
```

```
//compressão
```

```
campofc0d.setText("");
```

```
campofc90d.setText("");
```

```
campofcad.setText("");

campoAreaMinC.setText("");

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

bCisalhamento.setEnabled(false);

//Resultados
```

```
        areaTxtResultados.setText("");
    }

private void rbVerticalFocusLost(java.awt.event.FocusEvent evt) {

    painelAbas.setEnabledAt(2, false);

    painelAbas.setEnabledAt(3, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

    //Vertical

    campoComprimentoV.setText("");

    campoCargaNgV.setText("");

    campoCargaNqV.setText("");

    campoNsd1.setText("");

    campoNsd2.setText("");

    bpVertical.setEnabled(false);
```

```
//Horizontal  
  
campoComprimentoH.setText("");  
  
campoCargaGGH.setText("");  
  
campoCargaQQH.setText("");  
  
campoMsd1.setText("");  
  
campoMsd2.setText("");  
  
campoVsd1.setText("");  
  
campoVsd2.setText("");  
  
bpHorizontal.setEnabled(false);
```

```
//tração  
  
campoft0d.setText("");  
  
campoft90d.setText("");  
  
campoftad.setText("");  
  
campoAreaMinT.setText("");  
  
campoEsbeltezT.setText("");  
  
listaSecoesT.removeAllItems();
```

```
//compressão  
  
campoFc0d.setText("");  
  
campoFc90d.setText("");  
  
campoFcad.setText("");
```

```
campoAreaMinC.setText("");
```

```
campoEsbeltezC.setText("");
```

```
campoEstabilidadeC.setText("");
```

```
listaSecoesC.removeAllItems();
```

```
//Flexão
```

```
campoAlturaMinF.setText("");
```

```
campoInstabilidadeL.setText("");
```

```
campoTravamento.setText("");
```

```
listaAlturaF.removeAllItems();
```

```
bVerificaF.setEnabled(false);
```

```
bTravamento.setEnabled(false);
```

```
bpFlexao.setEnabled(false);
```

```
//Cisalhamento
```

```
campoCondCI.setText("");
```

```
campoRedimCI.setText("");
```

```
bRedimCI.setEnabled(false);
```

```
brCisalhamento.setEnabled(false);
```

```
//Resultados
```

```
areaTxtResultados.setText("");
```

```
}
```

```
private void rbHorizontalMousePressed(java.awt.event.MouseEvent evt) {
```

```
    painelAbas.setEnabledAt(2, false);
```

```
    painelAbas.setEnabledAt(3, false);
```

```
    painelAbas.setEnabledAt(4, false);
```

```
    painelAbas.setEnabledAt(5, false);
```

```
    painelAbas.setEnabledAt(6, false);
```

```
    painelAbas.setEnabledAt(7, false);
```

```
    painelAbas.setEnabledAt(8, false);
```

```
    //Vertical
```

```
    campoComprimentoV.setText("");
```

```
    campoCargaNgV.setText("");
```

```
    campoCargaNqV.setText("");
```

```
    campoNsd1.setText("");
```

```
    campoNsd2.setText("");
```

```
    bpVertical.setEnabled(false);
```

```
    //Horizontal
```

```
campoComprimentoH.setText("");  
  
campoCargaGGH.setText("");  
  
campoCargaQQH.setText("");  
  
campoMsd1.setText("");  
  
campoMsd2.setText("");  
  
campoVsd1.setText("");  
  
campoVsd2.setText("");  
  
bpHorizontal.setEnabled(false);
```

```
//tração
```

```
campoft0d.setText("");  
  
campoft90d.setText("");  
  
campoftad.setText("");  
  
campoAreaMinT.setText("");  
  
campoEsbeltezT.setText("");  
  
listaSecoesT.removeAllItems();
```

```
//compressão
```

```
campofc0d.setText("");  
  
campofc90d.setText("");  
  
campofcad.setText("");  
  
campoAreaMinC.setText("");
```



```
campoEsbeltezC.setText("");
```

```
campoEstabilidadeC.setText("");
```

```
listaSecoesC.removeAllItems();
```

```
//Flexão
```

```
campoAlturaMinF.setText("");
```

```
campoInstabilidadeL.setText("");
```

```
campoTravamento.setText("");
```

```
listaAlturaF.removeAllItems();
```

```
bVerificaF.setEnabled(false);
```

```
bTravamento.setEnabled(false);
```

```
bpFlexao.setEnabled(false);
```

```
//Cisalhamento
```

```
campoCondCI.setText("");
```

```
campoRedimCI.setText("");
```

```
bRedimCI.setEnabled(false);
```

```
brCisalhamento.setEnabled(false);
```

```
//Resultados
```

```
areaTxtResultados.setText("");
```

```
}
```

```
private void rbConiferaMousePressed(java.awt.event.MouseEvent evt) {
```

```
    painelPropriedades.setEnabledAt(1, false);
```

```
    painelAbas.setEnabledAt(4, false);
```

```
    painelAbas.setEnabledAt(5, false);
```

```
    painelAbas.setEnabledAt(6, false);
```

```
    painelAbas.setEnabledAt(7, false);
```

```
    painelAbas.setEnabledAt(8, false);
```

```
    //tração
```

```
    campoft0d.setText("");
```

```
    campoft90d.setText("");
```

```
    campoftad.setText("");
```

```
    campoAreaMinT.setText("");
```

```
    campoEsbeltezT.setText("");
```

```
    listaSecoesT.removeAllItems();
```

```
    //compressão
```

```
    campofc0d.setText("");
```

```
    campofc90d.setText("");
```

```
campofcad.setText("");

campoAreaMinC.setText("");

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

bCisalhamento.setEnabled(false);

//Resultados
```

```
        areaTxtResultados.setText("");
    }

private void rbDicotiledoneaFocusGained(java.awt.event.FocusEvent evt) {

    painelPropriedades.setEnabledAt(1, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

    //tração

    campoft0d.setText("");

    campoft90d.setText("");

    campoftad.setText("");

    campoAreaMinT.setText("");

    campoEsbeltezT.setText("");

    listaSecoesT.removeAllItems();

    //compressão
```

```
campofc0d.setText("");
campofc90d.setText("");
campofcad.setText("");
campoAreaMinC.setText("");
campoEsbeltezC.setText("");
campoEstabilidadeC.setText("");
listaSecoesC.removeAllItems();

//Flexão
campoAlturaMinF.setText("");
campoInstabilidadeL.setText("");
campoTravamento.setText("");
listaAlturaF.removeAllItems();
bVerificaF.setEnabled(false);
bTravamento.setEnabled(false);
bpFlexao.setEnabled(false);

//Cisalhamento
campoCondCI.setText("");
campoRedimCI.setText("");
bRedimCI.setEnabled(false);
brCisalhamento.setEnabled(false);
```

```
//Resultados

areaTxtResultados.setText("");

}

private void listaConiferasFocusGained(java.awt.event.FocusEvent evt) {

    painelPropriedades.setEnabledAt(1, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

//tração

campoft0d.setText("");

campoft90d.setText("");

campoftad.setText("");

campoAreaMinT.setText("");

campoEsbeltezT.setText("");

listaSecoesT.removeAllItems();
```

```
//compressão

campofc0d.setText("");

campofc90d.setText("");

campofcad.setText("");

campoAreaMinC.setText("");

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");
```

```
bRedimCl.setEnabled(false);

brCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void listaDicotiledoneasFocusGained(java.awt.event.FocusEvent evt) {

    painelPropriedades.setEnabledAt(1, false);

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

    //tração

    campoft0d.setText("");

    campoft90d.setText("");

    campoftad.setText("");

    campoAreaMinT.setText("");
```



```
campoEsbeltezT.setText("");
```

```
listaSecoesT.removeAllItems();
```

```
//compressão
```

```
campofc0d.setText("");
```

```
campofc90d.setText("");
```

```
campofcad.setText("");
```

```
campoAreaMinC.setText("");
```

```
campoEsbeltezC.setText("");
```

```
campoEstabilidadeC.setText("");
```

```
listaSecoesC.removeAllItems();
```

```
//Flexão
```

```
campoAlturaMinF.setText("");
```

```
campoInstabilidadeL.setText("");
```

```
campoTravamento.setText("");
```

```
listaAlturaF.removeAllItems();
```

```
bVerificaF.setEnabled(false);
```

```
bTravamento.setEnabled(false);
```

```
bpFlexao.setEnabled(false);
```

```
//Cisalhamento
```

```
campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

brCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void campoKmod3MousePressed(java.awt.event.MouseEvent evt) {

    lgBloqueioKmod = true;

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);

    //tração

    campoft0d.setText("");

    campoft90d.setText("");
```

```
campoftad.setText("");

campoAreaMinT.setText("");

campoEsbeltezT.setText("");

listaSecoesT.removeAllItems();

//compressão

campofc0d.setText("");

campofc90d.setText("");

campofcad.setText("");

campoAreaMinC.setText("");

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);
```

```
//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

brCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void campoKmod1FocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioKmod = true;

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);

    painelAbas.setEnabledAt(8, false);
```

//tração

campoft0d.setText("");

campoft90d.setText("");

campoftad.setText("");

campoAreaMinT.setText("");

campoEsbeltezT.setText("");

listaSecoesT.removeAllItems();

//compressão

campofc0d.setText("");

campofc90d.setText("");

campofcad.setText("");

campoAreaMinC.setText("");

campoEsbeltezC.setText("");

campoEstabilidadeC.setText("");

listaSecoesC.removeAllItems();

//Flexão

campoAlturaMinF.setText("");

campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

```
bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

brCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void campoKmod2FocusGained(java.awt.event.FocusEvent evt) {

    lgBloqueioKmod = true;

    painelAbas.setEnabledAt(4, false);

    painelAbas.setEnabledAt(5, false);

    painelAbas.setEnabledAt(6, false);

    painelAbas.setEnabledAt(7, false);
```

```
painelAbas.setEnabledAt(8, false);
```

```
//tração
```

```
campoft0d.setText("");
```

```
campoft90d.setText("");
```

```
campoftad.setText("");
```

```
campoAreaMinT.setText("");
```

```
campoEsbeltezT.setText("");
```

```
listaSecoesT.removeAllItems();
```

```
//compressão
```

```
campofc0d.setText("");
```

```
campofc90d.setText("");
```

```
campofcad.setText("");
```

```
campoAreaMinC.setText("");
```

```
campoEsbeltezC.setText("");
```

```
campoEstabilidadeC.setText("");
```

```
listaSecoesC.removeAllItems();
```

```
//Flexão
```

```
campoAlturaMinF.setText("");
```

```
campoInstabilidadeL.setText("");

campoTravamento.setText("");

listaAlturaF.removeAllItems();

bVerificaF.setEnabled(false);

bTravamento.setEnabled(false);

bpFlexao.setEnabled(false);

//Cisalhamento

campoCondCI.setText("");

campoRedimCI.setText("");

bRedimCI.setEnabled(false);

brCisalhamento.setEnabled(false);

//Resultados

areaTxtResultados.setText("");

}

private void listaExtensao90FocusGained(java.awt.event.FocusEvent evt) {

    campoEsbeltezC.setText("");

    campoEstabilidadeC.setText("");

    campoAreaMinC.setText("");
```



```
    campofc0d.setText("");
    campofcad.setText("");
    campofc90d.setText("");
    campoOutroC.setText("");
    listaSecoesC.setEnabled(false);
    listaSecoesC.removeAllItems();
    bVerificarC.setEnabled(false);
    brCompressao.setEnabled(false);
    painelAbas.setEnabledAt(8, false);
}

private void listaExtensaoOutroFocusGained(java.awt.event.FocusEvent evt) {
    campoEsbeltezC.setText("");
    campoEstabilidadeC.setText("");
    campoAreaMinC.setText("");
    campofc0d.setText("");
    campofcad.setText("");
    campofc90d.setText("");
    listaSecoesC.setEnabled(false);
    listaSecoesC.removeAllItems();
    bVerificarC.setEnabled(false);
    brCompressao.setEnabled(false);
}
```

```
        painelAbas.setEnabledAt(8, false);
    }

    private void listaSecoesCFocusGained(java.awt.event.FocusEvent evt) {

        lgBloqueioComp = true;

        campoEsbeltezC.setText("");

        campoEstabilidadeC.setText("");

        painelAbas.setEnabledAt(8, false);

    }

    private void listaSecoesTFocusGained(java.awt.event.FocusEvent evt) {

        lgBloqueioTra = true;

        campoEsbeltezT.setText("");

        painelAbas.setEnabledAt(8, false);

    }

    public static void main(String args[]) {
```

```
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new telainicial().setVisible(true);  
    }  
});  
  
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JTextArea areaTxtResultados;  
  
private javax.swing.JButton bCalculaCombH;  
  
private javax.swing.JButton bCalculaCombV;  
  
private javax.swing.JButton bCalcularKmod;  
  
private javax.swing.JButton bDimensionarC;  
  
private javax.swing.JButton bDimensionarFlexao;  
  
private javax.swing.JButton bDimensionarT;  
  
private javax.swing.JButton bIniciar;  
  
private javax.swing.JButton bRedimCI;  
  
private javax.swing.JButton bTravamento;  
  
private javax.swing.JButton bVerificaCI;  
  
private javax.swing.JButton bVerificaF;
```

```
private javax.swing.JButton bVerificarC;  
private javax.swing.JButton bVerificarT;  
private javax.swing.JButton bpClasseResistencia;  
private javax.swing.JButton bpFlexao;  
private javax.swing.JButton bpHorizontal;  
private javax.swing.JButton bpKmod;  
private javax.swing.JButton bpPaginaPrincipal;  
private javax.swing.JButton bpVertical;  
private javax.swing.JButton brCisalhamento;  
private javax.swing.JButton brCompressao;  
private javax.swing.JButton brTracao;  
private javax.swing.JTextField campoAlturaCI;  
private javax.swing.JTextField campoAlturaMinF;  
private javax.swing.JTextField campoAreaMinC;  
private javax.swing.JTextField campoAreaMinT;  
private javax.swing.JTextField campoBaseCI;  
private javax.swing.JFormattedTextField campoCargaGGH;  
private javax.swing.JFormattedTextField campoCargaNgV;  
private javax.swing.JFormattedTextField campoCargaNqV;  
private javax.swing.JFormattedTextField campoCargaQQH;  
private javax.swing.JFormattedTextField campoComprimentoH;  
private javax.swing.JFormattedTextField campoComprimentoV;
```

```
private javax.swing.JTextField campoCondCl;  
private javax.swing.JTextField campoEsbeltezC;  
private javax.swing.JTextField campoEsbeltezT;  
private javax.swing.JTextField campoEstabilidadeC;  
private javax.swing.JTextField campoFcdF;  
private javax.swing.JTextField campoFtdF;  
private javax.swing.JTextField campoFv0Cl;  
private javax.swing.JTextField campoInstabilidadeL;  
private javax.swing.JTextField campoKmod;  
private javax.swing.JTextField campoKmod1;  
private javax.swing.JTextField campoKmod2;  
private javax.swing.JTextField campoKmod3;  
private javax.swing.JTextField campoKmodC;  
private javax.swing.JTextField campoKmodCl;  
private javax.swing.JTextField campoKmodF;  
private javax.swing.JTextField campoKmodT;  
private javax.swing.JTextField campoMdF;  
private javax.swing.JTextField campoMsd1;  
private javax.swing.JTextField campoMsd2;  
private javax.swing.JTextField campoNdC;  
private javax.swing.JTextField campoNdT;  
private javax.swing.JTextField campoNsd1;
```

```
private javax.swing.JTextField campoNsd2;
private javax.swing.JTextField campoOutroC;
private javax.swing.JTextField campoOutroT;
private javax.swing.JTextField campoRedimCI;
private javax.swing.JTextField campoTravamento;
private javax.swing.JTextField campoVdCI;
private javax.swing.JTextField campoVsd1;
private javax.swing.JTextField campoVsd2;
private javax.swing.JTextField campoYwc;
private javax.swing.JTextField campoYwt;
private javax.swing.JTextField campofc0d;
private javax.swing.JTextField campofc90d;
private javax.swing.JTextField campofcad;
private javax.swing.JTextField campoft0d;
private javax.swing.JTextField campoft90d;
private javax.swing.JTextField campoftad;
private javax.swing.ButtonGroup cargaTeC;
private javax.swing.ButtonGroup classeResistencia;
private javax.swing.ButtonGroup grupoMadeira;
private javax.swing.JLabel imgPilar;
private javax.swing.JLabel imgPilarCarreg;
private javax.swing.JLabel imgViga;
```

```
private javax.swing.JLabel imgVigac;

private javax.swing.ButtonGroup inclinacaoCompressao;

private javax.swing.ButtonGroup inclinacaoTracao;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel14;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

private javax.swing.JLabel jLabel17;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JSeparator jSeparator1;
```

```
private javax.swing.JSeparator jSeparator10;  
private javax.swing.JSeparator jSeparator11;  
private javax.swing.JSeparator jSeparator12;  
private javax.swing.JSeparator jSeparator13;  
private javax.swing.JSeparator jSeparator14;  
private javax.swing.JSeparator jSeparator15;  
private javax.swing.JSeparator jSeparator16;  
private javax.swing.JSeparator jSeparator17;  
private javax.swing.JSeparator jSeparator18;  
private javax.swing.JSeparator jSeparator19;  
private javax.swing.JSeparator jSeparator2;  
private javax.swing.JSeparator jSeparator20;  
private javax.swing.JSeparator jSeparator21;  
private javax.swing.JSeparator jSeparator3;  
private javax.swing.JSeparator jSeparator4;  
private javax.swing.JSeparator jSeparator5;  
private javax.swing.JSeparator jSeparator6;  
private javax.swing.JSeparator jSeparator7;  
private javax.swing.JSeparator jSeparator8;  
private javax.swing.JSeparator jSeparator9;  
private javax.swing.JComboBox<String> listaAlturaF;  
private javax.swing.JComboBox<String> listaBaseF;
```



```
private javax.swing.JComboBox<String> listaCargaVariavelH;  
private javax.swing.JComboBox<String> listaCargaVariavelV;  
private javax.swing.JComboBox<String> listaConiferas;  
private javax.swing.JComboBox<String> listaDicotiledoneas;  
private javax.swing.JComboBox<String> listaExtensao90;  
private javax.swing.JComboBox<String> listaExtensaoOutro;  
private javax.swing.JComboBox<String> listaSecoesC;  
private javax.swing.JComboBox<String> listaSecoesT;  
private javax.swing.JTabbedPane painelAbas;  
private javax.swing.JTabbedPane painelCarregamentos;  
private javax.swing.JPanel painelCisalhamento;  
private javax.swing.JPanel painelClasseResistencia;  
private javax.swing.JPanel painelCoeficienteModificacao;  
private javax.swing.JPanel painelCompressao;  
private javax.swing.JPanel painelFlexao;  
private javax.swing.JPanel painelHorizontal;  
private javax.swing.JPanel painelPaginaInicial;  
private javax.swing.JTabbedPane painelPropriedades;  
private javax.swing.JPanel painelResultados;  
private javax.swing.JPanel painelTracao;  
private javax.swing.JPanel painelTriWood;  
private javax.swing.JPanel painelVertical;
```

```
private javax.swing.JRadioButton rb0C;  
private javax.swing.JRadioButton rb0T;  
private javax.swing.JRadioButton rb90C;  
private javax.swing.JRadioButton rb90T;  
private javax.swing.JRadioButton rbConifera;  
private javax.swing.JRadioButton rbDicotiledonea;  
private javax.swing.JRadioButton rbGrandeH;  
private javax.swing.JRadioButton rbGrandeV;  
private javax.swing.JRadioButton rbHorizontal;  
private javax.swing.JRadioButton rbOutroC;  
private javax.swing.JRadioButton rbOutroT;  
private javax.swing.JRadioButton rbPequenaH;  
private javax.swing.JRadioButton rbPequenaV;  
private javax.swing.JRadioButton rbVertical;  
private javax.swing.ButtonGroup sentidoPeca;  
private javax.swing.JLabel tabkmod1;  
private javax.swing.JLabel tabkmod2;  
private javax.swing.JLabel tabkmod3;  
private javax.swing.JLabel txtAlturaCI;  
private javax.swing.JLabel txtAlturaF;  
private javax.swing.JLabel txtAlturaMinF;  
private javax.swing.JLabel txtAreaMinC;
```

```
private javax.swing.JLabel txtAreaMinT;  
private javax.swing.JLabel txtBaseCI;  
private javax.swing.JLabel txtBaseF;  
private javax.swing.JLabel txtBemVindo;  
private javax.swing.JLabel txtCargaNgV;  
private javax.swing.JLabel txtCargaNqV;  
private javax.swing.JLabel txtCargagH;  
private javax.swing.JLabel txtCargaqH;  
private javax.swing.JLabel txtClasseResistencia;  
private javax.swing.JLabel txtComb;  
private javax.swing.JLabel txtCombH;  
private javax.swing.JLabel txtComprimento;  
private javax.swing.JLabel txtComprimentoV;  
private javax.swing.JLabel txtCondCI;  
private javax.swing.JLabel txtDCargaPerm;  
private javax.swing.JLabel txtDCargaPerm1;  
private javax.swing.JLabel txtDCargaVari;  
private javax.swing.JLabel txtDCargaVariH;  
private javax.swing.JLabel txtDadosC;  
private javax.swing.JLabel txtDadosCI;  
private javax.swing.JLabel txtDadosF;  
private javax.swing.JLabel txtDadosT;
```

```
private javax.swing.JLabel txtEsbeltezC;

private javax.swing.JLabel txtEsbeltezT;

private javax.swing.JLabel txtEstabilidadeC;

private javax.swing.JLabel txtFcdF;

private javax.swing.JLabel txtFtdF;

private javax.swing.JLabel txtKmod;

private javax.swing.JLabel txtKmodC;

private javax.swing.JLabel txtKmodCI;

private javax.swing.JLabel txtKmodF;

private javax.swing.JLabel txtKmodT;

private javax.swing.JLabel txtMdF;

private javax.swing.JLabel txtMsd1;

private javax.swing.JLabel txtMsd2;

private javax.swing.JLabel txtNdC;

private javax.swing.JLabel txtNdT;

private javax.swing.JLabel txtNsd1;

private javax.swing.JLabel txtNsd2;

private javax.swing.JLabel txtRedimCI;

private javax.swing.JLabel txtSecaoC;

private javax.swing.JLabel txtSecaoT;

private javax.swing.JLabel txtSelecionarSentido;

private javax.swing.JLabel txtSelecioneInclinacao;
```

```
private javax.swing.JLabel txtSelecioneInclinacaoC;  
  
private javax.swing.JLabel txtTipoCarga;  
  
private javax.swing.JLabel txtTipoCargaH;  
  
private javax.swing.JLabel txtTravamento;  
  
private javax.swing.JLabel txtUnC1;  
  
private javax.swing.JLabel txtUnC2;  
  
private javax.swing.JLabel txtUnC3;  
  
private javax.swing.JLabel txtUnMsd1;  
  
private javax.swing.JLabel txtUnMsd2;  
  
private javax.swing.JLabel txtUnNsd1;  
  
private javax.swing.JLabel txtUnNsd2;  
  
private javax.swing.JLabel txtUnT1;  
  
private javax.swing.JLabel txtUnT2;  
  
private javax.swing.JLabel txtUnT3;  
  
private javax.swing.JLabel txtUnVsd1;  
  
private javax.swing.JLabel txtUnVsd2;  
  
private javax.swing.JLabel txtUniAlt;  
  
private javax.swing.JLabel txtUniAlturaF;  
  
private javax.swing.JLabel txtUniBase;  
  
private javax.swing.JLabel txtUniFcdF;  
  
private javax.swing.JLabel txtUniFtdF;  
  
private javax.swing.JLabel txtUniFv0;
```

```
private javax.swing.JLabel txtUniRedimCl;  
  
private javax.swing.JLabel txtUniVdCl;  
  
private javax.swing.JLabel txtUnidadeAreaC;  
  
private javax.swing.JLabel txtUnidadeAreaT;  
  
private javax.swing.JLabel txtUnidadeCargaNgV;  
  
private javax.swing.JLabel txtUnidadeCargaNqV;  
  
private javax.swing.JLabel txtUnidadeCargagH;  
  
private javax.swing.JLabel txtUnidadeCargaqH;  
  
private javax.swing.JLabel txtUnidadeComprimento;  
  
private javax.swing.JLabel txtUnidadeComprimentoV;  
  
private javax.swing.JLabel txtVdCl;  
  
private javax.swing.JLabel txtVsd1;  
  
private javax.swing.JLabel txtVsd2;  
  
private javax.swing.JLabel txtYwc;  
  
private javax.swing.JLabel txtYwt;  
  
private javax.swing.JLabel txtdefkmod1;  
  
private javax.swing.JLabel txtdefkmod2;  
  
private javax.swing.JLabel txtdefkmod3;  
  
private javax.swing.JLabel txtfc0d;  
  
private javax.swing.JLabel txtfc90d;  
  
private javax.swing.JLabel txtfcad;  
  
private javax.swing.JLabel txtft0d;
```

```
private javax.swing.JLabel txtft90dT;  
  
private javax.swing.JLabel txtftadT;  
  
private javax.swing.JLabel txtfv0CI;  
  
private javax.swing.JLabel txtgrausC;  
  
private javax.swing.JLabel txtgrausT;  
  
private javax.swing.JLabel txtkNC;  
  
private javax.swing.JLabel txtkNT;  
  
private javax.swing.JLabel txtkNmF;  
  
private javax.swing.JLabel txtkmod1;  
  
private javax.swing.JLabel txtkmod2;  
  
private javax.swing.JLabel txtkmod3;  
  
private javax.swing.ButtonGroup variabilidadeH;  
  
private javax.swing.ButtonGroup variabilidadeV;  
  
// End of variables declaration  
  
}
```