

3

Implementação Computacional

3.1

Introdução

No capítulo anterior foram apresentados todos os procedimentos necessários para o dimensionamento, tanto à flexão quanto ao cisalhamento, de uma viga de edifício de concreto armado. O presente capítulo mostra os principais aspectos da implementação computacional desses procedimentos dentro do programa FTOOL. A estrutura de dados do programa é descrita, bem como as alterações feitas neste trabalho na estrutura de dados e na interface gráfica.

O FTOOL (*Two-dimensional Frame Analysis Tool*) é um programa gráfico-interativo que se destina ao ensino do comportamento estrutural de pórticos planos. Além de ser uma ferramenta simples, o programa une em uma única interface recursos para uma eficiente criação e manipulação do modelo (pré-processamento) aliados a uma análise ágil e transparente da estrutura e a uma visualização de resultados rápida e efetiva (pós-processamento). Uma descrição detalhada do funcionamento do programa FTOOL pode ser encontrada em MARTHA [11].

O FTOOL utiliza uma biblioteca de funções chamada HED (*Half-Edge Data Structure*) para representação interna dos dados, além do sistema de interface IUP e do sistema gráfico CD (*Canvas Draw*), desenvolvidos pelo Grupo de Tecnologia em Computação Gráfica (Tecgraf/PUC-Rio). Esta interface gráfica permite que o programa seja executado tanto no ambiente Windows quanto no ambiente Unix/X-windows.

Os elementos gráficos para interação com o usuário (diálogos, botões, caixas de texto, etc.) do FTOOL são confeccionados utilizando elementos e funções do IUP. O IUP é um sistema portátil de interface com usuário composto por uma Linguagem de Especificação de Diálogos (LED) e uma biblioteca de funções para a criação e manipulação de diálogos. Ele permite que um programa seja executado sem modificações em vários ambientes de interface, conferindo ao conjunto de ferramentas uma alta portabilidade.

A disposição dos elementos de interface dentro da tela do FTOOL é guardada em um arquivo texto escrito em LED que é lido ao se executar o programa. Este arquivo LED pode ser convertido em um arquivo “C” que é compilado com o restante do código do FTOOL, dispensando-se os arquivos LED.

3.2

Estrutura de Dados

A estrutura de dados do FTOOL se baseia na biblioteca de modelagem HED, que tem uma estrutura de dados centralizada em uma representação topológica completa de uma subdivisão planar, com busca eficiente de informações de adjacência entre as entidades do modelo.

Uma subdivisão do \mathbb{R}^2 é apresentada na Figura 3.1. Nela o plano é dividido em diversas regiões (faces f_1 , f_2 , f_3 e a face externa), sendo que cada uma delas é delimitada por um conjunto de segmentos de curva (arestas) que, por sua vez, são delimitados por dois vértices não necessariamente distintos. Os termos vértice, aresta e face são provenientes do fato de que subdivisões planares representam a topologia da fronteira de sólidos homeomorfos a uma esfera. Isto é, cada ponto da fronteira do sólido tem uma vizinhança que é mapeável para um disco aberto 2D.

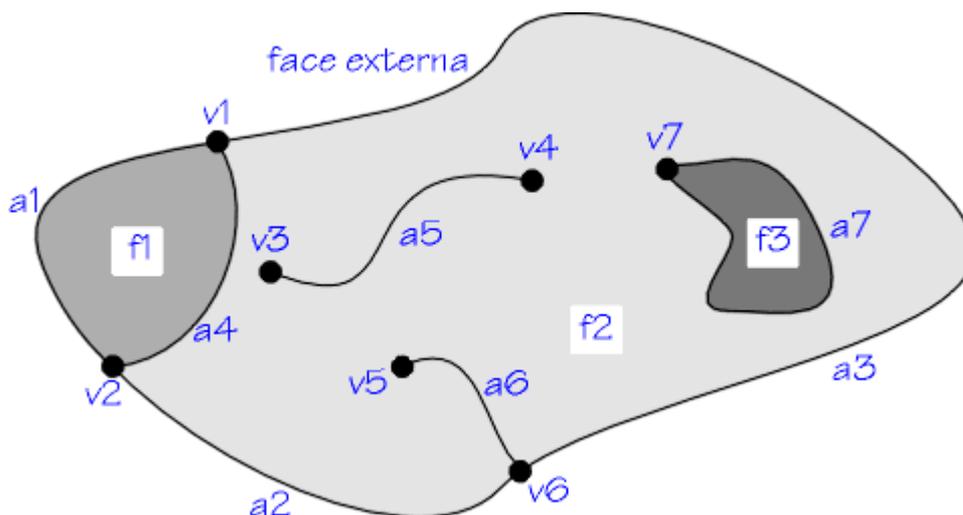


Figura 3.1 – Exemplo de uma subdivisão do \mathbb{R}^2 , adaptada de FERRAZ [12].

No processo de criação e manutenção de uma subdivisão planar, quando um segmento de curva for incluído, todos os pontos de interseção com os segmentos já pertencentes à subdivisão são calculados. Estes pontos se tornam vértices topológicos e são criadas arestas entre cada um destes vértices. No caso de se formarem regiões fechadas por arestas, novas faces são criadas na subdivisão.

As entidades topológicas que formam uma subdivisão planar são as seguintes:

- vértice (*vertex*): correspondente a um ponto único do \mathcal{R}^2 , não podendo haver dois vértices com a mesma localização geométrica;
- aresta (*edge*): segmento de curva (equivalente homeomorficamente a um segmento de reta) delimitado por dois vértices não necessariamente distintos;
- ciclo (*loop*): subconjunto conexo e ordenado de vértices e arestas alternados. Um loop pode ser degenerado para uma única aresta com um ou dois vértices, ou mesmo degenerado para um vértice;
- face: subconjunto conexo do \mathcal{R}^2 , cuja fronteira é formada por um ou mais ciclos, sendo que, com exceção da face externa, um destes contém os outros e representa sua fronteira externa;
- casca (*shell*): é uma porção conexa da fronteira de um volume. Uma casca pode ser degenerada para um único vértice e uma face.

No caso de modelagem de pórticos planos, os vértices da subdivisão planar são relacionados com os nós do quadro e as arestas são relacionadas com os membros do quadro (elementos de barra).

Pode parecer um exagero representar um quadro plano com uma estrutura de dados tão completa e sofisticada. A adoção da estrutura de dados HED no FTOOL se justifica por vários motivos. O principal deles é que a representação HED fornece de maneira bastante eficiente todas as informações de adjacência entre os nós (vértices) e barras (arestas) do quadro plano, fato este que é explorado em diversos algoritmos implantados no FTOOL. Por exemplo, o procedimento de determinação da armadura longitudinal em uma viga contínua (seqüência contígua de barras) descrito neste capítulo é baseado na adjacência entre nós e barras.

Na Figura 3.2 são apresentadas as nove relações de adjacência possíveis entre vértices, arestas e faces. De uma representação que tenha

suficiência de informação, pode-se extrair todos os relacionamentos de adjacência entre os elementos da subdivisão planar. Mas a estrutura de dados usada em uma representação não precisa necessariamente armazenar explicitamente as nove relações de adjacência, bastando armazenar alguma delas e, a partir destas, obter as restantes por derivação. O armazenamento explícito de todos os relacionamentos de adjacência exigiria uma quantidade maior de memória, apesar do rápido tempo de acesso às informações.

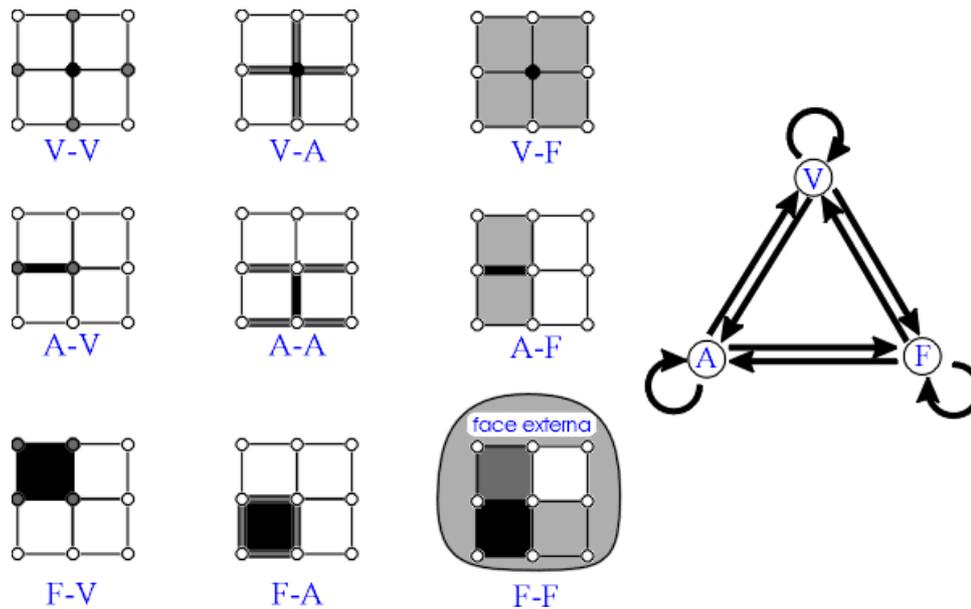


Figura 3.2 – As nove relações de adjacência entre vértices, arestas e faces, adaptada de FERRAZ [12].

As estruturas de dados topológicas mais utilizadas têm a aresta como elemento central (Figura 3.3). O motivo pelo qual isto ocorre é que a aresta é a única entidade cujo número de entidades adjacentes é limitado. Cada aresta é limitada por exatamente dois vértices não necessariamente distintos, e cada aresta é adjacente a exatamente duas faces não necessariamente distintas (Figura 3.2). Além disso, se for considerado que a adjacência aresta-aresta é caracterizada por arestas que compartilham tanto vértices quanto faces, o número de arestas adjacentes a uma dada aresta é limitado a quatro.

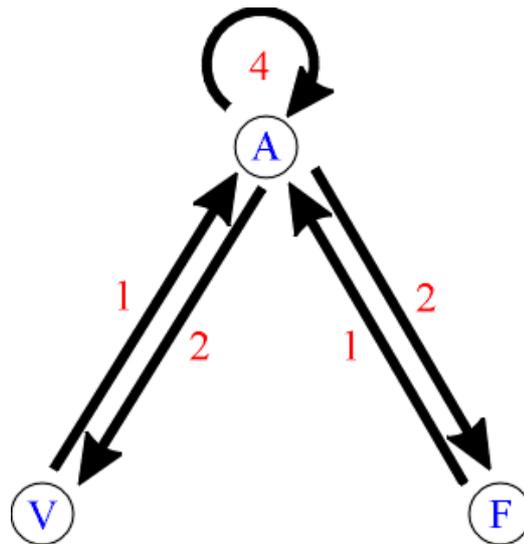


Figura 3.3 – Relações de adjacência explicitamente armazenadas em uma estrutura baseada em arestas, adaptada de FERRAZ [12].

A estrutura de dados HED do FTOOL é baseada em aresta. O nome HED significa Estrutura de Dados de Semi-aresta (*half-edge data structure*). Nessa estrutura a entidade central é uma semi-aresta (metade de uma aresta), que representa o *uso* de uma aresta por um ciclo (*loop*). Por esta razão, a entidade central também é chamada de *uso de aresta*.

Observa-se na Figura 3.4 que a estrutura HED é hierárquica e formada por listas circulares encadeadas, indo dos níveis mais altos de hierarquia (subdivisão planar) aos mais baixos (vértices). No que diz respeito ao armazenamento explícito das relações de adjacência, a estrutura *half-edge* não é completa, embora forneça alguns relacionamentos de forma já ordenada. Por exemplo, percorrendo-se o ponteiro *eu_nxt* ou o *eu_prv* (apontam o próximo uso de aresta e o anterior, respectivamente) varrem-se ordenadamente as arestas adjacentes de um ciclo da fronteira de uma face (Figura 3.5). Mesmo não sendo completa, esta estrutura de dados permite que se obtenha as relações de adjacência com um desempenho que é, no pior caso, proporcional ao número de entidades topológicas envolvidas na busca.

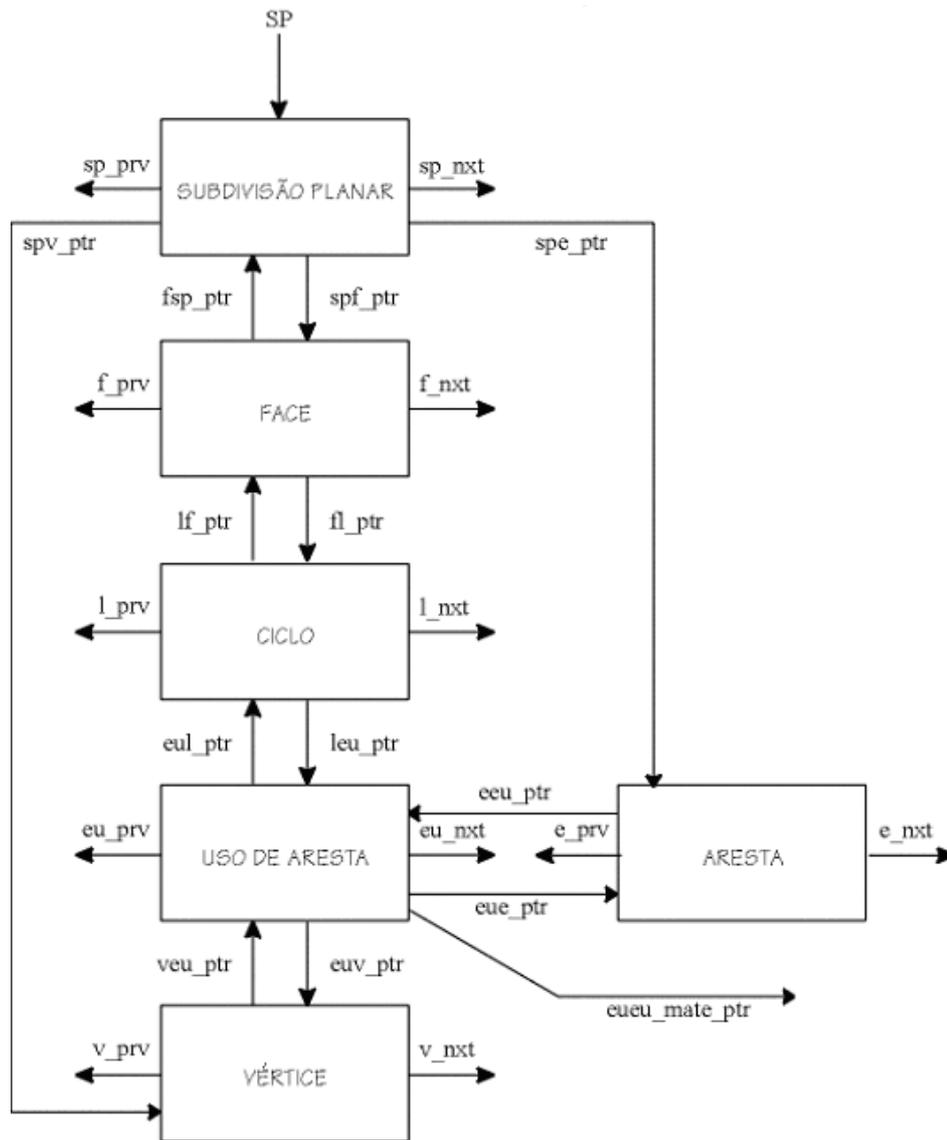


Figura 3.4 – Estrutura de dados HED, adaptada de FERRAZ [12].

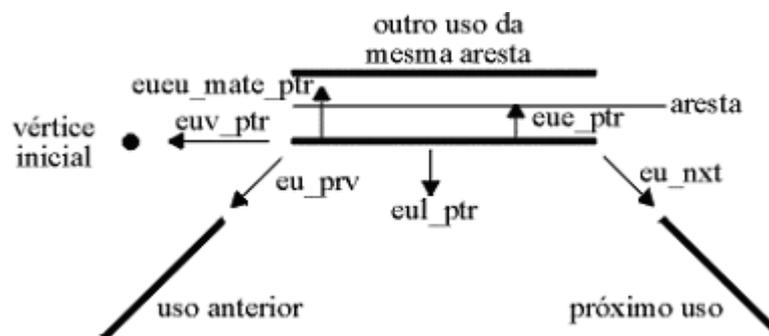


Figura 3.5 – Usos de uma aresta, adaptada de FERRAZ [12].

Além do encadeamento de ponteiros que estabelece as relações de adjacência, cada entidade possui também um ponteiro para um registro de atributos opcional. A subdivisão planar, a face, a aresta e o vértice podem ter atributos associados.

No caso do FTOOL, os registros de dados para forças e condições de apoios aplicados a nós, assim como para forças distribuídas aplicadas a barras, entre outras, são blocos de atributos “pendurados” nas entidades topológicas do HED: *solid* (modelo), *edge* (aresta, que corresponde a um elemento de barra) e *vertex* (vértice, que corresponde a um nó).

A Figura 3.6 descreve simplificada a estrutura de dados de atributos do FTOOL. Os retângulos pretos no fluxograma representam os nomes dos registros principais; os retângulos cinzas, os conjuntos de dados armazenados no FTOOL; e os brancos, os ponteiros que relacionam as entidades e registros de dados. A entidade “*SOLID*” representa um modelo ou uma subdivisão planar.

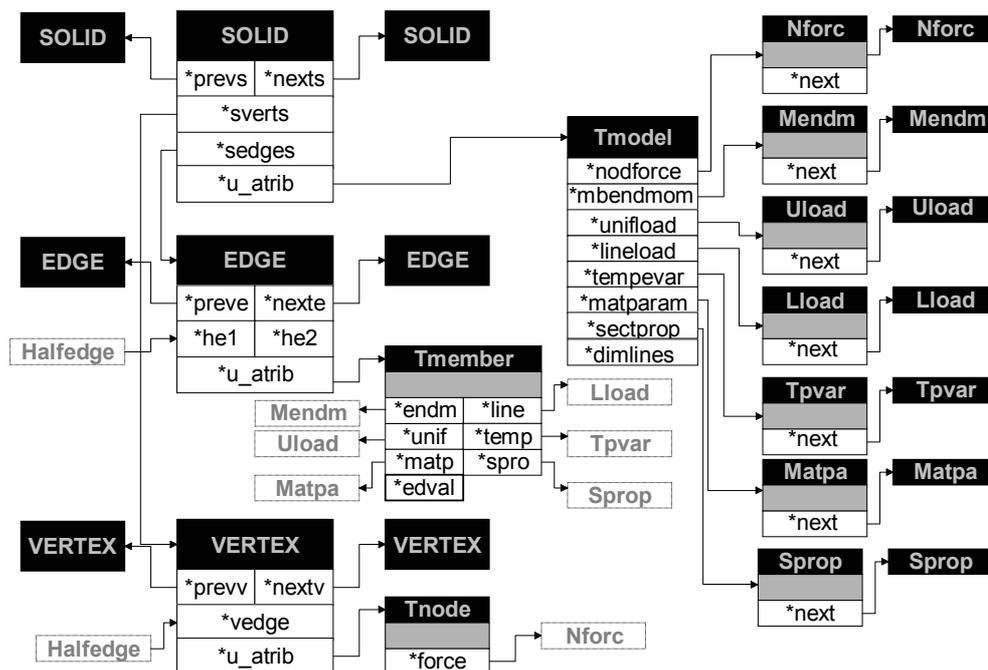


Figura 3.6 – Estrutura de dados de atributos do FTOOL, adaptada de KAEFER [1].

Uma descrição detalhada da estrutura de dados de atributos do FTOOL pode ser encontrada no trabalho de KAEFER [1]. Neste trabalho não foram feitas alterações na estrutura de dados do FTOOL. As implementações descritas na

seqüência deste capítulo foram feitas nos atributos “pendurados” nas arestas da estrutura de dados.

3.3

Implementação de Materiais e Seções Transversais em POO

Neste trabalho, o programa FTOOL foi estendido para considerar um novo tipo de material – concreto armado – e novas seções transversais relativas a uma viga de concreto armado – retangular, T, L e I. Essas implementações seguiram um paradigma de programação orientada a objetos (POO). Segundo COX [13], a principal vantagem de se adotar a programação orientada a objetos é que a expansão do programa fica mais simples e natural. Assim, as novas implementações têm um impacto mínimo no código existente. Desta maneira, a reutilização do código é maximizada. Além disso, comparando com a programação clássica estruturada, a utilização da programação orientada a objetos conduz para uma integração mais completa entre a teoria e a implementação computacional. Normalmente a POO é utilizada no desenvolvimento de programas grandes e complexos. Entretanto, para obter a máxima vantagem de se utilizar a POO, é necessário o completo entendimento da metodologia empregada na organização do programa.

Os cinco componentes chave do paradigma de orientação a objetos, de acordo com BORGES [14], são:

- Objeto: é uma abstração encapsulada que tem um estado interno dado por uma lista de atributos cujos valores são únicos para o objeto.
- Mensagem: é representada por um identificador que implica em uma ação a ser tomada pelo objeto que a recebe.
- Classe: é um modelo para a criação de um objeto. Inclui em sua descrição um nome para o tipo de objeto, uma lista de atributos e uma lista de mensagens com os métodos correspondentes que o objeto desta classe sabe responder.
- Instância: é um objeto que tem suas propriedades definidas na descrição da classe.
- Método: é uma lista de instruções que define como um objeto responde a uma mensagem em particular. Um método tipicamente consiste de expressões que enviam mais mensagens para objetos. Toda mensagem em uma classe tem um método correspondente.

Em uma visão simplificada, classes são estruturas, objetos são variáveis do tipo de alguma classe (instância de alguma classe), métodos são funções de classes e enviar uma mensagem para um objeto é chamar um método de um objeto.

Um conceito importante na POO, chamado encapsulamento, consiste no ato de esconder certas informações do cliente da classe (aquele que utiliza uma instância de classe). Os atributos dependentes da implementação não precisam e não devem estar disponíveis para o usuário de um objeto, pois este deve ser acessado exclusivamente através da interface definida.

Outro conceito importante na POO é a herança, que é um termo que se aplica apenas às classes. Herança permite que se construa e estenda continuamente classes já desenvolvidas sem nenhum limite. Começando da classe mais simples, pode-se derivar classes cada vez mais complexas.

A POO apresenta, ainda, como um de seus recursos mais poderosos, o polimorfismo, que é uma consequência da herança. O recurso de polimorfismo permite que todos os objetos de uma hierarquia de classes (da mais simples à mais complexa) sejam declarados e tratados como objetos da classe base (mais simples). Tal recurso permite trabalhar em um nível de abstração bem alto ao mesmo tempo que facilita a incorporação de novos pedaços em um sistema já existente.

Para a implementação do dimensionamento de vigas no programa FTOOL, foram criados o material concreto armado e as seções transversais mais comuns em uma viga de edifício de concreto armado (retangular, T, L ou I). Essas implementações foram feitas dentro do paradigma da POO.

3.3.1

Material Concreto Armado

Existem várias operações conceituais que são comuns para todos os tipos de materiais. Essas operações incluem: consultar os parâmetros do material, criar elementos de interface para receber os parâmetros do material, entre outras. Na versão anterior do FTOOL existiam três tipos de materiais: genérico (*Generic Isotropic*), concreto homogêneo (*Concrete Isotropic*) e aço (*Steel Isotropic*). Estes materiais continham os mesmos tipos de parâmetros: módulo de elasticidade, peso específico e coeficiente de dilatação térmica. Como para o material concreto armado são necessários diversos outros parâmetros, foi

criada uma “classe material” no contexto de POO (Figura 3.7). Essa classe tem como sub-classes os tipos de materiais criados na versão anterior, incluindo mais um material: o concreto armado (*Reinforced Concrete*). A Figura 3.8 mostra a interface gráfica utilizada para criar um material do tipo concreto armado no FTOOL.

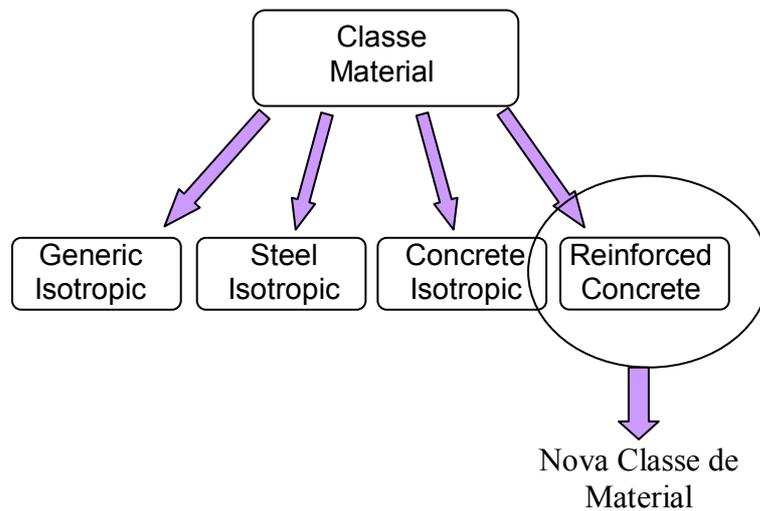


Figura 3.7 – Classes relativa ao material.

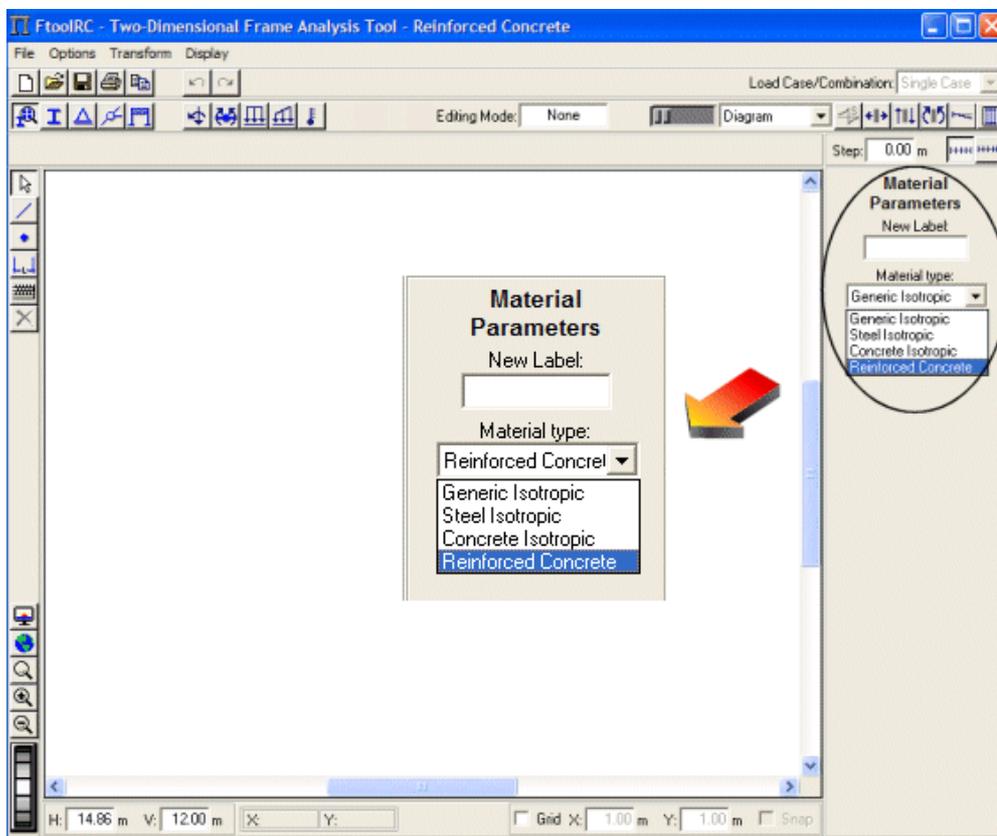
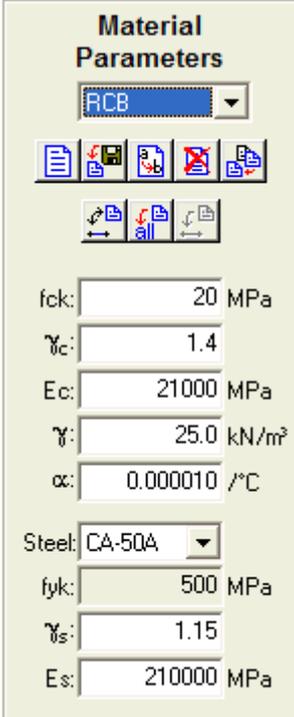


Figura 3.8 – Interface gráfica para seleção de um novo material.

A Figura 3.9 mostra todos os parâmetros referentes ao concreto e ao aço necessários para o dimensionamento de uma viga de concreto armado.



Material Parameters

RCB

fck: 20 MPa

γ_c : 1.4

Ec: 21000 MPa

γ : 25.0 kN/m³

α : 0.000010 /°C

Steel: CA-50A

fyk: 500 MPa

γ_s : 1.15

Es: 210000 MPa

Figura 3.9 – Parâmetros do material concreto armado.

Segundo a norma ABNT NBR 6118, 2003 [3], a resistência característica do concreto, f_{ck} , não deve admitir em estruturas valores menores do que $20MPa$. Os valores dos coeficientes de minoração γ_c e γ_s são determinados pelo usuário e geralmente assumem valores que estão indicados na Tabela 3.1. Por uma questão de simplicidade de interface com o usuário optou-se por colocar estes coeficientes, que são parâmetros de dimensionamento, junto com os parâmetros dos materiais.

Tabela 3.1 – Coeficientes de minoração

Combinações	γ_c	γ_s
Normais	1.4	1.15
Especiais ou de construção	1.2	1.15
Excepcionais	1.2	1.0

O módulo de deformação longitudinal tangente do concreto segundo a norma ABNT NBR 6118, 2003 [3] é calculado de acordo com:

$$E_c = 5600 \cdot \sqrt{f_{ck}} \quad (3.1)$$

O módulo secante é dado por:

$$E_{cs} = 0.85 \cdot E_c \quad (3.2)$$

O valor inicial do módulo de elasticidade que aparece na interface do programa quando o material concreto armado é selecionado está de acordo com a eq. (3.2). Ele pode ser editado a qualquer momento pelo usuário.

O módulo de elasticidade, o peso específico (γ) e o coeficiente de dilatação térmica (α) são parâmetros comuns a todos os tipos de materiais.

Os tipos de aço que aparecem são selecionados através de uma lista expansível, conforme a Figura 3.10. O prefixo *CA* indica aços para concreto armado e o número é o valor da tensão de escoamento característica, f_{yk} , expresso em kN/cm^2 . Os aços classe *B* não são mais utilizados. Eles foram deixados no programa para uma eventual verificação de comportamento de estruturas existentes.

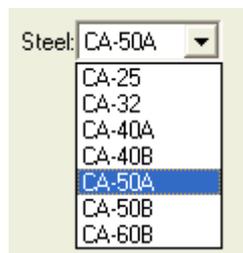


Figura 3.10 – Tipos de aço.

Segundo a norma ABNT NBR 6118, 2003 [3], na falta de ensaios ou valores fornecidos pelo fabricante, o módulo de elasticidade longitudinal dos aços para concreto armado pode ser admitido igual a $210GPa$. Este valor pode ser editado pelo usuário.

3.3.2

Seções Transversais de Vigas de Concreto Armado

Da mesma maneira que ocorre com os materiais, existem várias operações conceituais que são comuns para todos os tipos de seções transversais. Essas operações incluem, entre outras, o cálculo da área da seção transversal e do seu momento de inércia, além da criação de elementos de interface para receber os parâmetros da seção transversal. Na versão anterior do FTOOL existiam vários tipos de seções transversais, cada uma com seus respectivos parâmetros. Assim, já existia uma “classe seção transversal” no contexto de POO (Figura 3.11). Foi necessário somente acrescentar nesta classe as quatro seções transversais referentes ao dimensionamento de vigas de concreto armado: retangular (*RC-Beam Rectangle*), T (*RC-Beam T-shape*), L (*RC-Beam L-shape*) e I (*RC-Beam I-shape*). A Figura 3.12 mostra a interface gráfica para seleção de uma seção transversal retangular de concreto armado.

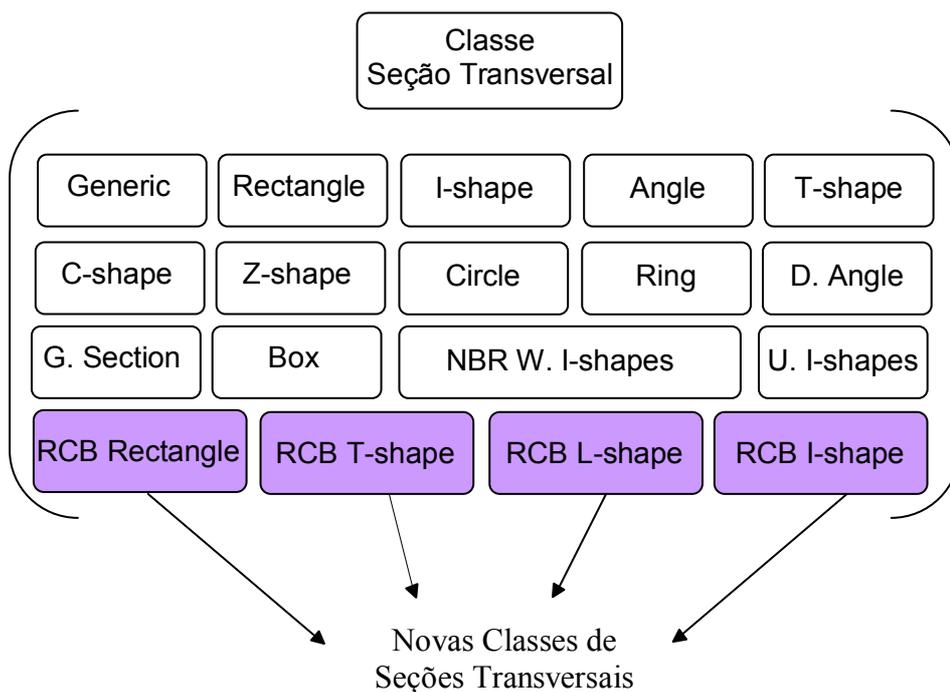


Figura 3.11 – Classes relativas às seções transversais.

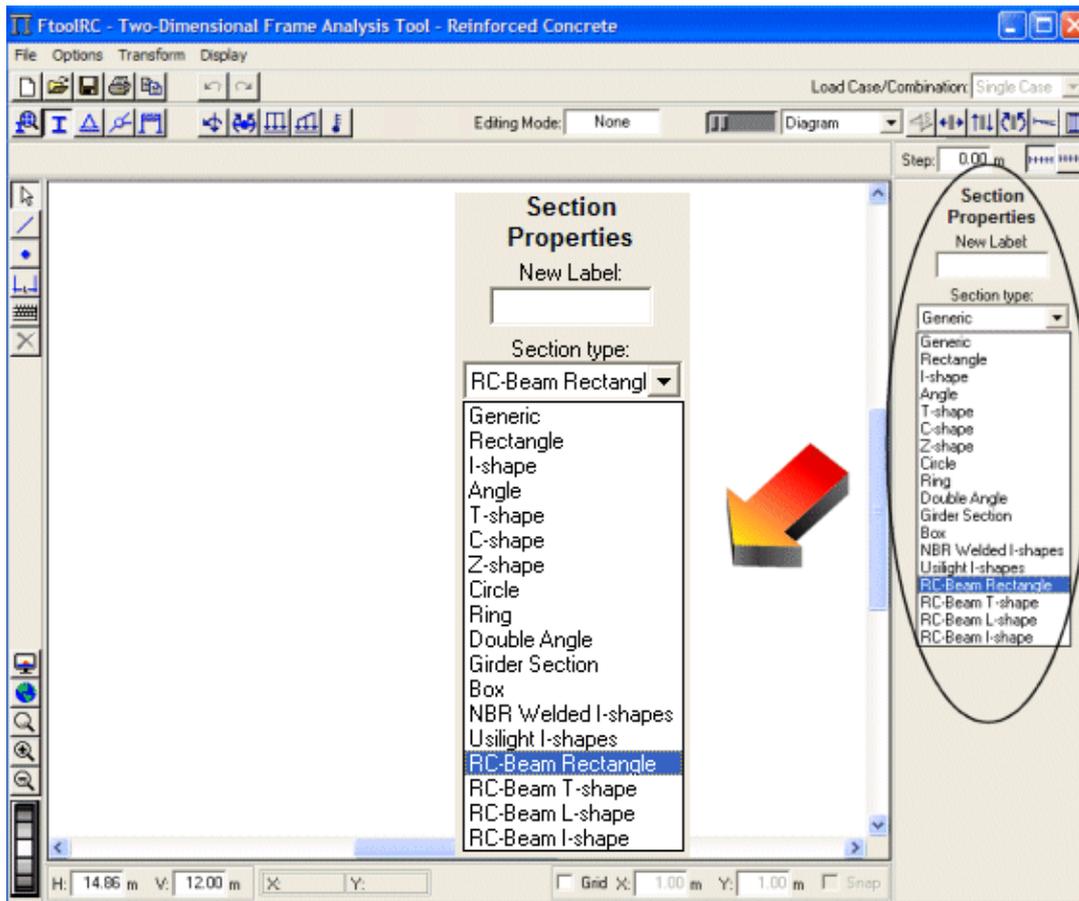


Figura 3.12 – Interface gráfica para seleção de seções transversais.

De acordo com o tipo de seção selecionada, na área lateral da tela deverão ser fornecidos os parâmetros que definem cada seção transversal. A Figura 3.13 mostra todos os parâmetros relativos às quatro seções transversais implementadas, necessários para o dimensionamento de uma viga de concreto armado. O usuário deve fornecer os valores para todos os parâmetros, com exceção dos três últimos, conforme explicado a seguir.

The image displays four panels for 'Section Properties' of reinforced concrete beams. Each panel includes a diagram of the cross-section and a list of parameters and calculated properties.

Section Properties	RCB-Rect	RCB-Tshape	RCB-Lshape	RCB-Ishape
Diagram	Rectangular	T-shaped	L-shaped	I-shaped
ϕ_{12}	16.0 mm	20.0 mm	12.5 mm	16.0 mm
ϕ_{11}	12.5 mm	20.0 mm	8.0 mm	10.0 mm
ϕ_t	6.3 mm	6.3 mm	8.0 mm	6.3 mm
Spacing	10.0 cm	10.0 cm	15.0 cm	10.0 cm
Single/Double	Single	Single	Single	Single
d_1	10.0	10.0	3.0	5.0
d_2	5.0	5.0	3.0	3.0
d_1	5.0	5.0	3.0	30.0
d_2	10.0	10.0	3.0	5.0
h	110.0	110.0	90.0	100.0
b	20.0	135.0	65.0	70.0
c	3.0	5.0	2.5	2.5
\bar{y}	55.0	73.9	55.8	50.0
A	2.20e+03	3.58e+03	1.85e+03	3.00e+03
I	2.2183e+06	4.2710e+06	1.4992e+06	3.7000e+06

Figura 3.13 – Parâmetros das seções transversais referentes ao dimensionamento de vigas de concreto armado.

Os parâmetros ϕ_{11} e ϕ_{12} correspondem às bitolas da armadura longitudinal inferior e superior e são selecionados através de uma lista expansível, conforme a Figura 3.14.

The image shows two expandable lists for selecting longitudinal reinforcement diameters. The first list is for ϕ_{11} (bottom reinforcement) and the second is for ϕ_{12} (top reinforcement).

ϕ_{11}	ϕ_{12}
12.5 mm	8.0 mm
6.3 mm	6.3 mm
8.0 mm	8.0 mm
10.0 mm	10.0 mm
12.5 mm	12.5 mm
16.0 mm	16.0 mm
20.0 mm	20.0 mm
25.0 mm	25.0 mm
32.0 mm	32.0 mm

Figura 3.14 – Diâmetros da armadura longitudinal inferior (ϕ_{11}) e superior (ϕ_{12}).

Além da escolha da bitola, o usuário também tem a possibilidade de escolher de quantas em quantas barras deseja desenhar a armadura (escalonamento). Nos exemplos apresentados no próximo capítulo o significado deste parâmetro ficará mais explícito.

O parâmetro ϕ_t , que corresponde ao diâmetro das barras dos estribos, também pode ser selecionado através de uma lista expansível, de acordo com a Figura 3.15. Usualmente, essas armaduras são formadas por estribos verticais de dois ramos (estribos simples). Se necessário, o usuário pode optar pela escolha de estribos duplos que possuem quatro ramos, tendo, portanto, o dobro da seção dos estribos simples.

Uma vez escolhida a bitola dos estribos, deve-se escolher a variação do seu espaçamento, conforme a Figura 3.15. O programa calcula automaticamente, de acordo com o item 2.10.6 o espaçamento necessário entre os estribos sendo que o usuário pode variar este espaçamento.

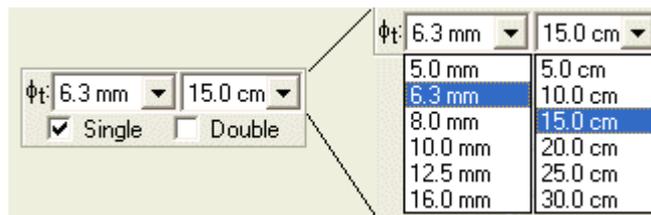


Figura 3.15 – Diâmetros da armadura transversal (ϕ_t) e espaçamentos.

Os parâmetros d'_1 e d'_2 , fornecidos pelo usuário, correspondem à distância do centro de gravidade da armadura longitudinal às bordas inferior e superior da seção, respectivamente, para momentos positivos e negativos.

Para cada tipo de seção transversal, o usuário deve fornecer os parâmetros geométricos da seção bruta (b, b_w, h, h_f, h_1 e h_2). Os campos \bar{y} (distância do centróide à borda inferior da seção), A (área da seção) e I (momento de inércia em relação ao centro geométrico da seção) são calculados automaticamente pelo programa.

Finalmente, o cobrimento da armadura c deve ser fornecido.

3.3.3

Verificações Realizadas para o Dimensionamento e Prescrições da NBR 6118, 2003

Segundo a norma ABNT NBR 6118, 2003 [3], o cobrimento nominal c de uma determinada barra deve ser, no mínimo, igual ao diâmetro da própria barra. A Tabela 3.2 mostra o valor do cobrimento nominal em função da classe de agressividade ambiental:

Tabela 3.2 – Cobrimento nominal

Componente ou elemento	Classe de agressividade ambiental			
	I	II	III	IV
	Cobrimento nominal (c) mm			
Viga	25	30	40	50

A agressividade do meio ambiente está relacionada às ações físicas e químicas que atuam sobre as estruturas de concreto, sendo I uma classe de agressividade fraca, II moderada, III forte e IV muito forte.

O diâmetro máximo do agregado graúdo utilizado no concreto não pode superar em 20% a espessura nominal do cobrimento, ou seja:

$$d_{\max} \leq 1.2 \cdot c \quad (3.3)$$

O diâmetro das barras deve ser escolhido procurando-se o menor desperdício. Deve-se dar preferência às barras mais finas, pois estas são favoráveis quanto à fissuração e ancoragem. Entretanto, um número elevado de barras de pequeno diâmetro pode não caber na largura da seção da viga, já que deve existir um espaçamento mínimo entre as barras. Nesse caso, as barras devem ser dispostas em mais de uma camada.

Para garantir que todas as barras da armadura longitudinal sejam envolvidas pelo concreto, evitando-se falhas de concretagem, devem ser respeitados os espaçamentos indicados na Figura 3.16 e a norma ABNT NBR 6118, 2003 [3].

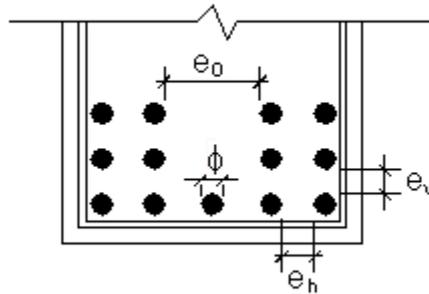


Figura 3.16 – Espaçamento mínimo das barras, adaptada de ARAÚJO [15].

O espaçamento mínimo das barras nas camadas horizontais é dado por:

$$e_h \geq \left\{ \begin{array}{l} 2cm \\ \phi \\ 1.2 \cdot d_{\max} \end{array} \right\} \quad (3.4)$$

onde ϕ é o diâmetro das barras e d_{\max} é o diâmetro máximo do agregado.

No plano vertical, deve-se respeitar o espaçamento mínimo:

$$e_v \geq \left\{ \begin{array}{l} 2cm \\ \phi \\ 0.5 \cdot d_{\max} \end{array} \right\} \quad (3.5)$$

Uma vez conhecidos o diâmetro das barras, o espaçamento entre as barras e o cobrimento, pode-se determinar o número máximo de barras que podem ser dispostas em uma mesma camada na seção da viga. Quando as barras estiverem dispostas em várias camadas, é aconselhável deixar um espaço livre e_0 para a passagem da agulha do vibrador.

De acordo com a ABNT NBR 6118, 2003 [3], se as barras estiverem dispostas em mais de uma camada, a distância y_0 (Figura 3.17) do centróide da armadura até a camada mais afastada da linha neutra tem que ser menor ou igual a $10\% \cdot h$. Caso contrário, o programa avisa que deve ser verificada a capacidade resistente da seção.

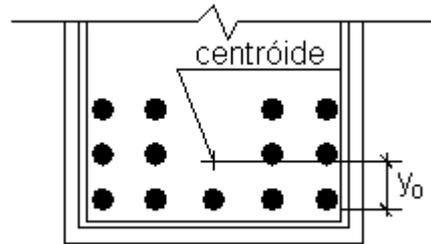


Figura 3.17 – Posição do centróide da armadura.

Além desta verificação, se a soma das armaduras longitudinais exceder a armadura máxima permitida, segundo a eq. (2.142), o programa exibe uma mensagem alertando que o usuário precisa modificar as dimensões da viga ou até mesmo aumentar o valor da resistência característica à compressão do concreto (f_{ck}).

No caso da armadura transversal, o programa verifica se a força cortante solicitante de cálculo é menor que a força cortante resistente de cálculo. Se esta verificação não for atendida, uma nova mensagem é exibida, tendo o usuário que modificar as dimensões da viga.

Dependendo da escolha da variação do espaçamento pelo usuário, este pode ser maior que o espaçamento máximo recomendado. O programa exibe uma mensagem aconselhando o usuário a escolher um espaçamento menor.

3.4

Modificações na Interface Gráfica

A interface gráfica do programa FTOOL precisou sofrer algumas modificações para que o dimensionamento pudesse ser feito.

Primeiramente, foram alterados os menus responsáveis pela entrada de dados referentes ao material utilizado e à seção transversal, de acordo com o item anterior. Tiveram que ser definidas unidades para as novas grandezas envolvidas (área da barra de aço e diâmetro da barra de aço), bem como um formato para a exibição dos valores numéricos associados a estes parâmetros (Figura 3.18).

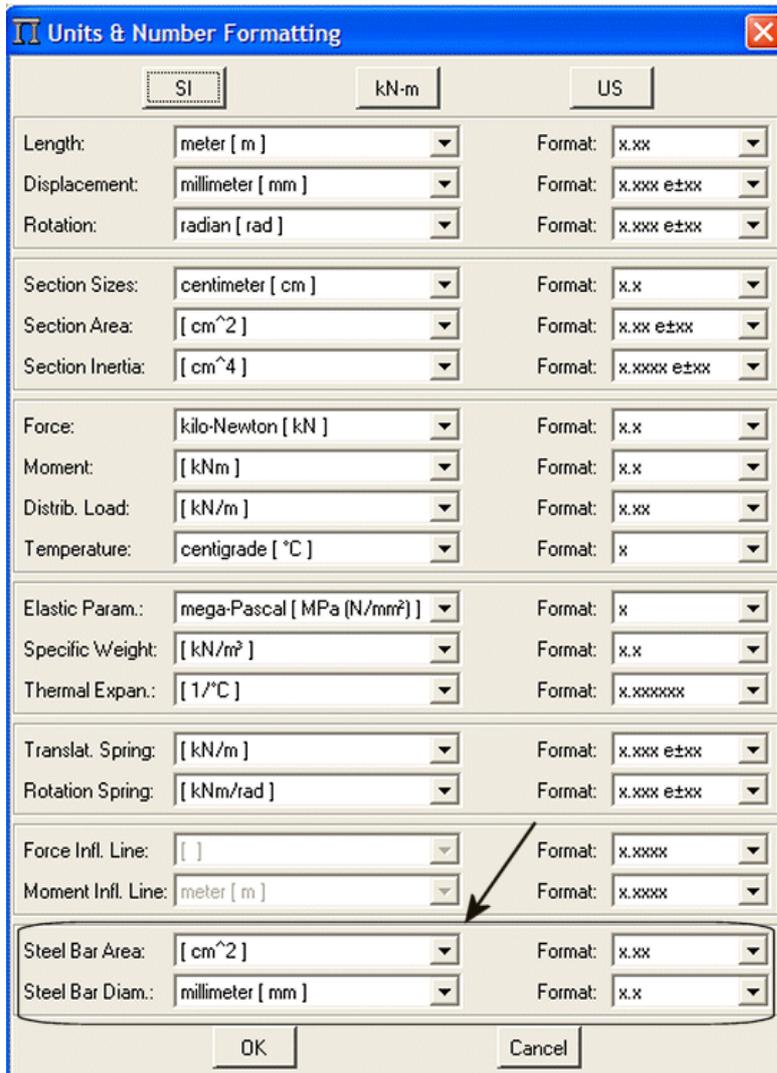


Figura 3.18 – Janela de configuração de unidades e formatos de valores numéricos.

Em seguida, foi criado um botão para acessar o módulo de dimensionamento de concreto armado, conforme a Figura 3.19 e Figura 3.20.

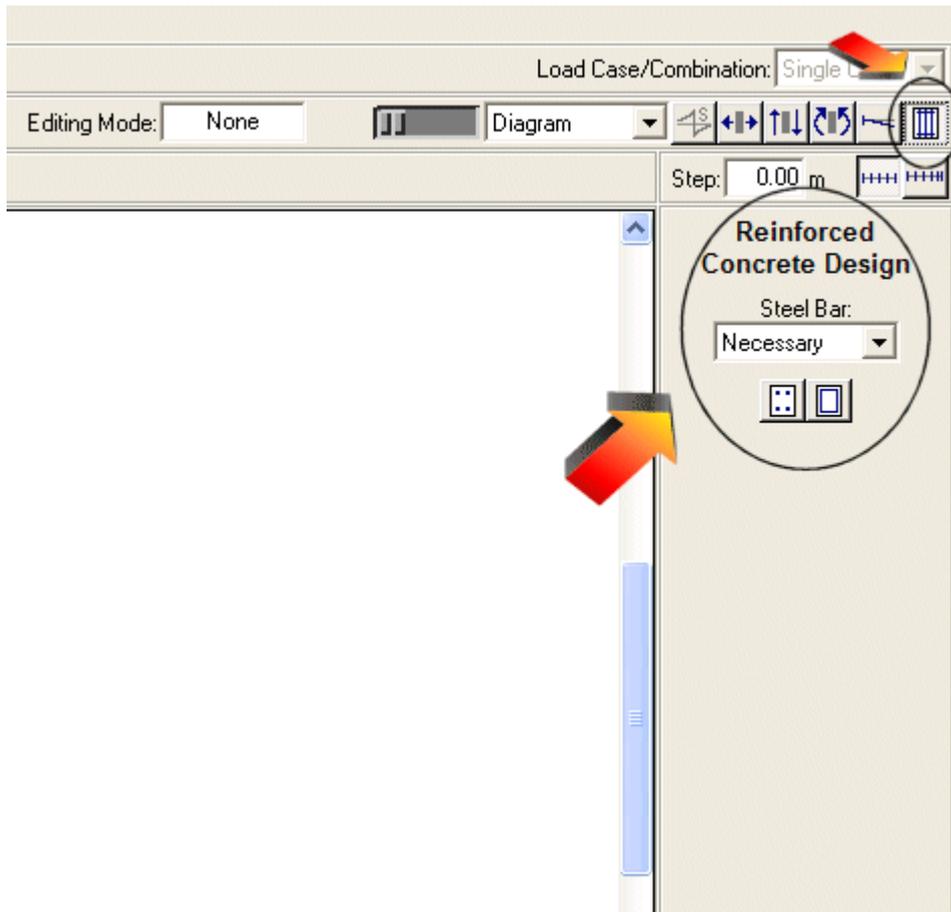


Figura 3.19 – Botão para acessar o módulo de dimensionamento de concreto armado.

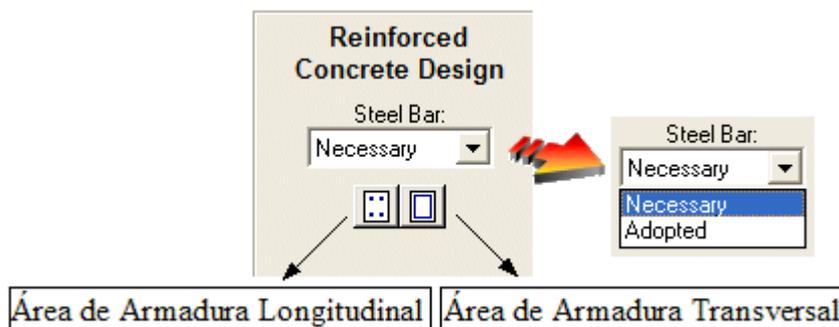


Figura 3.20 – Submenu responsável pelo dimensionamento de concreto armado.

O resultado que o programa desenvolvido neste trabalho oferece são dois diagramas: um deles contendo as áreas de armaduras longitudinais superior e inferior, calculadas para cada par de valores de momento fletor e esforço normal; e outro de área de armadura transversal, calculada para cada valor da força cortante. Existem dois modos de resultados da análise de dimensionamento. A

seleção do modo desejado é feita através de uma lista expansível, conforme a Figura 3.20, sendo o primeiro modo a armadura necessária (*Necessary*) e o segundo a armadura adotada (*Adopted*).

A armadura necessária corresponde ao cálculo de uma armadura para atender a todos os esforços solicitantes. O cálculo da armadura longitudinal é baseado no método de dimensionamento de Ferreira da Silva Junior e o da armadura transversal segue as considerações do modelo I da norma ABNT NBR 6118, 2003 [3], de acordo com o capítulo anterior. No caso da armadura longitudinal o diagrama é desenhado considerando-se a decalagem da armadura, segundo a eq. (2.165).

A armadura adotada corresponde ao cálculo da armadura, baseado na armadura necessária, obedecendo a um diâmetro de armadura, escolhido pelo usuário, e às prescrições da norma. Para armadura longitudinal, este cálculo é feito da seguinte maneira:

- se a armadura necessária for igual a zero, considera-se para a área de armadura adotada, segundo a ABNT NBR 6118, 1978 [9]:

$$A_{adotada} = 2 \cdot A_{\phi,t} \quad (3.6)$$

sendo $A_{\phi,t}$ a área da seção da barra escolhida para armadura transversal;

- se a armadura necessária for diferente de zero e menor que a armadura mínima imposta pela norma brasileira, adota-se para o cálculo da armadura adotada o valor da armadura mínima;
- calcula-se o número de barras, n_b , dividindo o valor da área de armadura necessária (A_{nec}) pela área da seção da barra escolhida ($A_{\phi,l}$):

$$n_b = \frac{A_{nec}}{A_{\phi,l}} \quad (3.7)$$

- se o resultado não for um número inteiro, este valor é arredondado para um número inteiro superior para que o cálculo esteja a favor da segurança;
- multiplica-se este número inteiro pela área da seção da barra escolhida, obtendo assim a armadura adotada ($A_{adotada}$). Este cálculo é feito para cada valor de armadura necessária.

$$A_{adotada} = n_b \cdot A_{\phi,l} \quad (3.8)$$

O diagrama de área de armadura longitudinal adotada considera, além da ancoragem, a decalagem das armaduras (eq. (2.165) e (2.167) respectivamente). Através deste diagrama pode-se ter uma idéia bastante aproximada do tamanho das barras.

No caso dos estribos, o cálculo da armadura adotada é feito da seguinte forma:

- se a armadura necessária for menor que a armadura mínima imposta pela norma brasileira, adota-se para o cálculo da armadura adotada o valor da armadura mínima;
- calcula-se o espaçamento s necessário de acordo com a bitola escolhida, conforme a eq. (2.161);
- se o espaçamento não for um inteiro, este número é arredondado para um número inteiro inferior e múltiplo de cinco;
- de acordo com o valor de espaçamento encontrado calcula-se a armadura adotada $A_{adotada}$:

$$A_{adotada} = n^{\circ} \text{pernas} \cdot \frac{A_{\phi,t}}{s} \quad (3.9)$$

Em qualquer um dos modos, Necessário ou Adotado, “clcando” em um ponto sobre a viga com o botão esquerdo do mouse, aparece na barra de mensagem acima da área de desenho o valor do correspondente diagrama para aquele ponto. Se for usado o botão direito do mouse, aparecem na área lateral informações adicionais sobre a viga com respeito ao diagrama visualizado. Se um passo (*Step*) para consulta de resultados estiver definido, na área lateral serão exibidos os resultados do diagrama ao longo da viga (Figura 3.21). Os valores para cada passo também podem ser mostrados no desenho dos diagramas (Figura 3.22).

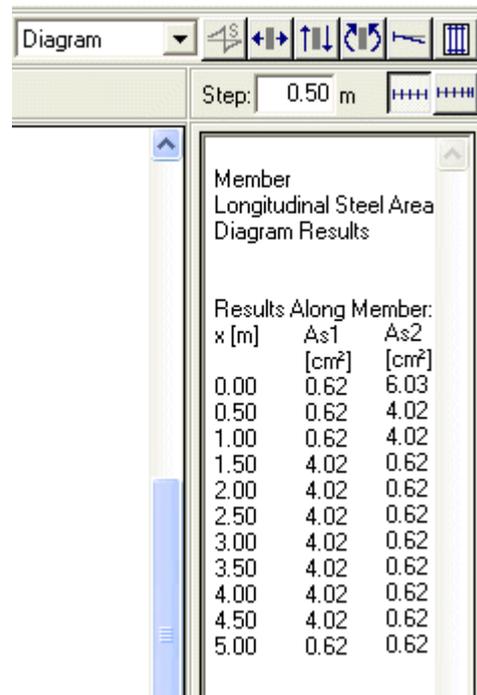


Figura 3.21 – Resultados ao longo da barra, de acordo com o passo (Step) estipulado.

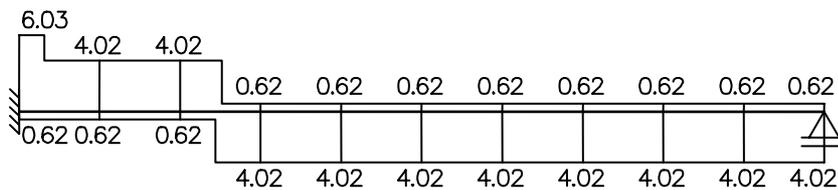


Figura 3.22 – Resultados de cada passo (Step) mostrados no diagrama.

3.5

Algoritmo de Decalagem e Ancoragem da Armadura Longitudinal

Para a implementação do algoritmo de decalagem da armadura longitudinal, a estrutura de dados do FTOOL não precisou sofrer alterações. As implementações foram feitas nos atributos “pendurados” nas arestas da estrutura de dados com a utilização da biblioteca de funções HED. Cada aresta corresponde a um elemento de barra do modelo de análise (Figura 3.23). No programa FTOOL, uma viga deve ser dividida em vários elementos de barra quando houver cargas concentradas no interior do vão ou cargas distribuídas que abrangem parcialmente o vão.

O problema do lançamento da armadura longitudinal, incluindo a decalagem e a ancoragem, é que este deve ser feito em cada vão como um todo. Como, no FTOOL, os resultados da análise (diagramas de esforços internos e configuração deformada) até então eram obtidos em cada elemento de barra isolado, um tratamento especial teve que ser desenvolvido para a armadura longitudinal.

Para tanto, foram criados dois novos algoritmos: um para encontrar automaticamente uma cadeia de elementos de barra que formam um vão; e outro para fazer o lançamento da armadura longitudinal no vão. Para o primeiro algoritmo, são zerados os diagramas de cada elemento de barra. Percorre-se a lista de elementos de barra do modelo. Para cada elemento que não foi tratado, define-se a cadeia de elementos de barra que formam um vão. O algoritmo utilizado para selecionar automaticamente um vão é descrito a seguir.

Partindo de uma dada aresta, percorre-se nos dois sentidos pela adjacência aresta-aresta e testa-se para ver quando duas arestas pertencem a um mesmo vão de uma viga de concreto armado. Para pertencer a um mesmo vão as duas arestas têm que ter o mesmo material (concreto armado) e as mesmas propriedades de seção transversal de concreto armado. Além disso, o vértice comum não pode ter nenhum tipo de restrição de apoio e incidência de outra aresta (bifurcação). O resultado é uma cadeia contínua de arestas. Todos os vértices no interior da cadeia só podem ser usados por duas arestas da cadeia. A cadeia de arestas é limitada por vértices que possuem restrições de apoio ou por arestas que representam pilares (Figura 3.23).

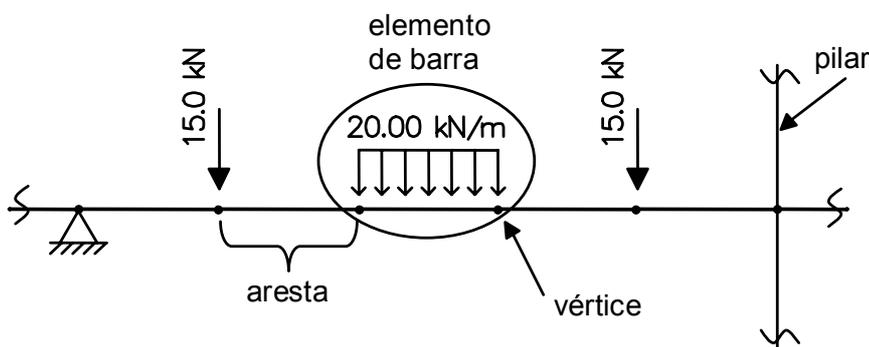


Figura 3.23 – Exemplo de uma cadeia de arestas.

Este algoritmo exemplifica a grande vantagem de se adotar uma estrutura de dados topológica, como é o HED. Esta estrutura provê todas as informações

de adjacência necessárias para definir de forma eficiente e localizada o vão de uma viga contínua, mesmo quando formado por vários elementos de barra.

Para o algoritmo de lançamento da armadura longitudinal, é necessário calcular o comprimento a_l de decalagem da armadura e o seu sentido. Também é necessário calcular o comprimento de ancoragem, dado pela eq. (2.167). Para o cálculo do comprimento a_l , conforme a eq. (2.165), é necessário encontrar o valor da força cortante máxima em cada trecho da viga. Para isso, o diagrama de esforços cortantes é determinado ao longo de cada vão (cadeia de elementos de barra), sendo retirados os valores da força cortante máxima (em valor absoluto) de cada trecho do vão. Para cada seção de um vão identifica-se a força cortante máxima a ser utilizada dependendo da localização da seção no vão.

O diagrama de armadura longitudinal adotada (considerando a decalagem e a ancoragem) é construído de maneira que a posição x de cada valor de armadura longitudinal adotada (superior ou inferior) seja acrescida ou decrescida de um comprimento $a_l + l_b$ no sentido desfavorável. Com isso, obtém-se uma nova abscissa (x_{al+lb}) para cada valor de armadura longitudinal adotada. O sentido desfavorável está relacionado com o sinal dos diagramas de momento fletor (M) e esforço cortante (Q). Se $M > 0$ e $Q > 0$ ou $M < 0$ e $Q < 0$, o valor de x é subtraído de $a_l + l_b$. Se $M > 0$ e $Q < 0$ ou $M < 0$ e $Q > 0$, o valor de x é acrescido de $a_l + l_b$ (Figura 3.24).

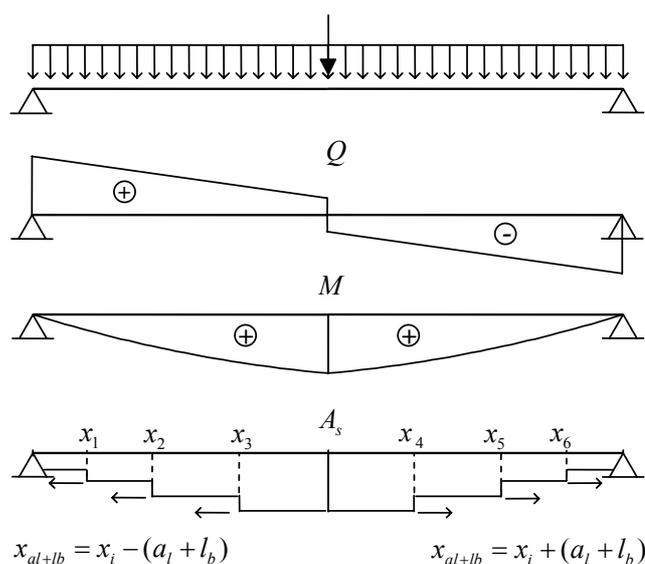


Figura 3.24 – Sentido desfavorável do diagrama de armadura.

A Figura 3.25 mostra esquematicamente como foi feito o algoritmo de lançamento da armadura longitudinal adotada considerando a decalagem e ancoragem da armadura em um vão da viga.

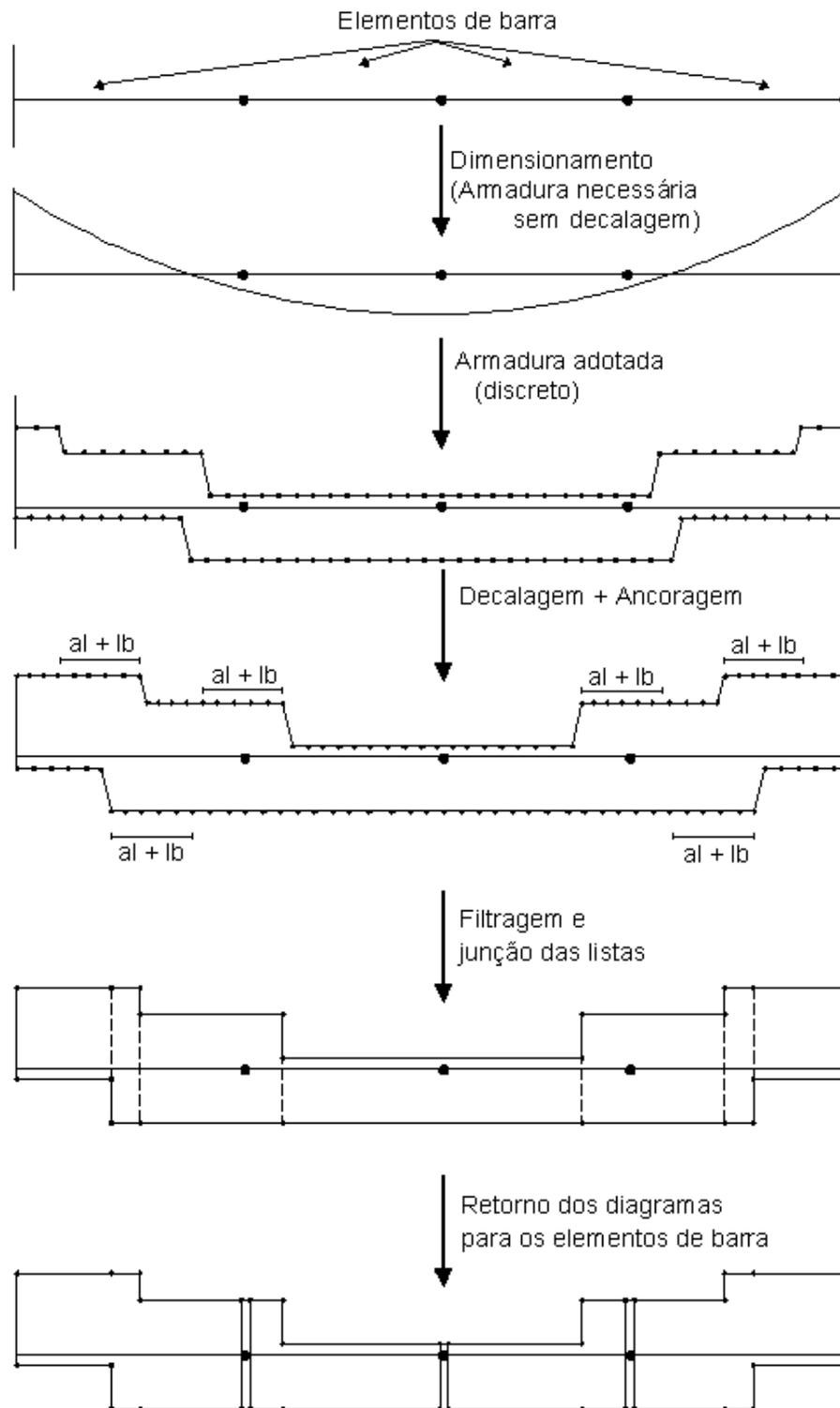


Figura 3.25 – Esquema do algoritmo de detalhamento da armadura longitudinal adotada em um vão.

O algoritmo cria inicialmente duas listas ordenadas independentes (uma para armadura superior e outra para armadura inferior) constituídas pelo valor da área de armadura adotada longitudinal e a posição x deste valor no vão. A partir de cada valor de armadura, a posição x deste valor é acrescida ou decrescida pela soma da decalagem com a ancoragem. Em seguida, o diagrama é filtrado com o objetivo de condensar uma seqüência de elementos consecutivos com o mesmo valor de área de aço em dois elementos extremos desta seqüência, além de tornar verticais as descontinuidades (rampas) existentes nos valores da área de aço (diagrama em degraus). Depois da filtragem ocorre a união das duas listas em uma e, finalmente, os diagramas são retornados para cada elemento de barra (aresta) do vão para que os resultados sejam visualizados da mesma maneira como todos os outros diagramas do FTOOL (desenha-se os resultados de cada elemento de barra de forma independente).

O algoritmo para o traçado do diagrama da armadura longitudinal necessária é semelhante ao algoritmo para a armadura adotada. A diferença é que o diagrama de armadura necessária não é em degraus. Portanto, a etapa de filtragem não é executada.