

**Luiz Liserre**

**SISTEMA COMPUTACIONAL COM GERAÇÃO DE  
DADOS E VISUALIZAÇÃO DE RESULTADOS PARA  
ESTRUTURA DE EDIFÍCIOS**

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do Título de Mestre em Engenharia de Estruturas.

**Orientador:** Prof. Dr. Marcio Antonio Ramalho

São Carlos  
2003

## **AGRADECIMENTOS**

---

Ao Prof. Dr. Marcio Ramalho por ter possibilitado e apoiado a elaboração desse trabalho e pela enorme paciência durante o desenvolvimento do projeto.

Aos professores, funcionários e colegas do Departamento de Engenharia de Estruturas que direta ou indiretamente muito contribuíram para elaboração desse trabalho.

À minha esposa Andréa pelo grande incentivo e compreensão dispensados durante essa fase da minha vida.

À minha família pelo apoio permanente mesmo nos momentos mais difíceis.

À USP e à FAPESP pela disponibilização de recursos que torna possível a elaboração de pesquisas favorecendo o desenvolvimento tecnológico deste país.

## RESUMO

LISERRE, L.;RAMALHO, M. A. (2003). *Sistema computacional com geração de dados e visualização de resultados para estrutura de edifícios*. Dissertação (MESTRADO) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2003.

Este trabalho trata do desenvolvimento de um sistema computacional, para a geração de dados e apresentação de resultados, específico para as estruturas de edifícios. As rotinas desenvolvidas devem trabalhar em conjunto com um sistema computacional para análise de estruturas com base no Método dos Elementos Finitos, contemplando tanto as estruturas de pavimentos; com a utilização de elementos de barra, placa/casca e molas; como as estruturas de contraventamento; com a utilização de elementos de barra tridimensional e recursos especiais como nó mestre e trechos rígidos. A linguagem computacional adotada para a elaboração das rotinas mencionadas é o Object Pascal do DELPHI, um ambiente de programação visual estruturado na programação orientada a objetos do Object Pascal. Essa escolha tem como objetivo, conseguir um sistema computacional onde alterações e adições de funções possam ser realizadas com facilidade, sem que o todo o conjunto de programas precise ser analisado e modificado. Por fim, o programa deve servir como um verdadeiro ambiente para análise de estruturas de edifícios, controlando através de uma interface amigável com o usuário uma série de outros programas já desenvolvidos em FORTRAN, como por exemplo o dimensionamento de vigas, pilares, etc.

Palavras-Chave: estruturas de edifícios; método dos elementos finitos; estruturas de contraventamento; pré-processador; pós-processador.

## ABSTRACT

LISERRE, L.;RAMALHO, M. A. (2003). *A computational system with input data generation and output data visualization for building structures*. M.Sc Dissertation – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2003.

This work deals with a pre and pos data processing computational system specific for building structures. This computational system has been developed to work together with a finite element program for structural analysis, and it must include elements for floor structures analysis, as bars, plates, membranes and springs; and elements for bracing structures, as 3D-bar and rigid elements. Borland's Delphi, a rapid application development environment based on Object Pascal, has been used in this work. The reason for this choice is to provide an easy way to future modifications and additions in the source code. Finally, the developed system should make possible the integration, through user friendly interfaces, with other programs already developed in Fortran, for instance, for designing beams, columns, and others structural elements.

Keywords: building structures, finite element method, bracing structures, pre data processing, pos data processing.

## SUMÁRIO

---

<b>CAPÍTULO 1- APRESENTAÇÃO .....</b>	<b>1</b>
1.1 Introdução .....	1
1.2 Objetivos .....	4
1.3 Justificativas .....	4
 <b>CAPÍTULO 2 – HISTÓRICO.....</b>	 <b>6</b>
 <b>CAPÍTULO 3 – ESTRUTURA GERAL DO SISTEMA.....</b>	 <b>10</b>
3.1 Metodologia .....	10
3.1.1 Programas para Geração dos Dados.....	10
3.1.2 Programas para Visualização de Resultados.....	13
3.2 Concepção do Sistema .....	14
3.3 Características do Delphi <i>versus</i> Estrutura do Sistema .....	20
 <b>CAPÍTULO 4 – MÓDULOS 1 e 2 - SISTEMA MESTRE.....</b>	 <b>24</b>
4.1 Estrutura Básica do Sistema MESTRE .....	24
4.2 MÓDULO 1 – Construção do Modelo Básico – Programa GeraMalha .....	29
4.3 MÓDULO 2 – Construção e Processamento do Modelo Final – Sistema LASER .....	35
 <b>CAPÍTULO 5 – MÓDULO 3 – Estrutura de Contraventamento .....</b>	 <b>45</b>
5.1 Módulo de Contraventamento Considerações Gerais.....	45
5.2 Ações Horizontais .....	50

5.2.1	Ações Devidas ao Vento.....	50
5.2.2	Ações Devidas ao Desaprumo .....	55
<b>CAPÍTULO 6 – MÓDULO 4 – Visualização de Resultados .....</b>		<b>57</b>
6.1	Estrutura Básica do Sistema VISUALIZA3D .....	57
6.2	Visualização Tridimensional – Biblioteca OpenGL .....	65
<b>CAPÍTULO 7 – PROJETO EXEMPLO .....</b>		<b>69</b>
7.1	Introdução .....	69
7.2	Sugestões de Modelagem .....	69
7.2.1	Definição dos Objetivos da Análise .....	71
7.2.2	Definição da Estrutura a ser Modelada .....	72
7.2.3	Escolha do Tipo do Elemento.....	72
7.2.4	Definição da Densidade da Malha .....	75
7.2.5	Demais Etapas do Processo de Análise .....	76
7.3	Estudo da arquitetura e concepção da forma estrutural.....	78
7.4	Sugestões para o pré-dimensionamento dos componentes....	80
7.5	Projeto Exemplo .....	82
7.5.1	Pavimento TIPO .....	71
7.5.2	Pavimento Ático .....	91
7.5.3	Estrutura de Contraventamento .....	93
7.6	Modelagem do Pavimento Tipo através do ANSYS .....	95
7.7	Comentários Finais.....	99
<b>CAPÍTULO 8 – RECOMENDAÇÕES E CONCLUSÕES .....</b>		<b>100</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>		<b>103</b>

## 1. APRESENTAÇÃO

---

### 1.1 Introdução

A análise de edifícios é um tópico de grande importância para engenharia de estruturas. Qualquer desenvolvimento técnico obtido para essa área produz, por efeito de escala, um benefício sensível para toda a sociedade. Isso porque esses edifícios são em número tão expressivo que deixam as outras estruturas em situação de menor destaque.

Até muito pouco tempo as estruturas de edifícios eram analisadas considerando-se separadamente todas as suas peças componentes. Assim sendo, um pavimento de edifício, por exemplo, era dividido em lajes que se apoiavam em vigas e que por sua vez iam descarregar em pilares. Todos esses componentes eram considerados como se trabalhassem isolados uns dos outros, facilitando sobremaneira a determinação dos esforços nas peças. Entretanto, o comportamento real dos pavimentos era bem diferente dessa idealização, produzindo algumas distorções de grande importância entre os valores calculados e aqueles que efetivamente estariam presentes na estrutura.

Com o aparecimento dos microcomputadores, tornou-se viável a aplicação de procedimentos mais sofisticados que consideram a interação entre esses elementos. Dessa forma, pouco a pouco as estruturas calculadas foram se aproximando do seu comportamento real, através da consideração do comportamento conjunto dos seus elementos componentes.

Assim, ainda para o caso dos pavimentos, as vigas, inicialmente consideradas isoladas, passaram a ser consideradas como um só conjunto, sendo todo o vigaento analisado como uma grelha. Outra melhoria significativa para os pavimentos foi a utilização do Método dos Elementos Finitos, que permitiu a inclusão das lajes e dos pilares no sistema a ser analisado. Dessa forma, todos esses elementos passaram a interagir, produzindo resultados mais verossímeis para a estrutura analisada.

Já para as ações horizontais os avanços foram semelhantes. Inicialmente, devido às dificuldades inerentes a essa análise, essas ações eram desconsideradas para a grande maioria dos casos. Quando não era possível ignorá-las, eram considerados apenas pórticos planos como componentes das estruturas de contraventamento, mesmo sendo o carregamento adotado para cada um deles um problema de difícil solução. Isso se dava pois a ação total atuante segundo uma determinada direção tinha que ser distribuída aos pórticos participantes de acordo com a rigidez de cada um. E a interação entre esses sistemas era difícil de ser considerada, em especial quando a estrutura não apresentava simetria em relação a essa direção.

Também nesse caso a evolução dos procedimentos de análise trouxe uma solução muito mais eficiente. A consideração da estrutura de contraventamento como sendo um pórtico tridimensional permite que todas as interações possam ser consideradas corretamente. O programa a ser utilizado, no entanto, deve possibilitar alguns recursos especiais, como o nó mestre, para a modelagem dos diafragmas rígidos que representam os pavimentos, e os trechos rígidos, para a consideração de pilares extensos.

Entretanto, é de suma importância ressaltar que, para se realizar análises eficientes com a consideração desses modelos, é necessário o desenvolvimento de ferramentas específicas para a geração dos dados e visualização dos resultados. Essas ferramentas devem permitir a utilização dessas técnicas mais sofisticadas sem grandes complicações para os usuários.

Aqueles que já estiveram envolvidos com essas análises sabem que a utilização de pacotes computacionais genéricos não é totalmente viável. Mesmo com os recursos de geração e visualização de resultados neles existentes, torna-se pouco confortável a análise das referidas estruturas devido as particularidades que necessariamente devem ser consideradas. Assim, é de grande importância que possam ser desenvolvidas essas ferramentas, possibilitando um avanço cada vez mais expressivo dos procedimentos de análise de edifícios. Nesse caso melhora-se a segurança dos dimensionamentos e também a economia dos diversos sistemas construtivos envolvidos.

Exatamente com esse intuito é que se propõe este trabalho, o de se desenvolver todo um sistema de geração de dados e visualização de resultados voltados especificamente para as estruturas de edifícios.

Finalmente, ressalta-se que, para a elaboração desses programas, é extremamente importante a escolha da linguagem a ser utilizada. Ela deve permitir que a interface com o usuário seja a mais amigável possível, de forma a facilitar ao



máximo a utilização de seus recursos e ainda permitir que modificações e adições posteriores possam ser feitas com grande eficiência.

Para oferecer essas qualidades, o ambiente de programação a ser utilizado deve ser do tipo RAD, sigla de "*Rapid Application Development*" encontrado nas linguagens visuais, e deve suportar também a programação orientada a objeto, em inglês "*Object Oriented Programming*", ou simplesmente OOP.

Um ambiente RAD combina mais de uma função associada às tarefas de programação, tais como: editor de código, módulo para montagem de formulários, caixas de recursos, compilador, depurador, etc. Oferece também, um conjunto completo de elementos de interface com o usuário pronto para ser utilizado na elaboração de aplicativos conhecidos como GUIs "*Graphical User Interface*", tais como: menus, caixas de ferramentas, controle da movimentação do mouse, botões de ação, quadros de opções, quadros de diálogo para abertura e gravação de arquivos, etc.

Já a OOP tem como objetivos principais propiciar a reutilização de código e garantir a organização dos dados e códigos para facilitar futuras modificações ou implementações. Isso é possível através da propriedade da hereditariedade, do encapsulamento de dados, da utilização de métodos virtuais, e do polimorfismo com ponteiros que a OOP nos fornece.

Dentre as linguagens que atendem plenamente os requisitos citados anteriormente, pode-se citar os produtos da Inprise (ex Borland) como é o caso do C++Builder (linguagem C++), do JBuilder (linguagem Java++) e do Delphi (linguagem Object Pascal).

Apesar das grandes vantagens da linguagem Java, em permitir o desenvolvimento de aplicativos para serem executados em qualquer sistema operacional e também do conjunto completo de ferramentas para a Internet que ela oferece, optou-se em utilizar uma linguagem que oferecesse um maior desempenho para desenvolvimento de aplicações gráficas. Com isso restaram as linguagens C++ e a Object Pascal, sendo escolhida esta última por possuir uma sintaxe mais simples e por permitir o aproveitamento imediato de códigos de trabalhos anteriores na linguagem Pascal que foram utilizados neste trabalho.

## 1.2 Objetivos

O principal objetivo deste trabalho é desenvolver um sistema computacional para a geração de dados e apresentação de resultados, específico para as estruturas de edifícios. As rotinas desenvolvidas devem trabalhar em conjunto com um sistema computacional para análise de estruturas com base no Método dos Elementos Finitos, RAMALHO(1990), contemplando tanto as estruturas de pavimentos; com a utilização de elementos de barra, placa/casca e molas; como as estruturas de contraventamento; com a utilização de elementos de barra tridimensional e recursos especiais como nó mestre e trechos rígidos.

Conforme já mencionado, a linguagem computacional adotada para a elaboração das rotinas é o Object Pascal, utilizando o ambiente de programação visual Delphi. Esse ambiente de programação é extremamente amigável ao desenvolvedor e fornece todos os recursos de uma linguagem visual com suporte à programação orientada a objeto.

Por fim, o programa deve permitir futuras implementações para que sirva como um verdadeiro ambiente para análise de estruturas de edifícios, permitindo o controle, através de uma interface amigável com o usuário, de uma série de outros programas já desenvolvidos em FORTRAN, como por exemplo o dimensionamento de vigas, pilares, etc.

## 1.3 Justificativas

Na Engenharia atual, em especial na Engenharia de Estruturas e no desenvolvimento de projetos, a utilização de computadores se faz obrigatória. Também é evidente a necessidade de que os modelos sejam aperfeiçoados de forma que o comportamento das estruturas possa ser avaliado de forma cada vez mais acurado.

Essa complexidade dos modelos leva a uma evidente maior complexidade dos dados de entrada e também na necessidade que recursos especiais para a observação dos resultados obtidos. É completamente impensável se fornecer dados para todos os nós e elementos que deverão constituir o modelo e obter resultados em termos de listagens de valores numéricos para deslocamentos nodais e esforços nas extremidades de elementos.

Por esses motivos, todos os programas de análise de estruturas que são voltados para uma utilização mais profissional possuem geradores de dados e recursos especiais para a visualização de resultados, como por exemplo o SAP90(1991), o SUPERSAP, o ANSYS(1996) e o NASTRAN(1995). Até mesmo

programas que tem uma utilização mais voltada para o ambiente acadêmico, como o LUSAS(1995), também possuem esses recursos que, conforme já se mencionou, são fundamentais para a viabilidade de sua utilização.

Entretanto, os usuários desses sistemas, que costumam analisar edifícios, sabem que a utilização dos mesmos para esse tipo de estrutura está longe de ser confortável. Diversos detalhes específicos fazem com que os edifícios sejam muito mais facilmente tratados quando se dispõem de funções especiais. A geração do pavimento, que normalmente possui aberturas, a colocação de vigas, pilares e o próprio carregamento proveniente de paredes são exemplos típicos que podem ser citados. Além disso, a rapidez e a confiabilidade do modelo aumentam bastante, se a geração do pavimento servir para a definição automática do modelo da estrutura de contraventamento; incluindo seus detalhes intrínsecos, como os nós mestres para a simulação dos diafragmas e os trechos rígidos para as vigas.

Exatamente pelos motivos ressaltados nos parágrafos anteriores é que se justifica a importância deste trabalho, no qual pretende colocar num só sistema todas essas ferramentas de grande importância e, o que é mais importante, possibilitando que atualizações e adições de recursos possam ser executadas com relativa facilidade. Isso fará com que programas já desenvolvidos para determinadas funções específicas tenham uma utilização muito maior, evitando-se que caiam em desuso pela dificuldade de aperfeiçoamento e utilização isolada. Além disso, há ainda o fato de se estar desenvolvendo um aplicativo para as plataformas Windows 95/98 e Windows 2000/NT evitando também, o desuso dos programas anteriormente desenvolvidos que dependem do sistema operacional DOS, que é uma interface cada vez menos utilizada e conhecida dos novos usuários.

## 2. HISTÓRICO

---

O MEF - Método dos Elementos Finitos, ou FEM – *Finite Element Method*, surgiu no início da década de 50 e hoje constitui um método de grande aplicação nas mais diversas áreas, tais como: estruturas metálicas, concreto armado, geotecnia, automobilística, mecânica naval, aeronáutica, etc. Entretanto, apesar das inúmeras vantagens do Método dos Elementos Finitos, os usuários deparam-se com um grande inconveniente: a grande quantidade de dados de entrada.

Segundo HABER et al. (1981), normalmente 80% do tempo gasto na análise de estruturas calculadas pelo Método dos Elementos Finitos é consumido pela entrada dos dados quando se utiliza o método convencional. Por esse motivo tem-se continuamente procurado desenvolver e aperfeiçoar programas para geração automática de malhas.

Como cita SOUZA JR. (1996), as primeiras técnicas usadas para desenvolver um gerador de malhas datam da década de 70, e já em 1976 podia-se listar aproximadamente 250 programas de geração de malha, B. FREDRIKSSON e J. MACKERLE apud GHASSEMI(1982). Segundo a pesquisa bibliográfica feita neste trabalho, esse número não para de crescer comprovando a dificuldade em se fazer um gerador de malha que venha a satisfazer plenamente o usuário final.

Vários pesquisadores dedicaram-se a partir de 1970 no desenvolvimento dessas ferramentas. Segundo levantamento feito por BATISTA (1994), os geradores de malhas desenvolvidos empregavam uma, ou combinação de duas das seguintes técnicas tradicionais SHEPHARD & YERRY (1983):

- Triangulação Automática CAVENDISH (1974), CAVENDISH et. al. (1985), SADEK (1980);
- Transformação de coordenadas BUELL & BUSH (1973), DUROCKER & GASPER (1979), GHASSEMI (1982), ZIENKIEWICZ & PHILIPS (1971);
- Procedimentos de suavização CAVENDISH (1974), CAVENDISH et. al (1985);

- Funções Combinadas GOLDMAN (1983), GORDON (1983), HABER et. al. (1981).

Ainda segundo BATISTA (1994), das quatro técnicas listadas anteriormente, transformação de coordenadas e funções combinadas são as mais utilizadas por produzirem malhas melhor distribuídas.

BATISTA (1994) também pesquisou entre os “softwares” para o cálculo de pavimentos através do Método dos Elementos Finitos, aquele que possuísse um gerador eficiente. Foram então pesquisados os programas SUPERSAP(1987), o COSMOS/M (1989) e o ANSYS (1990).

Concluiu-se que o SUPERSAP trabalha com sub-regiões, ou seja, para um pavimento com uma forma geométrica qualquer necessita-se que o mesmo seja discretizado em zonas quadrilaterais. Definindo-se a dimensão do elemento e tendo sido fornecido o número de divisões segundo os eixos X e Y, parte-se para gerar a rede de pontos através da divisão do quadrilátero básico. Esse programa admite também a existência de sub-regiões triangulares, só que na discretização automática de tais regiões, ocorre uma concentração de elementos num dos vértices do elemento, necessitando uma alteração manual muito trabalhosa.

Já o COSMOS/M trabalha com uma geração da geometria bastante aprimorada, possuindo alternativas automáticas de geração tais como: pontos, curvas (reta, arco, círculo, parábola, elipse e hipérbole), superfícies e volumes, etc. Com a definição das curvas definem-se as superfícies geradas por essas curvas e finalmente é feita a geração automática de pontos em cada superfície. O sistema ANSYS funciona de forma similar ao COSMOS/M.

Além dos exemplos citados anteriormente encontra-se à disposição, dentro do conjunto de pesquisas realizadas sobre o assunto no Departamento de Engenharia de Estruturas da Escola de Engenharia de São Carlos-USP, o programa de SOUZA JR. (1996). Nesse trabalho, desenvolveu-se um método de geração de malha utilizando a técnica de triangulação automática em que a estrutura é dividida em macro elementos. Essa técnica exige pouco tempo de processamento e define elementos bem distribuídos.

Além disso, SOUZA JR. (1996) desenvolveu várias ferramentas para a construção de malhas, que vão desde a construção da poligonal de contorno, discretização da malha em quadriláteros simples ou formados pela junção de dois triângulos, até a geração de vigas.

Tem-se também o programa desenvolvido por REZENDE (1990), onde se propôs um método que consiste em gerar malhas quadrilaterais onde pode haver a junção de duas ou mais malhas até formar o desenho desejado.

E dispõe-se, principalmente, do programa Geração de Redes Básicas para Pavimentos, desenvolvido originalmente por BAPTISTA(1994) em C. Esse programa trata da geração da rede básica de nós e elementos de placa ou casca para o pavimento de um edifício através do fornecimento de poligonais que descrevem contorno externo e contornos internos (vazios), além de pontos e segmentos de retas para eventuais posições pré-estabelecidas para nós ou mesmo linhas de nós.

Atualmente o mercado oferece outras soluções não citadas anteriormente. O ANSYS (1996), por exemplo, possui uma ferramenta para refinar a malha discretizada por elementos finitos identificando, a partir dos resultados de uma análise inicial, as regiões críticas com elevado gradiente de tensões. Para essas regiões o ANSYS permite refazer a malha de forma a representar melhor o comportamento físico do modelo.

Como o objetivo deste trabalho não foi o de desenvolver novas técnicas para geração de malha, mas sim de utilizar as técnicas mais adequadas disponíveis até o momento, melhorando e facilitando sua utilização pelo usuário através de recursos gráficos avançados, todas essas pesquisas foram de bastante utilidade.

Um outro problema levantado trata da geração da estrutura de contraventamento. Para isso foi utilizado o trabalho de PEREIRA (1997) que trata de detalhes de modelagem de estruturas de contraventamento de edifícios de concreto armado. PEREIRA (1997) fornece um procedimento para a modelagem das estruturas de contraventamento que utiliza os arquivos que contém as informações sobre cada pavimento. Dessa forma, pode-se gerar em detalhes todo o contraventamento através da conveniente montagem dos pavimentos da edificação.

No que diz respeito à análise de estruturas de contraventamento, a principal preocupação refere-se justamente à geração do modelo. A entrada de dados é a fase mais demorada e sujeita a erros, o que torna necessário o desenvolvimento de geradores com a finalidade de conferir maior rapidez, eficiência e confiabilidade.

Surge então o problema de se identificar, do ponto de vista computacional, qual é o processo mais adequado para realizar essa operação e justificar o processo utilizado por PEREIRA (1997).

PEREIRA (1997) cita que a maioria dos geradores disponíveis consiste basicamente em se fazer um número determinado de cópias ao longo da altura para os nós e elementos que compõem um nível da estrutura. Um caso típico é a geração de todos os nós e barras que compõem uma coluna de pilar a partir da definição geométrica de apenas uma barra e dos nós que a definem. As características elásticas e de conectividade estão associadas à barra e as

restrições aos deslocamentos estão associadas aos nós, sendo automaticamente repetidas para toda a altura. Pode-se citar o SAP90 (1988) como um “software” que se utiliza desse artifício.

Uma outra concepção de geração automática de pórtico tridimensional consiste na reprodução dos dados de um pavimento, ou seja, um modelo básico completo de pavimento é repetido  $np$  vezes, onde  $np$  representa o número de pavimentos do edifício em análise. Segundo BORTOLIN (1991) ganha-se tempo na etapa de aquisição de dados, com a redução do volume dos mesmos devido o reaproveitamento dos dados já fornecidos para o pavimento, e na etapa de conferência, que fica quase totalmente eliminada em função das consistências automáticas.

Por esse motivo é que PEREIRA (1997), introduzindo alguns aperfeiçoamentos, optou por essa última metodologia.

Finalmente, vale a pena destacar que todos os processamentos dos pavimentos em elementos finitos realizados neste trabalho são feitos através do programa LASER – “Linguagem para Análise de Estruturas Reticuladas” RAMALHO (1990), que é um sistema computacional geral em elementos finitos que permite a utilização de oito tipos diferentes de elementos, inclusive os elementos de placa e de barra, que interessam mais diretamente a essa pesquisa.

## **3. ESTRUTURA GERAL DO SISTEMA**

---

### **3.1 Metodologia**

O sistema desenvolvido teve que englobar quatro conjuntos de programas:

- I) Geração de dados
- II) Processamento dos modelos
- III) Visualização de resultados
- IV) Dimensionamento de componentes

Somente os conjuntos I e III são compostos por programas que foram elaborados ou reescritos em Object Pascal. Os conjuntos II e IV são compostos por programas já desenvolvidos em FORTRAN, sendo que no caso do conjunto II, apenas são executados através do programa controlador do sistema para que sua utilização seja facilitada.

A seguir faz-se uma breve descrição dos programas que compõem os grupos I e II, para explicitar melhor como o sistema em questão foi desenvolvido.

#### **3.1.1 Programas para Geração dos Dados**

São os seguintes os programas que dizem respeito à geração de dados para pavimentos e estruturas de contraventamento.

##### *a) Geração de Redes Básicas para Pavimentos – Programa GeraMalha*

Este programa foi originalmente desenvolvido por BAPTISTA(1994), em C, e trata da geração da rede básica de nós e elementos de placa ou casca para o pavimento de um edifício através do fornecimento de poligonais que descrevem o



contorno externo e os contornos internos (vazios), além de pontos e retas para eventuais posições pré-estabelecidas para nós ou mesmo linhas de nós.

É um programa que já atingiu uma boa eficiência de funcionamento, gerando com grande facilidade e rapidez modelos básicos para pavimentos de relativa complexidade. Um item que foi melhorado nessa pesquisa, permite que seus dados de entrada sejam gerados a partir de um arquivo do tipo DXF do AutoCAD, AUTODESK(1992). Dessa maneira, o sistema permite ao usuário fornecer esses dados a partir da própria arquitetura de um pavimento que tenha sido desenhada através do AutoCAD ou qualquer outro aplicativo que produza arquivos no formato DXF.

#### *b) Geração de Elementos, Detalhes Especiais e Carregamento para Pavimentos*

Existe um programa denominado GLS, desenvolvido em FORTRAN, que coloca sobre o modelo básico gerado pelo GeraMalha, as vigas, os pilares, as molas e as cargas pontuais ou lineares, além de executar algumas outras funções especiais como o levantamento ou abaixamento de grupos de elementos de placa de forma a simular trechos independentes do pavimento que se está gerando.

Essa operação gera um arquivo final, que pode ser chamado de modelo final, pronto para o processamento pelo Sistema LASER. Só que para realizar essa tarefa, esse programa necessita dos arquivos com as informações específicas de cada objeto citado, como o arquivo de pilares \*.P, o de vigas \*.V, o de cargas \*.C, e assim por diante. Estes por sua vez, necessitam de ferramentas próprias para serem gerados de forma eficiente.

De posse desses arquivos, o programa GLS extrai a definição do nó inicial e final de cada elemento ou carga a serem gerados. E, a partir desses dados, o programa executa automaticamente uma série de operações que simplificam sobremaneira o processo. Por exemplo, se um pilar é definido do nó i ao nó j o programa realiza as seguintes operações:

- Identifica todos os nós contidos no trecho.
- Coloca, se necessário, um nó adicional no centro.
- Gera a barra, perpendicular ao pavimento, que vai simular o pilar.
- Gera as barras, ou apenas modifica as características geométricas das que já existirem, sobre o trecho considerado de forma a considerar o trecho rígido que simula o comprimento do pilar.
- Realiza rebaixamento ou levantamento de grupos de elementos placa ou casca com a geração automática de elementos barra perpendiculares ao pavimento para simular a ligação entre as partes.

Como se pode perceber, tais operações são absolutamente importantes para a consideração de um modelo bem elaborado e que causam um grande desconforto se não puderem ser implementadas de forma automática. Entretanto, para que se possa utilizar o GLS de uma forma automatizada, é necessário fornecer uma ferramenta que, a partir da malha básica, permita ao usuário a montagem dos arquivos de entrada requisitados pelo programa GLS.

Para atender a essa deficiência é que foi proposto e elaborado um programa no ambiente DELPHI. Esse sistema permite ao usuário definir todo o modelo utilizando o desenho da malha básica no próprio vídeo do computador, oferecendo, para isso, os seguintes recursos:

- Criação e modificação de pontos nodais.
- Criação e modificação de elementos placa ou casca.
- Criação de vigas.
- Criação de pilares.
- Criação de molas.
- Criação de cargas nodais e linhas de cargas nodais.
- Criação de Lajes.
- Definição de espessuras e cargas distribuídas para elementos placa.
- Visualização para o conjunto de elementos gerados, inclusive cargas, com recursos de “zoom”.
- Geração dos arquivos de saída para o programa GLS.

### *c) Geração da Estrutura de Contraventamento*

Para a modelagem das estruturas de contraventamento é fundamental que se possa utilizar os arquivos que contém informações sobre cada pavimento. Somente através da conveniente montagem dos pavimentos da edificação e dados sobre os pés-direito a serem considerados é que se pode gerar todo o contraventamento em detalhes. Esse procedimento encontra-se desenvolvido em PEREIRA(1997) e produz resultados muito satisfatórios, permitindo inclusive que se façam modificações para a retirada e inclusão de barras e nós, modificações de trechos rígidos, etc.

Entretanto, trata-se de um programa escrito em FORTRAN que obtém seus dados de entrada de arquivos em formato ASCII, dificultando a sua utilização. Além disso, tal programa não oferece visualizador gráfico próprio, obrigando o usuário a gerar arquivos no formato DXF e visualizar o resultado através de outros aplicativos. Neste trabalho, boa parte deste programa foi reescrita em DELPHI melhorando significativamente a interface com o usuário.

A diferença é que se decidiu por dar ênfase na entrada de dados de forma amigável e na automatização da montagem dos pavimentos com a visualização tridimensional do conjunto gerado, sem entrar no âmbito da edição ou modificação dos elementos. Por esse motivo o sistema não contempla, diretamente no desenho apresentado no vídeo, recurso para retirada e adição de barras e nós a partir do conjunto básico formado pela sucessão de pavimentos.

No entanto, ele oferece recurso para geração automática do carregamento devido à ação dos ventos, segundo a NBR6123(1988), e ações devidas ao desaprumo, de acordo com a DIN1053 (1974) e da revisão da NBR-6118 (NB-1).

### **3.1.2 Programas para Visualização dos Resultados**

São os seguintes os programas que dizem respeito à visualização dos resultados para pavimentos e estruturas de contraventamento.

#### *a) Visualização de Resultados para Pavimentos – Programa CPLS*

A base para este visualizador de resultados é o programa desenvolvido pelo próprio autor em seu trabalho de iniciação científica, LISERRE(1993). Trata-se de um traçador de curvas de isovalor para momentos, tensões e deslocamentos, de funcionamento muito rápido e eficiente. O programa foi escrito em PASCAL e tem sido extensivamente utilizado desde a sua elaboração, funcionando basicamente para elementos placa ou casca.

Neste trabalho, tal programa foi reescrito em DELPHI para tornar sua interface com o usuário ainda mais confortável e para traçar curvas para o esforço cortante, que pode ser interessante principalmente quando se resolve pavimentos em laje nervurada utilizando-se elementos de placa com espessura equivalente.

Outro ponto de suma importância foi o de acrescentar rotinas para desenhar os diagramas de elementos barras, recurso que o programa não dispunha. Esses diagramas são apresentados através uma visualização tridimensional.

Todos esses resultados podem ser montados em arquivos DXF, de forma que possam ser impressos, na escala desejada, em impressoras ou “plotters”.

#### *b) Visualização de Resultados para Estruturas de Contraventamento*

Neste caso, a base para o programa que foi desenvolvido encontra-se em duas dissertações, PEREIRA(1997) e BARBIRATO(1997). Trata-se de um programa escrito em FORTRAN, que produz desenhos para esforços, momentos

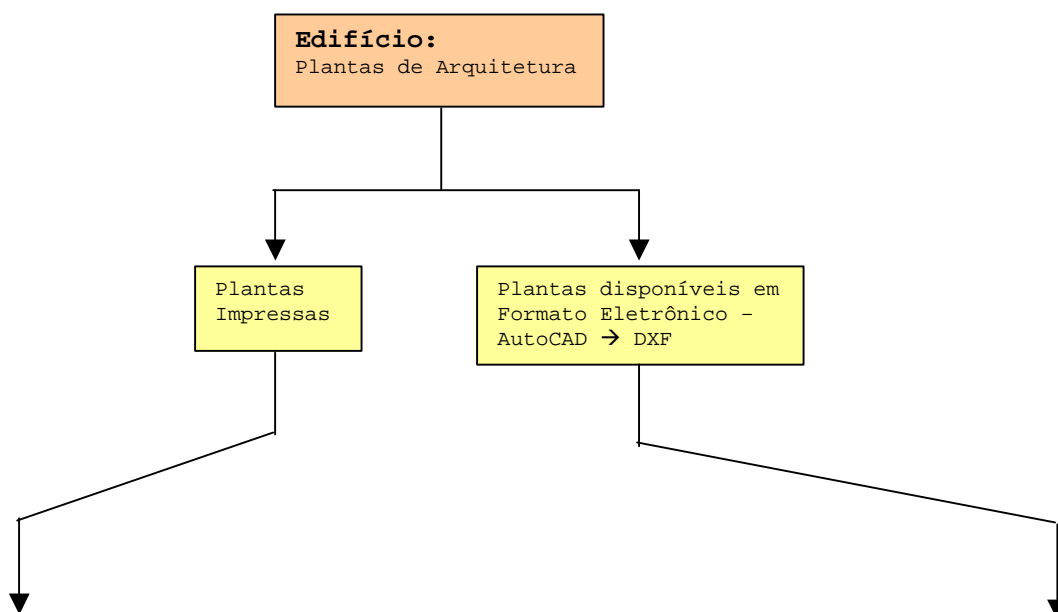
fletores e força cortante segundo os dois eixos principais dos elementos barra tridimensional, e configuração deslocada da estrutura. Todos esses resultados, entretanto, são montados em arquivos tipo DXF para posterior visualização através do AutoCAD. Esse procedimento se deve ao fato do FORTRAN, que fora utilizado, não possuir recursos gráficos convenientes para que a visualização pudesse ser realizada através do próprio programa.

Neste trabalho, a proposta foi de se reescrever as rotinas já existentes no DELPHI de modo a permitir uma melhor interface com o usuário e, principalmente, para que todas as visualizações pudessem ser feitas através do próprio programa. Dessa forma consegue-se uma agilidade muito maior na análise dos resultados, uma vez que possibilita, de forma instantânea, a seleção de vistas que podem ser impressas em escala para compor a memória de cálculo.

### 3.2 Concepção do Sistema

Conforme se pôde perceber pelos tópicos anteriores, o sistema desenvolvido vem suprir uma série de deficiências. Principalmente no que diz respeito à entrada de dados que dependem de geometria, e da integração entre os diversos módulos já existentes que compõem o processo da geração e análise dos pavimentos e da estrutura de contraventamento de um edifício em concreto armado.

Para que se possa ter uma visão global do processo envolvido, a seguir é apresentando um fluxograma, composto por quadros, que representa a base da concepção do sistema.



Entrada da geometria de forma manual

Entrada da geometria através de importação arquivo DXF

### Para cada Pavimento do Edifício

#### Módulo 1 → PROCESSO GMA:

Definição da geometria necessária para montagem do arquivo de entrada para o processamento GMA para geração da Malha Básica

- a)** Recursos para permitir a construção da geometria do modelo básico:  
Sistema de Coordenadas, entrada de coordenadas absolutas e relativas para a construção dos objetos ponto, segmento e contorno.
- Recursos para importar a linha de esqueleto da planta de arquitetura no formato DXF: Sistema de Coordenadas, Conversor de unidades, recursos de captura automática de pontos para a construção otimizada dos objetos ponto, segmento e contorno.
- b)** Recursos para Remoção de objetos ponto, segmento e contorno indesejáveis e criação de novos objetos a partir da captura automática de informações dos já existentes.
- c)** Gerar arquivo de entrada para o programa GMA
- d)** Executar Processamento GMA para gerar o modelo discretizado (Geração da Malha Básica)
- e)** Comunicação com o Módulo 2 para continuação do processo (Inicialização do Módulo 2 - Processo GLS e processamento LASER)

#### Módulo 2 → Processo GLS/LASER:

Montagem do Modelo de Elementos Finitos → Edição/Finalização do modelo discretizado com adição de pilares, vigas, carregamentos, ..., para geração dos arquivos necessários para o processo GLS e posterior processamento pelo sistema LASER

- a)** Leitura dos resultados do processo GMA → Malha Básica do modelo
- Recursos para permitir a edição finalização do modelo de elementos finitos para execução do processo GLS/LASER:  
- Criação/Remoção de Nós e Elementos  
- Adição/Remoção/Edição: Pilares, Vigas, Cargas, Molas, Lajes.  
- Especificação de pontos de forma otimizada através da captura automática de pontos diretamente na tela gráfica  
- Recursos de Zoom, renumeração de componentes, cadastro de carregamentos, visualização de propriedades, carregamentos, molas, etc.
- Recursos para importar e visualizar a planta de arquitetura completa, desde que disponível no formato DXF, para facilitar a construção do modelo de elementos finitos com a adição dos pilares, vigas, carregamentos, etc.
- c)** Gerar os arquivos de entrada (.B, .P, .V, .C, ...) e realizar o processamento do programa GLS para a montagem do arquivo final para o processamento pelo Sistema Laser
- d)** Realizar processamento pelo Sistema Laser
- e)** Comunicação com os Módulos 1, 3 e 4

1ª.  
Etapa

2ª.  
Etapa

### Módulo 3 - Sistema de contraventamento:

A partir da geometria e dos resultados obtidos para os diferentes pavimentos do edifício, esse módulo gera automaticamente a estrutura de contraventamento com os respectivos carregamentos devido a ação do vento na edificação e ações devidas ao desaprumo.

Leitura dos resultados do processamento do Sistema Laser para análise das ações horizontais (arquivo .XY) dos diferentes pavimentos que compõem o edifício.

Interface com o usuário para definição da ordem dos pavimentos e número de repetições de cada um deles

Interface com o usuário para especificação dos parâmetros para cálculo da ação do vento: Velocidade básica do vento, Classe e Categoria da Edificação, Fator Estatístico, Coeficiente de arrasto em x e y.

Geração da Estrutura de Contraventamento e visualização gráfica 3D.

Geração do arquivo de entrada para o processamento do Sistema LASER.

Comunicação com o Módulo 4 para visualização dos resultados

### Módulo 4 - Visualização de resultados:

Esse módulo atende tanto o módulo 2, para visualização dos resultados do modelo de elementos finitos dos diferentes pavimentos, quanto o módulo 4, para visualização dos resultados da estrutura de contraventamento do edifício.

Leitura dos resultados do processamento do Sistema Laser (Módulo 2 ou 3):

- geometria dos nós
- grupos de elementos
- casos de carregamentos
- esforços/tensões e deslocamentos

Seleção do tipo de elemento desejado para visualização:

- PLACA/CASCA
- BARRA

Seleção do tipo de resultado desejado em função do tipo de elemento escolhido na etapa anterior:

- P/ PLACA/CASCA → Tensões/Deslocamentos
- P/ BARRA → Esforços/Deslocamentos

Definição de parâmetros gerais para geração da visualização:

- Multiplicadores p/ Casos de Carregamentos
- Opções de desenho (2D/3D, tamanho texto, ...)

Geração, caso necessário, das curvas de isovalor

Visualização gráfica dos resultados

De acordo com os quadros anteriores, a primeira etapa do trabalho caracteriza-se pelo desenvolvimento de uma plataforma totalmente interativa com o usuário para possibilitar, a partir do desenho de um pavimento, a montagem da malha básica e complementação do modelo mediante a adição de vigas, pilares, molas, lajes e carregamentos. Essa primeira etapa deve ser realizada para cada um dos diferentes pavimentos, térreo, tipo, ático, etc, que integram o edifício em análise e é finalizada com o processamento do modelo de elementos finitos realizado pelo sistema LASER.

A malha básica citada refere-se à discretização do pavimento em elementos finitos que podem ser triangulares ou quadrangulares. Para a geração desse modelo, conforme citado anteriormente, aproveitou-se um programa já existente e desenvolvido em C. Trata-se do programa Geração de Redes Básicas para Pavimentos ou GeraMalha, desenvolvido originalmente por BAPTISTA(1994), e atualmente denominado GMA (Gerador de Malhas).

Definiu-se essa etapa de geração da malha básica como sendo o módulo 1.

A razão por se ter optado em utilizar o programa GMA na geração da malha, deve-se pelo fato de evitar o retrabalho computacional de se reescrever as mesmas rotinas em uma outra linguagem, por mostrar processamento consistente e eficiente, e principalmente, por oferecer uma entrada de dados simples e prática, necessitando apenas do fornecimento de poligonais que descrevem contorno externo e contornos internos (aberturas), além de pontos e retas, estas últimas denominadas no programa de segmentos.

Os segmentos são úteis para a indicação do posicionamento de uma carga linear ou mesmo de uma viga através da geração de uma linha de nós, figura 3.1. A presença do objeto segmento faz o programa GMA gerar no modelo, uma seqüência de nós no local onde o mesmo se encontra posicionado, para que, na fase seguinte, que é o módulo 2, o usuário possa dispor dos nós para o adequado posicionamento de uma viga ou de um carregamento linearmente distribuído.

Diante do exposto, conclui-se que, para a geração da malha básica, há a necessidade de se oferecer ferramentas específicas para que o usuário possa fornecer de maneira automatizada os parâmetros de entrada para a geração do arquivo de entrada, e posterior processamento pelo programa GMA.

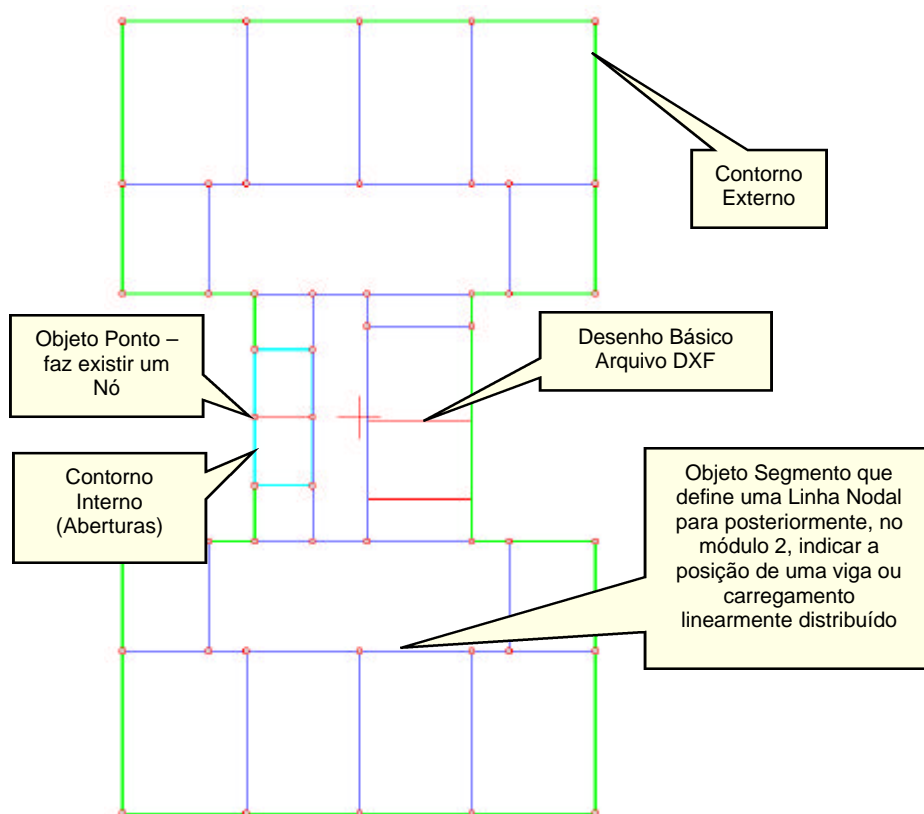


Figura 3.1. – Objetos que definem o modelo GMA

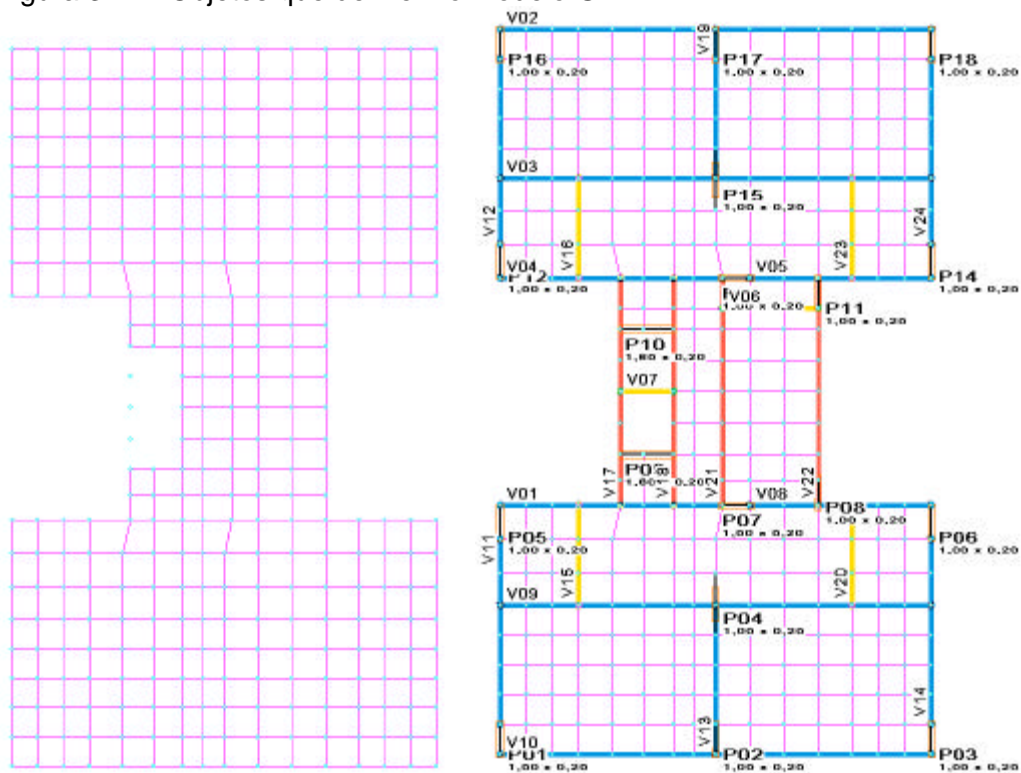


Figura 3.2. – À esquerda malha obtida após o processamento GMA do modelo da figura 3.1, e à direita mesma malha já acrescentado de vigas e pilares.



A fase seguinte, módulo 2, corresponde à leitura do arquivo de resultados gerado pelo programa GMA, isto é, a malha básica de elementos finitos, e sua edição pelo usuário, figura 3.2. Essa edição deve possibilitar a adição ou exclusão de nós e elementos, bem como completar o modelo com a definição de pilares, vigas, carregamentos, etc. Feito isso, é realizado um processamento utilizando-se um programa denominado GLS, também escrito em FORTRAN, que prepara efetivamente o arquivo final para o Sistema LASER.

Na seqüência desse processo, passa-se para a segunda etapa que compreende um terceiro módulo, que é o da definição da estrutura de contraventamento, e um quarto, que é de visualização de resultados. Destaca-se que o módulo 3, apenas é utilizado após terem sido devidamente montados e processados todos os diferentes pavimentos que compõem o edifício. Já o módulo 4, para visualização de resultados, pode ser utilizado a qualquer momento tanto pelo módulo 2 quanto pelo módulo 3, bastando, para isso, dispor dos resultados do processamento provenientes do sistema LASER.

Para a implementação dos módulos 1 e 2 conforme descrito anteriormente, foi necessário conceber um sistema capaz de gerenciar de forma eficiente todas as etapas do processo, isto é, de se fornecer as ferramentas necessárias para o usuário executar as tarefas relacionadas da forma mais automatizada possível.

O mesmo ocorreu para os módulos 3 e 4 que foram desenvolvidos em sistemas a parte. Destaca-se a importância dessa abordagem para o módulo 4, já que, estando totalmente independente dos módulos de geração de dados, acaba facilitando futuras modificações de tal sorte que o mesmo possa ser utilizado para a análise de resultados provenientes de outros programas de cálculo estrutural.

Tudo isso originou os seguintes sistemas desenvolvidos no ambiente de programação Delphi: Sistema MESTRE, para geração do modelo de elementos finitos de cada pavimento e o respectivo processamento pelo sistema LASER, contemplando os módulos 1 e 2; Sistema Edifício, para o módulo 3, com a montagem do modelo de contraventamento e o respectivo processamento pelo Sistema LASER; e o Sistema Visualiza3D, módulo 4, para a visualização de resultados dos pavimentos e do edifício.

Para completar a discussão sobre a concepção do sistema como um todo, o tópico seguinte trata de como a estrutura da linguagem utilizada acaba influenciando na definição da estrutura dos aplicativos.

E na seqüência, os capítulos seguintes tratam em detalhes cada um dos sistemas descritos neste tópico.

### 3.3 Características do Delphi *versus* Estrutura do Sistema

Esse tópico tem por finalidade fornecer uma visão geral de como o ambiente de programação pode, de certa forma, orientar ou até mesmo condicionar a elaboração dos aplicativos.

O Delphi possui três características marcantes que podem ser agentes influenciadores:

- a) É uma linguagem visual
- b) É uma linguagem baseada em eventos
- c) É uma linguagem orientada a objetos

Sobre o fato do Delphi ser uma linguagem visual, cita-se a maior facilidade para se criar o ambiente de trabalho do usuário, tanto para o programador, já que dispõem de componentes padrões como menus, botões, caixas de diálogos, caixas de textos, caixas de ferramentas, etc.; quanto para o próprio usuário, uma vez que esses componentes são comuns a maioria dos programas desenvolvidos para o Windows, facilitando a familiarização do mesmo com o sistema.

O Delphi também se caracteriza por ser um ambiente baseado na ocorrência de eventos, isso significa que o fluxo de execução do programa não segue necessariamente um sentido pré-estabelecido, como ocorre, por exemplo, em um programa estruturado na forma procedural, onde o início e o fim do programa estão sempre bem definidos. De forma diferente, no Delphi é o usuário quem define como o programa será executado, é ele quem estabelece a seqüência de tarefas a ser realizada. Isso ocorre através de eventos, como o movimento do mouse, o pressionamento de uma tecla ou do botão do mouse, um duplo clique na área de trabalho, e assim por diante. Para cada evento solicitado, o programa pode dar uma resposta diferente e realizar uma seqüência de processamentos diferentes. Portanto no Delphi, pode-se associar diretamente um código ao evento desejado, não sendo necessário criar rotinas que verifiquem a cada momento as ações realizadas pelo usuário. Vale lembrar que, para que ocorra um evento em um componente, o foco do sistema deve estar apontando para esse componente. Como exemplo, pode-se citar o caso do pressionamento da tecla ENTER para acionar um botão. Nesta situação, apenas se consegue acioná-lo, se o foco do sistema já estiver apontando para o mesmo. Para qualquer programa do Windows, o foco do sistema pode ser alterado de um componente para outro através de um clique com o mouse, ou pelo teclado através da tecla de tabulação (TAB).

Finalmente, o Delphi também se caracteriza por utilizar uma linguagem orientada a objetos e esse fato pode condicionar a forma de organizar internamente variáveis e códigos. Falando em variáveis e códigos, uma característica importante dos objetos são as denominações utilizadas.

A primeira definição é justamente com relação à utilização do termo objeto. No Delphi existe um outro termo denominado classe que anda lado a lado com o termo objeto. Uma classe é um tipo de dados definido pelo usuário, que possui um estado (sua representação ou características) e algumas operações (seu comportamento). Já o objeto é uma instância da classe, ou melhor, uma variável do tipo de dados definido pela classe. Os objetos são entidades reais e são eles que ocupam memória quando o programa é executado. O relacionamento entre objeto e classe é o mesmo que o existente entre variável e tipo. A classe é portanto o “molde” que dá a “forma” aos objetos.

Outra definição importante é que as variáveis são denominadas variáveis de instância ou campos, e as rotinas, independente de serem procedimentos ou funções, são denominadas métodos. O que justifica essa denominação diferente é o fato dos métodos levarem consigo uma referência de qual objeto eles pertencem e essa identificação é possível através da existência de um parâmetro implícito que apenas os métodos possuem, denominado Self.

As classes são estruturas que podem agrupar tanto os campos quanto os métodos que modificam ou retornam os valores desses campos. Seguindo essa definição, pode-se beneficiar de uma propriedade das classes denominada encapsulamento, pela qual os valores dos campos só podem ser modificados pelos próprios métodos da classe. Utilizar essa propriedade torna-se interessante na medida em que permite obter um maior controle sobre o valor e o estado das variáveis internas, uma vez que os métodos presentes na própria classe funcionam como “portas” de entrada e saída de informações.

Uma outra propriedade interessante da OOP é denominada hereditariedade que tem por finalidade tanto a redução da codificação dentro do programa, quanto a facilitação de novas implementações, como a ampliação do código escrito anteriormente. Isso ocorre pois é permitido definir novas classes baseadas em uma já existente. Utilizando essa propriedade, pode-se estruturar classes básicas de uso geral e a partir delas criar novas classes que assumam todas as suas características. A partir dessa situação, pode-se particularizar o comportamento das mesmas, com a adição de dados específicos, novos métodos ou até mesmo redefinição de métodos já existentes. A partir desse ponto pode-se beneficiar ainda de uma outra propriedade da OOP que é o polimorfismo, ou ligação tardia – *late binding*.

As funções e procedimentos nas linguagens comuns são normalmente baseadas em ligação estática, que também é chamada de ligação precoce, *early binding*. Isso significa que uma solicitação para a execução de uma rotina é resolvida no momento da compilação, isto é, a solicitação é substituída por uma chamada à posição da memória específica onde a função ou o procedimento

reside. Já as linguagens orientadas a objeto permitem o uso de outra forma de ligação, conhecida como ligação dinâmica ou ligação tardia, *late binding*. Nesse caso, o endereço real do método a ser chamado é determinado em tempo de execução com base no tipo da instância usada para fazer a chamada.

Em outras palavras esse recurso é conhecido como polimorfismo, que vem do grego, e significa “muitas formas”. No contexto de programação orientada a objeto, polimorfismo significa que uma espécie de ação pode ser feita de muitas maneiras específicas dependendo do objeto de classe que está em ação. Por esse princípio, durante a execução do programa, o sistema saberá exatamente qual método, evento ou campo deverá ser manipulado em função do objeto que estiver ativo dentro de uma determinada hierarquia, mesmo que os outros objetos dessa hierarquia possuam campos ou métodos com os mesmos nomes.

Apesar do princípio da programação orientada a objeto estar fundamentada na definição das classes e nas propriedades de encapsulamento, hereditariedade e polimorfismo, a implementação de um programa escrito no Delphi não é obrigada a seguir tais regras. Podendo, caso desejado, seguir os padrões tradicionais da programação procedural, como a definição de variáveis globais e rotinas desvinculadas de classes, também de uso geral. O mais importante é saber quais variáveis e rotinas podem ser representados como classes, para que efetivamente possa se beneficiar com a utilização das mesmas.

Por esse motivo, com exceção das ferramentas de interface com o usuário, o sistema desenvolvido não se encontra totalmente baseado na utilização de classes, sendo que muitas variáveis e rotinas, aproveitadas de outros programas, foram mantidas na sua forma original.

Uma última observação a respeito da programação em Delphi gira em torno da denominação de projeto e da utilização de formulários e unidades, figura 3.3. O formulário é um componente gráfico básico que nada mais é do que a área de trabalho ou tela de fundo, onde serão inseridos os demais componentes que efetivamente farão a interface com o usuário, como por exemplo: os menus, os botões, os textos informativos, as caixas de ferramentas, etc. Já as unidades são os arquivos com o código-fonte que podem estar vinculado a um formulário ou não, sendo que todo formulário possui uma unidade vinculada, mas a recíproca não é verdadeira. Já projeto é o programa como um todo, reunindo todos os formulários e unidades relacionados a ele. No caso em questão, o sistema Mestre é um projeto do Delphi.

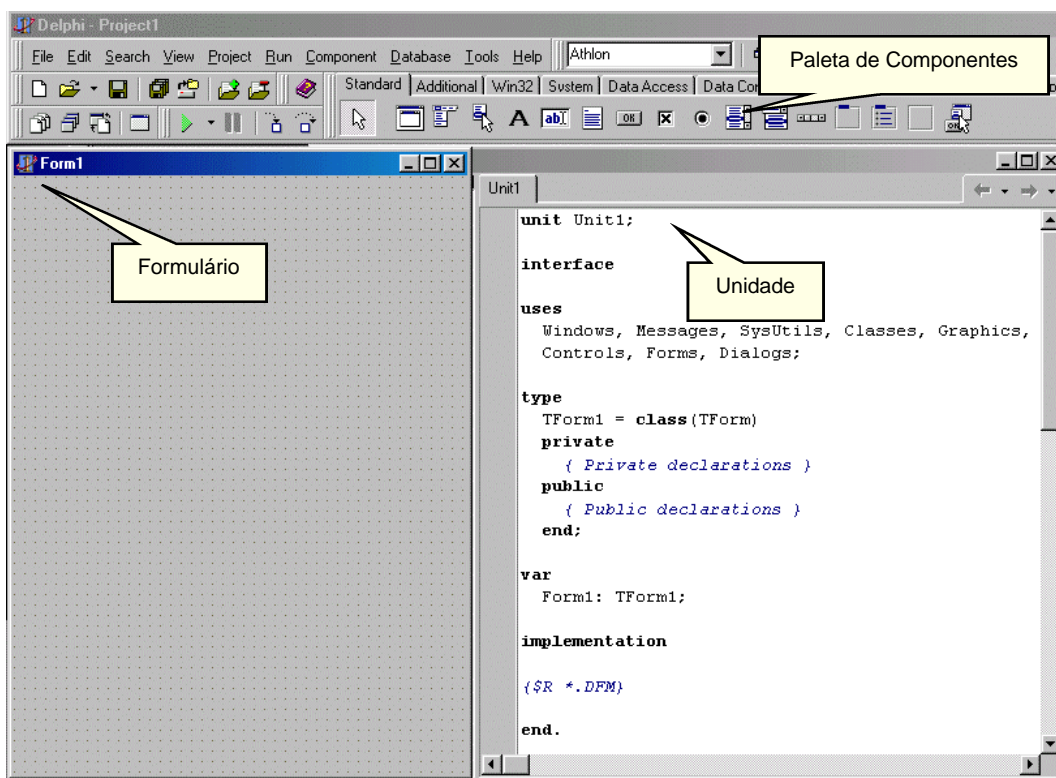


Figura 3.3- Ambiente de trabalho do Delphi com a visualização da paleta de componentes, o formulário inicial e sua respectiva unidade.

Enquanto o formulário armazena o “visual” do projeto, a unidade do formulário armazena todo o código relacionado a ele. Esse código em geral está presente nos eventos dos componentes, como, por exemplo, o código que deve ser executado ao se pressionar um botão. Para definir novos códigos desvinculados a eventos, pode-se criar novas unidades sem relação com os formulários.

Num projeto do Delphi deve-se relacionar apenas um formulário para cada unidade. Caso haja necessidade em se adicionar novos formulários em um mesmo projeto, para cada novo formulário haverá sempre uma nova unidade relacionada a esse formulário, e ela é criada automaticamente pelo Delphi. No caso de unidades independentes, apenas adiciona-se a unidade.

## **4. MÓDULOS 1 e 2 - SISTEMA MESTRE**

---

### **4.1 Estrutura Básica do Sistema MESTRE**

Quando se inicia um novo projeto no Delphi, abre-se um formulário limpo com a sua respectiva unidade, a partir do qual se inicia o desenvolvimento do programa.

O sistema Mestre foi organizado de forma a considerar esse formulário inicial como sendo a unidade principal do sistema. Portanto é essa unidade que cumpre o papel de gerenciar as demais.

Como a execução das tarefas foram divididas em módulos, módulo 1 para a preparação do arquivo para o processamento GMA e módulo 2 para edição da malha básica gerada pelo módulo anterior e posterior processamento pelo GLS e sistema LASER; esse formulário principal deve-se adequar para cada módulo em execução, fornecendo as ferramentas e todos os recursos necessários para o trabalho do usuário.

Já que os dois módulos tratam da edição de um desenho num ambiente gráfico, muitos comandos e funções são comuns aos dois. Citam-se como exemplo: definição da área gráfica de trabalho, a base de dados de informações de coordenadas de nós, barras e elementos, mapeamento dos elementos gráficos na tela para possível captura de coordenadas de pontos diretamente pelo mouse, funções de zoom, entrada de dados via teclado ou mouse, quadros de diálogos para abrir e salvar arquivos, e funções e procedimentos auxiliares.

Dessa forma há uma parte do projeto que é compartilhada entre os dois módulos, enquanto que outra, mais específica, diz respeito a informações e ferramentas exclusivas a cada um dos módulos. Baseado nessa situação, foram criadas variáveis gerais e unidades auxiliares no projeto com a finalidade de atender aos dois módulos, e unidades específicas contemplando apenas informações específicas de determinado módulo.

Com isso desenvolveu-se, no sistema MESTRE, as seguintes unidades gerais e unidades específicas por módulo:

*Unidade Principal :* UPrincipal\*

*Unidades gerais (Compartilhadas pelos dois módulos):*

UConversao

UDXF

UArquivos

UMapa

UObjetos

UImp\*

UProjeto\*

*Unidades específicas (Referentes aos Módulos)*

Módulo 1 – GMA

UGMA

UEntDados\*

UFDXF\*

Módulo 2 - GLS

UGLS

UDadosVigas\*

UDadosVigaTramos\*

UDadosPilar\*

UModifNos\*

UModifElemento\*

UCadastro\*

USubst\*

URestricaoNodal\*

UCadCarga\*

UCadMola\*

UCadLaje\*

OBS:

- A presença da letra U na frente do nome das unidades faz parte de uma organização interna para destacar que tal nome corresponde a uma unidade;
- O asterisco \* indica que a unidade possui um formulário associado.
- Uma descrição mais aprofundada dessas unidades pode ser encontrada no CD-ROM deste trabalho.

No caso do módulo 1, para montar o arquivo de entrada para o processamento pelo programa GMA e montagem da malha básica, o usuário deve fornecer um contorno externo que define a região a ser discretizada, e, caso seja necessário, deve fornecer os contornos internos para definir as aberturas, e lançar pontos dentro da malha de forma a forçar a passagem da discretização por tais posições, criando nós. Esse recurso também pode ser feito com a utilização de uma linha nodal, denominado no sistema de segmento.

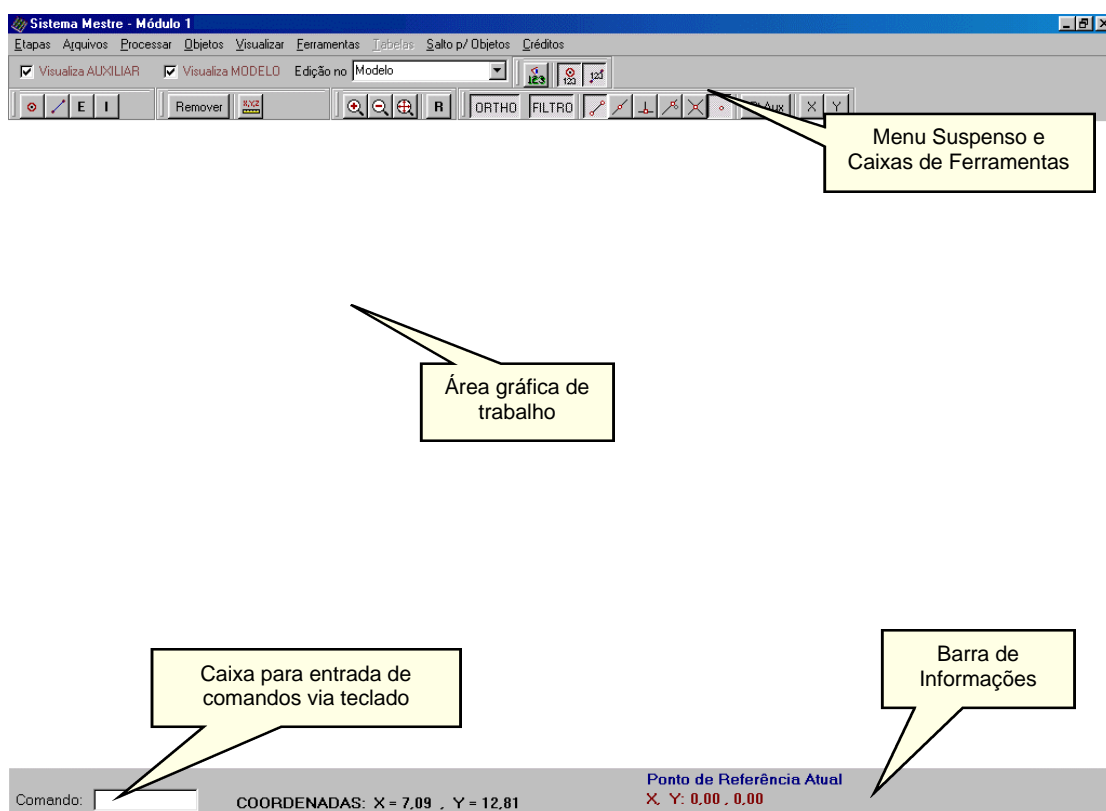


Figura 4.1 – Formulário principal do sistema Mestre.

A fim de possibilitar a construção gráfica que represente tal entrada, e posterior extração para um arquivo formatado adequadamente para o processamento GMA, desenvolveu-se um ambiente gráfico que possibilitasse entrar com todos esses objetos através da utilização do mouse ou teclado.

Neste ponto, torna-se oportuno comentar que, para disponibilizar os recursos para manipulação dos objetos gráficos através do mouse e teclado, tais como: adição, edição ou exclusão de nós e elementos e demais entidades gráficas; foi necessário realizar um mapeamento de todas as informações de posicionamento geométrico do modelo a fim de permitir uma rápida localização das entidades e objetos gráficos. Dessa necessidade surgiu uma estrutura de dados para mapear o posicionamento das informações dos objetos na tela gráfica, bem como um



conjunto de funções e procedimentos responsáveis pelas operações de buscas e atualização da base de dados.

Embora exija grande capacidade de armazenamento, considerou-se adequada tal metodologia uma vez que atualmente se dispõe de microcomputadores com grande capacidade de armazenamento e processamento. Outro ponto favorável é pelo fato de ser um processo que necessita pouco esforço computacional para sua implementação, e por atender satisfatoriamente as operações de busca para os modelos bidimensionais gerados através dos aplicativos desenvolvidos nesta pesquisa.

Como alternativa, pode-se recorrer ao estudo da Topologia para aproveitar melhor os recursos do equipamento, demandando, no entanto, mais tempo para concepção e implementação. Para esse tipo de abordagem PREPARATA & SHAMOS (1985) apresenta no capítulo 2 de seu trabalho, capítulo este intitulado "Geometric Searching", o problema da localização de pontos num dado espaço geométrico.

Inicialmente PREPARATA & SHAMOS (1985) define algumas denominações como "*file*" (arquivo) que é uma base de dados, as entidades geométricas como "*records*" ou registros, e a consulta como "*query*", que corresponde à pergunta que deve ser feita bem como o método de busca que deve ser utilizado para encontrar os registros de interesse dentro da base de dados.

Destaca-se ainda no trabalho que a escolha do método a ser utilizado para realizar a busca das informações deve ser avaliado através de quatro indicadores:

- Tempo para realizar a consulta;
- Quantidade de memória necessária para armazenar a estrutura de dados;
- Tempo de processamento necessário para organizar os dados para a busca; e
- Tempo necessário para atualizar a estrutura de dados, isto é, o tempo gasto para adicionar ou eliminar uma entidade da base de dados.

Considerando-se os indicadores sugeridos acima, pode-se avaliar a metodologia adotada neste trabalho como sendo grande consumidora de memória; eficiente para retornar os resultados da consulta para bases de dados de pequenas e médias dimensões; e tendendo a ficar lenta à medida que a quantidade de informações cresce demasiadamente. No entanto, essa situação não chega a ser crítica pois, para a análise de pavimentos correntes de concreto armado, dificilmente é necessário ultrapassar um número de dez mil elementos.

Processos mais inteligentes podem ser utilizados para otimizar a utilização de memória e o uso do processador como apresentado em PREPARATA & SHAMOS (1985):

- Princípio combinatório de inclusão e exclusão LIU (1968), que faz uso do princípio de vetor dominante.
- Problema de checagem de ponto contido ou não num dado polígono, que testa o número de interseções de uma reta que contém o ponto com as fronteiras do polígono. Através desse princípio, seguindo-se um sentido preestabelecido através dessa reta, contam-se quantas interseções ocorrem antes e depois do ponto, se o resultado for um número ímpar, garante-se que o ponto está contido no polígono, e caso seja par, então o ponto não está contido.
- O PSLG (Planar Straight-Line Graphs) , que é a localização de um ponto através de uma subdivisão do espaço geométrico em planos, onde cada plano passa através de um dos vértices do polígono. DOBKIN e LIPTON (1976)
- Outras técnicas, mais refinadas, aplicadas com base no PSLG utilizando o método da triangulação ou método do trapézio.

Após a apresentação geral da estrutura básica do sistema, os tópicos a seguir explicam de forma mais aprofundada os programas que são a base para o desenvolvimento do sistema Mestre, o GeraMalha para o módulo 1 e o Sistema Laser para o módulo 2. O CD-ROM em anexo completa esses tópicos, trazendo mais informações a respeito da codificação realizada para cada um desses módulos do sistema Mestre.

## 4.2 MÓDULO 1 – CONSTRUÇÃO DO MODELO BÁSICO – Programa GeraMalha

Este tópico tem por finalidade explicar sobre o funcionamento do programa GeraMalha, desenvolvido por BATISTA (1994). Para isso foram transcritos alguns trechos da dissertação de BATISTA (1994).

A finalidade do programa GeraMalha é o de implementar um algoritmo eficiente de geração de dados de forma a minimizar, para o usuário, as dificuldades da entrada de dados com os seguintes requisitos ZIENKIEWICZ & PHILIPS (1971):

- O usuário deve ser capaz de controlar a densidade de nós em qualquer região do pavimento;
- a distribuição dos elementos deve variar suavemente através do domínio, não apresentando grandes discrepâncias entre regiões vizinhas;
- o algoritmo deve capacitar a conexão de múltiplas regiões;
- o algoritmo deve produzir resultados apropriados para o sistema de análise a ser utilizado;
- o algoritmo deve ser econômico com respeito ao tempo computacional e esforço manual.

Para atender a esses requisitos, partiu-se do conceito da técnica de triangulação denominada *Modified Quadtree* SHEPHARD & YERRY (1983), que consiste em circunscrever uma figura por um quadrado e dividi-lo recursivamente em elementos quadriláteros. Obtém-se com esse método, uma malha com a forma final composta por quadriláteros de tamanhos variados, que também pode ser adaptada para cálculo com elementos triangulares.

Para entender essa técnica procedeu-se a uma revisão dos conceitos básicos utilizados na elaboração da mesma.

A partição da figura baseia-se originalmente no método Quadtree [MAZUMDER (1987) SAMET(1984) SHAFFER & SAMET (1987) WOODWARK (1982), onde a mesma é circunscrita por um quadrado. Esse quadrado é decomposto sucessivamente em outros quadrados, até que todo o domínio esteja totalmente discretizado por elementos de tamanhos variados. São utilizados três tipos de códigos para identificar a situação do quadrado em relação à figura, ou aos limites da mesma. Tem-se, então, três condições: vazio, totalmente preenchido ou parcialmente preenchido. Dá-se prosseguimento à decomposição nos elementos cuja situação seja preenchimento parcial, conforme demonstrado na figura 4.2. A forma de armazenamento dos dados é através de uma estrutura de árvore hierárquica. O quadrado geral é representado pela raiz da árvore, o qual é subdividido em quatro novos quadrados, que são representados como os “filhos” da

raiz da árvore, e assim, sucessivamente GONNET (1984) HOROWITZ & SAHNI (1979).

Através da figura 4.2 verifica-se que os elementos gerados na fronteira são muito menores que os gerados no interior da figura e que várias reentrâncias são introduzidas na representação do contorno. Para eliminar ou reduzir problemas desse tipo, uma significativa alteração foi realizada na técnica existente, uma vez que a mesma não produzia resultados satisfatórios quando se tratava do contorno. Assim, com o método modificado ou *Modified Quadtree*, os elementos gerados não precisam mais ser “quadrados perfeitos”, e podem assumir outras formas padronizadas, que representam mais coerentemente o objeto em estudo.

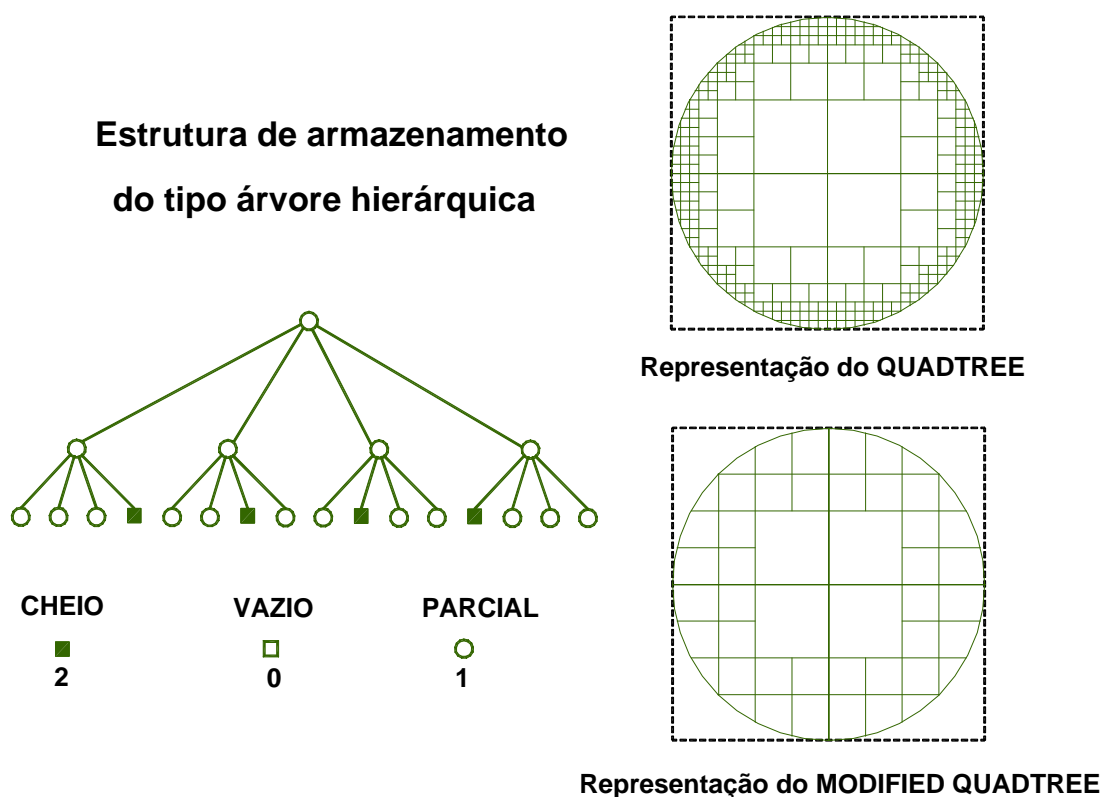


Figura 4.2. – A representação QUADTREE e MODIFIED QUADTREE de um círculo

Desse modo melhorou-se a representação geométrica do contorno da figura, porém as possibilidades que um elemento podia assumir continuavam fixas. Os pontos de intersecção desse elemento com o contorno só podiam ser alocados em determinados lugares, tais como: ou o vértice, ou o ponto médio, ou um quarto da aresta em análise.

Para remover algumas das limitações existentes, alterou-se a entrada de dados do objeto, tornando-a mais sofisticada SHEPHARD et al. (1987) SHEPHARD et al. (1988). As alterações realizadas eliminaram a restrição anterior através do armazenamento explícito dos vértices de intersecção com as arestas de contorno.

Além disso, acrescentou-se um controle sobre a existência de segmentos muito pequenos, alguns procedimentos de suavização e um refinamento das regiões mais solicitadas BAEHMANN et al. (1987).

Essas significativas e interessantes considerações na geração da malha, para o cálculo em elementos finitos, tornaram o procedimento mais demorado e trabalhoso, devido às funções matemáticas exponenciais utilizadas. A rede de polígonos gerada pelo método *Modified Quadtree* não apresentava ainda uma homogeneidade na dimensão dos elementos, principalmente para as figuras que possuíssem pequenas descontinuidades no seu interior. Conclui-se que é o método mais indicado e possui uma aplicação mais eficiente na Engenharia Mecânica, onde as peças possuem solicitações e tensões com valores bastante diferentes em regiões muito próximas.

Em vista dessas limitações e conhecendo-se outras técnicas de triangulação automática mencionadas no capítulo 2, partiu-se para o desenvolvimento do sistema GeraMalha, visando tanto propiciar a montagem de malhas com elementos distribuídos uniformemente, quanto de facilitar a implementação do algoritmo, sem aumentar demasiadamente o tempo de processamento.

A seqüência de operações realizadas pelo programa é a seguinte:

**Passo 1** – Ler o arquivo de dados com a extensão .GMA com as informações necessárias para a criação da rede sobre o pavimento, consistindo em:

- coordenadas cartesianas dos vértices de contorno e, eventualmente, pontos internos e externos ao pavimento;
- incidência dos segmentos retilíneos que delimitam o pavimento, ou que definam uma linha de nós a ser utilizada, ou seja, contornos e linhas nodais, respectivamente;
- dimensões máximas dos elementos a serem gerados segundo as direções X e Y.

Esse arquivo .GMA é fornecido pelo Módulo 1 do Sistema MESTRE, através da opção identificada por Geração do Arquivo Básico.

**Passo 2** - Para o pavimento discretizado através do seu contorno, circunscreve-se um Retângulo de Referência. A rede de pontos é criada com base nas dimensões fornecidas para os elementos segundo os eixos X e Y, conforme mostrado na figura 4.3. Essas dimensões podem ser ajustadas, para que a malha apresente um número inteiro de divisões dentro de cada intervalo de variação e segundo cada eixo.

**Passo 3** - A seguir determina-se a condição dos pontos da rede, ou seja, se os mesmos pertencem ou não ao domínio do pavimento. Para isso utilizou-se os

conceitos básicos da geometria vetorial, considerando todos os segmentos do contorno como vetores. Lembrando ainda que convencionou-se o contorno externo sendo definido no sentido anti-horário e os contornos internos no sentido horário, idealizou-se que o vetor normal a um segmento define uma superfície no lado oposto ao mesmo, conforme figura 4.4. Desse modo o domínio é automaticamente criado, bastando que se observe o sentido da declaração de incidência do segmento, conforme figura 4.5.

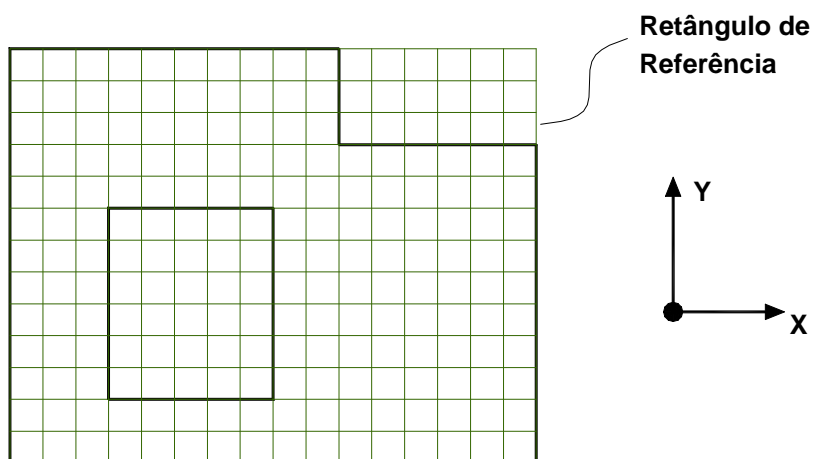


Figura 4.3 – Pavimento circunscrito por um Retângulo de Referência apresentando a rede de pontos inicial.

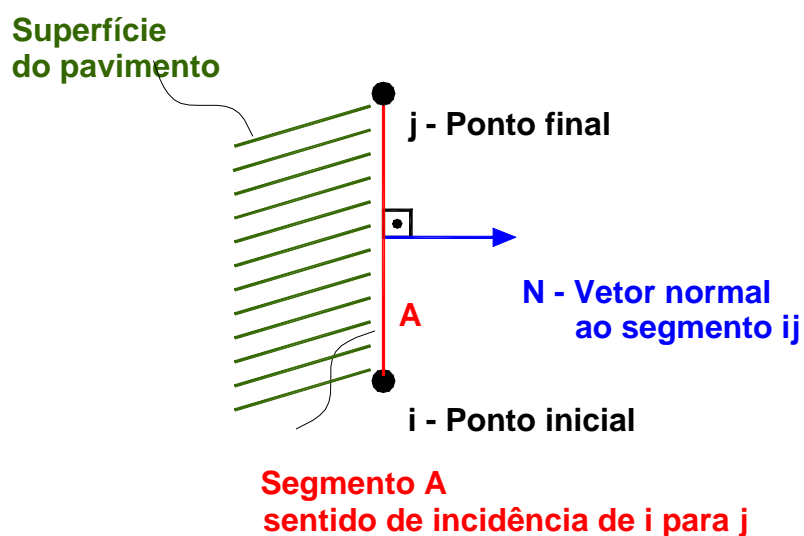
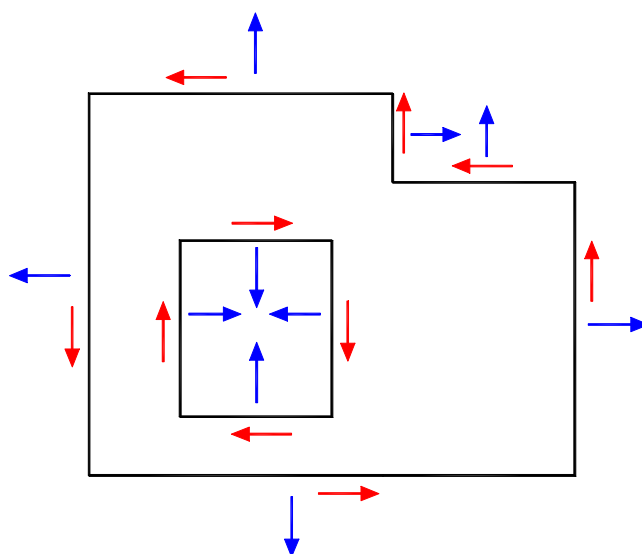


Figura 4.4. – Definição da superfície do pavimento para um segmento do contorno.



LEGENDA :

- N - Vetor normal
- Sentido da incidência do segmento

Figura 4.5. – Sentido de declaração dos segmentos de contorno e do vetor normal a cada um deles.

**Passo 4** - Após a definição da superfície passa-se a verificar se os pontos que compõem a rede pertencem ou não ao domínio. O procedimento de análise para verificar a posição de um ponto deve ser realizado em relação a todos os segmentos do contorno, considerando-se uma poligonal de cada vez. Essa verificação é feita através da criação de um vetor “distância”, que é uma perpendicular orientada do ponto em análise a um segmento de contorno no qual o mesmo se projeta, figura 4.6. De acordo com o sinal do resultado do produto escalar ente o vetor “distância” e a normal do segmento, consegue-se definir a localização do ponto.

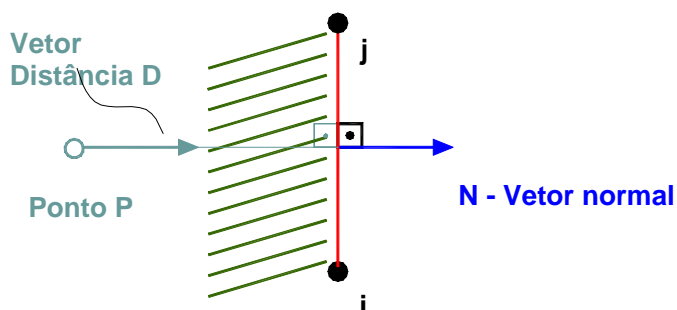


Figura 4.6. – Projeção de um ponto num segmento através do vetor distância.

**Passo 5** - A seguir faz-se um arranjo na rede de pontos de modo que a mesma fique preparada para a geração dos elementos de placa. Como protótipos dos elementos são criadas células, quadriláteros formados por quatro pontos vizinhos  $i$ ,  $j$ ,  $k$  e  $l$  (figura 4.7).

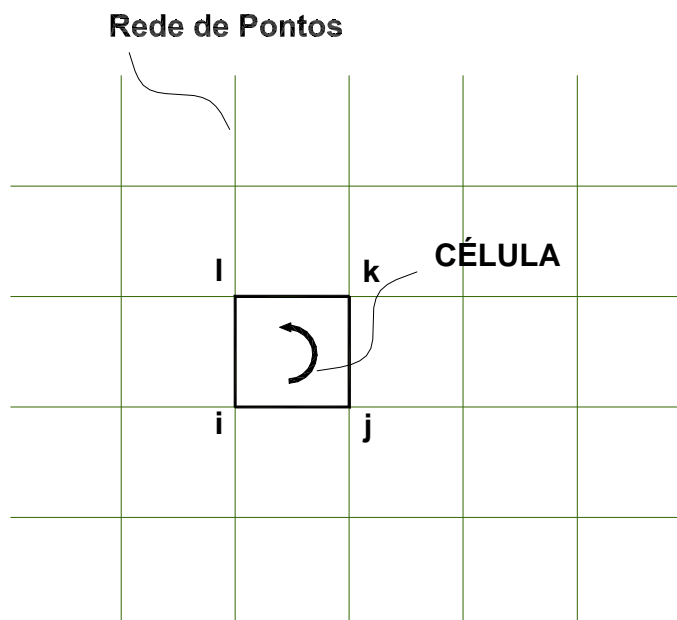


Figura 4.7. – Identificação de uma célula em uma rede de pontos.

**Passo 6** - Verifica-se se existem vértices no seu interior e/ou segmentos interceptando as suas arestas, tornando-as sujeitas a transformações. Verificam-se se as arestas da célula são interceptadas por segmentos do contorno. O número máximo de intersecções por aresta da célula é igual a dois. Se for encontrada apenas uma intersecção, o vértice mais próximo é trasladado. Se forem duas, cada vértice da aresta é transportado para a intersecção mais próxima.

**Passo 7** - Com a rede deformada finaliza-se o processo com a definição dos elementos nó, placa ou chapa e a gravação de um arquivo com a extensão .B (arquivo básico) para posterior retorno ao Módulo 2 do Sistema MESTRE, na opção Leitura do Arquivo Básico. Essa etapa permite editar a malha básica gerada e acrescentar vigas, pilares, lajes, carregamentos, etc, para só então realizar o processamento final.



### 4.3 MÓDULO 2 – CONSTRUÇÃO E PROCESSAMENTO DO MODELO FINAL - Sistema LASER

O programa utilizado pelo sistema MESTRE para realizar o processamento do cálculo estrutural propriamente dito é o programa LASER (Linguagem para Análise de Sistemas Estruturais Reticulados) RAMALHO (1990). O módulo de análise do programa LASER foi desenvolvido em linguagem FORTRAN, sendo constituído por um programa principal e 69 sub-rotinas. Todo o sistema estrutural apresenta algumas características comuns e outras próprias. Dentre aquelas que dizem respeito a um determinado sistema em particular, destaca-se a chamada biblioteca de elementos. O sistema LASER, possui oito tipos diferentes de elementos implementados. É a seguinte a relação desses tipos:

- Tipo 1 – elemento treliça
- Tipo 2 – elemento barra
- Tipo 3 – elemento membrana
- Tipo 4 – elemento plano (estado plano de tensão, deformação e axissimétrico)
- Tipo 5 – elemento tridimensional (sólido)
- Tipo 6 - elemento placa ou casca
- Tipo 7 – elemento contorno
- Tipo 8 – elemento sapata

Os elementos 1,2 e 7, são baseados numa formulação típica de análise matricial de estruturas. Sendo utilizado o processo dos deslocamentos, as respectivas matrizes de rigidez são montadas diretamente por equilíbrio. Um procedimento tão conhecido, dispensa maiores comentários. Esses elementos formam um conjunto muito potente para a solução de estruturas lineares. Elementos de treliça e barra para as peças propriamente ditas e o elemento de contorno para a definição de condições de apoio especiais. Aliados à grande capacidade de processamento do programa, podem ser muito úteis na análise de estruturas como pórticos e treliças tridimensionais com muitos graus de liberdade.

Já para os demais elementos, com exceção do elemento sapata que apresenta no seu desenvolvimento alguns aspectos um pouco diferentes da teoria tradicional, a utilização se volta para os meios contínuos, baseando-se esses elementos no Método dos Elementos Finitos. Sobre essa técnica, extremamente disseminada, também não será necessário tecer maiores comentários.

É interessante mencionar que, ao contrário dos elementos baseados em conceitos de pura análise matricial de estruturas, para os elementos finitos têm-se muitas formulações para resolver um mesmo problema. Como o método baseia-se na discretização do domínio total através da aplicação de funções aproximadoras por subdomínios, fica claro que para cada função haverá um comportamento

diferente, gerando por sua vez resultados que podem estar mais ou menos próximos dos valores exatos.

Portanto, se as funções aproximadoras de um determinado elemento são pobres é necessário uma discretização em maior número de elementos. Caso contrário, a rede pode ser menos densa, mas sempre haverá necessidade de um maior esforço computacional para tratar cada elemento em particular. Na verdade, o que se ganha por um lado se perde por outro.

O programa LASER adota para todos os casos elementos mais simples e que portanto necessitam de uma discretização mais fina. Acredita-se ser essa a solução mais vantajosa no cômputo geral. As razões dessa crença estão baseadas na maneira de se trabalhar com as estruturas que realmente ocorrem em casos práticos.

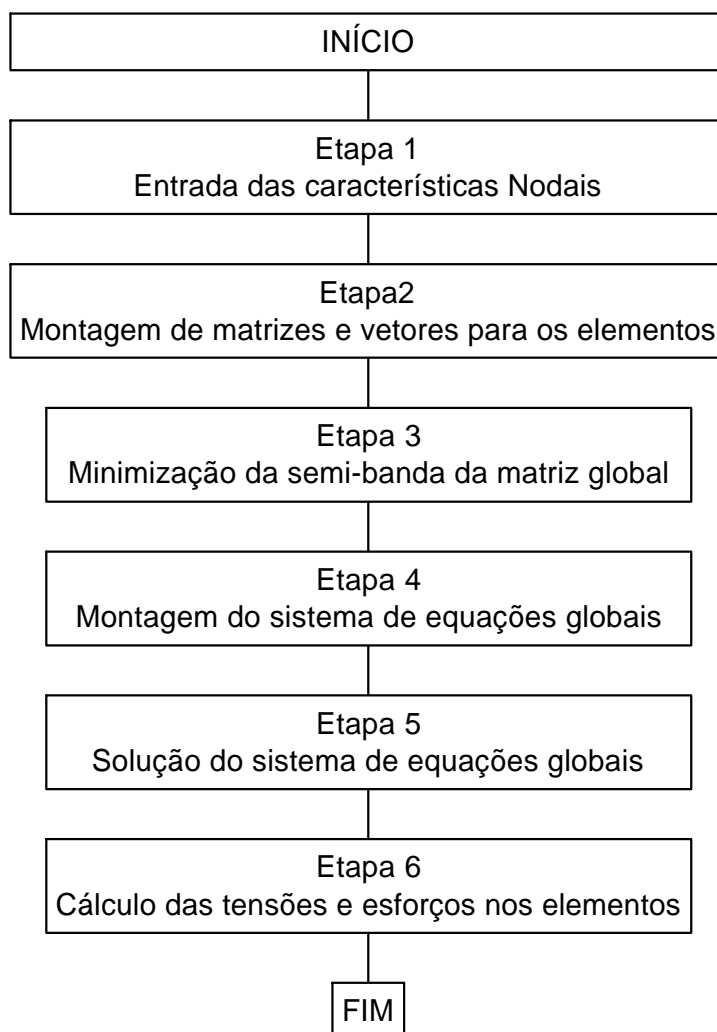
A verdade é que dificilmente são analisadas estruturas cujas condições de geometria e carregamento permitam o uso disseminado de grandes elementos. Pelo contrário, em casos usuais o que se verifica é a necessidade de discretização de detalhes e condições especiais, que praticamente exigem a utilização de pequenos elementos. Desse modo, pode haver um grande desperdício de esforço computacional, sem qualquer benefício em nível de resultados a serem obtidos.

Por outro lado, se os elementos são pobres, a grande desvantagem apontada é a dificuldade na entrada dos dados e análise dos resultados, já que devem ser utilizados muitos elementos. Problema este que se espera ter superado com a utilização do pré e pós-processadores desenvolvidos nesta pesquisa.

Além do que já foi mencionado, vale também ressaltar como característica especial do programa LASER, que o mesmo coloca à disposição do usuário um minimizador da semi-banda da matriz de rigidez global, que atua opcionalmente sobre as estruturas. Dessa forma não foi necessário criar dentro do sistema MESTRE, rotinas específicas para realizar essa tarefa.

Para compreender o processo envolvido e como o programa LASER organiza as suas diversas etapas de processamento durante a análise de uma estrutura, segue um fluxograma que reproduz esse caminho de forma simplificada.

## Etapas de processamento no sistema LASER



Outra informação importante, diz respeito ao arquivo de entrada de dados necessário para o processamento, e aos arquivos de saída gerados pelo programa LASER, que são utilizados pelo módulo de visualização de dados.

#### **a) Informações sobre o arquivo de entrada de dados para o Sistema LASER**

Para o processamento de um pavimento gerado pelo Sistema MESTRE, o programa LASER necessita de um único arquivo, sem extensão, especialmente formatado com todas as informações relativas às características nodais, aos elementos e aos carregamentos.

Esse arquivo é montado com base nos arquivos gerados pelo Sistema MESTRE, como o arquivo básico de geometria, com a extensão .B, e os arquivos com as informações de vigas \*.V, de pilares \*.P, de molas \*.M, de cargas \*.C, e de lajes \*.L. Existe um programa denominado GLS que se encarrega de ler as informações em cada um desses arquivos para montar corretamente o arquivo de entrada de dados para o programa LASER.

A partir dos arquivos com a definição dos nós iniciais e finais de pilares, vigas, carregamentos, etc, o programa GLS gera automaticamente os elementos ou cargas correspondentes, executando uma série de operações que simplificam o processo, como por exemplo: identifica todos os nós contidos no trecho de determinado objeto (viga, pilar ou carregamento); adiciona nós caso seja necessário; no caso de pilar, gera barra, perpendicular ao pavimento, ou modifica as características geométricas das que já existirem para considerar o trecho rígido que simula o comprimento do pilar; transfere os carregamentos definidos entre um nó inicial e final como cargas aplicadas nos nós que compõem o respectivo trecho; etc.

No Sistema MESTRE essa etapa faz parte do Módulo 2, e é identificada por Processamento Final, localizado dentro do menu Processamento. É importante salientar que essa opção se torna disponível somente após o usuário efetuar o processamento GLS para o modelo em que estiver trabalhando.

## **b) Informações sobre os arquivos de saída do Sistema LASER**

Com relação aos arquivos de saída, existem 3 arquivos montados pelo programa LASER para posterior utilização pelos pós-processadores. Todos eles possuem como prefixo o nome definido para o arquivo de dados da estrutura, aqui representado pela palavra PREFIX. Já o sufixo é determinado de acordo com a saída que eles contém. Os próximos itens fornecem informações mais detalhadas a respeito das características e conteúdos desses arquivos, ressaltando-se apenas que I4, R4 e R8 são símbolos usados para indicar inteiros de 4 bytes, reais de 4 bytes e reais de 8 bytes, respectivamente.

### *1) Arquivo PREFIX.GO*

O arquivo de sufixo GO guarda as informações relativas à geometria da estrutura. É um arquivo em formato binário com tamanho de cada registro fixado em 44 bytes. Considerando cada linha como um registro, sua estrutura de gravação é a seguinte:

```

nn,nge
1,IR,X1,X2,X3,T
2,IR,X1,X2,X3,T
..
..
n,IR,X1,X2,X3,T
nte,ne,nmn
1,NI,NJ,....
2,NI,NJ,....
..
..
n,NI,NJ,....

```

} NGE vezes

onde:

nn : número de pontos nodais (I4)  
nge : número de grupos de elementos (I4)  
IR : matriz de restrições nodais (I4)  
X1,X2,X3 : coordenadas dos nós (R4)  
T : temperatura nodal (R4)  
nte : número do tipo dos elementos do grupo (I4)  
ne : número de elementos do grupo (I4)  
nmn : número máximo de nós por elemento (I4)  
NI,NJ,... : número dos nós que definem o elemento (I4)

## 2) Arquivo PREFIX.DO

O arquivo de sufixo DO guarda os resultados dos deslocamentos nodais obtidos para cada caso de carregamento. É um arquivo em formato binário com tamanho de cada registro fixado em 48 bytes. Seguindo a mesma metodologia anterior, considerando cada linha como um registro, sua estrutura de gravação é a seguinte:

```

nn,nc
1,DX1,DX2,DX3,RX1,RX2,RX3
2,DX1,DX2,DX3,RX1,RX2,RX3
..
..
n,DX1,DX2,DX3,RX1,RX2,RX3

```

} nc vezes

onde,

nn : número de pontos nodais (I4)  
 nc : número de carregamentos (I4)  
 DX1,DX2,DX3 : translações nodais segundo X1,X2,X3 (R8)  
 RX1,RX2,RX3 : rotações nodais em torno de X1,X2,X3 (R8)

### 3) Arquivo PREFIX.SO

O arquivo de sufixo SO contém informações a respeito das tensões e ou esforços que ocorrem nos elementos. É um arquivo em formato binário com tamanho de cada registro fixado em 104 bytes. Sua estrutura é um pouco mais complexa que a dos arquivos anteriores e por isso será apresentada por partes.

Os primeiros registros são montados no programa principal e tem a seguinte estrutura:

nge,nc		
nrl	}	nge números
..		
..		
nrn		

onde,

nge : número de grupos de elementos (I4)  
 nc : número de casos de carregamento (I4)  
 nrl a nrn : número do registro onde tem início a gravação das tensões ou esforços para cada grupo de elementos (I4)

Após esses registros iniciais, são gravadas as tensões e ou esforços propriamente ditos. Esses valores são gravados por elemento, para cada caso de carregamento e cada localização existente. Um registro genérico para essa parte do arquivo seria o seguinte:

ne,nc,nl,ntl,TE(1),TE(2),.....,TE(ntl)

onde,

ne : número do elemento (I4)  
 nc : número do carregamento (I4)  
 NL : número da localização (I4)  
 ntl : número de tensões ou esforços (I4)  
 TE : valores de tensões ou esforços (R8)

Tomando-se, como exemplo, uma situação específica de um grupo de apenas um elemento, com seis tensões e ou esforços em dois locais, e supondo a existência de dois casos de carregamento, os registros para esse grupo seriam os seguintes:

1,1,1,6,TE(1),TE(2),TE(3),TE(4),TE(5),TE(6)

1,1,2,6,TE(1),TE(2),TE(3),TE(4),TE(5),TE(6)

1,2,1,6,TE(1),TE(2),TE(3),TE(4),TE(5),TE(6)

1,2,2,6,TE(1),TE(2),TE(3),TE(4),TE(5),TE(6)

### c) Informações adicionais sobre os elementos barra e placa/casca.

Para completar as informações referentes aos arquivos de saída, seguem abaixo algumas considerações sobre os elementos tipo barra (tipo 2) e tipo placa/casca (tipo 6), que efetivamente são utilizados pelo sistema de visualização de dados para exibir os resultados, bem como o sistema global de coordenadas adotados para nós, figuras 4.8 e 4.9.

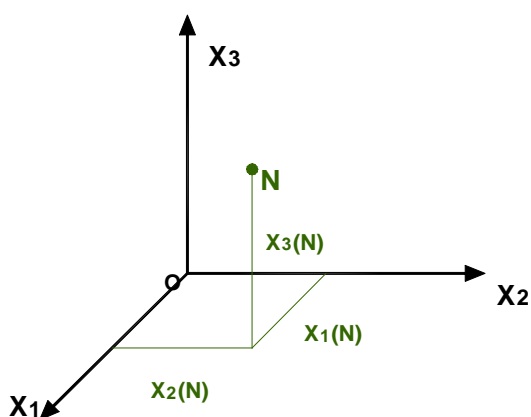


Figura 4.8. – Coordenadas do nó “N” .

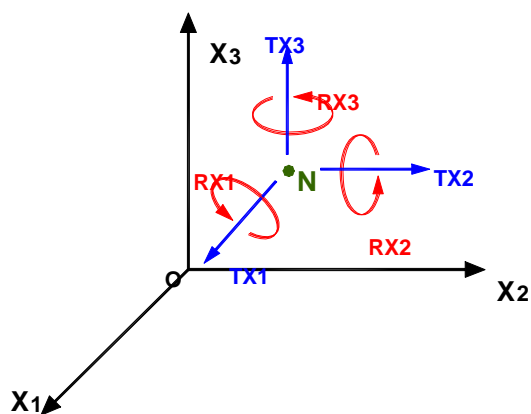


Figura 4.9. – Graus de liberdade do nó “N” no espaço.

### 1) Elemento BARRA (tipo 2)

O elemento barra é definido por dois nós de extremidade e pode ser colocado em qualquer posição do espaço definido pelo sistema de referência global da estrutura. De modo geral, tem seis graus de liberdade por nó, três translações e três rotações, perfazendo um total de 12 parâmetros conforme mostra a figura 4.11.

Um ponto de grande importância é a definição de planos de referência para o elemento. Isso é necessário pois a barra trabalha tanto no seu eixo longitudinal, quanto segundo a dois planos ortogonais. Assim sendo, o usuário precisa definir a posição desses planos.

A definição desses planos está ligada à definição do sistema local de referência. Como se pode observar na figura 4.10, o eixo local  $x_1$  é definido pelo eixo longitudinal da barra, com sentido orientado do nó "I" para o nó "J". Os outros dois eixos, coincidentes com os eixos principais de inércia da seção transversal, é que precisam ser definidos pelo usuário. Existem duas maneiras de se fazer essa definição. Na primeira basta fornecer o eixo global da estrutura ao qual o eixo  $x_2$  é paralelo. Entretanto, se esse paralelismo não existir o usuário pode fornecer um terceiro nó, na figura 4.10 o ponto K, que define, juntamente com os nós "I" e "J" o plano que contém o eixo  $x_2$ . A partir daí o sistema local dextrorso estará construído, sendo o eixo  $x_3$  adotado normal ao plano formado pelos dois anteriormente definidos. Mais informações podem ser obtidas em RAMALHO (1990).

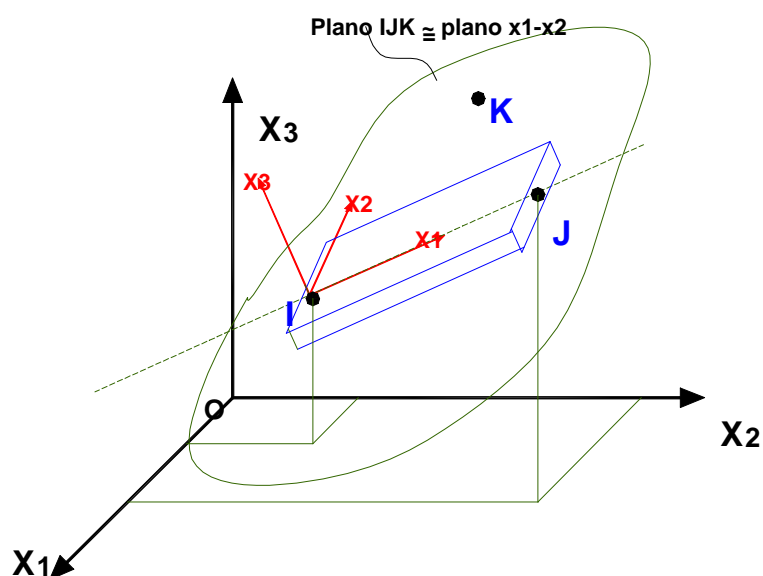


Figura 4.10 – Sistema local do elemento barra



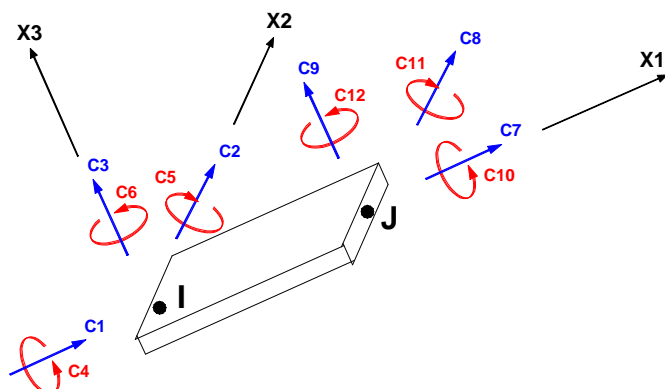


Figura 4.11. – Coordenadas locais do elemento barra.

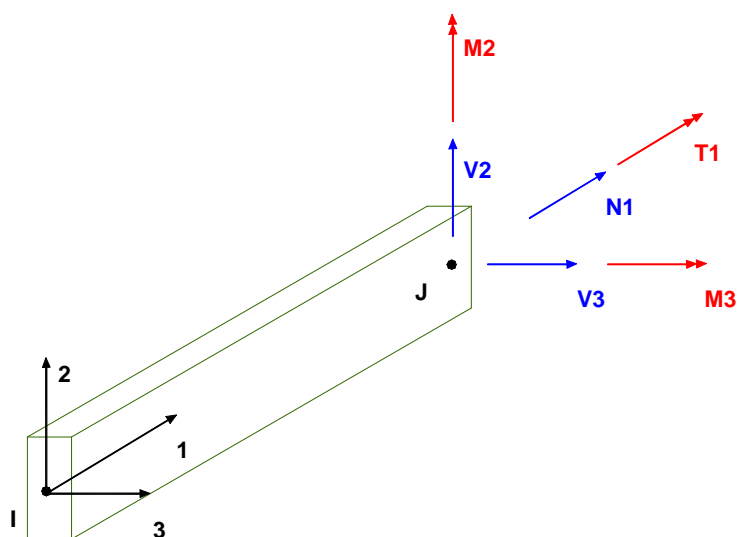


Figura 4.12. – Coordenadas locais em termos de esforços para o elemento barra, com a mesma notação utilizada no programa de visualização de resultados.

## 2) Elemento PLACA ou CASCA (tipo 6)

O elemento tipo 6 pode ser usado para modelagens de placas ou cascas de espessura relativamente pequena. Esse elemento, no caso do programa LASER, nada mais é que a composição de um elemento utilizado para modelagem de placas fletidas e um elemento utilizado para discretização de estados planos de tensão.

Desse modo, as solicitações que forem normais ao plano do elemento são resistidas pela rigidez fornecida pelo elemento placa. Já as que estiverem contidas no próprio plano, pela rigidez de membrana. Essa composição equivale a definir,

sobre uma malha de nós, elementos dos dois tipos com os mesmos nós de extremidade.

Entretanto, se o desejo do usuário é a utilização de um elemento para modelagem de estruturas que possam ser definidas exclusivamente como placas, basta que o programa, avisado dessa condição, deixe de somar às equações do problema de rigidez do elemento membrana. Esta é a razão de mencionar o elemento tipo 6 como sendo adequado à modelagem de placas ou cascas.

O elemento pode ser triangular ou quadrangular, estando definido em qualquer posição do espaço. No caso das cascas, cinco graus de liberdade são definidos para cada nó: duas translações no próprio plano, devidas ao componente membrana, e mais uma translação normal ao plano e duas rotações, devidas ao componente placa. Já para o caso de um elemento exclusivamente placa, obviamente somente restarão os graus de liberdade a ele associados, no caso o deslocamento normal ao plano e as rotações, perfazendo um total de três graus de liberdade para cada nó.

Tanto para o elemento triangular como para o quadrangular são definidos os sistemas locais de referência mostrados na figura 4.13. Essa referência é composta por um eixo  $x_1$ , definido no nó "I" para o nó "J", e ainda  $x_2$  e  $x_3$ , respectivamente no plano e normal ao plano do elemento, formando um sistema dextrorso.

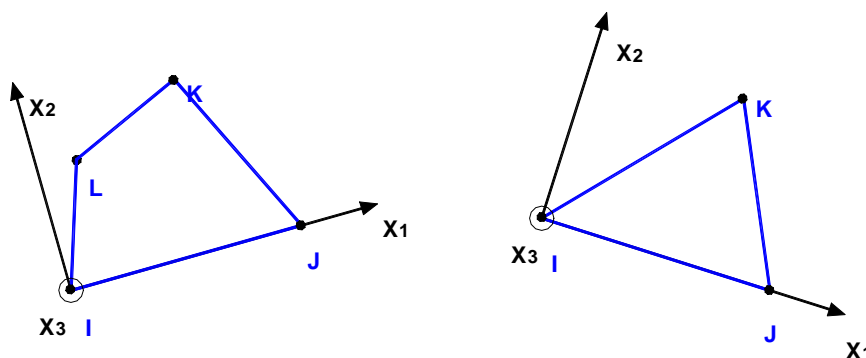


Figura 4.13. – Sistemas locais de referência para o elemento placa ou casca.

## **5. MÓDULO 3 – ESTRUTURA DE CONTRAVENTAMENTO**

---

### **5.1. Módulo de Contraventamento – Considerações Gerais**

Em razão das diferenças de escopo entre os módulos 1 e 2, para geração do pavimento, e este módulo, para montagem da estrutura de contraventamento, optou-se em se fazer esse módulo independente dos dois primeiros de forma a introduzir maior flexibilidade nos recursos a serem oferecidos, bem como para facilitar manutenção e futuras implementações.

A metodologia empregada na construção desse módulo e as premissas relacionadas ao assunto, foram retiradas do trabalho “Contribuições à Análise de Estruturas de Contraventamento de Edifícios em Concreto Armado” desenvolvido por PEREIRA (1997).

Um primeiro conceito importante, para efeito de análise de ações horizontais, diz respeito à definição de elementos de contraventamento e elementos contraventados. Segundo FUSCO (1986), seriam considerados elementos de contraventamento os pilares de grandes dimensões, as paredes estruturais e as treliças ou pórticos de grande rigidez. A compatibilização dos deslocamentos de todos esses elementos seria realizada pelas lajes dos pavimentos, que se transformariam em elementos de importância fundamental para o trabalho conjunto da estrutura.

Os elementos contraventados seriam, por hipótese, tratados como se pertencessem a uma estrutura indeslocável. Então, definida a estrutura de contraventamento, deveria-se verificar se a mesma apresentava rigidez suficiente para assegurar a validade desta hipótese e garantir assim a estabilidade global de toda a estrutura. Por fim, analisaria uma eventual ocorrência de instabilidade localizada, separadamente em cada um dos elementos da estrutura.

Durante muito tempo, vários pesquisadores defenderam que nos edifícios altos, não seria recomendável que todos os elementos estruturais participassem do sistema de contraventamento, já que, contando com recursos limitados de

hardware e software, tal operação resultaria em um projeto excessivamente trabalhoso e com resultados pouco confiáveis.

Sabe-se entretanto que atualmente os recursos de hardware e software são praticamente ilimitados se comparados ao universo de alguns anos atrás, e sabe-se também que quando um edifício é solicitado pela ação lateral do vento, toda a sua estrutura é mobilizada e que os esforços internos desenvolvidos em cada elemento são diretamente relacionados a sua rigidez. Torna-se importante, portanto, identificar quais elementos podem realmente ser desprezados e quais que merecem ser considerados.

Seguindo essa linha de raciocínio, pode-se citar as vigas secundárias, ou seja, vigas que não se apoiam diretamente em pilares. Essas vigas não são usualmente consideradas para efeito de contraventamento, mas em alguns casos é fundamental que elas sejam consideradas para que se possa criar um modelo que represente, com um bom grau de aproximação, o comportamento real da estrutura.

E é nesse contexto que se insere esse programa para gerar a estrutura de contraventamento, a de facilitar a montagem do modelo permitindo a inclusão do maior número de detalhes, como a consideração de vigas secundárias, nós mestres para considerar a rigidez do pavimento e a consideração de "OFFSETS" ou trechos rígidos, para representar uma ligação viga-pilar.

Para fazer isso, o programa dispõe de resultados parciais gerados por pavimento pelo programa GLS. Ao receber os arquivos com o modelo final do sistema Mestre, o arquivo básico \*.B, o de vigas \*.V, o de pilares \*.P, o de lajes \*.L, e o de carregamentos \*.C; o programa GLS formata todas as informações necessárias para o processamento do pavimento pelo sistema LASER e paralelamente a isso, também gera um arquivo básico específico para a análise das ações horizontais. Esse arquivo possui a extensão .XY e é chamado modelo básico de pavimento, possuindo informações apenas dos elementos que participam da estrutura de contraventamento.

A partir desse arquivo de modelo básico do pavimento, pode-se construir outro modelo que consiste na repetição desse pavimento  $n$  vezes, de forma a simular um edifício constituído de  $n$  pavimentos iguais. A informação de como é feita essa montagem, como o número de repetições; vigas e pilares que devem ser considerados na análise; bem como dados gerais como a velocidade básica do vento; os coeficientes topográfico e estatístico; a classe e a categoria da edificação; os coeficientes de arrasto segundo as direções X e Y; entre outros, ficam armazenadas em um arquivo de geração (\*XY.GEV).

Como no programa disponível esse arquivo baseia-se apenas no esquema de reprodução de um único modelo básico de pavimento, o gerador de estruturas

de contraventamento deste trabalho avança no sentido de permitir a possibilidade de considerar vários modelos de pavimentos ao montar um edifício. Assim pode-se considerar pavimentos específicos para representar cada tipo de pavimento como, por exemplo, o térreo o tipo e a cobertura.

A seguir é apresentada a tela de interface com o usuário desenvolvida para esse módulo, onde observa-se que é possível especificar o número de pavimentos diferentes e para cada pavimento diferente, o sistema permite especificar o nome do arquivo de entrada dados e o número desejado de repetições. A informação do pé-direito de cada pavimento exibida na tela é obtida do próprio arquivo de dados uma vez que este já contém as barras que representam os pilares.

NPD	Arquivo de dados do pavimento	Nome do pavimento	Repetições
1	C:\Modelos\Pavimentos\T1	Térreo	1
2	C:\Modelos\Pavimentos\T2	Pavimento Tipo	10
3	C:\Modelos\Pavimentos\T3	Cobertura	1

Figura 5.1 – Tela de interface com o usuário do sistema edifício.

Após a especificação dessas informações é possível visualizar o resultado do modelo clicando-se no botão Visualizar, figura 5.2.

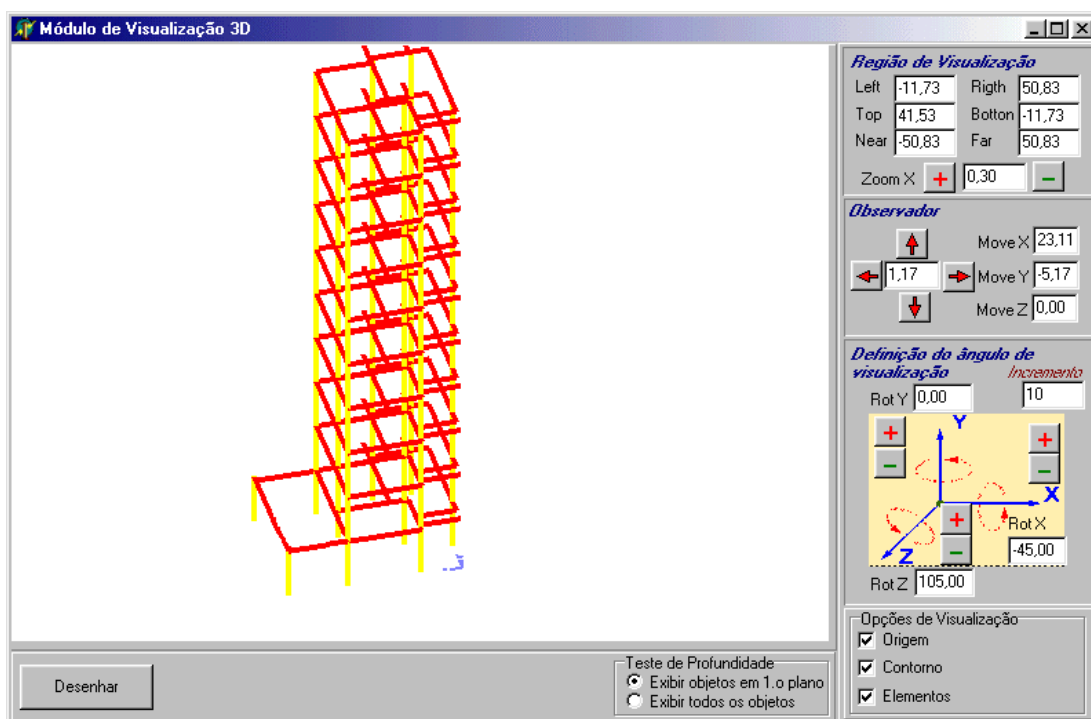


Figura 5.2 – Visualização de uma estrutura de contraventamento gerada pelo sistema Edifício.

Com isso a formatação do arquivo \*XY.GEV foi alterada de forma a indicar os arquivos do modelo básico de pavimento (\*.XY) que compõem o modelo final do edifício, bem como o número de repetições que devem ser consideradas para cada pavimento.

A estrutura do sistema Edifício é relativamente simples sendo composta por apenas dois formulários, o formulário de entrada de dados (figura 5.1) e o formulário para visualização da estrutura gerada (figura 5.2). Com relações às unidades de código, existem as unidades associadas aos formulários mencionados anteriormente, nos quais estão concentrados a maior parte do código desse sistema, e uma unidade auxiliar para manipulação de arquivos.

Todo o código relacionado a esse módulo encontra-se no CD-ROM em anexo, onde também é possível obter mais informações sobre o módulo de visualização utilizado nesse módulo, tópico - O Visualizador do Sistema – Visualiza3D.

Na página seguinte é apresentado um esquema resumindo o funcionamento do Sistema Edifício, e, no tópico a seguir, são apresentadas as ações horizontais consideradas no carregamento da estrutura de contraventamento.

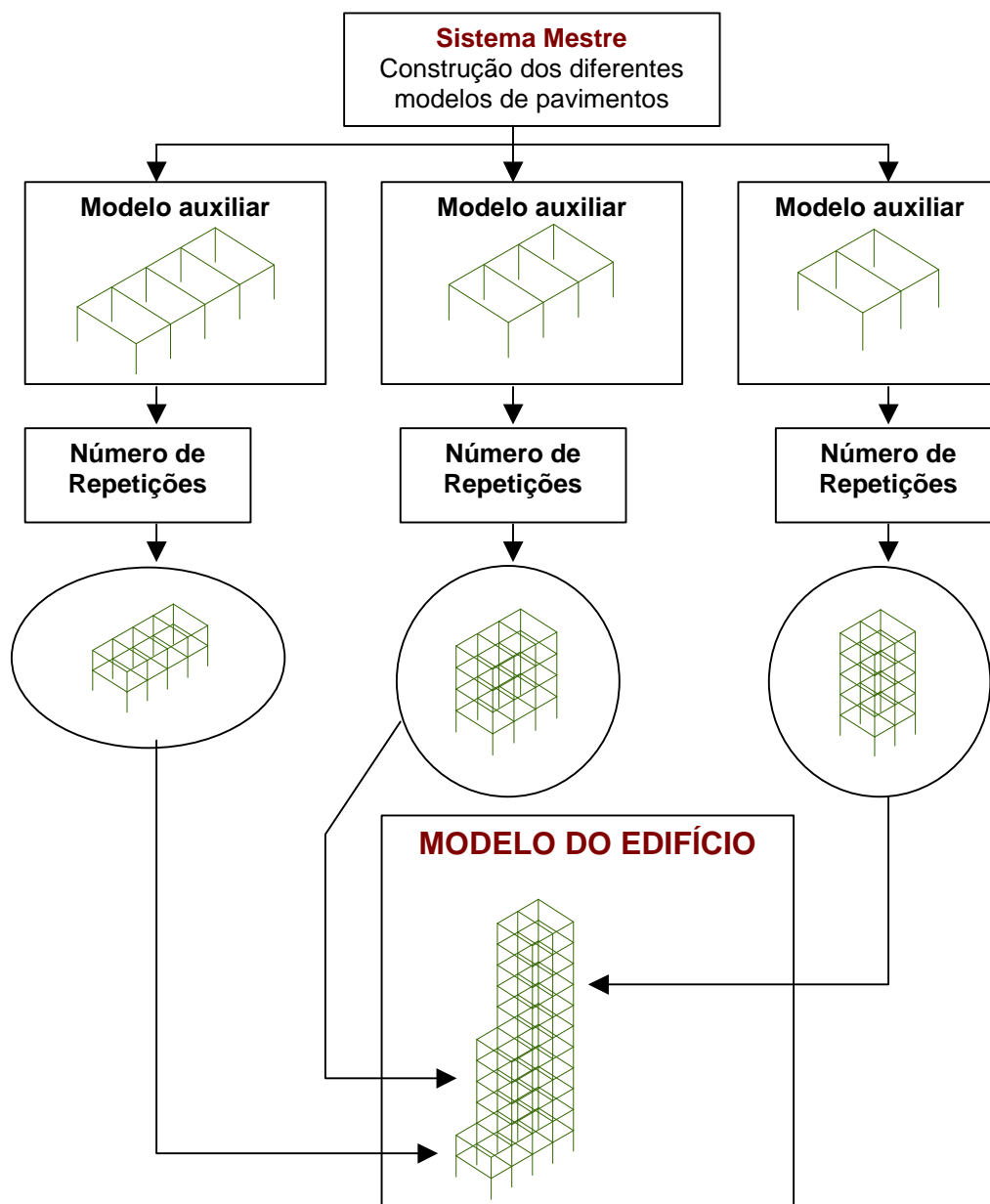


Figura 5.3 – Esquema da metodologia utilizada no Sistema Edifício para a montagem da estrutura de contraventamento.

## 5.2. Ações Horizontais

Para efeito de ações horizontais, este trabalho considera apenas as ações devidas ao vento e as ações devidas ao desaprumo. Segundo a revisão da NBR 6118 (NB-1), a consideração da ação do vento deve ser realizada de acordo com o prescrito na NBR 6123, sem exceção, seja qual for a relação entre a altura e a menor largura da edificação, e o número de filas de pilares.

Como no Brasil não é comum a ocorrência de abalos sísmicos e terremotos, estes tipos de ações não são considerados neste trabalho. Caso seja necessário a definição dessas forças, deve-se consultar normas específicas que sejam válidas para o local onde será construída a edificação.

Os itens a seguir descrevem como as ações horizontais foram consideradas neste trabalho.

### 5.2.1 Ações devidas ao vento

O módulo de contraventamento deste trabalho determina as forças devidas a ação do vento segundo o que prescreve a NBR 6123 (1988) – Forças devidas ao vento em Edificações.

Essa norma indica como quantificar a ação do vento em estruturas de edifícios, normalmente aplicada estaticamente. A rigor a ação do vento sempre deveria ser considerada como uma solicitação dinâmica, mas devido as dificuldades inerentes desse tipo de análise e devido a grande variabilidade existente na ação do vento, este tipo de análise, mais refinado, limita-se apenas a construções muito esbeltas, extensas e flexíveis.

A resposta dinâmica de uma edificação está relacionada à frequência natural, ao grau de amortecimento dos materiais empregados e à rigidez global, sendo portanto uma particularidade de cada edificação. Esses efeitos, quando significativos, podem aumentar em muito as solicitações e a estrutura pode entrar em colapso. Por esse motivo não se deve desprezar os efeitos dinâmicos do vento quando estes são significativos.

Em razão da maioria dos edifícios correntes de concreto armado não serem estruturas tão sensíveis aos efeitos dinâmicos do vento, este trabalho considera apenas as forças estáticas devidas à ação do vento.

Segundo a NBR 6123 (1988), para as estruturas de edifícios paralelepípedicos (item 6.6.2 da norma), o projeto deve levar em conta:



- As forças devidas ao vento agindo perpendicularmente a cada uma das fachadas;

- As excentricidades causadas por vento agindo obliquamente ou por efeitos de vizinhança. Os esforços de torção oriundos daí são calculados considerando estas forças agindo, respectivamente, com as seguintes excentricidades em relação ao eixo vertical geométrico:

- Edificações sem efeitos de vizinhança

$$e_a = 0,075.a \quad e \quad e_b = 0,075.b$$

- Edificações com efeitos de vizinhança

$$e_a = 0,15.a \quad e \quad e_b = 0,15.b$$

Ainda segundo a Norma, os efeitos de vizinhança são considerados somente até a altura do topo da ou das edificações situadas nas proximidades, dentro de um círculo de diâmetro igual à altura da edificação em estudo, ou igual a seis vezes o lado menor da edificação,  $b$ , adotando-se o menor destes dois valores.

Conforme mencionado, considera-se que o vento atua sobre as paredes que estão dispostas na perpendicular à sua direção. Estas passam a ação às lajes dos pavimentos que distribuem, de acordo com a rigidez, aos painéis de contraventamento.

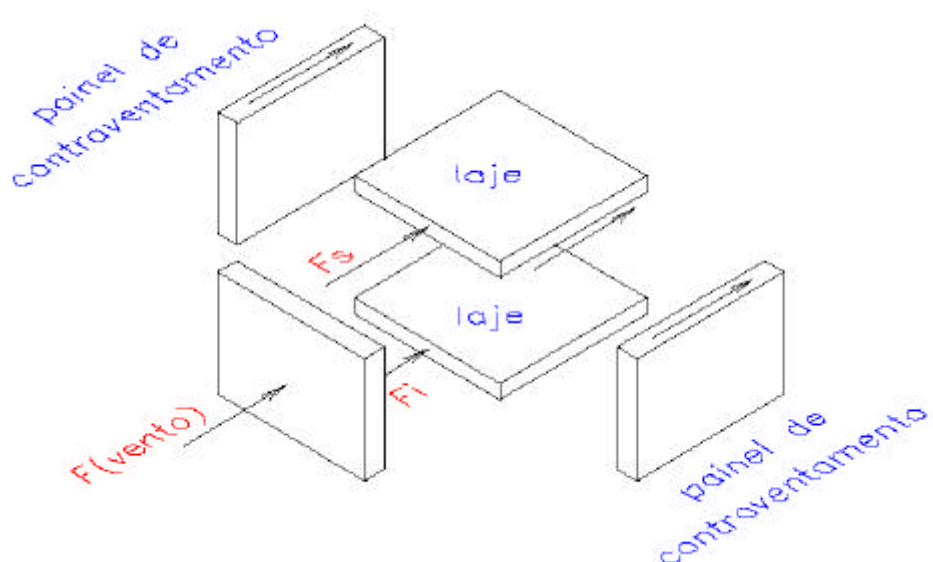


Figura 5.4 - Atuação do vento e distribuição para os painéis de contraventamento

Para determinar a ação horizontal a ser associada a cada uma das direções do vento, deve-se seguir as prescrições apresentadas a seguir:

**a) Determinar a velocidade característica  $v_k$**

A velocidade característica é calculada pela expressão:

$$v_k = S1 \cdot S2 \cdot S3 \cdot v_o$$

onde

$v_o$  : velocidade básica ( isopletas da figura 1 da NBR 6123 )

$S1$  : fator topográfico ( item 5.2 da NBR 6123 )

$S2$  : fator de rugosidade e regime ( equação 2.4 ou tabela 2 da NBR 6123 )

$S3$  : fator estatístico ( expressão 2.5 ou tabela 3 da NBR 6123 )

Dos fatores apresentados o mais importante é o  $S2$  . Para a sua determinação é necessário conhecer a categoria do terreno, a classe da edificação e a altura sobre o terreno do ponto considerado.

A categoria do terreno é dada pela sua rugosidade, conforme as definições que se seguem:

- Categoria I : superfícies lisas de grandes dimensões, mais de 5 km na direção e sentido do vento incidente.
- Categoria II : terreno aberto em nível com poucos obstáculos isolados, como árvores ou pequenas construções.
- Categoria III : terrenos planos com obstáculos como muros, edificações baixas e esparsas.
- Categoria IV : terreno com obstáculos numerosos e pouco espaçados em zonas florestal, industrial e urbanizada.
- Categoria V : terreno com obstáculos numerosos, altos e pouco espaçados, como centro de grandes cidades.

Já a classe de uma edificação é definida pela sua maior dimensão, e guarda relação com o intervalo de cálculo de sua velocidade média. São utilizados intervalos de 3, 5 e 10s, respectivamente :

- classe A : edificações com maior dimensão menor que 20 m.
- classe B : edificações com maior dimensão entre 20 e 50 m.
- classe C : edificações com maior dimensão maior que 50 m.

Z (m)	Categoria														
	I			II			III			IV			V		
	Classe			Classe			Classe			Classe			Classe		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
≤ 5	1,06	1,04	1,01	0,94	0,92	0,89	0,88	0,86	0,82	0,79	0,76	0,73	0,74	0,72	0,67
10	1,10	1,09	1,06	1,00	0,98	0,95	0,94	0,92	0,88	0,86	0,83	0,80	0,74	0,72	0,67
15	1,13	1,12	1,09	1,04	1,02	0,99	0,98	0,96	0,93	0,90	0,88	0,84	0,79	0,76	0,72
20	1,15	1,14	1,12	1,06	1,04	1,02	1,01	0,99	0,96	0,93	0,91	0,88	0,82	0,80	0,76
30	1,17	1,17	1,15	1,10	1,08	1,06	1,05	1,03	1,00	0,98	0,96	0,93	0,87	0,85	0,82
40	1,20	1,19	1,17	1,13	1,11	1,09	1,08	1,06	1,04	1,01	0,99	0,96	0,91	0,89	0,86
50	1,21	1,21	1,19	1,15	1,13	1,12	1,10	1,09	1,06	1,04	1,02	0,99	0,94	0,93	0,89
60	1,22	1,22	1,21	1,16	1,15	1,14	1,12	1,11	1,09	1,07	1,04	1,02	0,97	0,95	0,92

Tabela 5.1 - Valores do coeficiente S2 - NBR 6123

**b) Determinar a pressão de obstrução**

A pressão de obstrução é calculada pela expressão:

$$q = 0,613 \cdot vk^2$$

onde:

$q$  : pressão de obstrução em N/m<sup>2</sup>

$vk$  : velocidade característica em m/s

**c) Determinar a força de arrasto**

A força de arrasto é calculada para cada direção considerada:

$$F = Ca \cdot q \cdot Ae$$

onde

$q$  : pressão de obstrução

$Ae$  : área da superfície onde o vento atua ( área de influência )

$Ca$  : coeficiente de arrasto

O coeficiente de arrasto  $Ca$  depende da direção e do regime do vento. Se tratar-se de vento de baixa turbulência deve-se utilizar a figura 4 da NBR 6123., aqui reproduzida com figura 5.5. Caso se trate de vento de alta turbulência deve-se utilizar a figura 5 da NBR 6123, aqui reproduzida como Fig 5.6.

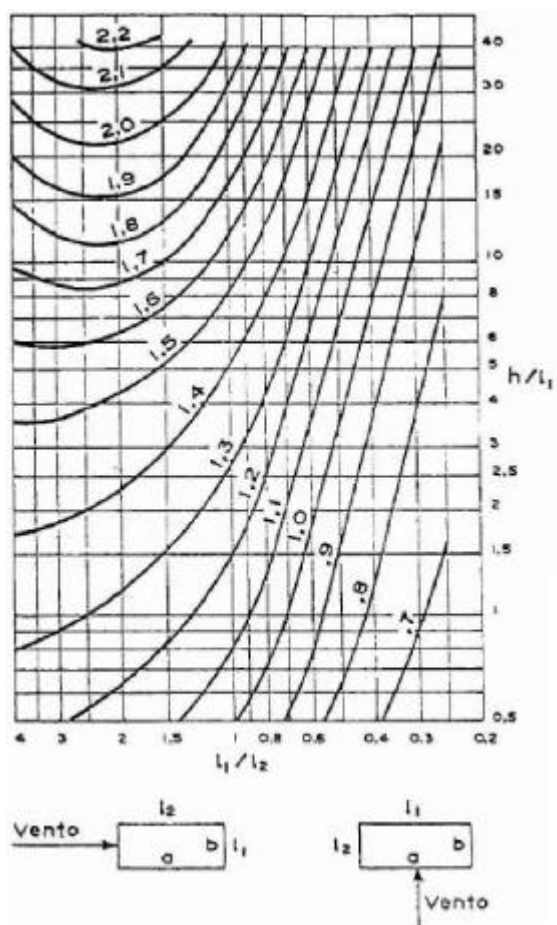


Figura 5.5 - Coeficiente de arrasto  $C_a$  para vento de baixa turbulência

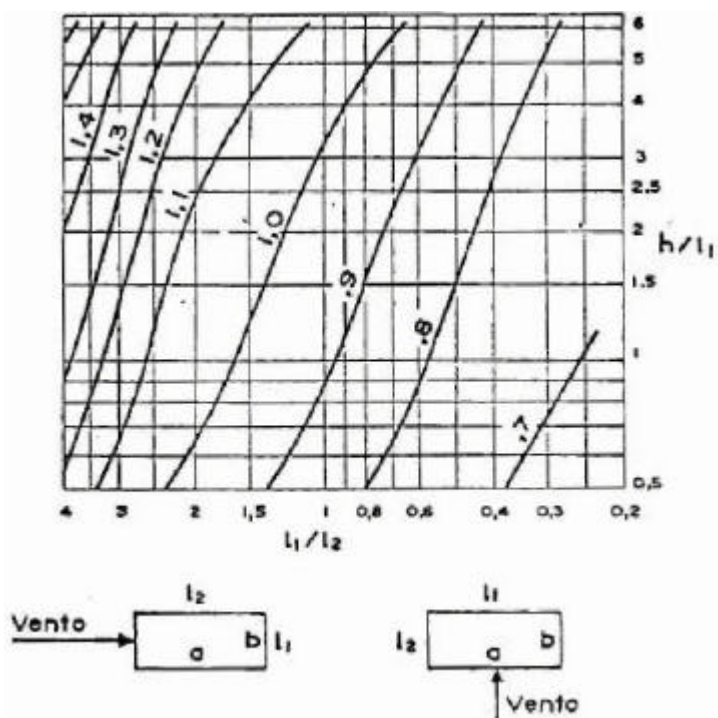


Figura 5.6 - Coeficiente de arrasto  $C_a$  para vento de alta turbulência

O coeficiente de arrasto para vento de alta turbulência é muito mais favorável à economia que o de baixa turbulência. Entretanto, como coloca a Norma “uma edificação pode ser considerada em vento de alta turbulência quando a sua altura não excede duas vezes a altura média das edificações da vizinhança, estendendo-se estas, na direção e sentido do vento incidente, a uma distância mínima de :

- 500 m para edificações de até 40 m de altura.
- 1000 m para edificações de até 55 m de altura.
- 2000 m para edificações de até 70 m de altura.
- 3000 m para edificações de até 80 m de altura.”

Uma vez determinadas as forças de arrasto a serem aplicadas em cada nível do pavimento, o modelo tridimensional, que representa a estrutura de contraventamento, é carregado com essas forças para o posterior processamento.

Para finalizar, aproveita-se para destacar a vantagem em se utilizar um modelo tridimensional na determinação dos esforços solicitantes devidos às ações horizontais de vento em edifícios, uma vez que, levando em conta o aspecto tridimensional, podem aparecer efeitos geométricos de torção global, além da possibilidade de introdução do vento em quaisquer direções.

### 5.2.2 Ações devidas ao Desaprumo

Inicialmente este trabalho tomou por base a norma alemã, a DIN-1053, mas como a revisão da NBR-6118 (NB-1) prescreve o procedimento para essa análise, considerou-se adequado adotar o que foi definido nesta última.

Segundo a revisão da NBR-6118 (NB-1), na análise global de estruturas contraventadas ou não, deve ser considerado um desaprumo dos elementos verticais conforme ilustra a figura 5.7.

$$q_1 = \frac{1}{100 \cdot \sqrt{H}}$$

$$j = q_1 \cdot \sqrt{1 + \frac{1}{n}}$$

onde

$\theta_{1\min} = 1/400$  para estruturas de nós fixos

$\theta_{1\min} = 1/300$  para estruturas de nós móveis e imperfeições locais

$\theta_{1\min} \leq 1/200$

$n$  = número de prumadas de pilares  
Os ângulos  $\theta$  e  $\varphi$  são dados em radianos

Ainda segundo a Norma, “o desaprumo mínimo  $\theta_{1\min}$  não deve necessariamente ser superposto ao carregamento de vento. Entre os dois, vento e desaprumo, pode ser considerado apenas o mais desfavorável, que pode ser definido pelo que provoca o maior momento total na base da construção.”

Através do ângulo  $\varphi$ , pode-se determinar uma ação lateral equivalente ao desaprumo, a ser aplicada ao nível de cada pavimento, através da expressão :

$$F_d = DP \cdot \varphi$$

onde

DP : peso total do pavimento considerado

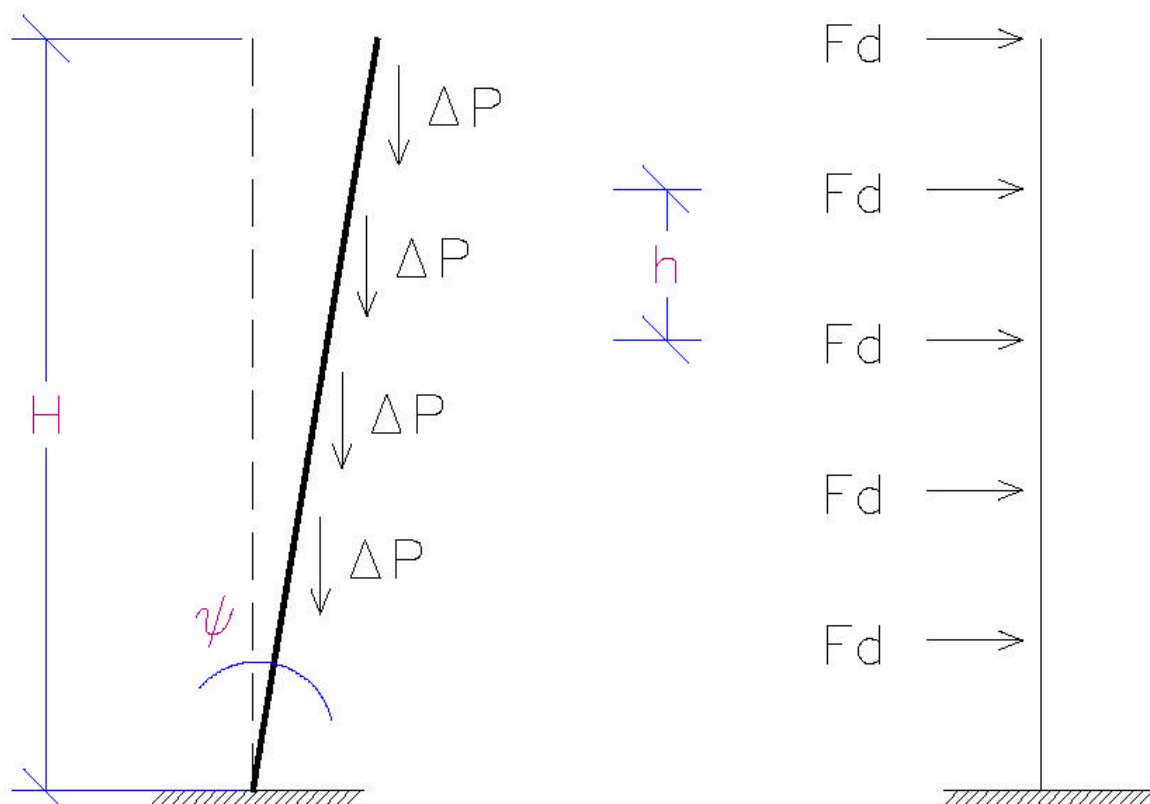


Figura 5.7 - Ação lateral equivalente à consideração do desaprumo – Consideração das imperfeições geométricas globais

## **6. MÓDULO 4 – VISUALIZAÇÃO DE RESULTADOS**

---

### **6.1 Estrutura Básica do Sistema VISUALIZA3D**

Em uma boa parte o módulo de visualização de resultados é baseado no programa CPLS desenvolvido em linguagem PASCAL, LISERRE (1993). Esse programa é um traçador de curvas de isovalor com saída gráfica no espaço bidimensional para momentos, tensões e deslocamentos, funcionando basicamente para elementos placa ou casca.

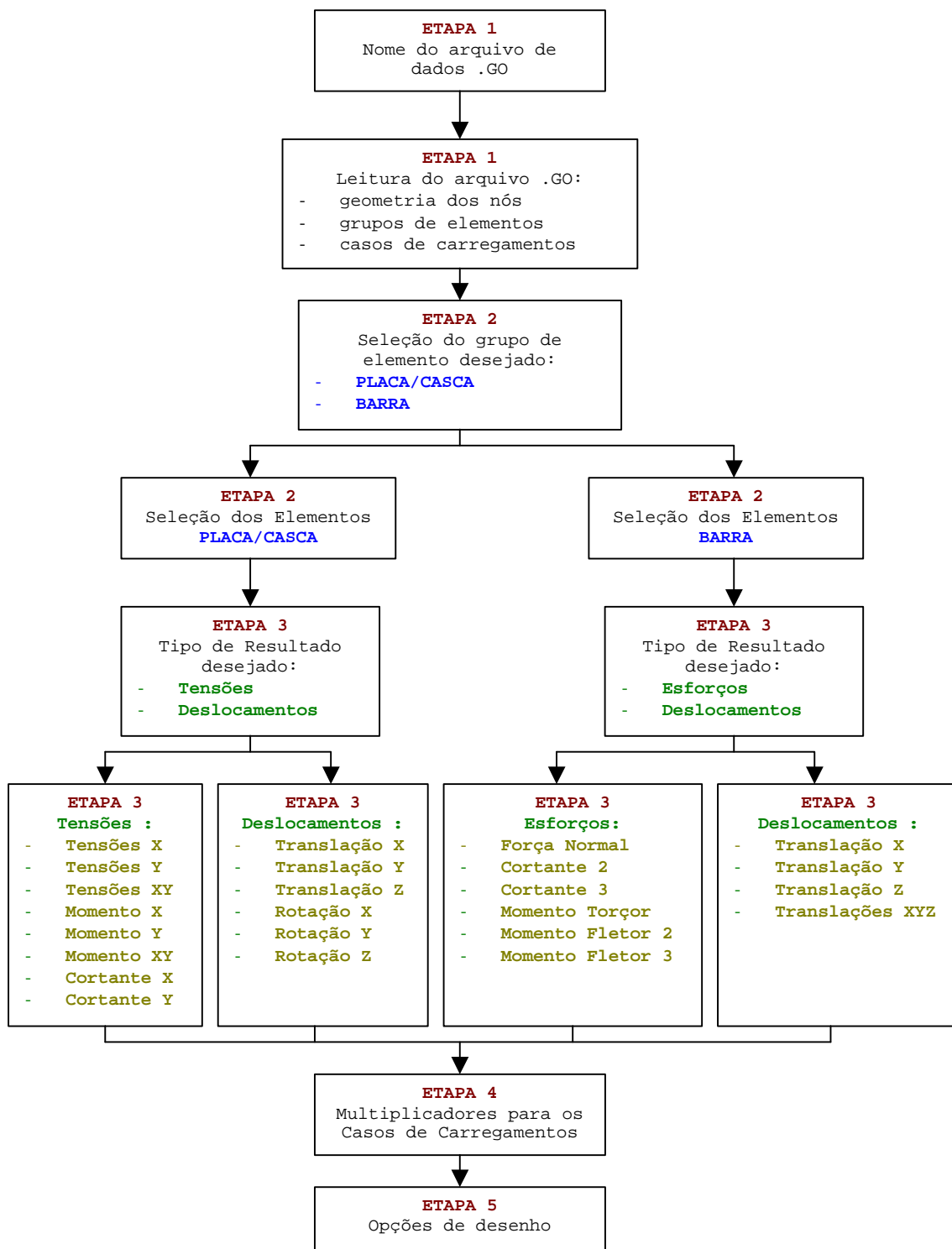
Já o novo módulo de visualização desenvolvido, refere-se a uma interface totalmente remodelada com novos recursos, onde se destacam o suporte ao desenho de diagramas de esforços solicitantes para elementos barras e a visualização tridimensional tanto para placas e cascas quanto para barras. Esse novo módulo oferece também o suporte para a visualização de resultados para estruturas de contraventamento, produzindo desenhos para esforços, momentos fletores e força cortante segundo os dois eixos principais dos elementos barra tridimensional, bem como a configuração deslocada da estrutura. Vale ressaltar que as curvas de isovalor são geradas apenas para o grupo de elementos placa/casca.

Devido a possibilidade de se utilizar esse visualizador em outros programas de cálculo estrutural sem vínculos com o sistema Mestre, decidiu-se por produzir esse módulo em um novo aplicativo totalmente independente do sistema Mestre, denominado então Visualiza3D. No entanto, tal aplicativo pode ser executado diretamente do sistema Mestre a partir do menu Módulos.

Esse novo aplicativo utiliza as mesmas rotinas para geração de curvas de isovalor desenvolvidas no programa CPLS, só que com alterações profundas na entrada de dados por parte do usuário e algumas alterações adicionais para se adequar a nova linguagem Object Pascal. Além disso o novo aplicativo possui uma série de novas rotinas responsáveis pelas saídas tridimensionais e para o desenho dos diagramas de esforços nas barras.

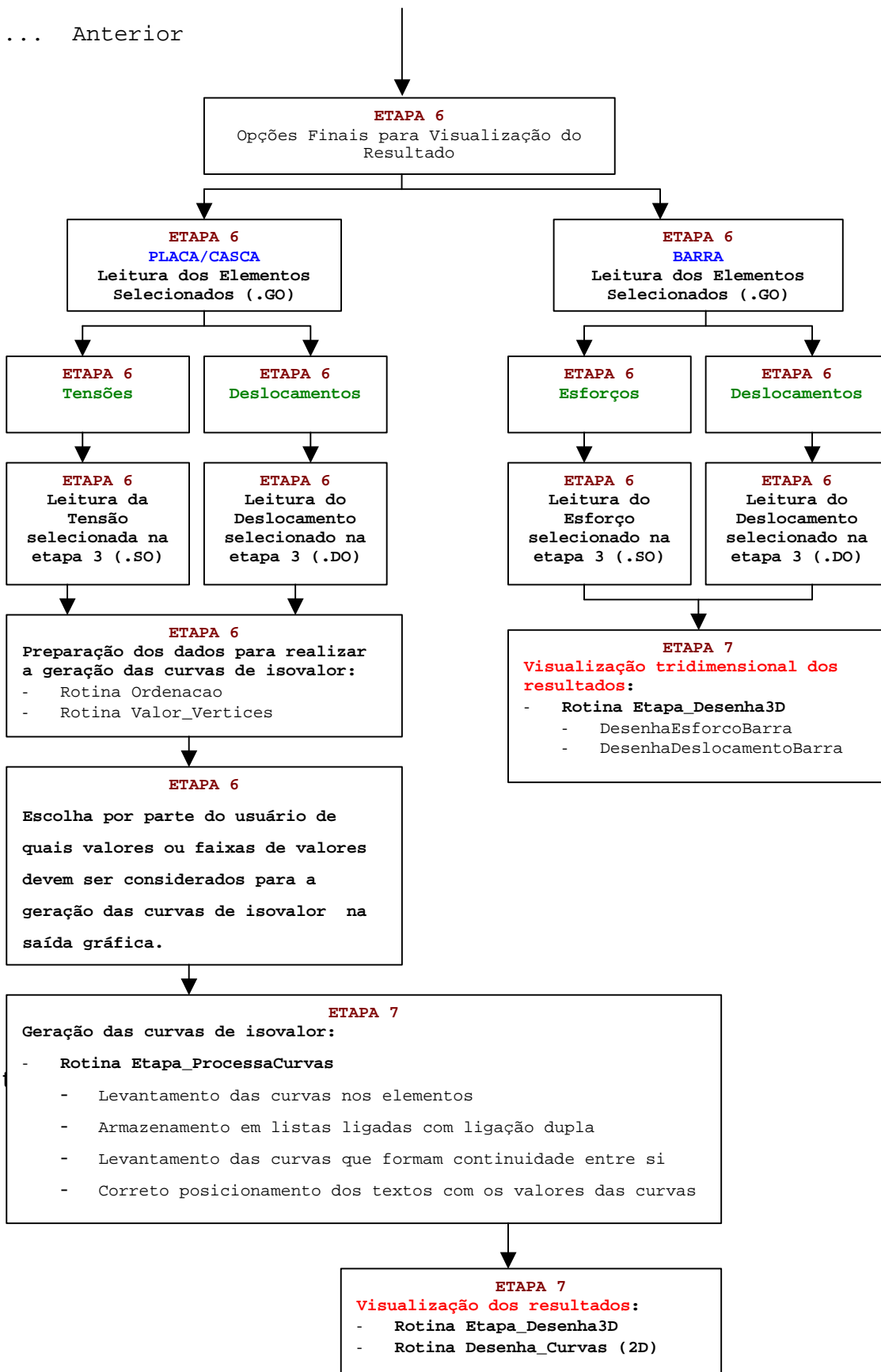
Para ficar mais claro a seqüência de tarefas realizadas para as saídas gráficas, a seguir é mostrado o fluxograma das etapas envolvidas no processo desde a entrada de dados até o processamento e visualização dos resultados.

Fluxograma do processo de visualização de resultados:



Continua na próxima página ....





E a seguir segue a descrição detalhada de cada etapa:

**Etapa 1** – O usuário é solicitado a informar qual é o arquivo de entrada que deseja visualizar. Esse arquivo é o resultado do processamento pelo sistema LASER e possui a extensão .GO (arquivo de geometria), sendo que, além desse arquivo, devem estar presentes também no mesmo diretório, os arquivos com as extensões .SO (arquivo de tensões e esforços) e .DO (arquivo de deslocamentos), ambos resultados do processamento do sistema LASER.

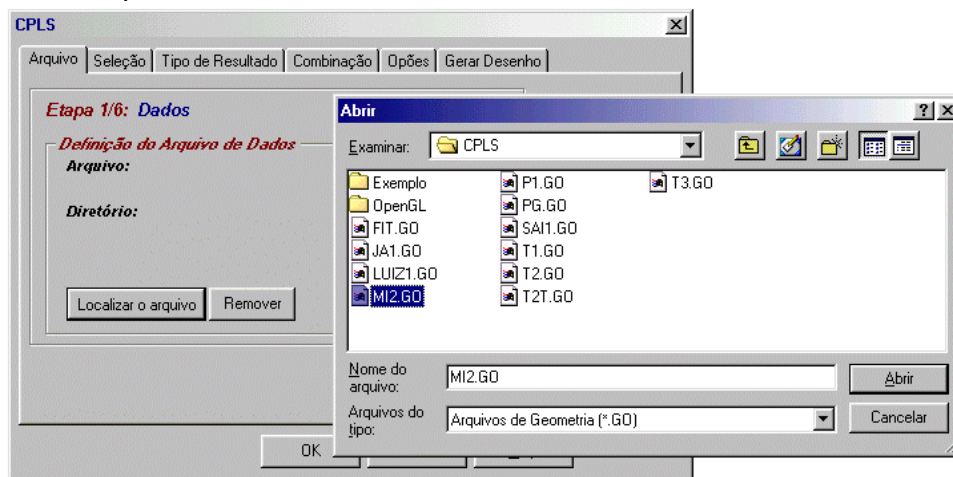


Figura 6.1 – Processo de Visualização de Resultados – ETAPA 1/6

**Etapa 2** – A etapa 2 consiste na escolha do grupo de elemento, barra ou placa/casca, desejado para se realizar a análise, bem como pela definição de quais elementos farão parte da visualização. Com relação à essa última opção, é possível selecionar todos os elementos de uma única vez através do botão **Seleciona Todos**, ou selecionar apenas os elementos ou intervalo de elementos desejados através do botão **Seleção**.

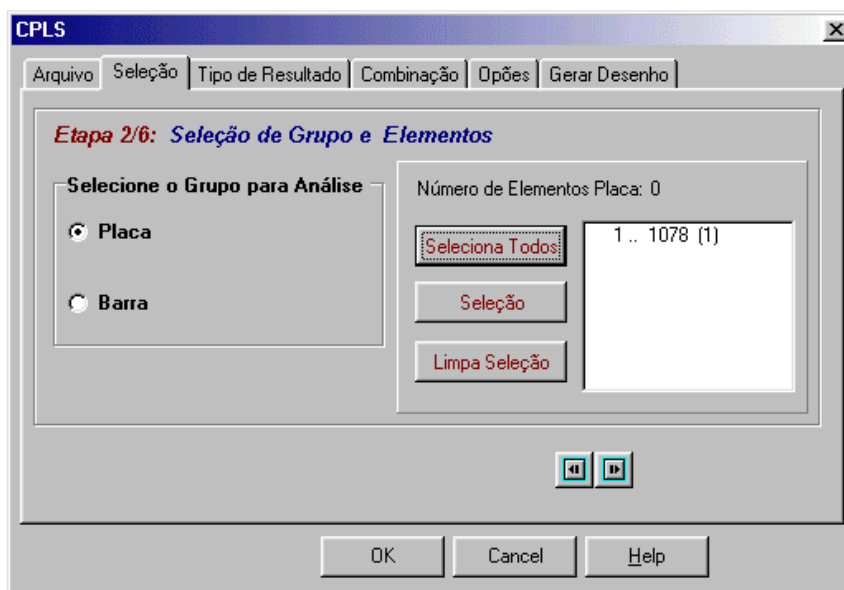


Figura 6.2 – Processo de Visualização de Resultados – ETAPA 2/6

**Etapa 3** – A etapa 3 consiste na definição do tipo de resultado desejado para a visualização. O tipo de resultado a ser selecionado varia de acordo com o tipo de grupo de elemento, barra ou placa/casca, selecionado na etapa 2. Sendo o seguinte:

**- Para barra:**

( saída gráfica na forma de diagramas para os esforços e estrutura deformada para os deslocamentos )

- **Esforços** : Força Normal, Cortante 2, Cortante 3, Momento Torçor, Fletor 2, Fletor 3.
- **Deslocamentos** : Translação X, Translação Y, Translação Z  
Translações X,Y e Z juntas

**- Para placa/casca:**

( saída gráfica na forma de curvas de isovalor )

- **Tensões** : Tensões X, Tensões Y, Tensões XY, Momento X, Momento Y, Momento XY, Cortante X, Cortante Y
- **Deslocamentos** : Translação X, Translação Y, Translação Z, Rotação X, Rotação Y, Rotação Z

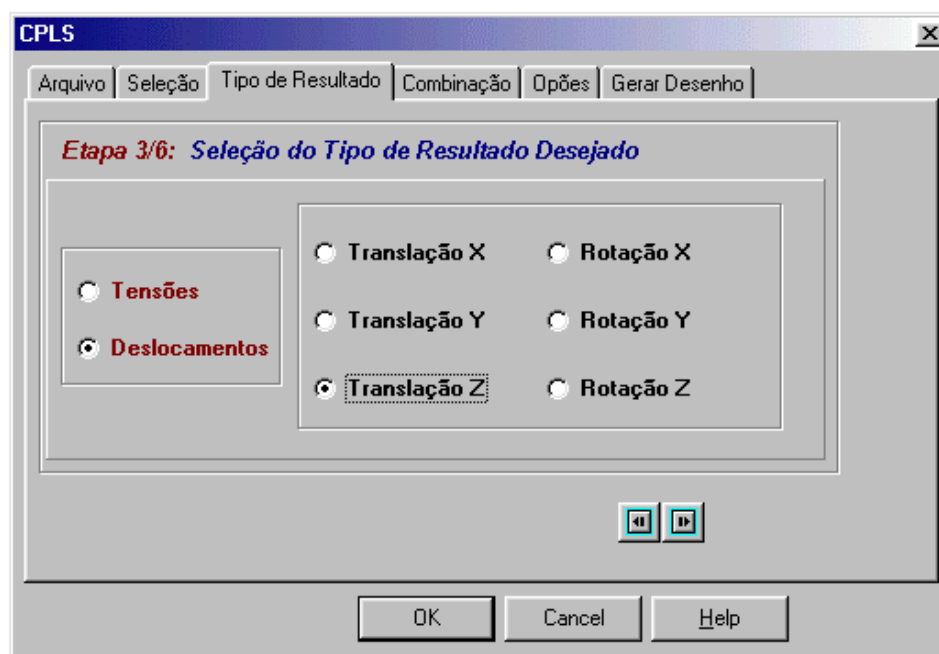


Figura 6.3 – Processo de Visualização de Resultados – ETAPA 3/6

**Etapa 4** – A etapa 4 consiste na correta definição dos multiplicadores para os casos de carregamentos existentes.

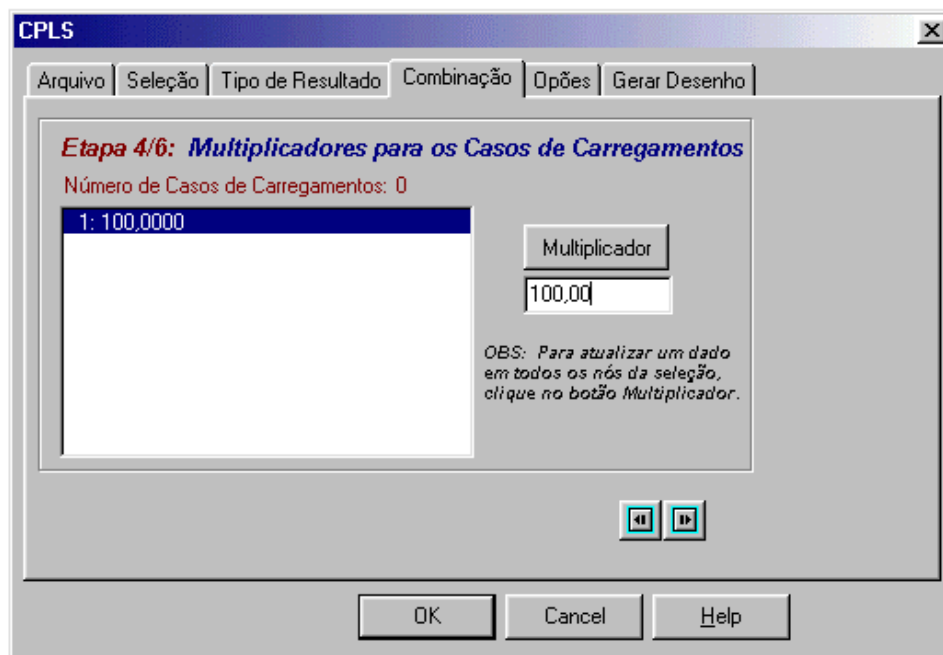


Figura 6.4 – Processo de Visualização de Resultados – ETAPA 4/6

**Etapa 5** – O passo 5 é opcional e permite alterar algumas opções de desenho, como optar por uma saída 2D ou 3D, incluir ou não a malha dos elementos na saída gráfica, alterar o tamanho do texto e número de casas decimais.

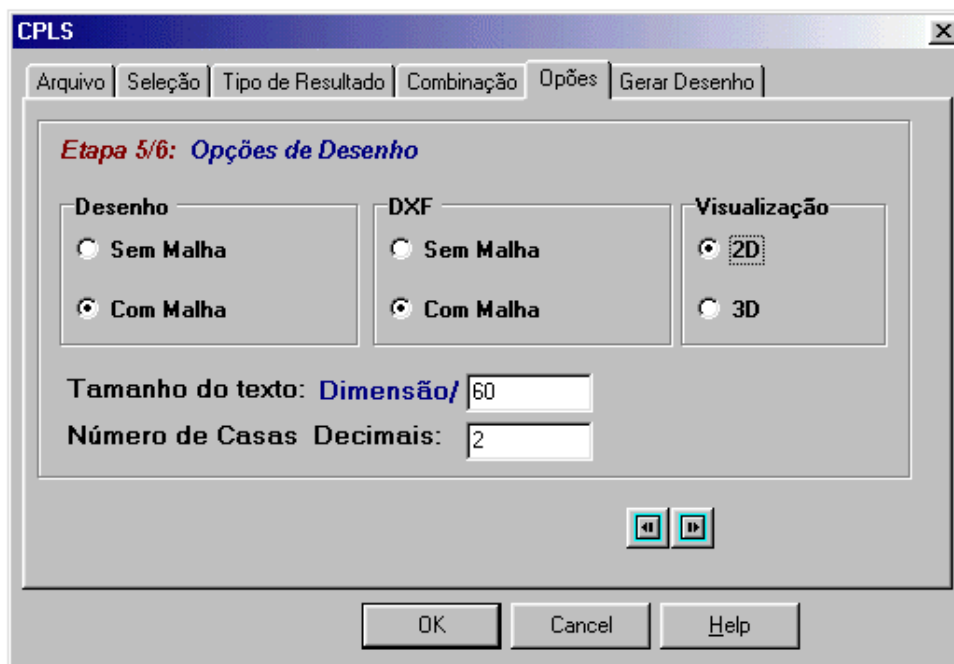


Figura 6.5 – Processo de Visualização de Resultados – ETAPA 5/6

**Etapa 6** – O passo 6 é o que finaliza o processo de definição de opções antes de gerar a saída gráfica. É nessa etapa que são lidos os arquivos .SO e .DO com os resultados das tensões, esforços e deslocamentos provenientes do resultado do processamento do sistema LASER.

Nessa etapa, os valores lidos são ajustados de acordo com o multiplicador indicado para cada caso de carregamento. Após a leitura, o resultado final é a correta combinação dos casos de carregamentos existentes, já que é a somatória de cada valor de caso de carregamento corrigido pelo seu respectivo multiplicador.

Após a leitura e ajuste dos valores lidos, no caso de elementos do tipo barra, o sistema está pronto para chamar o visualizador gráfico 3D onde serão efetivamente gerados os desenhos dos diagramas ou da estrutura deformada. Já para os elementos do tipo placa/casca, o sistema determina os valores extremos (máximo e mínimo) encontrados nos nós e nos centróides dos elementos, e permite ao usuário escolher quais valores ou faixas de valores devem ser considerados para a geração das curvas de isovalor na saída gráfica.

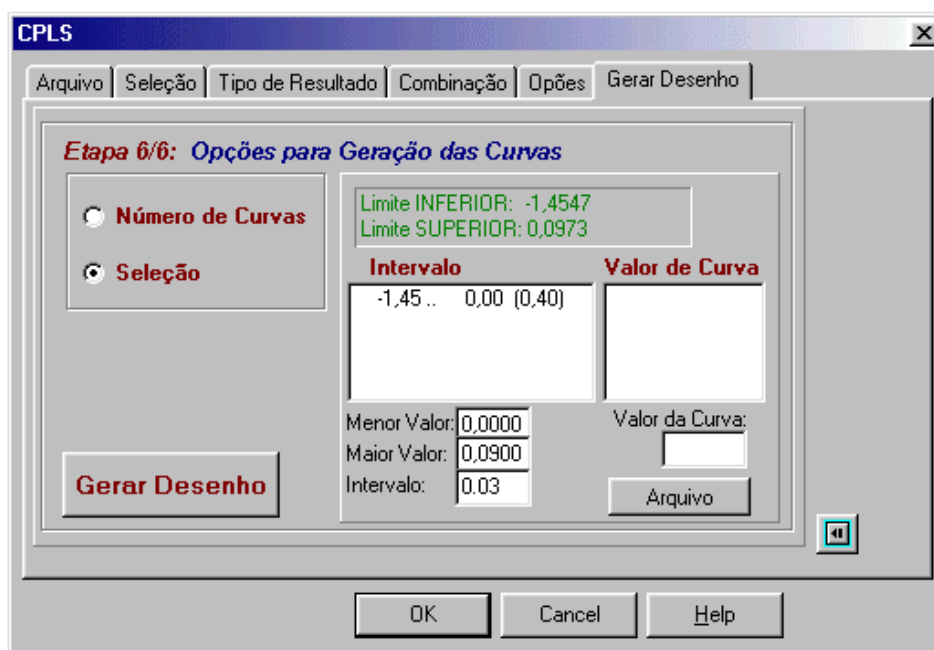


Figura 6.6 – Processo de Visualização de Resultados – ETAPA 6/6

**Etapa 7** – Uma vez selecionados, para o caso de placas/cascas, os valores das curvas ou as faixas de valores de curvas, pode-se prosseguir para a visualização dos resultados clicando-se no botão Gerar Desenho.

Nessa fase novas tarefas são realizadas principalmente no caso da geração das curvas de isovalor, já que a determinação do traçado das curvas envolve a

utilização de estruturas de listas. Para mais informações pode-se consultar no CD-ROM em anexo o tópico sobre o módulo de visualização de resultados.

A figura abaixo exibe um exemplo de uma saída em 2D com as curvas de isovalor para o deslocamento na direção Z.

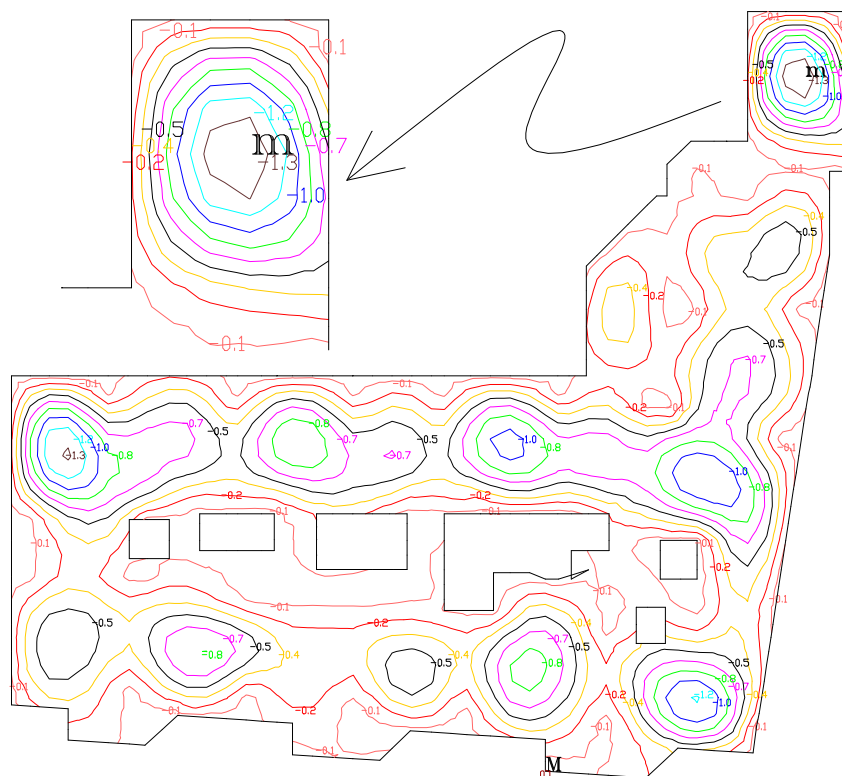


Figura 6.7. – Resultado típico em 2D, do processamento de um pavimento com elementos do tipo placa, com levantamento das curvas de isovalor para o deslocamento na direção Z e o adequado posicionamento dos textos que identificam os valores das curvas.

No próximo tópico serão dadas as informações relacionadas à saída tridimensional.

## 6.2 Visualização Tridimensional – Biblioteca OpenGL

Para gerar esse módulo tridimensional, inicialmente surgiu uma grande dúvida com relação aos recursos que poderiam ser oferecidos e de que forma seriam oferecidos. Isso porque a manipulação de dados num espaço tridimensional é extremamente mais complexa do que no espaço bidimensional. Cuidados que antes não havia por que se preocupar, como vistas em perspectivas, noção de profundidade, planos de corte para visualizar regiões específicas, rotações no espaço tridimensional, preenchimento de objetos com variação de cor, etc, passaram a ser fundamentais para que se pudesse oferecer um ambiente de visualização tridimensional realmente satisfatório.

Como o desenvolvimento de rotinas específicas, para realizar o controle de todas essas operações, demandaria um tempo muito maior do que o disponível para realizar essa pesquisa, decidiu-se procurar alguma biblioteca gráfica já disponível que oferecesse tal gama de recursos.

Uma solução seria adotar o formato da Internet para visualização de modelos geométricos 3-D, chamado VRML – *Virtual Reality Modeling Language*. Este padrão é bem difundido e é inclusive padronizado desde 1997 através da norma ISO/IEC 14772. A versão atual é a 2.0 e é suportado por visualizadores de VRML como o Cosmos Player. Ocorre que ao se adotar esse padrão, não seria possível utilizar recursos específicos e necessários para uma melhor visualização dos resultados do processamento dos modelos gerados pelo Sistema Mestre e Edifício. Dessa forma procurou-se uma outra alternativa que permitisse criar um visualizador próprio, sem a dependência de visualizadores genéricos desenvolvidos por terceiros.

Felizmente existe uma biblioteca de rotinas gráficas, denominada OpenGL, especialmente desenvolvida para trabalhar em duas e três dimensões. Essa biblioteca possui recursos até mais avançados do que os necessários para o desenvolvimento do visualizador gráfico desejado neste trabalho, permitindo por exemplo a aplicação de efeitos fotorealísticos tais como a aplicação de texturas em objetos, mapeamento de imagens, aplicação de luzes e sombras, efeito de neblina e outros efeitos especiais, o que faz dessa biblioteca a base de muitos jogos os populares “*Games*”, programas de animação e modelagem 3D, sistemas CAD/CAM, sistemas de realidade virtual, etc.

A OpenGL “*Open Graphics Library*” é uma biblioteca gráfica criada pela Silicon Graphics Inc. em 1992, e desde então controlada e continuamente atualizada por um consórcio independente formada por várias empresas de destaque no segmento de tecnologia de aplicações gráficas, tais como: Silicon Graphics, Intergraph, 3Dlabs, ATI, Compaq, Evans & Sutherland, Hewlett-Packard, IBM, Intel, Nvidia, Microsoft, e SGI. Esse consórcio da OpenGL é conhecido como

ARB – *Architecture Review Board*, cuja função principal é a de especificar, testar e aprovar todos os melhoramentos a serem incorporados à OpenGL. Com relação à licença de uso da biblioteca, todos os usuários finais que utilizarem a OpenGL em sistemas homologados pelo consórcio possuem direito à livre utilização do produto.

A OpenGL nada mais é do que um software desenvolvido para ter uma interface direta com o hardware gráfico, no caso a placa de vídeo. Isso faz com que muitas operações sejam processadas diretamente pelo hardware da placa de vídeo ao invés de se utilizar o processador principal da placa-mãe. Essa solução tem o seu lado positivo uma vez que quase todas as operações de transformações tridimensionais e armazenamento de dados são realizadas pela placa de vídeo, liberando o processador principal para realizar outras tarefas o que gera uma saída gráfica extremamente rápida. Por outro lado tem o seu lado negativo uma vez que, para se tirar proveito total dessa capacidade, há a necessidade de se ter instalado no computador uma placa aceleradora 3D com suporte ao OpenGL, e na falta da mesma, as operações que deveriam ser realizadas pelo hardware gráfico são emuladas via software e executadas pelo processador principal o que torna as saídas gráficas mais lentas. De qualquer forma essa solução traz muito mais vantagens do que desvantagens.

Para o programador, a OpenGL é um conjunto de comandos que permite especificar a geometria de objetos em duas ou três dimensões ao mesmo tempo que disponibiliza uma série de recursos em mais de cem procedimentos e funções para produzir visualizações e imagens gráficas de alta qualidade. Além disso, existem versões da OpenGL para os diversos sistemas operacionais, como UNIX, Windows 95/98, Windows NT/2000, Mac OS, OS/2, Linux, entre outros, sendo que as rotinas, independente da linguagem de programação utilizada, são sempre as mesmas. O que difere o uso de OpenGL em uma plataforma UNIX ou Windows, ou em linguagens de programação como Delphi, C++, Java, FORTRAN, ou Visual Basic, é a forma de inicializar os recursos, de definir a região de exibição e de integrar a biblioteca OpenGL com cada linguagem, além, obviamente, das diferenças de sintaxe e notação de uma linguagem para outra.

Como a tarefa de integrar OpenGL a uma linguagem visual nem sempre é trivial, existem diversas ferramentas para esse fim, especialmente para C++Builder, Delphi e Visual Basic. Estas ferramentas são componentes ou OCXs, no caso do Visual Basic, que encapsulam a OpenGL, e tornam mais fácil a integração com estas linguagens. Para mais informações sobre a OpenGL e estas ferramentas pode-se consultar <http://www.opengl.org>.

Da aplicação da OpenGL nesta pesquisa, vale a pena descrever os recursos utilizados e a forma como foram utilizados. O CD-ROM deste trabalho apresenta mais informações de como utilizar a OpenGL em conjunto com uma linguagem de programação.



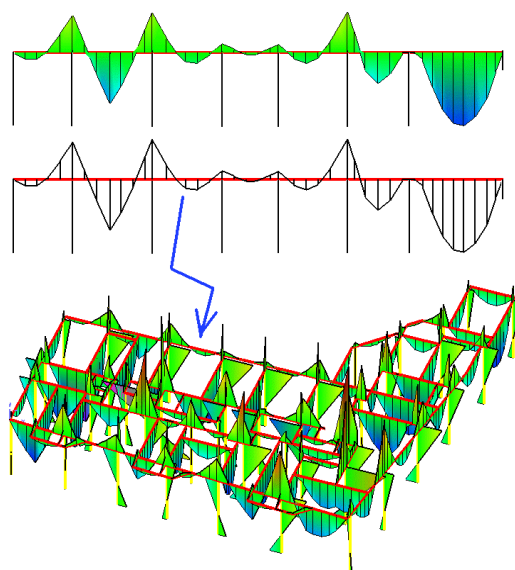


Figura 6.8– Visualização dos diagramas de momento fletor nas vigas e nos pilares

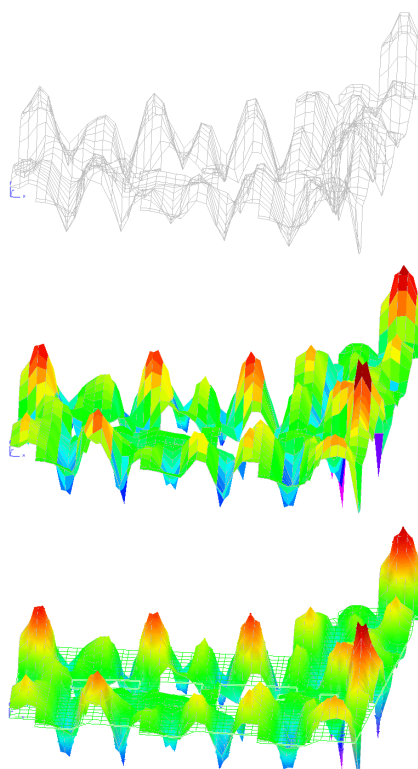


Figura 6.9– Sequência da visualização do resultado do momento fletor, segundo a direção X em um pavimento, modificando alguns parâmetros de visualização. No topo, visualiza-se apenas os elementos acompanhando os valores do resultado; no centro, visualiza-se o degradê sem suavização de cores; e abaixo, visualiza-se o degradê com suavização de cores e o desenho dos elementos no nível do pavimento. Nota-se que eles foram propositalmente pintados de verde justamente para representar o nível zero.

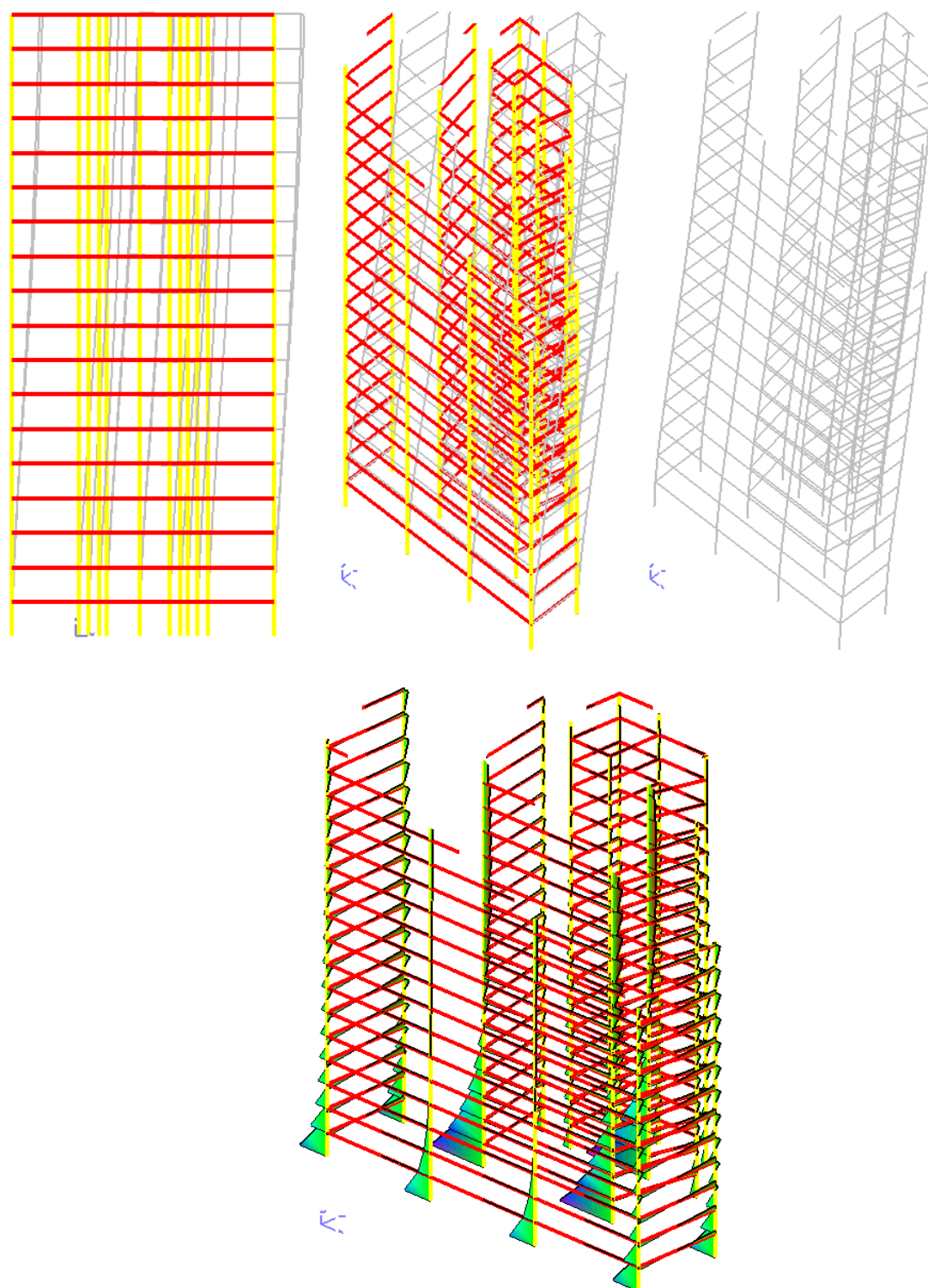


Figura 6.10– Seqüência de resultados para uma estrutura de contraventamento. No topo visualiza-se a estrutura deformada, primeiro segundo o plano XZ, depois numa vista em perspectiva e por último com a mesma vista mas sem o desenho dos elementos. Abaixo visualiza-se os diagramas de momento fletor correspondentes à estrutura deformada na direção Y.

## **7. PROJETO EXEMPLO**

---

### **7.1 Introdução**

Com o intuito de permitir uma melhor avaliação do sistema e dos recursos que o mesmo oferece, considerou-se adequado acrescentar um exemplo prático para a modelagem dos pavimentos e a estrutura de contraventamento de um edifício residencial em concreto armado que, para este exemplo, é composto de 10 pavimentos tipo mais o ático.

Além do exemplo propriamente dito, este capítulo traz, inicialmente, algumas considerações a respeito de sugestões de modelagem, cuidados na concepção da forma estrutural, e alguns critérios úteis a serem considerados na definição das dimensões dos principais componentes estruturais para a fase de pré-dimensionamento. Informações estas consideradas fundamentais para o início de qualquer atividade de modelagem estrutural.

No final, é apresentado, para efeito de comparação com o sistema Mestre, o processo de geração do pavimento tipo realizado no ANSYS, onde fica claro o maior esforço na geração do modelo quando não se dispõem de processos e ferramentas especialmente desenvolvidas para tratar da solução de pavimentos de estruturas de concreto armado.

### **7.2 Sugestões de Modelagem**

O propósito fundamental de uma análise de uma estrutura através do Método dos Elementos Finitos é de se representar matematicamente o comportamento do modelo físico com a maior exatidão possível.

Este capítulo tem como objetivo orientar e fornecer algumas sugestões aos usuário dos sistemas desenvolvidos neste trabalho, a respeito dos cuidados que se deve ter ao conceber os modelos estruturais para solução pelo Método dos Elementos Finitos.

Como apresenta ALVES FILHO (2000), é sabido que a grande popularidade do Método dos Elementos Finitos nos dias de hoje deve-se justamente pelo seu conceito básico, a discretização, produzir inúmeras equações algébricas simultâneas que são geradas e resolvidas com o auxílio de computadores digitais. Assim, ao se utilizar desse método com o auxílio de computadores, o usuário acaba tendo acesso a procedimentos que não envolvem explicitamente decisões de engenharia. Grosseiramente falando, basta gerar uma geometria que reproduza o modelo desejado e alguns outros comandos a mais, para discretizar o modelo, adicionar carregamentos e definir condições de contorno, muitas vezes sem critério algum, que em poucos minutos o programa já estará “cuspidando” resultados.

Diante disso, salienta-se a importância de que o usuário, ou melhor, o engenheiro, deve estar muito seguro com relação à ferramenta que está lidando e, também, a respeito dos conhecimentos que deve possuir para a correta utilização de um software baseado no Método dos Elementos Finitos. ALVES FILHO (2000) resume o que foi dito até aqui em apenas uma frase: “Se o engenheiro não sabe modelar o problema sem ter computador, ele não deve fazê-lo tendo o computador”.

Além disso destaca-se o fato que a maioria das decisões de engenharia devem ser tomadas pelo engenheiro ainda na etapa de concepção do modelo, antes de se realizar qualquer tipo de análise de resultados. Deve-se, por exemplo, definir um sistema de coordenadas adequado; selecionar elementos apropriados que representem os comportamentos físicos desejados além de se garantir a correta conectividade entre eles; definir o tamanho adequado da malha a ser utilizado na subdivisão da estrutura em elementos finitos; adotar carregamentos representativos e coerentes com o problema físico; e especificar condições de contorno compatíveis com o tipo de sistema estrutural que se está modelando.

BAPTISTA (1994), reforça essa afirmação em seu trabalho, destacando que se forem consideradas as quatro fases bem definidas de um projeto estrutural, tais como: concepção, análise estrutural, dimensionamento e detalhamento; talvez a que necessite mais atenção seja a de concepção. Salienta-se a sua importância devido ao fato de que uma estrutura má concebida pode comprometer seriamente todas as fases posteriores.

Seguindo em sua análise, BAPTISTA (1994) salienta as várias dificuldades que são encontradas na modelagem de pavimentos por um usuário típico de programas computacionais:

- Identificação do caminho realizado pelas cargas desde o ponto de aplicação até a fundação ou limite da estrutura.
- Conhecimento das limitações da aplicação da teoria e da preparação de um modelo satisfatório

- Correta representação dos principais componentes com os tipos de elementos disponíveis, como por exemplo, elementos tipo placa que simulam lajes e elementos barra que correspondem às vigas e pilares e suas ligações.
- Correta representação dos carregamentos e das condições de contorno.

Assim, sempre ao se iniciar a geração de um novo modelo as seguintes questões devem ser consideradas:

- Quais são os objetivos da análise?
- Há necessidade de se modelar o todo ou apenas uma parte do sistema físico?
- Que tipos de elementos adotar?
- Qual densidade para a subdivisão da malha é a adequada?
- Deseja-se priorizar o tempo de processamento ou a precisão dos resultados?

São as decisões a essas questões, tomadas ainda na fase da concepção, que indicarão o sucesso ou fracasso da análise a ser realizada.

Para auxiliar na obtenção das respostas dessas e outras questões, os itens a seguir trazem algumas informações e recomendações de caráter prático.

### **7.2.1 Definição dos Objetivos da Análise**

O primeiro passo do processo de análise consiste em estar plenamente ciente que não se deve confiar demais na capacidade dos programas baseados no Método dos Elementos Finitos, ao contrário, deve-se confiar na própria formação, experiência e julgamento profissional.

Somente o engenheiro pode determinar quais são os objetivos de sua análise, isto é, qual o tipo de resposta que o modelo investiga já que os objetivos estabelecidos no início do processo influenciarão as demais decisões para a geração do modelo.

Como destaca BAPTISTA (1994), nessa etapa as perguntas a seguir e outras quantas forem necessárias devem ser respondidas satisfatoriamente:

- Quais são os carregamentos?
- Quais são as condições de contorno?
- Quais ações são importantes e quais não são?
- O problema é estático ou dinâmico?
- Pode-se ou não adotar para o concreto comportamento do material no regime elástico?

- O material é isótropo?
- Existe alguma não-linearidade que necessite atenção?
- Deseja-se analisar esforços ou deslocamentos?

Salienta-se para o fato que antecipando-se os resultados também é possível definir os objetivos. Isso é verdadeiro pois se alguns comportamentos esperados forem previamente levantados, pode-se modelar regiões mais críticas de forma mais refinada; enquanto que se o objetivo da análise for somente o de se conhecer deformações sem a necessidade de analisar tensões, uma malha mais grosseira pode ser suficiente.

### **7.2.2 Definição da Estrutura a ser Modelada**

Ao se analisar um modelo pode-se aproveitar de sua forma simétrica, quando existir, para reduzir o tamanho do modelo a ser discretizado pelo Método dos Elementos Finitos. Essa metodologia traz uma série de benefícios uma vez que permite reduzir a entrada de dados, reduz o tempo de processamento e utilização de recursos do equipamento, além de permitir, caso desejado, a construção de um modelo mais refinado. Destaca-se que deve haver simetria não somente em relação à geometria, mas também em relação a carregamentos, vinculações e materiais.

Já no caso da necessidade de análises locais mais refinadas, pode-se ao invés de reanalisar a estrutura inteira, fazer uma análise separada apenas da porção da estrutura que contém a malha a ser refinada. Nesta situação ela deve ser carregada pela solicitação inicial e, ao longo do contorno, da onde a mesma foi separada do resto da estrutura, deve ser carregada pelos deslocamentos calculados nas análises precedentes do conjunto da estrutura.

### **7.2.3 Escolha do Tipo do Elemento**

Um modelo de elementos finitos pode ser categorizado como bi-dimensional ou tri-dimensional, contendo elementos lineares, elementos de área, ou elementos sólidos. Evidentemente pode-se associar diferentes tipos de elementos conforme a necessidade, mas deve-se tomar o cuidado em sempre se manter a compatibilidade entre os graus de liberdade nos pontos nodais dos elementos conectados.

O elemento estrutural é linear quando uma das dimensões, em geral o comprimento, é muito maior que as outras duas, altura e largura. Na formulação dos elementos finitos, tais elementos são denominados elementos tipo viga ou barra. Dependendo do software de análise que se utiliza, existem pequenas diferenças entre eles, o ANSYS, por exemplo, adota apenas a denominação viga, já

o NASTRAN adota barra e viga, sendo que a diferença neste caso é que o elemento barra possui seção transversal constante e o elemento viga permite a seção ser “*tapered*” (afilada) ou, em outras palavras, que existem diferentes seções transversais no início e fim do elemento. O programa desenvolvido nesta pesquisa utiliza a denominação barra e trabalha apenas com seção transversal constante. Este elemento é suportado pelo Sistema LASER e transmite forças axiais, cortantes e momentos fletores e torçores. O sistema LASER também oferece o elemento treliça, não utilizado no trabalho desenvolvido, cuja característica é de transmitir apenas forças axiais de tração e compressão.

O elemento estrutural é de área, ou de superfície, quando uma das dimensões, a altura ou espessura, é muito menor que as outras duas, o comprimento e a largura que, por sua vez, possuem aproximadamente a mesma ordem de grandeza. Dependendo do comportamento estrutural a que se propõem considerar e do software que se esteja utilizando, pode ser denominado chapa, placa, casca ou membrana. Aconselha-se checar a documentação do software em uso para se certificar das considerações adotadas.

Para efeito deste trabalho, considera-se as seguintes definições:

- Elemento Chapa e Membrana: Resistem a esforços aplicados apenas em seu plano médio (Estado plano de tensão)
- Elemento Placa: Resiste a esforços normais ao seu plano médio (Predominância de efeitos de flexão)
- Elemento Casca: Combina os comportamentos dos elementos Placa e Membrana.

O elemento estrutural é dito sólido quando as três dimensões, comprimento, largura e altura, possuem a mesma ordem de grandeza. Este tipo de elemento não é oferecido no programa desenvolvido neste trabalho.

Exemplos de aplicações:

- i) Elementos do tipo barra podem ser utilizados para representar vigas, barras de treliças ou pórticos, pilares e elementos de contraventamento. Esses elementos resistem a tração, compressão, torção e momento fletor.
- ii) Elementos do tipo chapa, podem por exemplo ser utilizados para representar painéis verticais como paredes de edifícios de alvenaria estrutural sujeitas às ações verticais
- iii) Elementos do tipo placa podem simular o comportamento de uma laje sujeita a ações verticais (normais ao plano da laje)
- iv) Para representar uma situação mais genérica, através da superposição de efeitos, pode-se compor num mesmo elemento o

comportamento de placa e chapa resultando no elemento casca. Essa situação foi discutida no tópico 4.3 e é a característica que possui o elemento de superfície existente no Sistema LASER e utilizado pelo programa desenvolvido neste trabalho.

- v) Elementos 3D sólidos podem ser utilizados para modelar blocos e sapatas de fundação.

Comentários adicionais:

- Para o elemento barra, vale destacar que se deve tomar cuidado ao representar vigas de paredes delgadas utilizando-se esse tipo de elemento. Isso ocorre pois do ponto de vista de flexão, as vigas sólidas e as vigas de paredes delgadas apresentam comportamentos semelhantes, mas quanto à torção, seus comportamentos são muito diferentes. As vigas de parede fina empenam ao sofrerem torção, podendo resultar no aparecimento de tensões normais na torção, fenômeno não coberto pela teoria elementar de vigas. Apesar de ser um tipo de componente pouco comum no caso de estruturas de concreto armado, aparecendo mais comumente em estruturas metálicas e estruturas de projetos mecânicos em geral, as vigas de parede delgadas devem ser modeladas por elementos de casca, que permitem representar o comportamento detalhado das chapas que constituem a alma e as abas dos perfis delgados.

- Para o elemento placa, a formulação é baseada na teoria de placas delgadas, ou seja, a espessura é bem inferior às duas outras dimensões. Assim, caso a espessura seja representativa com relação às demais dimensões, se faz necessário considerar os efeitos da deformação pelo esforço cortante. Para isso os softwares de elementos finitos oferecem elementos que consideram resistência à força cortante. Destaca-se que o Sistema LASER, que é utilizado pelo programa desenvolvido neste trabalho, não oferece elemento placa com resistência ao esforço cortante, sendo, portanto, sempre necessário discretizar a malha levando-se em consideração que a classificação de placa delgada deve ser satisfeita:

$$\frac{h}{d} < \frac{1}{5}$$

onde:

h: espessura

d: menor dimensão do plano médio da placa

- Destaca-se novamente a importância de se manter a consistência do número de graus de liberdade entre elementos conectados. Quando os elementos não são consistentes uns com os outros, a solução pode não transferir adequadamente forças ou momentos entre os elementos que são diferentes. Para serem consistentes, diferentes elementos que estejam conectados, devem possuir os mesmos tipos e números de graus de liberdade ("degrees of freedom" DOFs).



## 7.2.4 Definição da Densidade da Malha

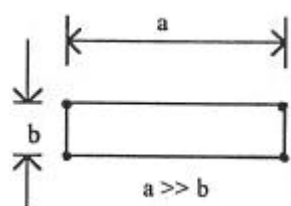
A questão que freqüentemente surge numa análise de elementos finitos é:

Qual a densidade da malha deve ser considerada de forma a obter resultados satisfatórios?.

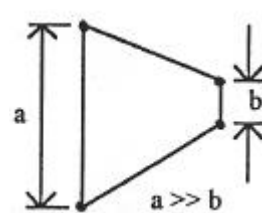
Infelizmente a resposta a essa pergunta é que ela deve ser resolvida pelo próprio engenheiro. Porém algumas técnicas e recomendações podem ajudar a obter um modelo mais representativo.

Antes de qualquer coisa, destaca-se que um elemento desempenha melhor a sua função quando a sua forma é compacta e regular. Um elemento tende a ficar rígido e perder precisão quando:

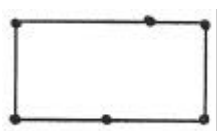
- a razão entre os seus lados aumenta;
- os ângulos dos cantos se tornam diferentes entre si;
- os lados se tornam curvos;
- os nós laterais (se presentes) tornam-se espaçosamente não uniformes



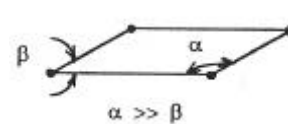
a) Grande relação entre os lados



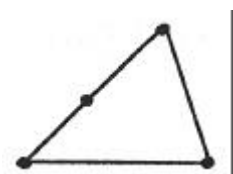
b) Quase triangular



c) Nós fora do centro



d) Grande inclinação



e) Quadrilátero triangular



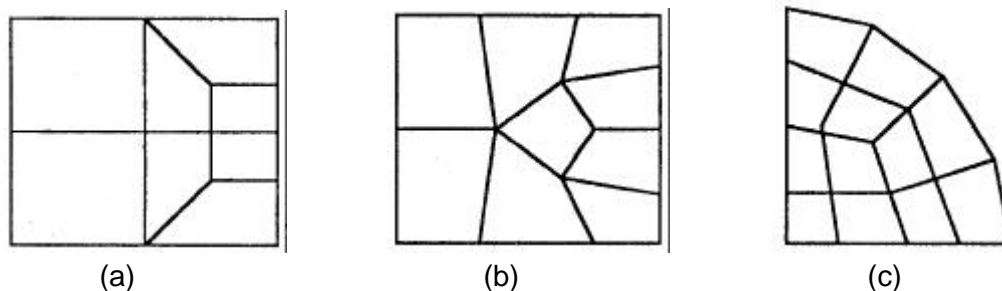
f) Lado curvo

Figura 7.1 – Algumas formas de elementos indesejáveis

As regras gerais recomendadas para a distribuição do elemento são vagas, podendo-se considerar o seguinte:

- guardar as relações entre os lados próximas da unidade
- ângulos dos cantos dos quadriláteros próximos de  $90^\circ$

- nós laterais no meio dos lados
- lados retos
- Para malhas não uniformes, distribuir os elementos de forma gradual sem produzir grandes discrepâncias no tamanho de elementos adjacentes



- (a → b) Transformação de uma malha inadequada para uma mais criteriosa onde não há alteração abrupta de formas
- (c) Possível malha de quadriláteros para representar um quadrante de círculo

Figura 7.2 – Distribuição de Malhas

Além disso, nos casos em que interessa a obtenção dos esforços, principalmente dos momentos negativos de placa sobre os pilares, deve-se refinar mais a distribuição da malha nos pontos onde sabidamente há maior gradiente de tensões. De forma semelhante pode-se adotar uma distribuição mais grosseira nas regiões onde o gradiente varia pouco ou é constante. Essas recomendações favorecem tanto na melhora da precisão dos resultados do modelo quanto na diminuição do tempo de processamento.

Caso não se tenha uma idéia inicial das regiões onde possa haver regiões críticas, sugere-se a partir dos resultados obtidos em uma primeira análise, modificar a malha refinando-se melhor essas regiões críticas. No entanto, se for constatado que apenas uma pequena região do modelo necessita uma melhor discretização, pode-se então analisá-la em separado através de uma análise local.

### 7.2.5 Demais Etapas do Processo de Análise

Para finalizar o processo de análise estrutural, deve-se ainda considerar os seguintes passos:

#### a) Definir os carregamentos e as condições de contorno.

Os carregamentos devem ser bem representados através dos mecanismos disponíveis para especificação dos mesmos nos programas de Elementos Finitos:

cargas pontuais para representar forças e momentos concentrados; cargas lineares para representar, por exemplo, o carregamento devido uma parede em uma laje; e cargas distribuídas por área, para representar pesos próprios e ações que possam ser absorvidos pela laje como um todo. A ação do vento (ação horizontal), pode também ser especificada como uma ação concentrada ao nível de cada pavimento ou distribuída por área, tendo seu valor determinado para cada nível do edifício.

Cuidado especial deve ser dado às vinculações da estrutura de forma a reproduzir corretamente as ligações vigas-pilares e pilares-fundações.

#### **b) Verificar o modelo em Elementos Finitos construído**

Essa é uma etapa importante pois após finalizada a construção do modelo deve-se, utilizando os recursos de pré-processador disponibilizados pelo programa de elementos finitos, checar todos as entradas realizadas: vinculações, carregamentos, propriedades dos elementos, materiais associados, conectividade entre os elementos, offsets, etc.

#### **c) Processar a análise pelo Método dos Elementos Finitos**

Estando tudo OK submete-se o modelo ao processamento. Nessa etapa eventuais problemas com o modelo podem impedir que o processamento seja concluído com êxito, por isso é extremamente importante que o modelo tenha sido verificado cuidadosamente no passo anterior.

#### **d) Verificar os resultados obtidos e avaliar a sua coerência com as expectativas de comportamento estabelecidas no início do processo de análise**

Antes de utilizar os resultados em qualquer tipo de dimensionamento, deve-se checar através dos recursos de pós-processador disponibilizados pelo programa de elementos finitos se os resultados estão compatíveis com o esperado.

#### **e) Caso a estrutura não atenda aos critérios estabelecidos, modificar o modelo proposto em função das alterações geométricas introduzidas na estrutura e reiniciar a análise.**

Esse processo é um ciclo que pode ser repetido quantas vezes for necessário até que se atinjam os objetivos especificados no início da análise.

### 7.3 Estudo da arquitetura e concepção da forma estrutural

Para iniciar a elaboração das formas estruturais de um edifício, faz-se necessário dispor das plantas arquitetônicas que representam tal estrutura. Vale destacar, como é o caso do exemplo tratado neste capítulo, que o trabalho pode ser extremamente facilitado caso já se disponham de tais plantas no formato eletrônico do AutoCAD. Com isso é possível gerar as linhas básicas da forma estrutural no próprio AutoCAD, e utilizar o sistema Mestre apenas para definir o que cada uma dessas linhas representam na etapa da geração da malha básica de elementos finitos, isto é, contorno externo, contornos internos e pontos e linhas nodais. Caso contrário, deve-se indicar o posicionamento do contorno externo, contornos internos e linhas nodais, utilizando os próprios recursos do sistema Mestre. Posteriormente, já dispondo da malha básica, são posicionadas as vigas, os pilares, as lajes, os carregamentos, etc.

Ainda antes de utilizar o sistema Mestre, de posse das plantas arquitetônicas, o engenheiro deve iniciar um estudo detalhado das diferentes plantas dos pavimentos, sub-solo, térreo, tipo, cobertura, etc, para definir o posicionamento dos pilares do edifício, seguido das vigas e lajes. Recomenda-se que a definição da forma da estrutura sempre seja iniciada com o posicionamento dos pilares, pois isso ajuda a evitar as interferências com os diferentes pavimentos que compõem o edifício, evitando, por exemplo, a adoção de pavimentos de transição. Esse procedimento garante também a adoção de distâncias econômicas para vigas, além de já definir os pórticos que irão resistir às ações horizontais devido à ação do vento. Na verdade uma vez definidos os pilares, os posicionamentos das vigas e das lajes já ficam praticamente definidos.

Vale lembrar, também, que o projeto completo de um edifício envolve outros como o de instalações-hidráulicas, elétricas, paisagismo, ar-condicionado, etc, e que tais projetos devem da mesma forma ser considerados ao se definir o projeto estrutural a fim de se eliminar qualquer tipo de interferência.

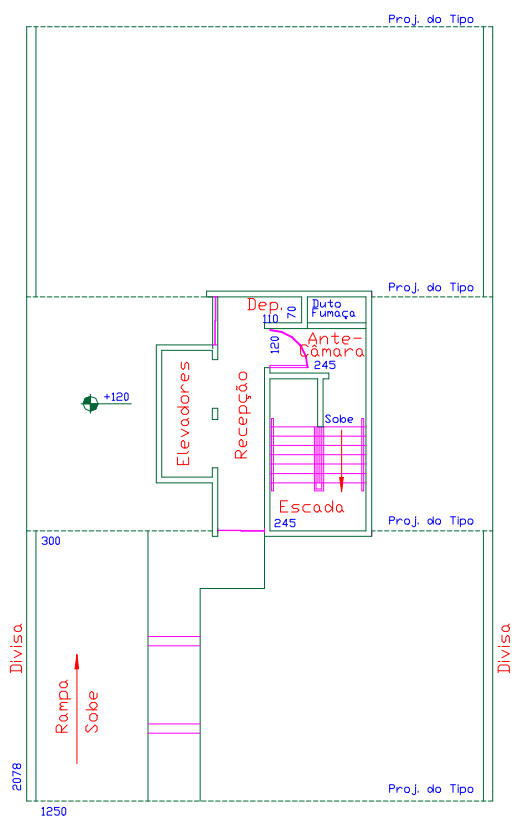


Figura 7.1 – Edifício Exemplo: Planta Baixa Têrreo

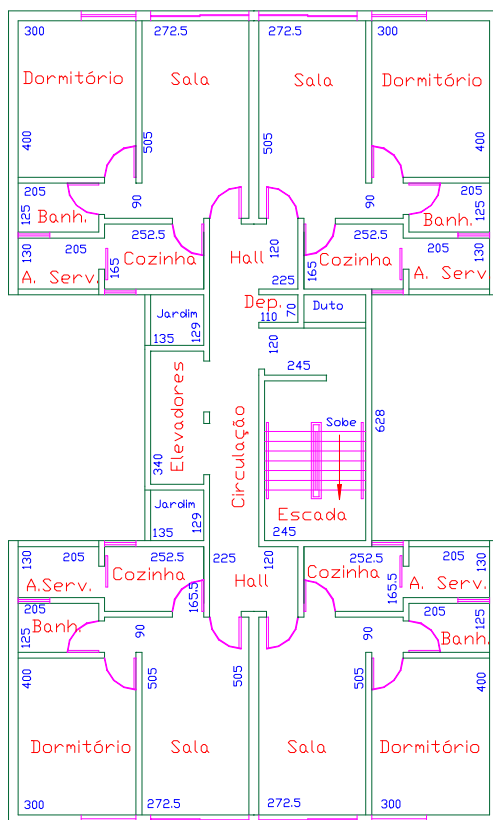


Figura 7.2 – Edifício Exemplo: Planta Baixa Pavimento Tipo (x10)

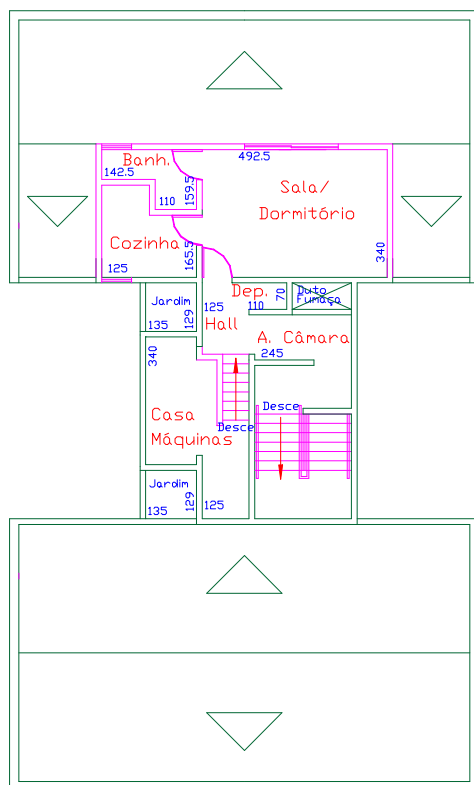


Figura 7.3 – Edifício Exemplo: Planta Baixa Ático

Para a definição da forma estrutural desse exemplo, procurou-se respeitar o projeto arquitetônico, eventuais passagens de tubulações e os critérios de economia durante o posicionamento dos componentes estruturais principais. Mas como o objetivo principal do projeto exemplo é referente à exemplificação da utilização do sistema Mestre, não houve uma preocupação em se realizar uma análise aprofundada com relação à adoção de dimensões econômicas para os componentes estruturais nessa fase de concepção e de pré-dimensionamento.

No entanto, o próximo item traz alguns critérios que podem ser considerados na fase de pré-dimensionamento visando uma abordagem econômica. Destaca-se ainda que tais critérios podem, futuramente, ser integrados ao sistema, de forma a auxiliar o engenheiro nessa fase de pré-dimensionamento.

#### 7.4 Sugestões para o pré-dimensionamento dos componentes

Segundo Giongo et. al. (1994) em uma coletânea com textos de vários autores, para uma abordagem econômica, as alturas das vigas não devem ultrapassar um treze avos ( $1/13$ ) e um onze avos ( $1/11$ ) do vão livre no caso de vãos internos e um onze avos ( $1/11$ ) a um nove avos ( $1/9$ ) no caso de vãos externos. Para as vigas contínuas com alturas diferentes, pode-se utilizar, como indicativo para as alturas das vigas, o valor de um décimo ( $1/10$ ) do vão livre.

Ainda segundo Giongo et. al. (1994), para cada viga deve-se prever área de influência de lajes com valores compreendidos entre  $3,5 \text{ m}^2$  a  $4,2 \text{ m}^2$  de laje por unidade de comprimento de viga. Com relação aos pilares, os mesmos devem ser posicionados sempre que possível a uma distância entre 4,5 m e 5,5 m, e a área de influência econômica deve resultar entre  $15 \text{ m}^2$  a  $20 \text{ m}^2$  de área de pavimento por pilar.

Todas esses critérios tem como objetivo resultar em taxas de armaduras para as vigas na ordem de 80Kg a 105 Kg por unidade de volume e entre 2% a 3% para os pilares, e com consumo de aço no primeiro pavimento, compreendido entre 130Kg a 220Kg por unidade de volume de concreto. No caso das lajes, as taxas de armaduras devem resultar entre 2,4Kg a 3,2Kg de aço por unidade de área e 28 a 53 quilos de aço por unidade de volume de laje. Os vãos considerados econômicos podem ser adotados em torno de 4m, resultando aproximadamente áreas de  $15 \text{ m}^2$  a  $20 \text{ m}^2$ .

Vale ressaltar que tais índices e critérios são meros indicativos para as estruturas convencionais de edifícios em concreto armado e podem sofrer consideráveis variações dependendo das particularidades de cada projeto.

Na seqüência, apresenta-se a utilização do sistema Mestre para a construção do modelo de elementos finitos e para a adição dos componentes estruturais e carregamentos.

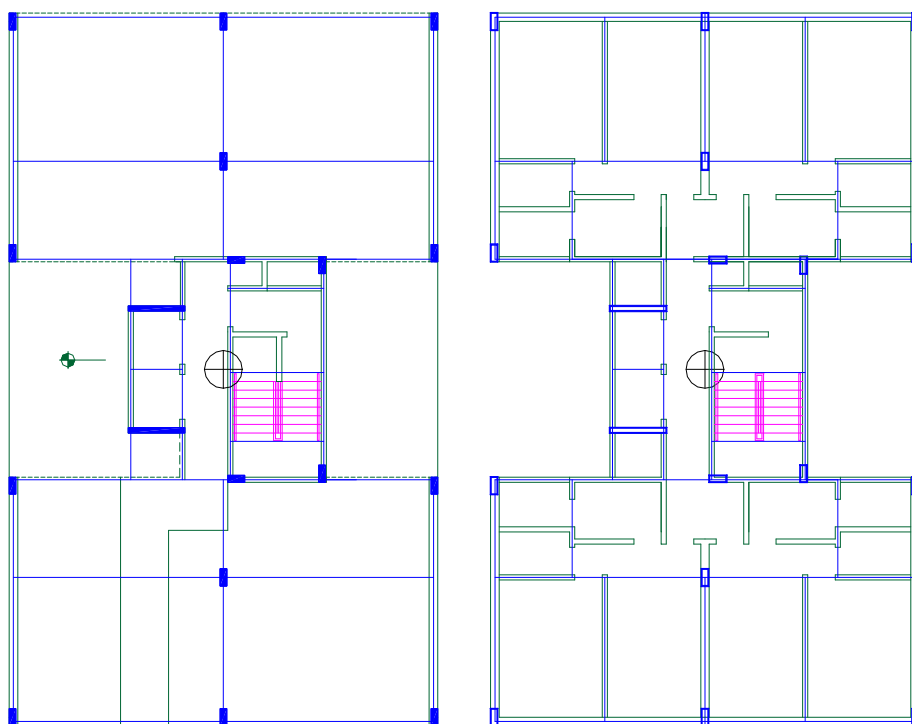


Figura 7.4 – Edifício Exemplo: Formas estruturais para os Pavimentos Térreo e Tipo

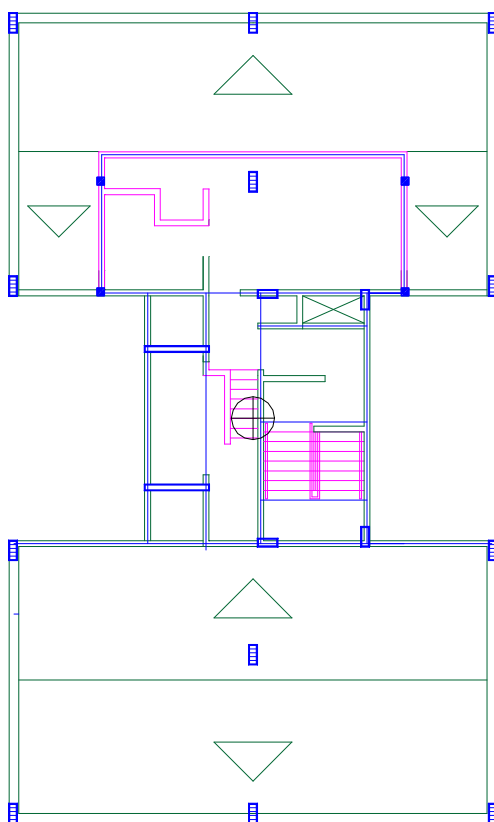


Figura 7.5 – Edifício Exemplo: Forma estrutural para o Pavimento Ático

## 7.5 Projeto Exemplo

Como mencionado no início do capítulo, esse projeto exemplo trata da modelagem dos pavimentos e a estrutura de contraventamento de um edifício residencial em concreto armado composto de 10 pavimentos tipo e o ático. Para o cálculo do vento, supôs-se que tal edifício será construído em terreno plano na região central de uma cidade de médio porte, sendo a velocidade básica de 40 m/s.

Uma vez realizado o estudo para a definição das formas estruturais, pode-se utilizar o sistema Mestre para a montagem dos modelos que representam os pavimentos que compõem o edifício.

Ressalta-se que para a posterior integração desses pavimentos no módulo de estruturas de contraventamento do sistema Mestre, é necessário que a origem do sistema de coordenadas para cada pavimento seja exatamente a mesma.

Nesse exemplo, adotou-se como ponto para origem do sistema de coordenadas o ponto identificado pelo símbolo  $\oplus$ .



Para facilitar o processo de montagem dos pavimentos no sistema Mestre, pode-se seguir o roteiro descrito a seguir:

1) Criar um diretório para o edifício e sub-diretórios para cada pavimento

- C:\Projetos\EdificioExemplo → Diretório do edifício
- ...\EdificioExemplo\Terreo
- ...\EdificioExemplo\Tipo
- ...\EdificioExemplo\Atico
- ...\EdificioExemplo\Contraventamento



Subdiretórios para cada pavimento e para a análise do vento

- 2) Caso se disponha das plantas arquitetônicas no AutoCAD, pode-se utilizá-los diretamente para a construção do modelo. Para isso, basta criar uma camada denominada GMA contendo as linhas que definem a forma e as posições dos elementos estruturais e carregamentos a serem gerados no sistema Mestre. Destaca-se que tal desenho deve ser exportado no formato DXF para que o mesmo possa ser lido pelo sistema Mestre.

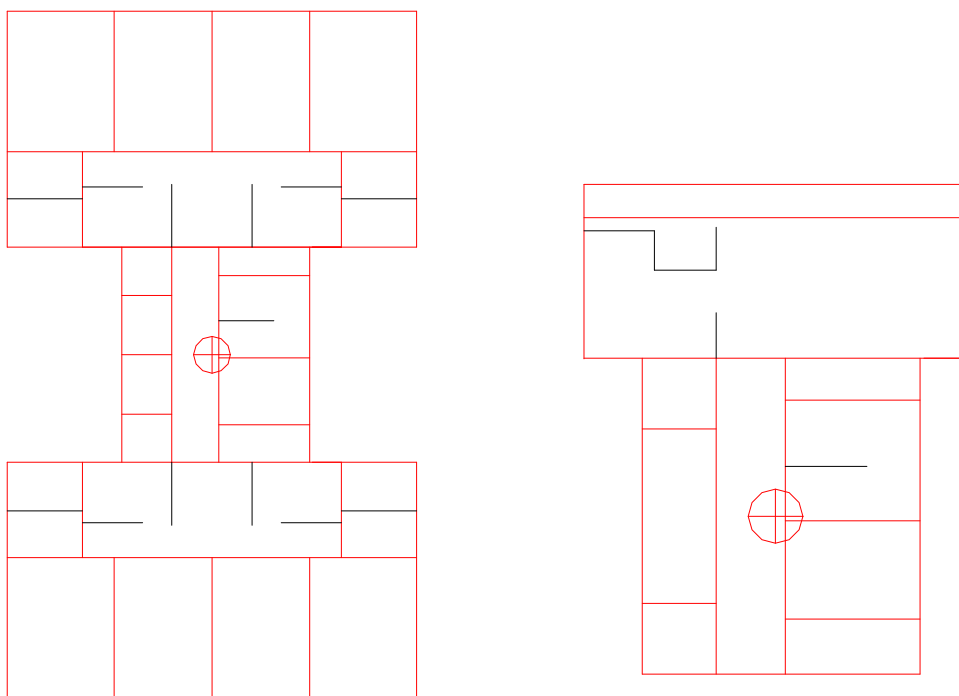


Figura 7.6 – Visualização das linhas criadas na camada GMA para os pavimentos tipo e ático. Os arquivos devem ser gerados no formato DXF do AutoCAD para serem lidos posteriormente pelo Sistema Mestre. O símbolo ⊕ indica a origem do sistema de coordenadas.

3) Etapas a completar para cada um dos pavimentos :

- a) Entrar no sistema Mestre e criar um novo modelo no subdiretório correspondente.
- b) Módulo 1 – Preparação do arquivo para o processamento GMA – Montagem da malha básica de elementos finitos
  - I) Importar o arquivo DXF com as linhas básicas do pavimento
  - II) Especificar no sistema Mestre o contorno externo, os contornos internos, linhas e pontos nodais que serão utilizados no processamento GMA para a construção da malha básica de elementos finitos (formada apenas pelos elementos placa).
  - III) Gerar o arquivo .GMA e executar o processamento GMA para a montagem da malha básica.
- c) Módulo 2 – Edição da malha básica e adição de vigas, pilares e carregamentos.
  - I) Importar o arquivo gerado pelo módulo anterior
  - II) Checar a malha básica gerada e editá-la se necessário.
  - III) Inserir os pilares, vigas com suas respectivas propriedades e materiais
  - IV) Inserir os carregamentos
  - V) Acertar as vinculações caso necessário
  - VI) Realizar o processamento GLS para gerar os arquivos para o processamento final a ser realizado pelo programa LS5H.EXE (Sistema LASER). O programa GLS gera um arquivo com os offsets para as vigas, trechos rígidos dos pilares, e com os devidos carregamentos associados.
  - VII) Processamento final → Programa LS5H para gerar os resultados de tensões e deslocamentos para a visualização de resultados.
- d) Módulo 4 – Visualização dos Resultados. Esse módulo permite visualizar, no caso dos pavimentos, as curvas de isovalor para deslocamentos, momentos, cortantes e tensões, através de recursos 2D e 3D.
- e) Módulo 3 – Estrutura de Contraventamento. Ao final de todas as análises e acertos nos modelos, parte-se para a montagem da estrutura de contraventamento onde se define como o edifício é constituído. Para isso cria-se o modelo que representa o edifício

no subdiretório Contraventamento utilizando os arquivos existentes nos demais subdiretórios que representam os pavimentos.

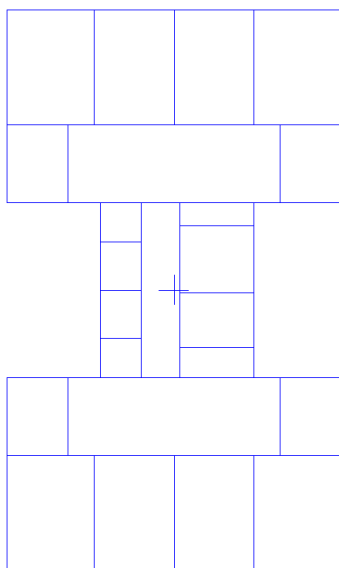
A seguir é apresentada uma série de ilustrações mostrando o resultado dos passos descritos neste tópico, bem como algumas informações adicionais. E, no final, faz-se uma comparação do processo do sistema Mestre com o processo de geração do pavimento tipo no software ANSYS, onde fica claro o maior esforço na geração do modelo quando não se dispõem de processos e ferramentas especialmente desenvolvidas para tratar da solução de pavimentos de estruturas de concreto armado.

## 7.5.1 Pavimento TIPO

### 1) Criando o Modelo

Nessa tela são fornecidas as informações gerais do modelo, tais como: nome do projeto, nome do cliente, unidades de trabalho, Fck e Fyk, módulo de elasticidade, peso específico e módulo de Poisson.

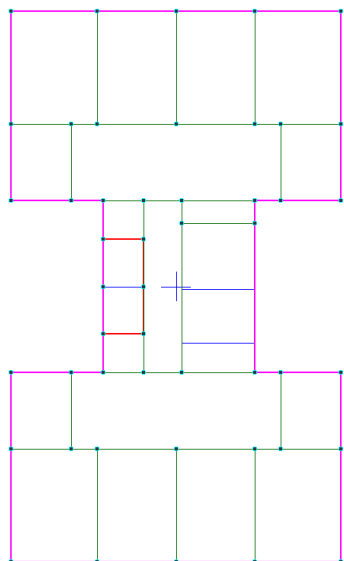
### 2) Carregando o arquivo DXF



Uma vez criado o modelo DXF com a camada GMA com as linhas de esqueleto do pavimento, pode-se importá-lo no sistema Mestre através do menu *Arquivos* opção *Importar*.

Após a importação, o Sistema Mestre permite obter diretamente com o mouse informações de coordenadas para o posicionamento das entidades geométricas que representam o contorno externo, contornos internos, segmentos e pontos nodais. Essas entidades geométricas são utilizadas no processamento GMA para a geração da malha básica de elementos finitos, isto é, a malha composta apenas pelos elementos placa.

### 3) Definindo o contorno externo, contornos internos e linhas e pontos nodais



Utilizando os recursos do sistema Mestre: Criar Contorno Externo, Contornos Internos, Segmentos e Pontos Nodais; foi possível gerar o resultado ilustrado na figura ao lado para o posterior processamento do programa GMA.

### 4) Gerando e processando o arquivo GMA

**Construção do Modelo Básico - GMA**  
 Informações Gerais para a geração da Malha

<b>Projeto</b>	Pav. Tipo	<i>Distância de Captura para pontos em torno dos vértices - dcapv</i>	0,00
<b>Espessura dos elementos - ESP</b>	0.1	<i>Distância de captura para pontos em torno dos lados - dcapl</i>	0,00
<b>Carga Normal ao plano dos elementos - CAR</b>	-1.5	<i>Porcentagem a ser desprezada no início e fim de cada diagonal - ddiag</i>	0,00
<b>Espaçamento do reticulado Básico - DELTA</b>	1	<i>Ângulo mínimo em graus para se considerar um triângulo como bem formado - angl</i>	0,00

*Unidade de Comprimento : m*  
*Unidade de Força : KN*

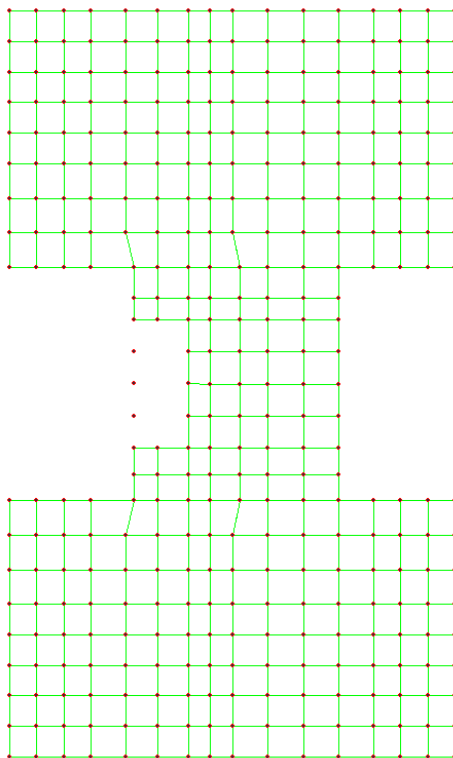
Gera Arquivo      Cancela

Adicionalmente, antes da execução do processamento GMA, são solicitadas algumas informações adicionais, tais como: espessura desejada para os elementos placa (espessuras das lajes), carregamento vertical aplicado e espaçamento desejado para o reticulado básico (dimensão dos elementos). Opcionalmente é possível especificar outros parâmetros para controlar a geração da malha, onde o valor 0.00 (recomendado) é o padrão.

## 5) Passando para o Módulo 2

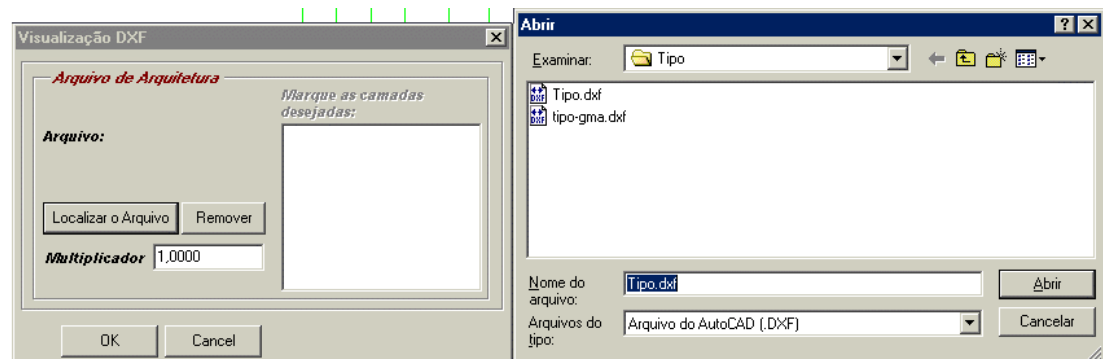


## 6) Carregando o arquivo com a malha básica gerada pelo Módulo 1



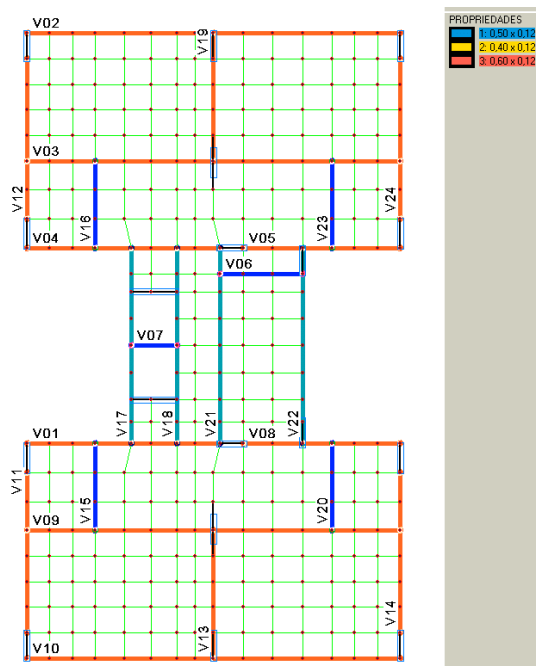
Após o processamento GMA, deve-se importar a malha básica gerada para realizar os ajustes necessários e incluir os componentes estruturais pilares e vigas, bem como os demais carregamentos. Destaca-se que todos os parâmetros da malha gerada podem ser editados pelo módulo 2, desde coordenadas e vinculações de nós, incidência de elementos, até espessura e carregamentos associados aos elementos.

## 7) Carregando o arquivo DXF com a planta de arquitetura





### 9) Definindo e renumerando as vigas



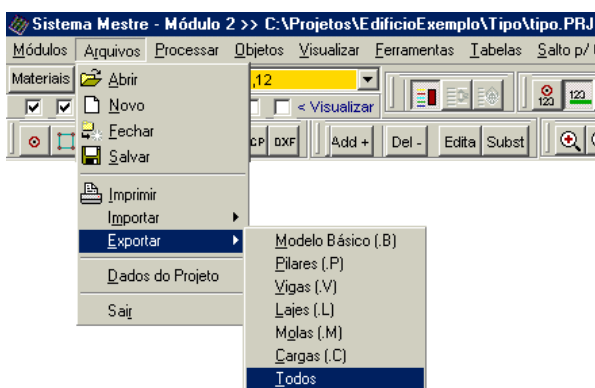
Assim como para os pilares, as vigas são geradas com o botão adicionar estando o modo Viga ativo. Neste caso, o sistema também disponibiliza uma interface especializada que conduz o usuário no processo de geração da viga. Destaca-se que, para o caso das vigas, o Nó K é definido automaticamente, isto é, o eixo y do sistema local é orientado na direção Z do sistema global.

Uma observação importante é com relação à especificação de vigas contínuas. Mesmo havendo pilares de canto na direção da viga formados por 2 nós, pode-se definir a incidência

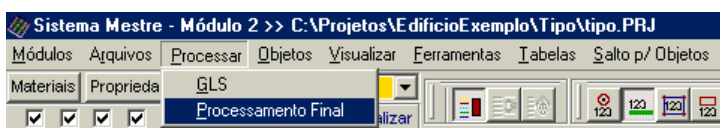
da viga a partir do nó de canto fazendo a viga atravessar o pilar. Isso é possível pois o sistema gerencia automaticamente estas situações e realiza os acertos necessários com relação aos vãos teóricos.

Da mesma forma que ocorre com os pilares, a qualquer momento, durante as adições das vigas, pode-se executar o comando renumerar para refazer a numeração das mesmas.

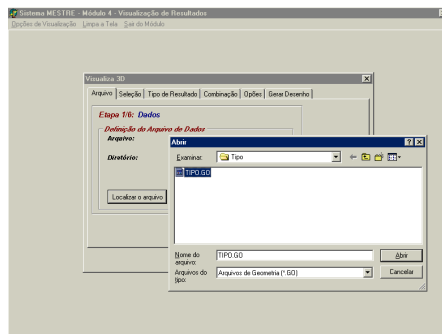
### 10) Executando o processamento GLS e o processamento final LS5H (LASER)



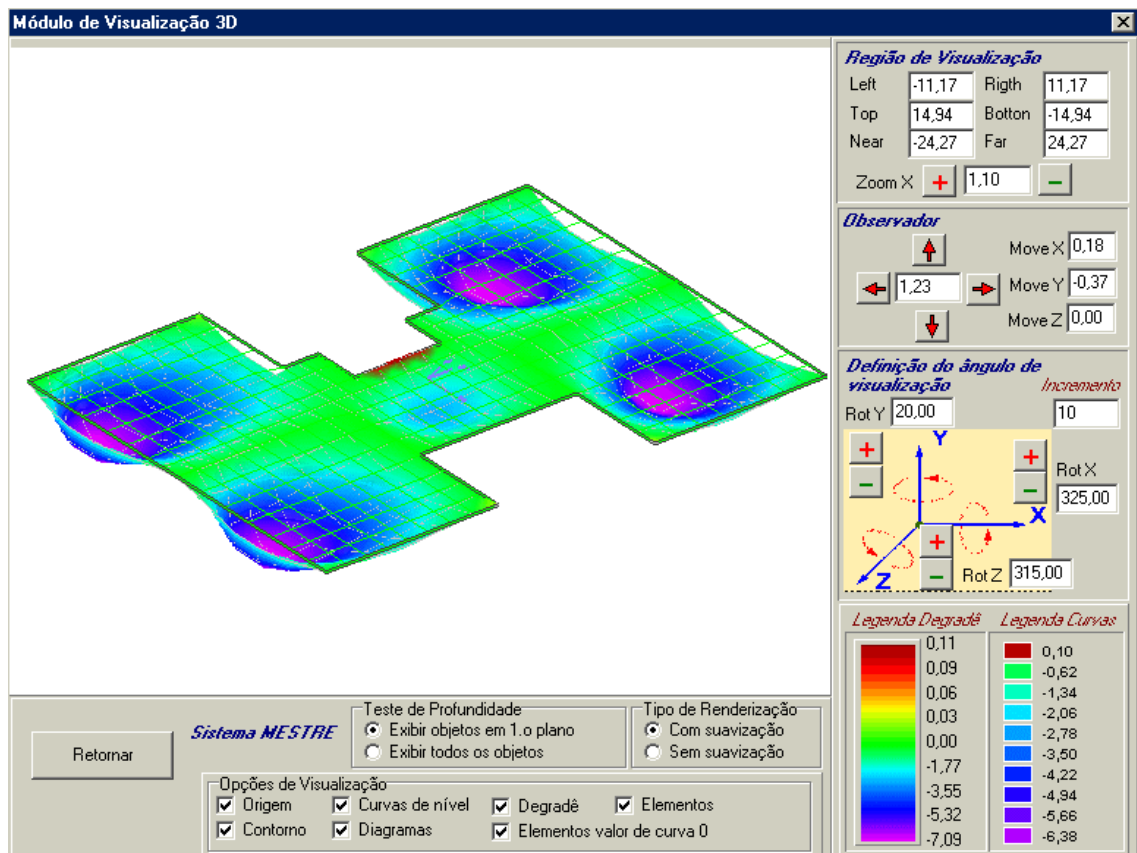
Após a finalização do modelo, com os componentes viga, pilar e carregamentos adicionais quando houver necessidade, basta realizar o processamento GLS e na seqüência o processamento final (LS5H). Destaca-se que antes de realizar o processamento GLS, é necessário gerar os arquivos com as informações do modelo através da geração dos arquivos: .B, .P, .V, .M e .C.



## 11) Visualizando os resultados no Módulo 4



Uma vez finalizado o processamento final, pode-se utilizar o módulo 4, Visualiza 3D, para a visualização dos resultados gerados referentes a tensões, momentos, cortantes e deslocamentos para os elementos do modelo. Tal módulo permite definir desde os elementos a serem considerados na visualização, até a combinação dos casos de carregamento com a aplicação de multiplicadores específicos para cada caso.

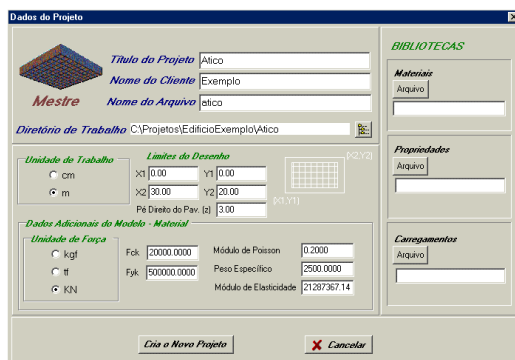




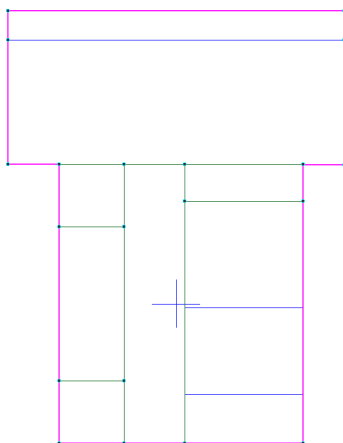
## 7.5.2 Pavimento Ático

Como a seqüência para geração do modelo do Ático é a mesma para o pavimento tipo, apresentam-se a seguir apenas as figuras ilustrando os resultados das etapas realizadas até a definição dos pilares e das vigas.

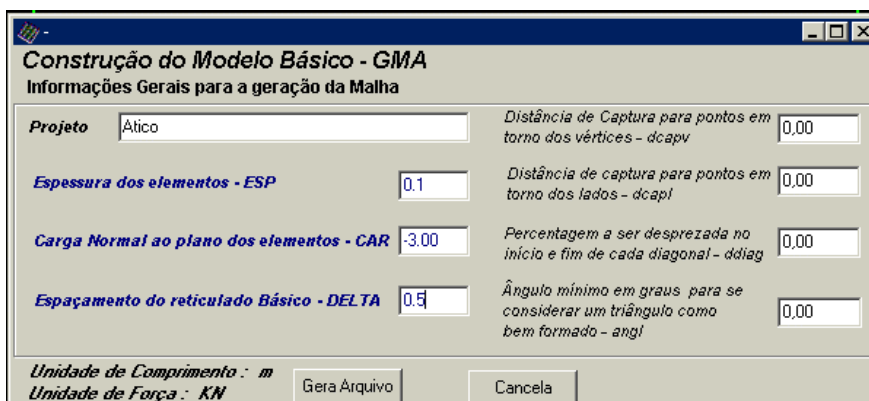
### 1) Criando o Modelo



### 2) Carregando o arquivo DXF e definindo o contorno externo, contornos internos e linhas e pontos nodais.



### 3) Gerando e processando o arquivo GMA





### 7.5.3 Estrutura de Contraventamento

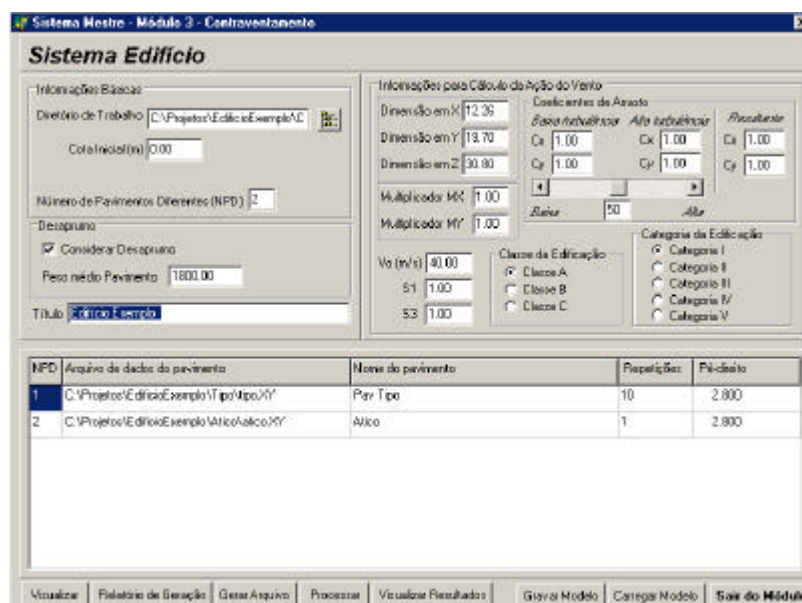
Ao finalizar a montagem e processamento dos pavimentos citados anteriormente, pode-se acessar o módulo 3 para definir a estrutura de contraventamento.



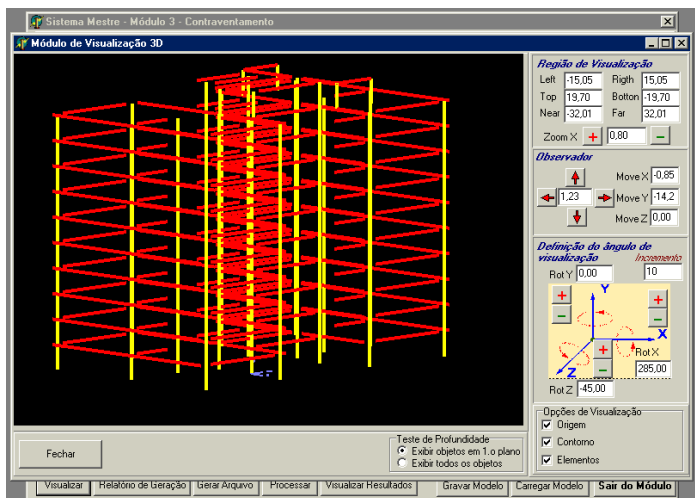
Etapas a serem realizadas:

- 1) Definir os dados para o cálculo do vento do edifício e carregar os arquivos com as informações de cada pavimento.
  - Definição da cota inicial para consideração do efeito do vento,
  - Número de pavimentos diferentes,
  - Consideração ou não dos efeitos de desaprumo
  - Dimensões da edificação segundo as direções x,y e z,
  - Coeficiente de arrasto (combinação dos efeitos de baixa e alta turbulência),
  - Velocidade básica e fatores topográfico S1, de rugosidade S2 e estatístico S3,
  - Selecionar os pavimentos e definir o número de repetições a ser aplicado em cada pavimento,

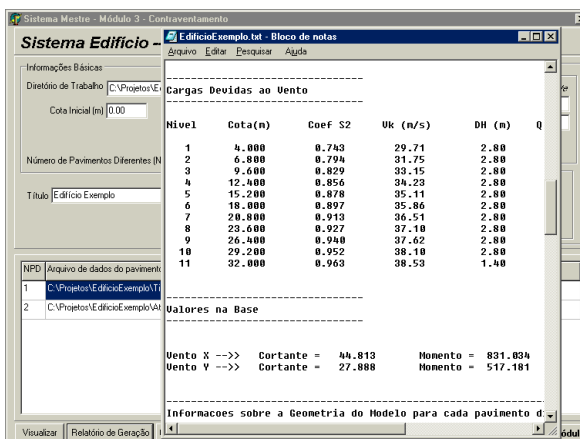
OBS: A ordem de entrada dos pavimentos define a seqüência da montagem da estrutura do edifício.



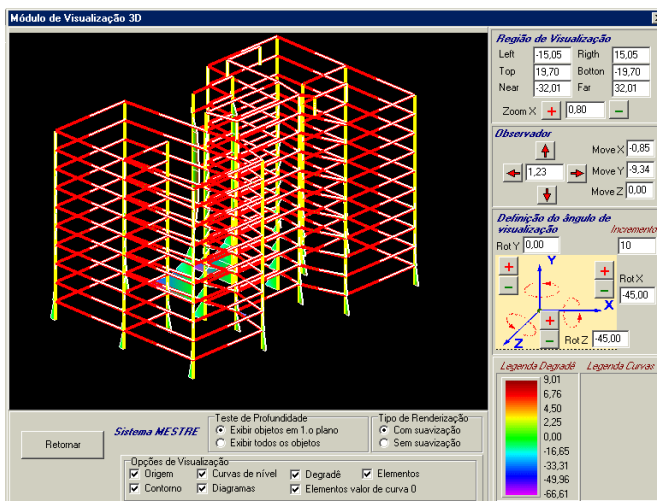
2) Visualizando a estrutura gerada → botão Visualizar



3) Visualizando o relatório de geração → botão Relatório de Geração



4) Visualizando os resultados pelo módulo de visualização Botões Processar e Visualizar Resultados



## 7.6 Modelagem do Pavimento Tipo através do ANSYS

Como forma de comparar o esforço necessário para a geração de um pavimento de um edifício quando não se dispõem de ferramentas especialmente criadas para esse fim, modelou-se o mesmo pavimento tipo através do ANSYS.

Dispondo dos dados de entrada utilizados para gerar o modelo no sistema MESTRE, tais como: geometria do pavimento tipo no formato eletrônico do AutoCAD, dados de materiais, propriedades de vigas, espessura de laje, carregamentos, e vinculações, procurou-se modelar o pavimento de forma parecida do que foi feito no sistema MESTRE.

Como simplificação, decidiu-se não entrar com os tramos dos pilares, assim os nós que estavam conectados aos pilares tiveram seus graus de liberdade totalmente restringidos.

As tabelas a seguir apresentam alguns dos dados de entrada utilizados:

<b>Materiais</b>	<b>Propriedades Físicas (unidades em KN e cm)</b>			
	Módulo Elasticidade.	Módulo de Cisalhamento.	Poisson	Peso específico
Concreto C20	4760	1983,3	0,2	2,50E-5

<b>Propriedades Geométricas</b>	<b>Dimensões (cm)</b>		<b>Propriedades</b>			<b>Offsets</b>	
	h	w	A	I <sub>yy</sub>	I <sub>zz</sub>	DY1	DZ1
Viga 60x12 (Dz neg)	60	12	720,0	216000,0	8640,0	0,0	-30,0
Viga 50x12 (Dz neg)	50	12	600,0	125000,0	7200,0	0,0	-25,0
Viga 40x12 (Dz neg)	40	12	480,0	64000,0	5760,0	0,0	-20,0
Lajes	10	-	-	-	-	-	-

<b>Carregamentos</b>	<b>Carga Distribuída</b>
	(KN/cm <sup>2</sup> )
Sobrecarga Utilização.	-1.50E-4
Peso próprio Laje	-2.50E-4
Total	-4.00E-4

Tabela 7.1 – Dados de entrada utilizados no pavimento TIPO

A primeira dificuldade enfrentada é que o ANSYS utilizado não disponha de filtro para leitura de arquivos no formato DXF, ele trabalha com formatos de outros programas de CAD e o formato neutro IGES, que no entanto a versão do AutoCAD utilizada desconhece. Assim foi necessário fazer um levantamento dos pontos que definem o contorno do pavimento, além dos pontos nas interseções de vigas e dos localizados sobre pilares. Esse levantamento deu origem à tabela 7.2

Ponto	X	Y	Z	Ponto	X	Y	Z
1	-612,50	-1026,50	0,00	29	0,00	606,50	0,00
2	-612,50	-606,50	0,00	30	0,00	1026,50	0,00
3	-612,50	-321,50	0,00	31	20,00	-321,50	0,00
4	-612,50	321,50	0,00	32	20,00	-177,50	0,00
5	-612,50	606,50	0,00	33	20,00	0,00	0,00
6	-612,50	1026,50	0,00	34	20,00	101,50	0,00
7	-387,50	-606,50	0,00	35	20,00	177,50	0,00
8	-387,50	-321,50	0,00	36	20,00	321,50	0,00
9	-387,50	321,50	0,00	37	292,50	-1026,50	0,00
10	-387,50	606,50	0,00	38	292,50	-606,50	0,00
11	-292,50	-1026,50	0,00	39	292,50	-321,50	0,00
12	-292,50	-606,50	0,00	40	292,50	-177,50	0,00
13	-292,50	606,50	0,00	41	292,50	0,00	0,00
14	-292,50	1026,50	0,00	42	292,50	101,50	0,00
15	-270,00	-321,50	0,00	43	292,50	177,50	0,00
16	-270,00	-177,50	0,00	44	292,50	321,50	0,00
17	-270,00	0,00	0,00	45	292,50	606,50	0,00
18	-270,00	177,50	0,00	46	292,50	1026,50	0,00
19	-270,00	321,50	0,00	47	387,50	-606,50	0,00
20	-195,00	-177,50	0,00	48	387,50	-321,50	0,00
21	-195,00	177,50	0,00	49	387,50	321,50	0,00
22	-120,00	-321,50	0,00	50	387,50	606,50	0,00
23	-120,00	-177,50	0,00	51	612,50	-1026,50	0,00
24	-120,00	0,00	0,00	52	612,50	-606,50	0,00
25	-120,00	177,50	0,00	53	612,50	-321,50	0,00
26	-120,00	321,50	0,00	54	612,50	321,50	0,00
27	0,00	-1026,50	0,00	55	612,50	606,50	0,00
28	0,00	-606,50	0,00	56	612,50	1026,50	0,00

Tabela 7.2 – Tabela com as coordenadas dos principais pontos do pavimento TIPO

No ANSYS, com os dados da tabela 7.2, foram criados os pontos e as linhas que posteriormente deram origem aos elementos lineares tipo BEAM, representando as vigas. Essas linhas também serviram para delimitar as áreas que deram origem aos elementos SHELL representando as lajes.

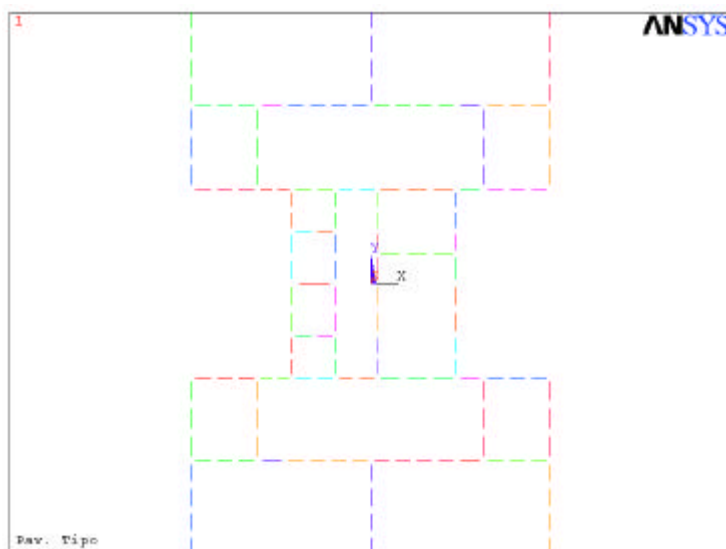


Figura 7.7 – Visualização das linhas básicas criadas no modelo do ANSYS

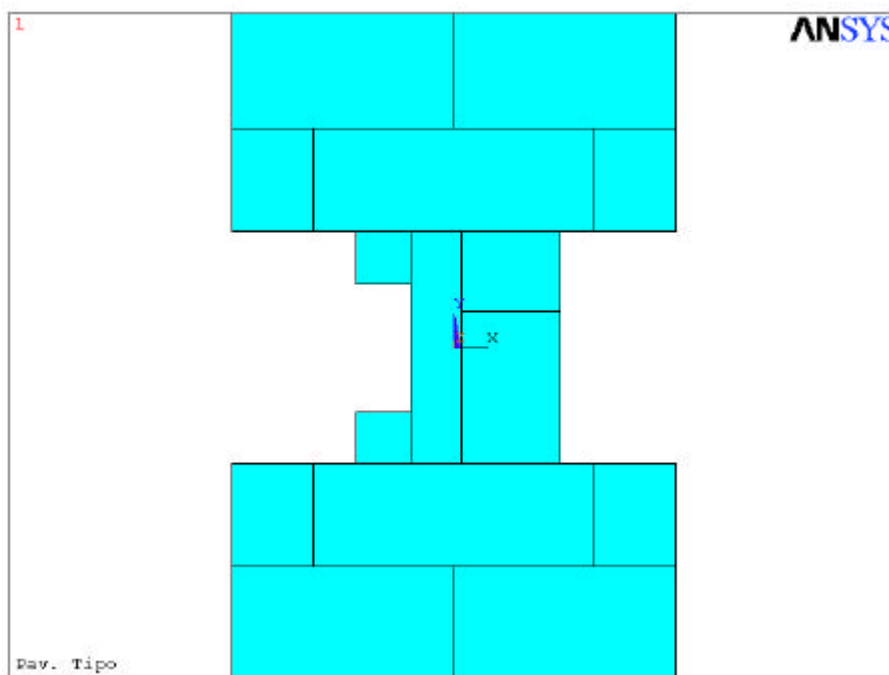


Figura 7.8 – Visualização das áreas criadas no modelo do ANSYS a partir das linhas básicas

Após a finalização da definição da geometria, iniciou-se especificação dos dados do material (no caso isótropo), e os tipos de elementos utilizados para a análise. Foram utilizados apenas dois tipos de elementos: o BEAM 44 para modelar as vigas e o SHELL 63 para modelar as lajes, sendo que ambos os elementos possuem seis graus de liberdade por nó, as três translações e as três rotações.

Decidiu-se pelo BEAM 44 pois o mesmo permite especificar excentricidades “Offsets” nas extremidades do elemento, assim é possível representar a situação em que o eixo da viga fica abaixo do plano médio da laje, simulando a situação em que o contato entre a viga e a laje ocorre na face superior da viga.

Uma operação que mereceu cuidado foi ao se criar as propriedades para representar as vigas ou mesmo pilares, pois houve necessidade de se prestar atenção na orientação a ser adotada na geração das barras para não inverter as seções e considerar inércias trocadas,. No sistema MESTRE esse problema é minimizado pois as vigas sempre são orientadas da mesma maneira e os pilares são orientados no posicionamento através da definição do Nó K. O ANSYS também permite a definição do terceiro nó, ou Nó K, mas sua definição é mais trabalhosa.

Finalizado a definição dos elementos, materiais e propriedades, passou-se para a discretização propriamente dita. Talvez essa tenha sido a etapa mais trabalhosa de todo o processo.

Para criar uma malha parecida com a que foi gerada pelo sistema Mestre, onde se especificou apenas o contorno externo, contornos internos e linhas nodais, houve a necessidade de associar o número de divisões desejáveis para cada linha básica. Como as áreas foram associadas a essas linhas, elas acabaram sendo discretizadas, automaticamente, segundo o número de divisões definido para os seus lados. No entanto, antes de realizar a discretização, houve a necessidade de se utilizar o comando GLUE para associar os diversos objetos linhas uns com os outros, assim como os objetos áreas. Essa associação foi fundamental, pois garantiu uma discretização compatível entre as diferentes áreas, além de não duplicar nós localizados nas mesmas coordenadas.

O resultado da discretização pode é ilustrado na figura a seguir:

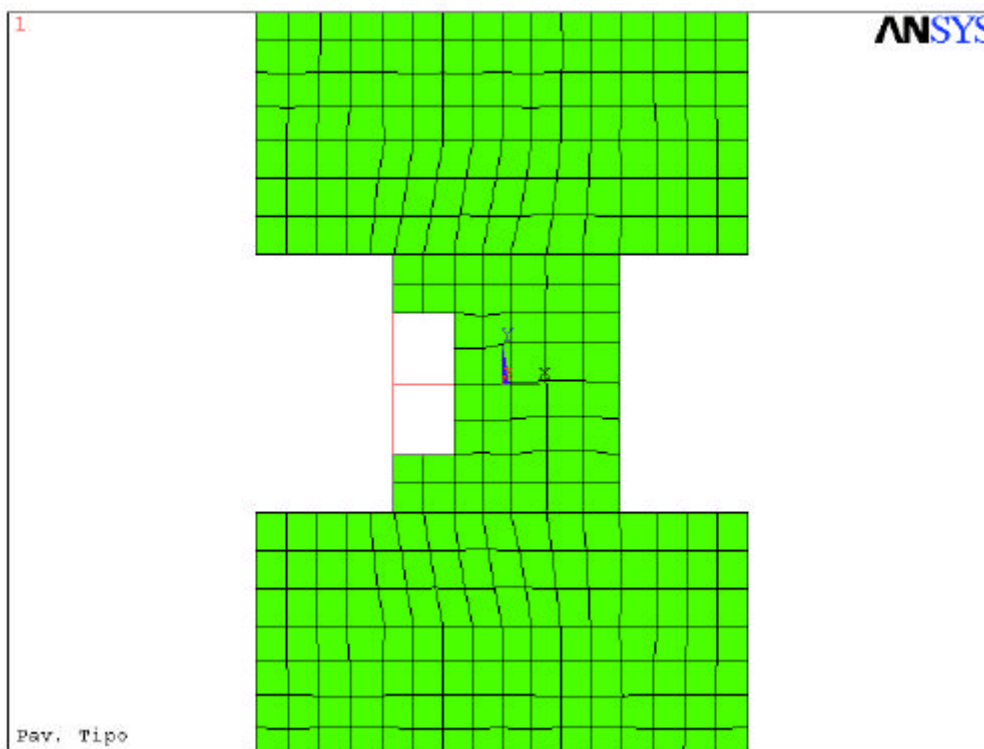


Figura 7.9 – Visualização da malha de elementos finitos gerada no ANSYS

Finalizada a malha, passou-se para a definição das vinculações, processo esse totalmente automático no sistema MESTRE, e a aplicação dos carregamentos. Para realizar essa última operação, aproveitaram-se os objetos áreas nos quais foram aplicados um carregamento distribuído com intensidade de  $-4,00E-4$  KN/cm<sup>2</sup> na face 2 (face superior do elemento).

Após finalizado essas etapas partiu-se para o processamento e, assim como exibido para o sistema MESTRE, plotou-se a figura com os deslocamentos nodais na direção Z.



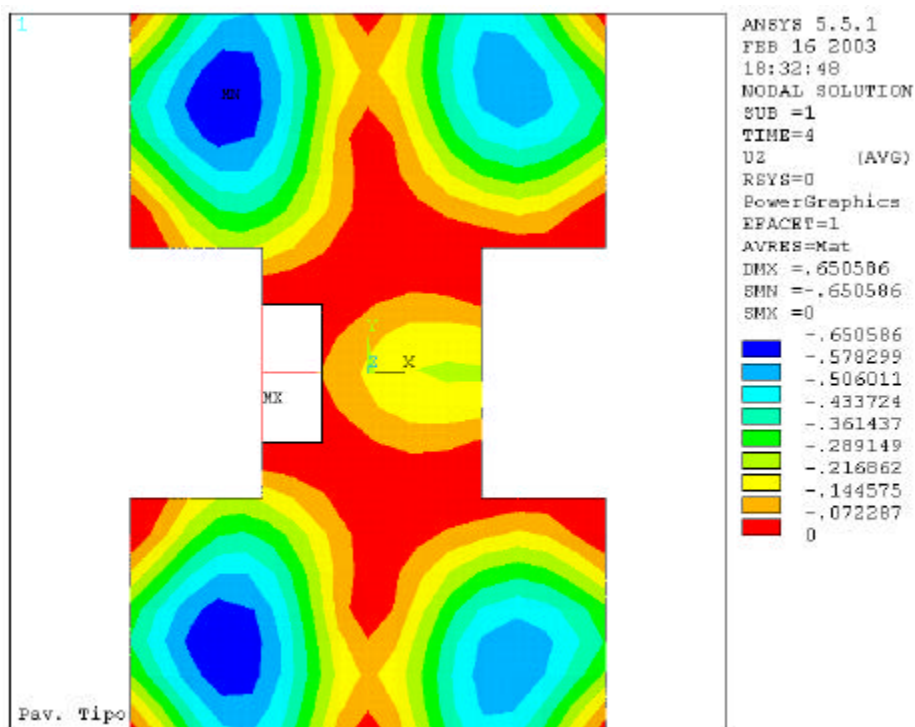


Figura 7.10 – Visualização dos deslocamentos nodais para a translação na direção z do modelo gerado no ANSYS

## 7.7 Comentários Finais

Como pôde ser visto através do projeto exemplo e da modelagem do pavimento tipo através do ANSYS, fica claro como uma ferramenta computacional especialmente desenvolvida para tratar de estruturas de edifícios de concreto armado, traz inúmeras vantagens com relação à redução do ciclo de desenvolvimento, melhoria da qualidade, aumento da confiabilidade e, conseqüentemente redução de custo.

A redução do ciclo de desenvolvimento é direta, uma vez que o sistema permite chegar a um resultado satisfatório com menor esforço e tempo. Já a melhoria da qualidade e o aumento da confiabilidade, são decorrências naturais dessa última, visto que a facilidade e a rapidez em gerar modelos reduz a ocorrência de erros e permite a realização de diversas análises em busca da melhor solução.

Com isso reforça-se a importância para o desenvolvimento de aplicativos desse tipo, destacando-se, principalmente, o fato de poderem ser utilizados como base para o desenvolvimento de pesquisas mais específicas, onde o grande “gargalo” se concentra na entrada de dados e visualização de resultados.

## 8. RECOMENDAÇÕES E CONCLUSÕES

---

Mesmo tendo-se obtido resultados satisfatórios com o desenvolvimento deste sistema, acredita-se que o mesmo pode ainda ser melhorado para incorporar recursos visando orientar, facilitar e otimizar o trabalho do engenheiro. Cita-se por exemplo, como foi visto no capítulo do projeto exemplo, a possibilidade de integração ao sistema de critérios de economia para o posicionamento e dimensionamento dos elementos estruturais num estágio de pré-dimensionamento. Com isso o sistema poderia interagir com o usuário indicando, com base em tais critérios, quais seriam as dimensões mais adequadas a serem adotadas na fase de pré-dimensionamento.

Poderia também aproveitar mais informações da planta de arquitetura existente no AutoCAD, através de novos parâmetros passados na camada GMA. Como exemplo cita-se a possibilidade do sistema ler diretamente os carregamentos associados às linhas de paredes através da adoção de um padrão de cores, onde cada cor representaria um valor de carga linearmente distribuída. São implementações que requerem um tempo adicional de desenvolvimento, mas perfeitamente factíveis.

Outro exemplo seria de se adicionar um módulo de banco de dados para cadastro e gerenciamento de materiais, propriedades e, principalmente, de carregamentos. Esse módulo também seria responsável por todas as operações de conversões de unidades permitindo ao engenheiro trabalhar de forma mais organizada e segura. Boa base para esse módulo inclusive já foi implementada.

Pode-se ainda transformar esse sistema num banco de dados com a história de todos os projetos desenvolvidos, isto é, o próprio sistema gerenciaria o acesso a todos os projetos já desenvolvidos permitindo, inclusive, obter índices estruturais para efeito de comparação ou até mesmo para auxiliar na elaboração de novos projetos. Seria uma forma de permitir um acesso rápido e permanente ao “histórico” do engenheiro.

Outra frente que poderia ser melhor explorada, refere-se a geração e leitura dos arquivos de dados para outros sistemas baseados no método dos elementos finitos. Atualmente o sistema Mestre possui unidades específicas para essas operações de entradas e saídas, só que qualquer novo formato a ser gerado ou lido, necessita nova codificação. Como forma de melhoria, poder-se-ia utilizar saídas padronizadas aceitas por diversos sistemas ou criar interpretador interno para permitir o usuário definir externamente o formato dos arquivos.

Um melhoramento interessante poderia também ser realizado no módulo de visualização 3D para a visualização de elementos barras, uma vez que atualmente o sistema não dispõe de recursos para relacionar os elementos barras aos componentes pilar e viga que, com isso, não permite visualizar os resultados de uma viga ou de um pilar em específico.

Por fim salienta-se o poder da implementação em OpenGL, que pode ser melhor explorada, e da lacuna deixada para a geração de modelos 3D no formato VRML, que podem ser facilmente visualizados em qualquer parte do mundo.

No entanto, apesar do aplicativo desenvolvido ainda não contar com todos esses recursos, acredita-se que o desenvolvimento do mesmo contribuiu para uma boa automatização do processo de montagem, processamento e análise de estruturas pelo Método dos Elementos Finitos.

Como pôde ser visto através do projeto exemplo, o aplicativo atende de forma satisfatória no que se refere a ferramentas de modelagem e visualização de resultados para modelos de elementos finitos. Desde a construção da malha básica até a inclusão ou edição de componentes estruturais no modelo, o usuário é sempre bem assistido, podendo intervir na construção do modelo de forma a representar o comportamento estrutural da forma mais correta possível. Facilidades como visualização de plantas arquitetônicas, consideração de recursos especiais como o nó mestre e os trechos rígidos, e geração e processamento dos modelos de forma integrada num único ambiente, oferece potencialidade, agilidade e confiabilidade na manipulação das informações.

Além disso o módulo para montagem da estrutura de contraventamento oferece facilidades sem precedentes, já que, com poucas informações de entrada, gera toda a estrutura, incluindo os carregamentos devidos à ação do vento e do desaprumo, processa o modelo e ainda permite a visualização dos resultados, tudo em questão de minutos.

Do comparativo entre o Sistema Mestre e o ANSYS, ficou claro o enorme ganho de produtividade quando um sistema é especialmente desenvolvido para atender a determinadas necessidades. E não se pode deixar de mencionar que, nos dias atuais, com um mercado tão competitivo, qualquer ganho desta natureza se reverte em grandes vantagens.

Talvez esse tenha sido o primeiro passo de muitos outros na busca de um sistema que consiga reunir todas as fases que envolvem um projeto de estruturas de edifícios, indo desde a planta de arquitetura até os desenhos finais de detalhamento dos componentes estruturais.

Além disso, com a experiência adquirida neste trabalho, percebe-se que com os recursos computacionais disponíveis atualmente é possível avançar muito mais no desenvolvimento de ferramentas para facilitar o trabalho do engenheiro. Com uma arquitetura visando a integração com outros sistemas já desenvolvidos, a lista de diferentes problemas que podem ser tratados e automatizados torna-se bastante extensa.

Acredita-se também que as ferramentas desenvolvidas neste trabalho podem auxiliar em muito o desenvolvimento de outros trabalhos mais específicos, que necessitem de ferramentas de geração e visualização de dados para análise por elementos finitos. Foi inclusive com esse intuito que optou-se por fazer módulos separados para a geração e visualização de dados.

Diante de tudo isso se tem a convicção que grande parte dos objetivos desse trabalho foram alcançados e que o mesmo pode ainda alavancar a concretização de muitos outros.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- ALGOR SUPERDRAW SYSTEM (1987). *Reference manual*. Algor Interactive System, Inc.
- ALMEIDA, F. J. (1992). *Projeto e implementação de pré/pós processador gráfico para exibir tensões calculadas pelo método dos elementos finitos*. São Carlos, Dissertação (Mestrado), EESC – USP.
- ALVES FILHO A. (2000). *Elementos Finitos a Base da Tecnologia CAE* - São Paulo, Érica.
- ANSYS – 386/ED – Revision 4.4 – Reference Manual, (1990). Swanson Analysis System, Inc., aug.
- ANSYS(1996). *Introduction to ANSYS Release 5.3*. USA, SAS IP Inc.
- ASHLEY, R. (1992). *Cobol Estruturado*. São Paulo, Érica.
- AUTODESK (1992). *AutoCAD Reference Manual Release 12*. USA, by AutoDesk Inc.
- AUTORYU, L.; RIGHETTO, A. M. (1995). *Geração de malha de elementos finitos para domínios complexos - aplicação na modelagem de um ciclone na costa noroeste da Austrália*. Recife, Abrh.
- BABUSKA I., (1983). *Accuracy estimates and adaptive refinements in finite element computation*. Chichester, John Wiley.
- BABUSKA, I. (1972). *Survey lectures on the mathematical foundations of the finite element method*. New York, Academic Press.
- BAPTISTA, S.M. (1994). *Análise de pavimentos de edifícios com a utilização do método dos elementos finitos*. São Carlos, Dissertação (Mestrado), EESC-USP.
- BARBIRATO, C. B. C. (1997). *Contribuição a análise de pavimentos de edifícios em laje nervurada*. São Carlos, Dissertação (Mestrado), EESC-USP.

- BARBIRATO, J. C. C. (1991). *Formulação do método dos elementos de contorno para sólidos elásticos tridimensionais, baseada na solução fundamental de Mindlin*. São Carlos, Dissertação (Mestrado), EESC-USP.
- BATHE, K. (1996). *Finite element procedures*. Englewood Cliffs, Prentice Hall.
- BORLAND(1996). *C++ Programmer's Guide*. Scotts Valley, CA, Borland International Inc
- BORLAND(1996). *C++ User's Guide*. Scotts Valley, CA, Borland International Inc.
- BORTOLIN, A.A. (1991). *Sobre a racionalização e a automação do cálculo de edifícios de concreto armado*. São Carlos. Dissertação (Mestrado) – EESC –USP
- BUEL, W.R.; BUSH, B.A. (1973). *Mesh generation – a survey*. Journal of Engineering for Industry, Transactions of the ASME, Series B, v.95, p.332-338, Feb.
- BUGEDA, G; ONATE, E; (1993). *A methodology for adaptive mesh refinement in optimum shape design problems*. Barcelona, Centro Internacional de Metodos Numericos en Ingenieria.
- CANTÚ, M. (1996). *Dominando o Delphi*. São Paulo, Makron Books.
- CARVALHO, J. (1989). *Projeto e implementação de pré-pós processador gráfico para o método dos elementos finitos*. São Carlos, Tese (Doutorado), EESC –USP.
- CAVENDISH, J.C. (1974). *Automatic triangulation of arbitrary planar domains for the Finite Element Method*. International Journal for Numerical Methods in Engineering, V.21, p-329-347.
- CAVENDISH, J.C.; FIELD, D.A.; FREY, W.H. (1985). *An approach to automatic three-dimensional finite element mesh generation*. International Journal for Numerical Methods in Engineering, v.21, p.329-347.
- CORRÊA, M. R. S.; RAMALHO, M. A. (1993). *Cálculo de pavimentos de edifícios utilizando-se o método dos elementos finitos*. In: *26a Jornadas Sudamericanas de Ingenieria Estructural*. Montevideo.
- CORRÊA, M. R. S.; RAMALHO, M. A. (1995). *Modelos numéricos para análise estrutural de edifícios*. In: *19 Seminário sobre Não-Linearidade Física E Geométrica das Estruturas de Concreto*. São Paulo, IBRACON.
- CORRÊA, M.R.S. (1991). *Aperfeiçoamento de modelos usualmente empregados no projeto de sistemas estruturais de edifícios*. São Carlos. Tese (Doutorado), EESC-USP.
- COSMOS/M – Introduction to COSMOS/M (1989). Structural research and analysis corporation, 2nd. Edition, Version 1.52a.

- DELPHI(1997) *Delphi 3.0 Object Pascal Guide*. Scotts Valley, CA, Borland International Inc
- DELPHI(1997) *Delphi 3.0 User's Guide*. Scotts Valley, CA, Borland International Inc
- DIN1053(1974). *Masonry: Design and Construction*. Berlim, Deutsche Norm.
- DUROCHER, L. L.; GASPER, A. (1979). *A versatile two-dimensional mesh generator with automatic bandwidth reduction*. Computers & Structures, v.10, n.4, p 561-575.
- FORTRAN(1992) *Microsoft Fortran PowerStation User's Guide*. USA, Microsoft Corp
- FORTRAN(1993) *Microsoft Fortran PowerStation Language Guide*. USA, Microsoft Corp.
- FUSCO(1986) *Estruturas de concreto: solicitações normais*. Rio de Janeiro, Guanabara Dois.
- GEORGE, P. L. (1991). *Automatic mesh generation :application to finite element*. New York, Wiley.
- GIONGO, J. S. (1994). *Concreto Armado: Projeto Estrutural de Edifícios*. São Carlos, EESC-USP.
- GUASSEMI, F (1982). *Automatic mesh generation scheme for a two or three dimensional triangular curved surface*. Computer's & Structures.v.15,n.6,p 613-626.
- HABER, R.; SHEPHARD, M.S.; ABEL, J.F.; GALLAGHER, R.H.; GREENBERG, D.P. (1981). *A general two-dimensonal graphical finite element preprocessor utilizing Discrete Trasfinite Mappings*. International Journal for Numerical Methods in Engineering, v.17, p-1015-1044.
- HERMAN, M. R.; LEBENSZTAJN, L. (1994). *Geração de malha auto-adaptativa para o método dos elementos finitos*. São Paulo, EPUSP.
- KAWANO, A. (1991). *Pré e pós processamento gráfico de dados para o método dos elementos finitos para sistemas baseados em microcomputadores*. São Paulo. Dissertação (Mestrado), EPUSP
- LEBENSZTAJN, L. (1989). *Desenvolvimento de pré e pós processadores para o método dos elementos finitos aplicados a conversão eletromecânica de energia*. São Paulo. Dissertação (Mestrado), EPUSP.
- LISERRE, L.; (1993). *Traçador de curvas de isovalor - saída no padrão DXF do Autocad*. In: Simpósio de Iniciação Científica, 1., São Paulo, 1993, São Paulo, Universidade de São Paulo.

- LISERRE, L; (1993). *Programa Integrado para análise gráfica de estruturas geradas pelo método de elementos finitos*. In: XII Congresso de Iniciação Científica e Tecnológica em Engenharia, 1., São Paulo, 1993, São Carlos, Universidade de São Paulo.
- LUSAS(1995). *Finite element system, version 11*, Survey, UK, FEA. McGraw-Hill.
- MICALI, J. F. M. (1994). *Modelagem de figuras planas e geração de malhas usando quadtree*. Dissertação (Mestrado). São Paulo, EP-USP.
- MICALI, J. F. M.; ANDRE, J. C. (1995). *Conceitos básicos sobre geração automática de malha*. São Paulo, EPUSP.
- NASTRAN(1995). *Quick Start Guide*. USA, MSC-Software.
- NBR6123 (1987). *Forças devidas ao vento em edificações: procedimentos*. Rio de Janeiro, Associação Brasileira de Normas Técnicas.
- O'BRIEN, S. K.(1992). *Turbo Pascal 6 : Completo e Total*. São Paulo, Makron,
- PEREIRA, G. S. (1997). *Contribuições a análise de estrutura de contraventamento de edifícios em concreto armado*. São Carlos, Dissertação (Mestrado), EESC-USP.
- PREPARATA F. P; SHAMOS, M. I.. (1985). *Computational Geometry An Introduction*. USA – Springer-Verlag New York Inc..
- RAMALHO,M.A. (1990) *Sistema para análise de estruturas considerando interação com o meio elástico*. São Carlos. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.
- RESENDE, M.N (1990). *Análise de pavimentos de edifícios pelo método dos elementos finitos em microcomputador*. São Carlos, Dissertação (Mestrado) – EESC-USP  
Rio de Janeiro, Editora Ciência Moderna Ltda.
- SABONNADIÈRE, J.; COULOMB, J. (1987). *Finite element methods in CAD*. London, North Oxford Academic.
- SADEK, E.A. (1980). *A scheme for the automatic generation of triangular finite elements*. International Journal for Numerical Methods in Engineering, v.15, p.1813-1882.
- SAP90(1991) *A series of computer programs for the finite element analysis of structures*. Berkeley, Computer and Structures Inc.
- SHEPHARD, M.S.; YERRY, M.A. (1983). *Approaching the automatic generation of finite element meshes*. Computers in Mechanical Engineering, p.49-56.



SOUSA JR., E. (1996). *Aplicativo para o ambiente windows para aquisição de dados para análise de pavimentos de edifícios via método dos elementos finitos*. São Carlos, Dissertação (Mestrado), EESC-USP.

SUPERSAP: USER'S MANUAL (1987). Algor Interactives Systems, Inc

WEISKAMP, K. e HEINY, L. (1992). *Programação Gráfica em Turbo Pascal 6.0*.

ZIENKIEWICZ, O.C.; PHILIPS, D.V. (1971). *An automatic Generation scheme for plane and curved surfaces by "Isoparametric" Co-ordinates*. International Journal for Numerical Methods in Engineering, v.3, p.519-528.