

**UM APLICATIVO PARA O AMBIENTE WINDOWS  
PARA AQUISIÇÃO DE DADOS PARA ANÁLISE DE  
PAVIMENTOS DE EDIFÍCIOS VIA MÉTODO DOS  
ELEMENTOS FINITOS**

**EDGARD SOUSA JUNIOR**

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. João Batista de Paiva**

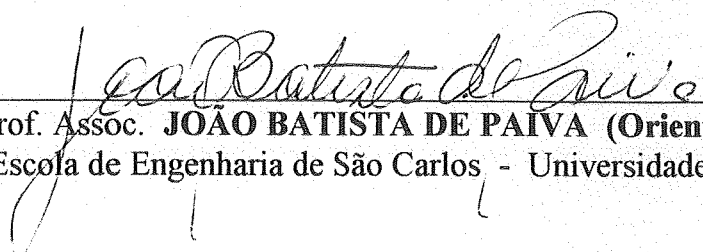
**São Carlos**


**1996**


**DEPARTAMENTO DE ESTRUTURAS  
ESCOLA DE ENGENHARIA DE SÃO CARLOS**


**FOLHA DE APROVAÇÃO**

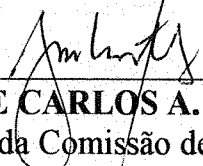
Dissertação defendida e aprovada em 29-10-1996  
pela Comissão Julgadora:

  
\_\_\_\_\_  
Prof. Assoc. **JOÃO BATISTA DE PAIVA (Orientador)**  
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

  
\_\_\_\_\_  
Profa. Doutora **AGMA JUCI M. TRAINA**  
(Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo)

  
\_\_\_\_\_  
Prof. Doutor **MARCIO ANTONIO RAMALHO**  
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

  
\_\_\_\_\_  
Prof. Dr. **MOUNIR KHALIL EL DEBS**  
Coordenador da área - Engenharia de Estruturas

  
\_\_\_\_\_  
**JOSE CARLOS A. CINTRA**  
Presidente da Comissão de Pós-Graduação

*A minha esposa, Fabiana Lopes de Oliveira,  
companheira na vida e nos estudos.*

## **AGRADECIMENTOS**

A Deus, por me permitir concluir mais uma etapa da minha vida.

A meus pais, por todo apoio e orientação que me proporcionaram.

A meus sogros, por toda ajuda e incentivo durante todo o curso de pós-graduação.

Ao Professor João Batista de Paiva, pela orientação fornecida durante a elaboração deste trabalho.

À CAPES, pela bolsa de estudo concedida.

À Universidade Federal do Maranhão - UFMA por ter concedido afastamento para realização do mestrado.

A todos os colegas, professores e funcionários do Departamento de Estruturas da EESC/USP pela colaboração.



# SUMÁRIO

LISTA DE FIGURAS .....	i
RESUMO .....	iv
<i>ABSTRACT</i> .....	v
1 INTRODUÇÃO .....	1
2 HISTÓRICO .....	4
3 METODOLOGIA UTILIZADA .....	15
3.1 A escolha do ambiente Windows .....	15
3.2 A linguagem de programação adotada .....	17
4 APRESENTAÇÃO DO APLICATIVO .....	18
4.1 Janela principal .....	19
4.2 Janela texto .....	24
4.3 Janela gráfica .....	25
4.3.1 Dados gerais .....	25
4.3.2 Poligonal de contorno .....	26
4.3.3 Pontos pré-definidos .....	27
4.3.4 Gerador de malha .....	30
4.3.5 Gerador de vigas .....	38
4.3.6 Lista de pontos nodais .....	45
4.3.7 Lista de elementos de placa .....	46
4.3.8 Lista de elementos de viga .....	47
4.3.9 Diálogo visão/informação .....	49
4.3.10 Menu flutuante .....	50

4.3.11 Gerador de relatórios .....	59
4.4 Esquema gráfico representativo do aplicativo PEC .....	77
5 MANIPULAÇÃO INTERNA DOS DADOS .....	78
6 EXEMPLOS DE UTILIZAÇÃO DO APLICATIVO .....	83
6.1 Edifício Andrade .....	83
6.2 Edifício Firenze .....	86
6.3 Edifício Casa Blanca .....	88
6.4 Folha poliédrica .....	93
7 CONCLUSÕES .....	95
REFERÊNCIAS BIBLIOGRÁFICAS .....	97

## LISTA DE FIGURAS

Figura 01 - Malha gerada pela técnica Transformação de Coordenadas .....	5
Figura 02 - Geração dos pontos internos segundo a densidade .....	6
Figura 03 - Malha gerada por Triangulação mais Suavização .....	7
Figura 04 - Geração dos elementos nos cantos da estrutura .....	8
Figura 05 - Método de Triangulação proposto por SADEK (1980) ...	8
Figura 06 - Malha gerada utilizando-se Mapeamento Transfinito ....	9
Figura 07 - Malha de um objeto circular obtida com a Técnica <i>Quadtree</i> .....	10
Figura 08 - Elementos utilizados na técnica <i>Modified Quadtree</i> .....	11
Figura 09 - Malha gerada pela técnica <i>Modified Quadtree</i> .....	11
Figura 10 - Malha gerada pelo gerador proposto por KULAA & KLINGE (1995) .....	13
Figura 11 - Diálogo de abertura do aplicativo .....	19
Figura 12 - Janela principal com vários projetos abertos .....	20
Figura 13 - Diálogo Sobre o PEC .....	20
Figura 14 - Diálogo Abrir .....	21
Figura 15 - Janela texto ativa .....	24
Figura 16 - Caixa de diálogo Dados .....	25
Figura 17 - Caixa de diálogo Poligonais de contorno .....	26
Figura 18 - Exemplos de uma planta feita com poligonais .....	27
Figura 19 - Lista de pontos pré-definidos .....	29
Figura 20 - Representação gráfica dos pontos pré-definidos .....	29
Figura 21 - Caixa de diálogo Define Malha .....	30
Figura 22 - O gerador de malha .....	31
Figura 23 - Malha gerada com os dados da figura 21 .....	31
Figura 24 - Malha gerada por dois macro elementos sendo que um tem como altura a metade da altura do outro .....	33

## LISTA DE FIGURAS (continuação)

Figura 25 - Malha gerada por um macro elemento .....	34
Figura 26 - Malha gerada por três macro elementos .....	34
Figura 27 - Dados da malha do Exemplo 1 .....	35
Figura 28 - Gerador de Malha antes da malha ser gerada .....	35
Figura 29 - Gerador de Malha após a malha ter sido gerada .....	36
Figura 30 - Pergunta sobre os pontos pré-definidos .....	36
Figura 31 - Representação gráfica da malha do Exemplo 1 .....	37
Figura 32 - Definição de vigas .....	39
Figura 33 - Esquema representativo da geração de uma viga .....	40
Figura 34 - Final da geração das vigas da figura 32 .....	42
Figura 35 - Vigas em conjunto geradas com os dados da figura 32 .....	44
Figura 36 - Representação gráfica das vigas do Exemplo 1 .....	44
Figura 37 - Lista de pontos nodais .....	45
Figura 38 - Lista de elementos de placa .....	47
Figura 39 - Lista de elementos de viga .....	48
Figura 40 - Diálogo visão/informação .....	49
Figura 41 - Menu flutuante da janela gráfica .....	50
Figura 42 - Edição de pontos nodais .....	51
Figura 43 - Edição de elementos de placa .....	52
Figura 44 - Edição de elementos de viga .....	52
Figura 45 - Seleção de alguns elementos .....	52
Figura 46 - Define ponto nodal .....	53
Figura 47 - Define elemento de placa .....	54
Figura 48 - Retirando-se um elemento da malha .....	55
Figura 49 - Localização do ponto a ser gerado .....	55
Figura 50 - Definição de um novo ponto nodal .....	56
Figura 51 - Resultado da geração do ponto nodal .....	56

## LISTA DE FIGURAS (continuação)

Figura 52 - Geração do primeiro elemento triangular .....	57
Figura 53 - Geração do segundo elemento triangular .....	57
Figura 54 - Geração do terceiro elemento triangular .....	57
Figura 55 - Geração do quarto elemento triangular .....	57
Figura 56 - Resultado da geração dos elementos triangulares .....	58
Figura 57 - Resultado final da alteração .....	58
Figura 58 - Malha final do Exemplo 1 .....	59
Figura 59 - Gerador de relatórios .....	68
Figura 60 - Seleção do arquivo de formato .....	69
Figura 61 - Gerador de relatórios com um formato escolhido .....	69
Figura 62 - Gerador de relatórios: final de uma geração .....	70
Figura 63 - Formato DXF do Exemplo 1 mostrado pelo AutoCAD ...	72
Figura 64 - Exemplo 1 visto pelo sistema NEXP .....	72
Figura 65 - Exemplo 1 visto pelo sistema SAP 90 .....	73
Figura 66 - Exemplo 1 com uma malha de 220 elementos de placa .....	75
Figura 67 - Planta de fôrma do edifício Andrade .....	84
Figura 68 - Edifício Andrade, somente elementos de viga .....	85
Figura 69 - Malha do edifício Andrade .....	85
Figura 70 - Malha do edifício Firenze .....	86
Figura 71 - Planta de fôrma do edifício Firenze .....	87
Figura 72 - Malha do edifício Casa Blanca .....	88
Figura 73 - Planta de fôrma do edifício Casa Blanca .....	89
Figura 74 - Malha do edifício Casa Blanca vista pelo sistema NEXP .....	90
Figura 75 - Malha da folha poliédrica .....	93
Figura 76 - Malha da folha poliédrica vista no AutoCAD .....	94

## RESUMO

SOUSA JUNIOR, E. *Um aplicativo para o ambiente Windows para aquisição de dados para análise de pavimentos de edifícios via método dos elementos finitos*. São Carlos, 1996. 99p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

Apresenta-se um pré-processador para análise de pavimentos de edifícios via método dos elementos finitos desenvolvido como um aplicativo do ambiente Windows. O programa é interativo, permitindo identificar problemas de imediato na criação do modelo a ser estudado. Possui recursos gráficos básicos para a definição geométrica da estrutura. A saída de dados é programável para que o pré-processador não se prenda a um só programa de cálculo. Assim, os mesmos dados de uma estrutura podem ser analisados por diferentes programas de cálculo o que possibilita a escolha do programa mais adequado, dependendo do tipo de análise desejado. O aplicativo tem recursos para definir elementos finitos de barra e placa. Como o número desses elementos normalmente é grande, eles são gerados de uma forma automatizada.

Palavras-chave: elementos finitos, gerador de malha, Windows, Borland C++.

## **ABSTRACT**

SOUSA JUNIOR, E. *A Windows preprocessor for building slab analysis by Finite Element Method*. São Carlos, 1996. 99p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

It is presenting a Windows preprocessor for building slab analysis by Finite Element Method. It is an interactive program, allowing problems to be identified while the model is generated. It has graphical resources to define the structure. The program output is defined by the user; therefore it is possible to use the preprocessor with many other finite element programs. This allows the engineer to choose the appropriate program depending on the kind of analysis is wanted. The program can define frame and shell finite elements. As slabs usually need large numbers of these elements to be properly analyzed, they are generated using automatic meshing.

Keywords: finites elements, mesh generation, Windows, Borland C++.

## 1 INTRODUÇÃO

O método dos elementos finitos é muito utilizado por engenheiros na análise de pavimentos de edifícios. Entretanto, o grande volume de dados de entrada torna necessário o desenvolvimento de pré-processadores. Nestes pré-processadores, o ponto mais importante é o gerador de malhas de elementos finitos, pois dela depende a melhor ou pior representação do pavimento.

Em 1976, B. FREDRIKSSON e J. MACKERLE *apud* GHASSEMI (1982) listaram aproximadamente 250 programas de geração de malha. Segundo a pesquisa bibliográfica feita neste trabalho esse número não para de crescer. Isso mostra a dificuldade em se fazer um gerador de malha que venha a satisfazer plenamente o usuário final.

Com o desenvolvimento da indústria da informática, existem hoje novas maneiras de entrada de dados para o computador. Sistemas operacionais baseados em um ambiente gráfico e multitarefa estão se tornando padrão para os microcomputadores, equipamentos muito utilizados pelos projetistas, pois já oferecem desempenho e capacidade de armazenamento suficientes para o cálculo estrutural que antes era efetuado apenas em computadores de grande porte. Com esse panorama é inevitável que surjam geradores de malhas cada vez mais simples de serem usados e mais eficientes.

O método dos elementos finitos consiste em subdividir a estrutura a ser analisada em partes menores e mais simples, denominadas elementos



finitos. A conexão entre os elementos finitos é efetuada pelos nós ou pontos nodais que podem estar localizados nos vértices e nos lados dos elementos, dependendo do tipo de elemento finito. Associam-se aos elementos finitos uma função aproximadora dos deslocamentos em seus domínios. Os deslocamentos internos ao elemento ficam determinados exclusivamente em função dos deslocamentos nodais. Após a minimização da energia potencial total da estrutura, obtém-se um sistema de equações lineares cuja resolução fornece os deslocamentos nodais e a partir desses as tensões ou esforços são obtidos. Esta solução é mais simples do que a proposta pela teoria da elasticidade que estuda o equilíbrio de um elemento infinitesimal da estrutura e a solução do problema recai na resolução de sistemas de equações diferenciais de derivadas parciais.

Nota-se que a solução da estrutura pela Teoria da Elasticidade fornece um resultado exato, porém, a imposição das condições de contorno e a resolução final das equações diferenciais quase sempre se tornam um trabalho difícil e, na maioria das vezes, impraticável. Mas, o Método dos Elementos Finitos, quando convenientemente aplicado, fornece resultados bem próximos do exato. Um dos fatores para que isso ocorra é que a estrutura seja corretamente modelada.

Os programas que utilizam-se do Método dos Elementos Finitos manipulam um grande volume de dados. Normalmente, quanto mais for dividida a estrutura em elementos finitos mais próximo do resultado exato se chegará e maior será o volume de dados do problema. Estudos estatísticos têm revelado que 75% das falhas estruturais dos modelos podem ser atribuídas a erros humanos, STEWART & MELCHERS (1989). Normalmente 80% do tempo total gasto na análise pelo método dos elementos finitos é consumido pela entrada dos dados quando se usa o método convencional, HABER et al. (1981). Isto demonstra a importância de desenvolver pré-processadores para a aquisição destes dados.

As primeiras técnicas usadas para desenvolver um gerador de malhas automático datam da década de 70. Hoje já se possui ferramentas

computacionais bem mais avançadas e que devem ser utilizadas no desenvolvimento de aplicativos para a análise estrutural.

Este trabalho aborda o desenvolvimento de um pré-processador para programas de elementos finitos voltado para a análise de pavimentos de edifícios. A linguagem de programação utilizada foi Borland C++ 5.0 e o aplicativo foi desenvolvido para o ambiente Windows 95.

São apresentados alguns métodos de geração de malha de elementos finitos, porém, neste trabalho, é utilizada a técnica de triangulação automática em que a estrutura é dividida em macro elementos. Este método foi considerado ser o de melhor eficiência para o tipo de malha gerada pelo pré-processador.

O programa usufrui das facilidades do ambiente gráfico oferecido pelo Windows, e ainda, possui um interpretador que é usado para gerar relatórios com os dados da estrutura discretizada. Ou seja, a saída de dados é programável, definida pelo usuário em tempo de execução, para que o pré-processador não se prenda a um só programa de cálculo. Além disso, com poucas modificações, o programa poderá ser utilizado para problemas de chapas.

No capítulo 2 é feito um histórico das técnicas de geração de malha de elementos finitos desenvolvidas desde 1971 até o presente.

A seguir, no capítulo 3, descreve-se a metodologia utilizada no trabalho, quando se justifica a escolha do ambiente Windows e a linguagem de programação adotada.

O capítulo 4 apresenta o aplicativo PEC (Projeto Estrutural por Computador) e descreve os módulos que o compõe. É bom deixar claro que apesar do nome PEC sugerir um aplicativo em que se efetue um projeto estrutural completo assistido por computador, neste trabalho é apresentado apenas a primeira parte de um projeto maior, que é o gerador de dados para a análise de estruturas via método dos elementos finitos.

No capítulo 5 é demonstrada a manipulação interna dos dados.

O capítulo 6 contém exemplos de utilização do aplicativo PEC.

O capítulo 7 apresenta as conclusões gerais deste trabalho.

## 2 HISTÓRICO

Uma das primeiras técnicas usadas para gerar malha de elementos finitos foi a Transformação de Coordenadas que produz malhas bem distribuídas. O procedimento mais popular para essa técnica é o baseado no conceito de Coordenadas Isoparamétricas, que utiliza um mapeamento das superfícies planas ou curvas através de funções de interpolação de polinômios. A estrutura é dividida em super-elementos quadrilaterais que não possuam aberturas em suas superfícies. Depois, cada super-elemento é dividido em elementos finitos tomando-se o cuidado de manter o número de nós nas extremidades de cada super-elemento coincidente com a do seu vizinho, garantindo a consistência da malha. Por último são analisados os pontos nodais que possuem coordenadas iguais e a junção dos super-elementos é efetuada, formando uma só malha. ZIENKIEWICZ & PHILLIPS (1971) apresentaram um esquema de um gerador automático de malhas para superfícies planas e curvas usando coordenadas isoparamétricas. Um outro exemplo de gerador de malha que utiliza essa técnica é o proposto por DUROCHER & GASPER (1979) (figura 1). WU (1980), após testar esse gerador, notou que para elementos triangulares, na junção dos super-elementos ocorria uma disposição não apropriada dos elementos finitos. A causa do problema foi encontrada e ele propôs uma sub-rotina para ser acrescentada ao programa inicial e evitar esse problema. Pouco tempo depois WU (1982) propôs técnicas para evitar nós duplicados na numeração dos super-elementos, o que era a causa da má disposição dos elementos finitos no programa de DUROCHER & GASPER (1979). Porém,

as correções feitas por WU (1982) eram válidas somente para elementos triangulares. LIU & CHEN (1987), apresentaram uma técnica simples para eliminar nós duplicados. Essa técnica é válida tanto para elementos triangulares como para elementos quadriláteros e exige menos tempo de processamento do que a proposta por WU (1982). Depois, LIU & CHEN (1988), propuseram mais uma otimização no gerador de malha de DUROCHER & GASPER (1979). Quando esse gerador era aplicado a problemas de elementos finitos com uma disposição complexa dos super-elementos onde era compartilhado mais de um contorno ou face entre dois super-elementos, a malha gerada era incorretamente construída. As modificações propostas funcionam perfeitamente com a técnica sugerida anteriormente para eliminação de nós duplicados.

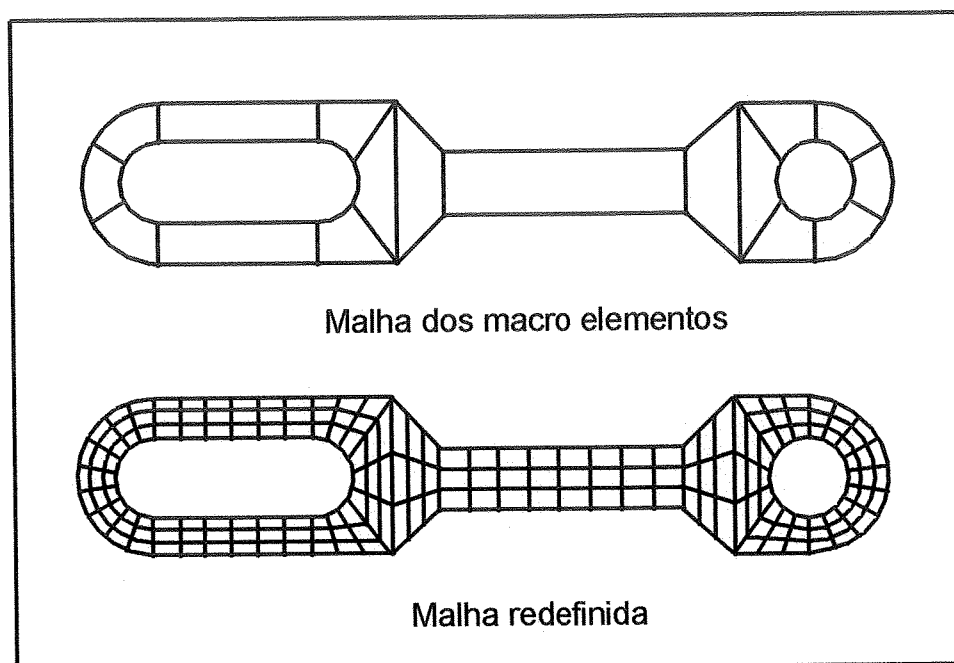


Figura 1 - Malha gerada pela técnica Transformação de Coordenadas

CAVENDISH (1974) expôs um novo algoritmo baseado em modificação do gerador de malha para elementos finitos de FUKUDA & SUHARA *apud* CAVENDISH (1974) que utiliza a técnica de Triangulação. O algoritmo de CAVENDISH (1974) é indicado para estruturas planas. Neste algoritmo primeiramente são definidos os pontos dos elementos que ficam

no contorno da estrutura. O algoritmo não gera os pontos do contorno, assim esses pontos devem ser definidos pelo usuário ou deve-se utilizar um processo que tendo os vértices, seja possível gerar os pontos intermediários por interpolação. São definidas regiões que indicam a densidade da malha a ser gerada. Essa densidade indica a distância média que um ponto deve ter de outro ou do contorno. O algoritmo então gera automaticamente os pontos internos consistentes com a densidade definida anteriormente (figura 2). O programa então liga os pontos tentando formar triângulos o mais próximo possível de triângulos equiláterais (figura 3). A modificação de CAVENDISH (1974) sobre o gerador de malha de FUKUDA & SUHARA *apud* CAVENDISH (1974) foi acrescentar à técnica de Triangulação o procedimento definido como Suavização. Esse procedimento visa regularizar a triangulação, formando triângulos mais uniformes. Em 1985 foi publicado um método de gerar malhas tridimensionais de elementos finitos por CAVENDISH et al. (1985) que tem como base a técnica exposta por ele em 1974. A malha é formada por camadas planas de elementos triangulares que depois são interligadas formando elementos tetraédricos.

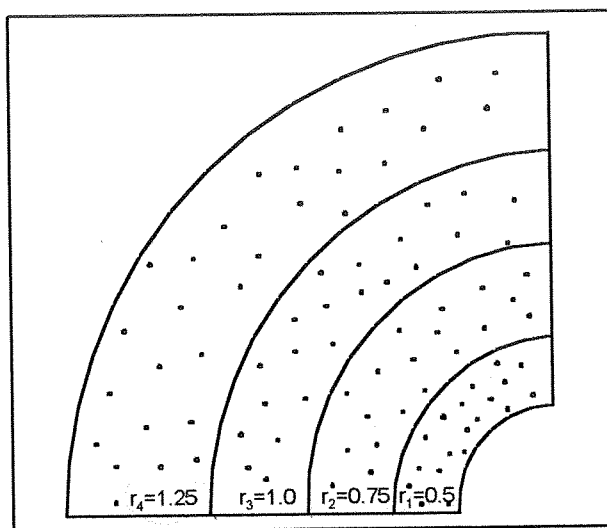


Figura 2 - Geração dos pontos internos segundo a densidade

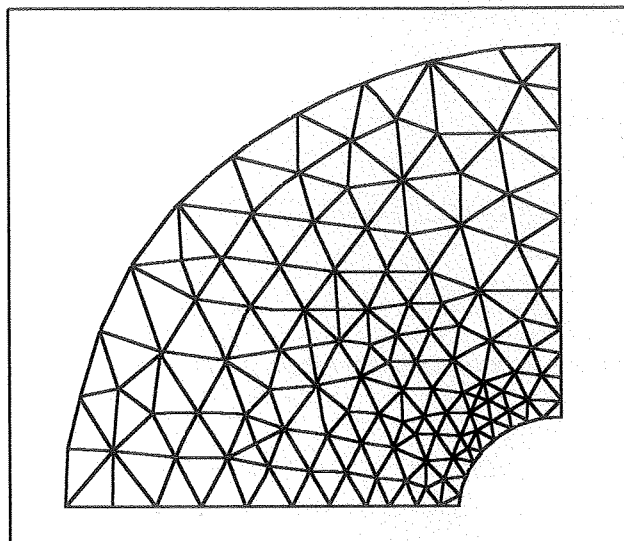


Figura 3 - Malha gerada por Triangulação mais Suavização

SADEK (1980) propôs um método de triangulação automática para planos definidos por múltiplos lados arbitrariamente. Como é um método que não funciona em contornos curvos, logo no início foi proposto que deveria ser usado em conjunto com o método sugerido por ZIENKIEWICZ & PHILLIPS (1971) que utiliza o conceito de coordenadas isoparamétricas. A essência do esquema consiste em formar elementos finitos triangulares nos cantos da estrutura. Consideram-se canto um ponto no contorno que coincida com um vértice e possua um ângulo diferente de  $180^\circ$ . Terminando-se esse passo, considera-se os elementos formados como se fossem eliminados da área em estudo e o mesmo conceito é aplicado aos outros cantos que surgiram no domínio (figura 4). E assim, o processo é feito até formar elementos em todo o domínio (figura 5).

Um pré-processador bidimensional feito para uso genérico foi desenvolvido por HABER et al. (1981) utilizando Mapeamento Transfinito. No início do artigo é feita uma comparação entre três métodos: Laplaciano, Coordenadas Isoparamétricas e Mapeamento Transfinito. O método Laplaciano utiliza equações para definirem as coordenadas dos pontos internos da malha. Essas equações, que podem ser interpretadas como o operador de diferenças finitas Laplaciano, em seu conjunto formam um sistema de equações não lineares, que pode ser resolvido via técnica

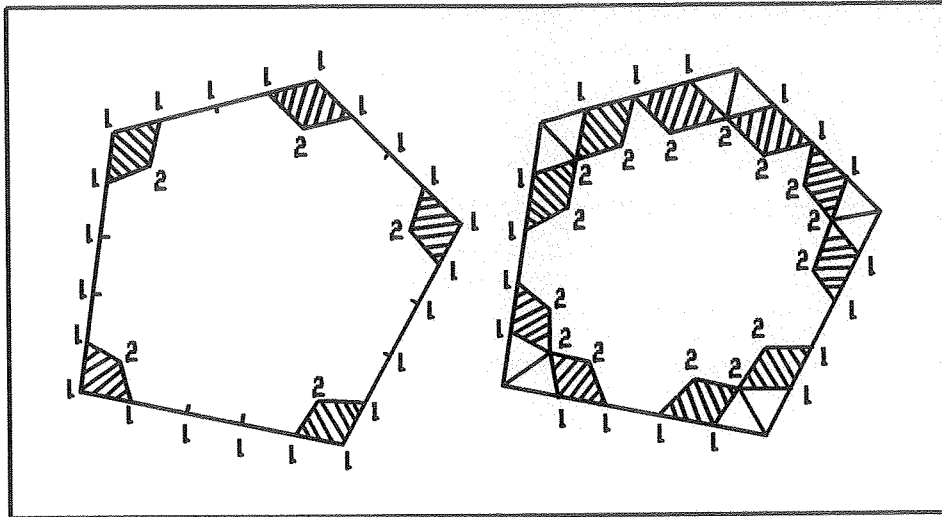


Figura 4 - Geração dos elementos nos cantos da estrutura

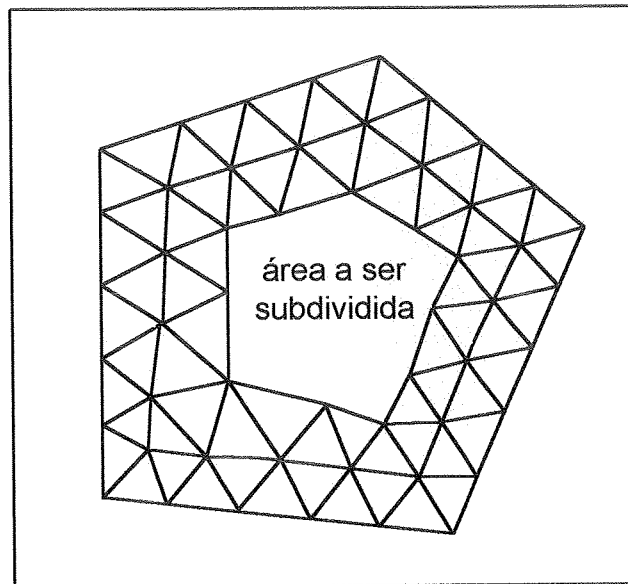


Figura 5 - Método de Triangulação proposto por SADEK (1980)

interativa como a de Gauss-Seidel ou interação de Jacobi. Esse método requer um grande esforço computacional na geração das malhas de elementos finitos. O método de Coordenadas Isoparamétricas já foi discutido anteriormente neste trabalho, sendo assim, falar-se-á logo do método do Mapeamento Transfinito. Esse método, desenvolvido inicialmente por GORDON & HALL *apud* HABER et al. (1981) em 1973, foi idealizado para gerar coordenadas curvilíneas em domínios arbitrários. O método do Mapeamento Transfinito gera uma superfície ou volume igual, ou

tão próximo o quanto se deseja, à superfície ou volume real da estrutura em estudo com uma quantidade infinita de pontos. Essa propriedade é que dá origem ao termo Mapeamento Transfinito. Isso contrasta com o mapeamento isoparamétrico em que se define a superfície com um número finito de pontos, isto é, os pontos usados na interpolação. No caso bidimensional o Mapeamento Transfinito define um modelo exato do contorno da estrutura em estudo, e assim não são introduzidos erros geométricos no mapeamento. No exemplo de uma região formada por quatro lados, primeiramente definem-se as quatro curvas que representam o contorno. Duas curvas na direção  $u$  e duas na direção  $v$ . Depois, utilizando-se conceitos de álgebra linear, são feitas duas interpolações, uma na direção  $u$  e outra em  $v$ . O resultado é uma malha formada por elementos quadriláteros que acompanham o traçado das curvas do contorno (figura 6). Elementos triangulares podem ser criados dividindo-se os elementos formados na diagonal. Este método é mais indicado em regiões com contornos curvos nos quatro lados. O mapeamento isoparamétrico é obtido como um caso específico de mapeamento transfinito quando as curvas do contorno são definidas por polinômios de Lagrange.

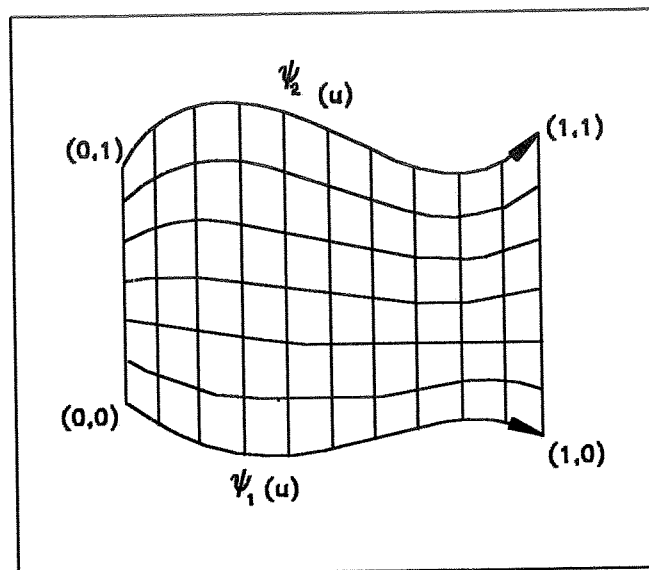


Figura 6 - Malha gerada utilizando-se Mapeamento Transfinito



YERRY & SHEPHARD (1983) apresentaram uma nova técnica de triangulação: *Modified Quadtree*. A técnica *Quadtree* consiste em desenhar o objeto de interesse dentro de um quadrado. O quadrado é então dividido em quatro quadrantes. Depois cada quadrante é testado para ver se está totalmente dentro da figura, parcialmente dentro ou totalmente fora. Se o quadrante estiver fora, ele é eliminado. Se estiver parcialmente dentro da figura, ele é subdividido em quatro quadrantes. A análise se repete para cada novo quadrante formado até todos os quadrados estarem dentro da figura (figura 7). Quando há curvas no contorno da figura, a malha fica mal desenhada. Surge o efeito de uma escadinha formada pelos cantos dos quadrados e a malha fica muito densa nessa região. Para resolver esse

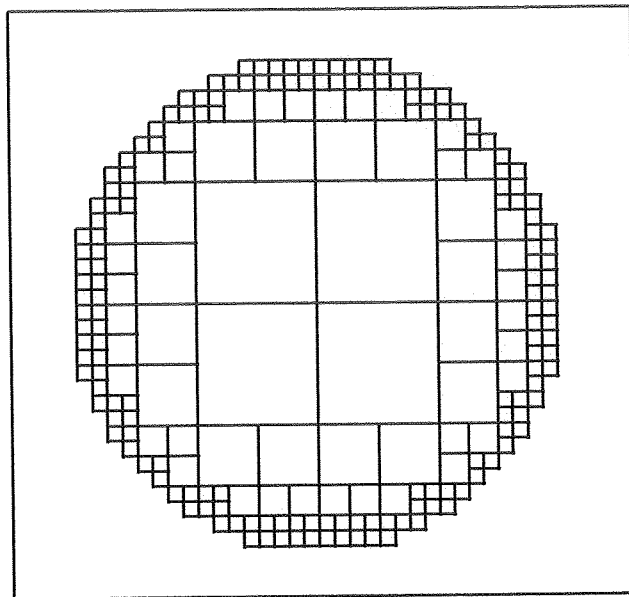


Figura 7 - Malha de um objeto circular obtida com a Técnica *Quadtree*

problema foi desenvolvido a técnica *Modified Quadtree*, onde os elementos gerados não precisam mais ser quadrados perfeitos, podendo assumir outras formas. As novas formas dos elementos são obtidas cortando-se o elemento quadrado por uma linha ligada à quarta parte, metade ou ao ponto final de dois dos seus lados (figura 8). O resultado final é uma malha com o contorno mais suave (figura 9). Mesmo assim, a posição dos pontos no contorno ainda precisa de um ajuste para coincidir com o contorno real

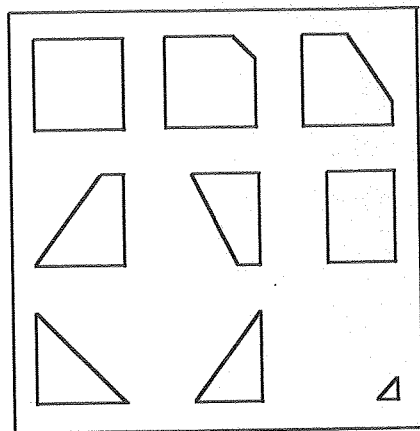


Figura 8 - Elementos utilizados na técnica *Modified Quadtree*

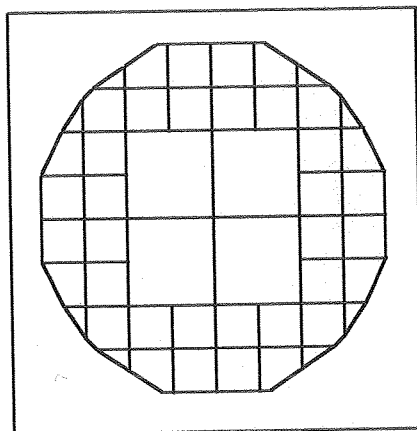


Figura 9 - Malha gerada pela técnica *Modified Quadtree*

da figura. E depois, é necessário utilizar-se a técnica de suavização para distribuir melhor a área dos elementos, formando elementos mais uniformes. Em 1987 foi apresentado por BAEHMANN et al. (1987) um gerador de malha automático bidimensional em que a técnica básica era derivada da apresentada por YERRY & SHEPHARD (1983). Foram feitas modificações com o propósito de acrescentar mais potencialidade à técnica.

Tem-se também o método proposto por REZENDE (1990). Consiste em gerar malhas quadrilaterais onde pode haver a junção de duas ou mais malhas até formar o desenho desejado. Nesse gerador também é possível eliminar elementos do interior da malha definindo-se um outro quadrilátero que indica a área onde não deve ter elementos finitos. É um processo feito para gerar pavimentos típicos de edifícios, ou seja, pavimentos formados

por partes quadrilaterais. É muito veloz, porém, torna-se complicado quando a geometria do pavimento possui curvas.

BAPTISTA (1994) propôs um gerador de malha de elementos finitos. Primeiramente é desenhado o contorno geométrico do objeto sem nenhum tipo de restrição quanto à sua forma, podendo apresentar vários ângulos, reentrâncias e uma ou mais aberturas no seu interior. Depois é gerada uma malha retangular sobre esse objeto. Utilizando-se conceitos básicos de Geometria Vetorial são verificados os pontos que não pertencem ao domínio e, então, esses pontos são eliminados. A partir daí, analisam-se os vértices do contorno que não coincidam com pontos nodais. O ponto nodal mais próximo do vértice que se encontra nesta situação tem suas coordenadas ajustadas para coincidirem com as do vértice e solucionar o problema de não ter um ponto nodal em um vértice.

Foi encontrado também na pesquisa bibliográfica o gerador de malha proposto por KULLAA & KLINGE (1995). Ele é um gerador desenvolvido para gerar malhas de elementos finitos de cascos de navios mas, pode ser usado sem problemas em estruturas similares como pontes ou placas, pois nele são usados vários tipos de elementos como elementos de vigas e pilares. Primeiramente, o usuário define a estrutura com objetos geométricos como pontos, linhas e áreas. Depois o gerador de malha corta a malha definida pelo usuário com linhas tentando formar elementos quadrilaterais, quando isto não é possível, ele forma elementos triangulares. É possível definir um espaçamento mínimo entre dois pontos. Se forem gerados pontos próximos entre si com espaçamento menor que o estabelecido, os pontos convergem para um só, evitando assim formação de micro elementos. Quando existe um ponto no interior de um elemento finito, o elemento é subdividido em elementos menores. Dependendo da situação o elemento pode ser dividido em elementos quadrilaterais ou triangulares (figura 10). A grande vantagem desse gerador de malha é que os elementos de vigas ou pilares podem ser postos depois da malha de elementos de placa ter sido gerada.

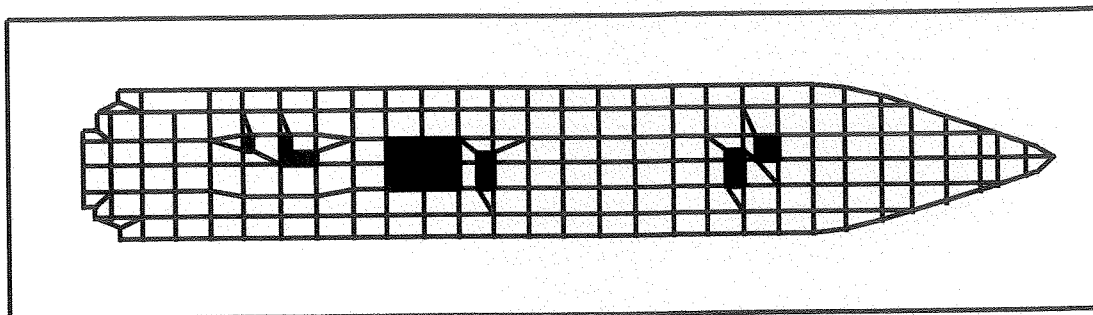


Figura 10 - Malha gerada pelo gerador proposto por KULLAA & KLINGE (1995)

As técnicas encontradas para geradores de malhas de elementos finitos foram quatro: Triangulação Automática, Transformação de Coordenadas, Procedimentos de Suavização e Funções Combinadas. Foi comum encontrar geradores de malha que utilizam uma combinação de duas das técnicas. A escolha da técnica a ser utilizada em um pré-processador depende do tipo de projeto em que ele vai ser mais utilizado. Deve haver um equilíbrio entre tempo de processamento e qualidade da malha. Normalmente um processo que gera uma malha bem distribuída demora muito para ser elaborado, e em alguns casos não compensa o tempo perdido no processamento quando uma malha simples pode dar resultados equivalentes aos de uma malha bem trabalhada. Na análise de problemas de engenharia mecânica onde encontram-se variações muito grandes de tensões em regiões muito próximas, as malhas devem ter algumas áreas bem mais refinadas que outras. Já na análise de pavimentos de edifícios as malhas podem ser mais uniformes e raramente algumas regiões do projeto exigem uma densidade maior de elementos.

No aplicativo desenvolvido neste trabalho (PEC - Projeto Estrutural por Computador), a técnica utilizada para gerar a malha está classificada, segundo a pesquisa bibliográfica, como sendo de Triangulação Automática. As técnicas Transformação de Coordenadas e Funções Combinadas não foram utilizadas de início por exigirem mais tempo de processamento já que utilizam cálculos avançados de geometria, porém nada impede que se venha a acrescentar no aplicativo novas funções para gerar malhas que

utilizem essas técnicas. A técnica Procedimentos de Suavização que normalmente é utilizada junto com uma das outras três técnicas também não está sendo utilizada pelo aplicativo. A técnica de Triangulação Automática foi escolhida porque ela exige pouco tempo de processamento e define elementos bem distribuídos.

## 3 METODOLOGIA UTILIZADA

### 3.1 A escolha do ambiente Windows

Optou-se em fazer um aplicativo para o ambiente Windows porque este está se tornando o padrão mais aceito e usado no mercado da microinformática. Isso se dá pelas facilidades que um ambiente gráfico multitarefa pode oferecer. Um sistema com uma interface gráfica bem elaborada exige pouco treinamento e pouco esforço do usuário pois, esse tipo de sistema pode ser tão intuitivo que com apenas a lógica e associações venha a se operá-lo. A multitarefa, capacidade de executar mais de um aplicativo ao mesmo tempo, facilita a troca de dados entre programas, é possível transferir dados de uma forma simples com a utilização do *Clipboard* - Área de Transferência - , ou através de métodos mais sofisticados, como o DDE - *Dynamic Data Exchange* - e o OLE - *Object Linking and Embedding*.

O programador para Windows não precisa se preocupar em que equipamento seu programa será executado. A comunicação entre aplicativo e máquina é feita através do Windows que utiliza *drivers* - controladores de dispositivos - para cada periférico. Desta forma o Windows consegue proporcionar gráficos independentes de dispositivos. O programador deve utilizar as rotinas da GDI - *Graphic Device Independent* - do Windows para que o programa possa desenhar em qualquer dispositivo utilizando o mesmo conjunto de chamadas, por exemplo, a rotina *Arc* da GDI é chamada para desenhar um arco tanto na tela como em impressoras,

independente de o monitor ser VGA, SVGA ou outro e do modelo da impressora.

O Windows é um ótimo gerenciador de memória. Quando se executa mais de uma cópia ou instância de um mesmo aplicativo, ele faz com que o seguimento de código seja compartilhado entre elas. Isto reduz muito a memória utilizada. Os seguimentos de um programa Windows são classificados como descartáveis, móveis e fixos. Quando o Windows precisa em um determinado instante de uma certa quantidade de memória e não encontra, ele libera os seguimentos descartáveis - aqueles que podem ser recuperados a partir do executável em disco - que não estão sendo usados no momento. Depois o Windows defragmenta a memória movendo os seguimentos classificados como móveis e assim ele consegue um seguimento contínuo de memória livre. Além disso, o Windows consegue criar memória virtual em disco de maneira transparente para o programador. Outro recurso de gerenciamento de memória utilizado pelo Windows são as DLL's - *Dynamic Link Libraries* - que são um formato especial de arquivo executável para exportar e importar funções. Quando um programa é compilado, as sub-rotinas ou funções são vinculadas ao código do programa executável. Isso se chama vinculação estática. Desta forma, se programas diferentes que utilizam funções iguais mas, que foram compiladas utilizando a vinculação estática, forem executados ao mesmo tempo, haverá um desperdício de memória pois, duas funções iguais estarão na memória ocupando espaços diferentes. Com o uso de DLL's, ou seja, bibliotecas de vinculação dinâmicas as funções são compiladas separadas do código de programa executável e a vinculação é feita em tempo de execução. Com as DLL's os aplicativos podem compartilhar tanto funções como recursos, o que representa mais uma economia, e quando nenhum programa as estiver usando elas são descartadas.

### 3.2. A linguagem de programação adotada

A linguagem de programação adotada foi C++ porque é uma linguagem que oferece recursos para a Programação Orientada a Objetos. Isso ajuda a elaborar projetos grandes e complexos. A manutenção de projetos orientados a objetos não exige grande esforço. É importante salientar que o próprio Windows foi projetado em C, sendo também, um dos motivos para a escolha dessa linguagem. O compilador disponível é o Borland C++ 5.0. Esse compilador oferece OWL 5.0 - *Object Windows Library 5.0* -, uma biblioteca orientada a objetos que simplifica e agiliza o desenvolvimento de aplicativos para Windows. Com essas ferramentas é possível compilar programas para o Windows 95. Ou seja, é possível desenvolver aplicativos de 32 bits. Isso significa que pode-se utilizar matrizes com dimensões bem maiores do que em aplicativos de 16 bits que são limitados a blocos de 64 k.



## 4 APRESENTAÇÃO DO APLICATIVO

O aplicativo intitulado de PEC - Projeto Estrutural por Computador - é dividido em módulos, são eles: PEC.EXE, DADOS.DLL, LISTAS.DLL, MALHA.DLL, GERADOR.DLL, EDITA.DLL e RELAT.DLL. O módulo principal é PEC.EXE, que é responsável pelo gerenciamento da janela principal. Para se ativar o aplicativo basta executar este módulo e os demais são executados automaticamente. Os outros módulos contém funções específicas, o módulo DADOS.DLL contém as classes ou objetos responsáveis pelos dados gerais da estrutura como nome do projeto, coeficiente de Poisson e módulo de Young; o módulo LISTAS.DLL é composto pelas classes responsáveis pela visualização e edição das listas de pontos nodais, elementos de placa e elementos de viga; o módulo MALHA.DLL possui as classes responsáveis pela entrada de dados para a geração automática das malhas de elementos finitos de placa e de elementos de viga; o módulo GERADOR.DLL possui as classes responsáveis pela geração automática da malha de elementos finitos de placa e de elementos de viga; o módulo EDITA.DLL contém as classes de edição dos dados diretamente na janela gráfica, essas funções são chamadas pelo menu flutuante que será detalhado mais tarde; e finalmente o módulo RELAT.DLL, as funções do interpretador, aquelas responsáveis pela geração dos relatórios de saída de dados.

## 4.1 Janela principal

Ao se executar o aplicativo é mostrada uma janela de apresentação (figura 11) e depois a janela principal é desenhada. A comunicação do aplicativo com o usuário é através de uma interface MDI - *Multiple Document Interface* -, ou seja, ele suporta trabalhar com mais de um arquivo de dados abertos ao mesmo tempo (figura 12). O usuário tem acesso aos

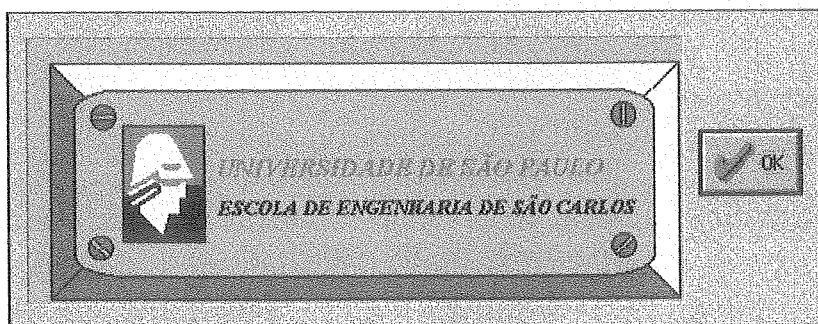


Figura 11 - Diálogo de abertura do aplicativo

comandos através de um menu localizado no topo da janela principal, um menu flutuante que é ativado quando se toca em uma janela com o botão direito do *mouse* e por duas barras de botões, uma horizontal no topo e outra vertical localizada do lado direito. Os significados dos ícones das barras de botões estão nas tabelas 1 e 2. Quando o usuário arrasta o cursor do *mouse* por cima dos ícones a sua descrição é apresentada na barra de *status* localizada na parte inferior da janela principal. Se o cursor do *mouse* ficar estacionado sobre um botão por um curto intervalo de tempo é apresentado logo abaixo do botão o nome do comando ativado por ele. Após o último botão da barra horizontal existe um espaço reservado para mostrar as coordenadas X e Y da malha quando o *mouse* passa por uma janela gráfica ou o total de linhas quando o *mouse* passa por uma janela texto. O diálogo "Sobre o PEC" é mostrado na figura 13.

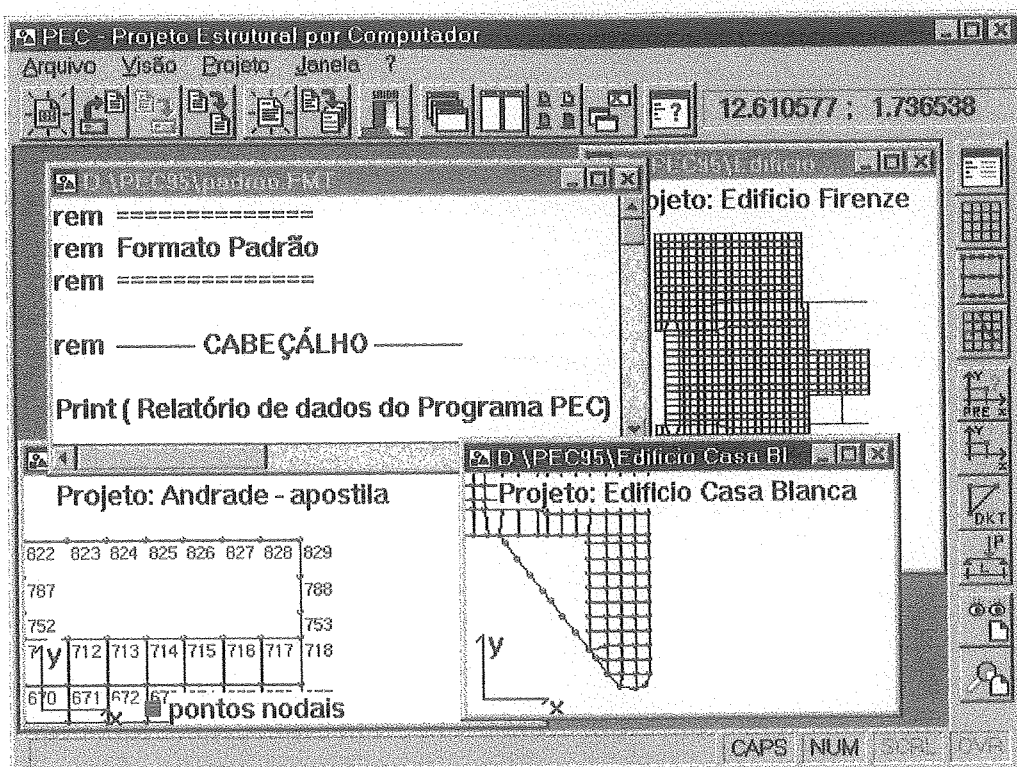


Figura 12 - Janela principal com vários projetos abertos

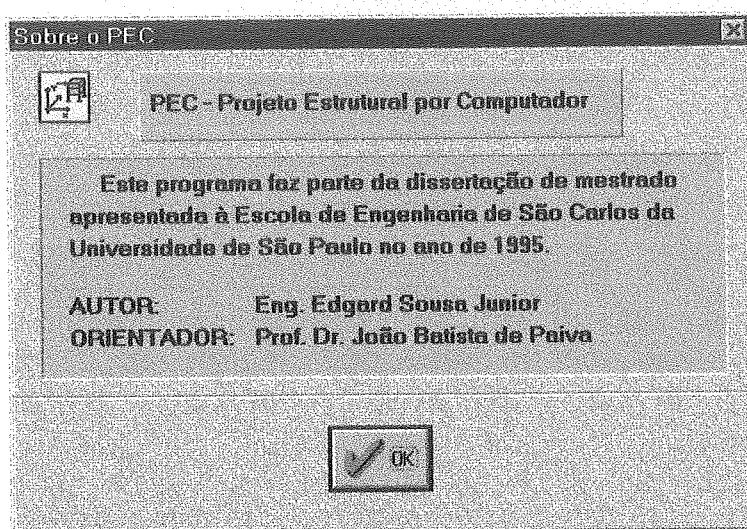


Figura 13 - Diálogo Sobre o PEC

O aplicativo trabalha com nomes de arquivos com até 255 caracteres, incluindo espaços e acentos. Através do diálogo Abrir é possível renomear, apagar e copiar arquivos. Basta tocar no nome do arquivo com o botão direito do *mouse* (figura 14). É possível abrir dois tipos de janelas, uma gráfica e outra de texto. Quando um arquivo é aberto a extensão do nome é

verificada, se for PEC o aplicativo verifica uma assinatura localizada no início do arquivo. Essa assinatura contém o nome do aplicativo e a sua versão. A seguir os dados do arquivo são lidos e se nenhum erro for encontrado é aberta uma janela gráfica e desenhada a malha. Caso seja outra extensão, é aberta uma janela texto.

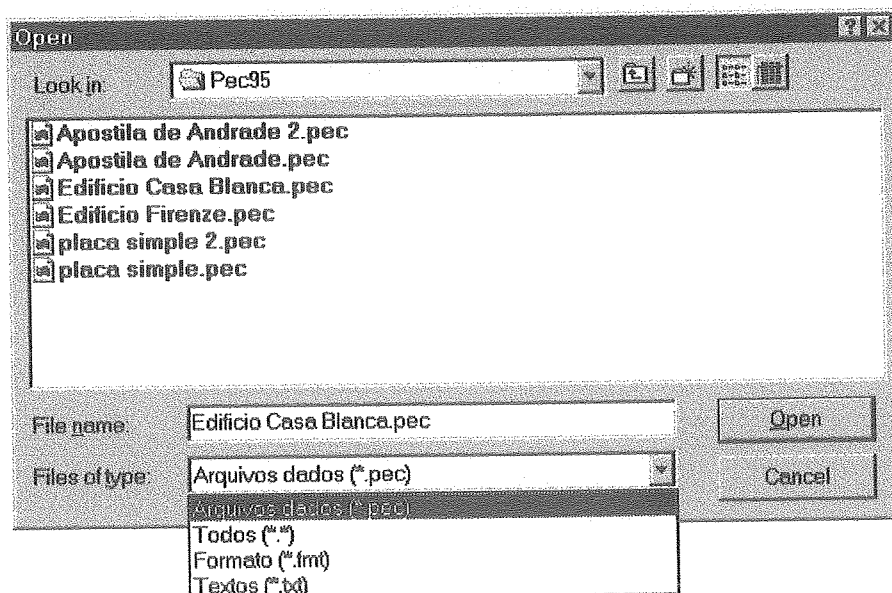


Figura 14 - Diálogo Abrir

Tabela 1 - Ícones da barra de botões horizontal

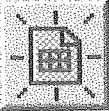
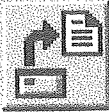
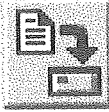
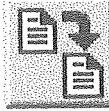
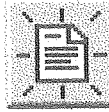
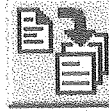

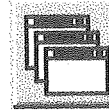
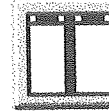
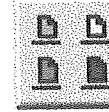


ÍCONE	COMANDO	DESCRIÇÃO NA BARRA DE STATUS
	Novo	Cria uma janela nova
	Abre	Abre uma janela com dados de um arquivo já existente
	Salva	Salva os dados de uma janela em um arquivo
	Salva como	Salva os dados de uma janela em um arquivo com novo nome
	Relatório novo	Cria uma janela de relatório nova
	Gerar relatório	Gera relatório de saída de dados
	Sair	Sai do aplicativo
	Cascata	Arruma as janelas abertas em cascata
	Lado a lado	Arruma as janelas abertas lado a lado
	Organiza ícones	Arruma os ícones na janela principal
	Fecha todas	Fecha todas as janelas abertas
	Informação	Controle de visão/informação

Tabela 2 - Ícones da barra de botões vertical

ÍCONE	COMANDO	DESCRIÇÃO NA BARRA DE <i>STATUS</i>
	Dados	Entrada dos dados básicos do projeto
	Gera malha	Gerador de elementos de placa
	Gera viga	Gerador de elementos de viga
	Poligonal de contorno	Definição da poligonal de contorno
	Pontos pré-definidos	Lista de pontos pré-definidos
	Lista de pontos nodais	Lista de pontos nodais (inclusão e alteração)
	Lista de elem. de placa	Lista de elementos de placa (inclusão, exclusão e alteração)
	Lista de elem. de viga	Lista de elementos de viga (inclusão, exclusão e alteração)
	Zoom total	Zoom total - mostra toda a estrutura
	Zoom janela	Zoom janela - visualiza uma parte definida por uma janela

## 4.2 Janela texto

Ao se abrir um arquivo do tipo texto os comandos de projeto são desativados e o menu da janela principal se modifica para o menu do editor de texto onde se encontram comandos básicos de edição como: Procurar, Trocar, Copiar, Colar e outros. O editor de texto é semelhante ao Bloco de Notas do Windows. Ele trabalha com textos sem formatação especial. A figura 15 mostra uma janela texto ativa em que foi tocada pelo botão direito do *mouse* e o menu flutuante do editor de texto foi ativado.

O editor de texto é usado para se criar e editar os arquivos de formato que são utilizados pelo interpretador na geração dos relatórios de dados.

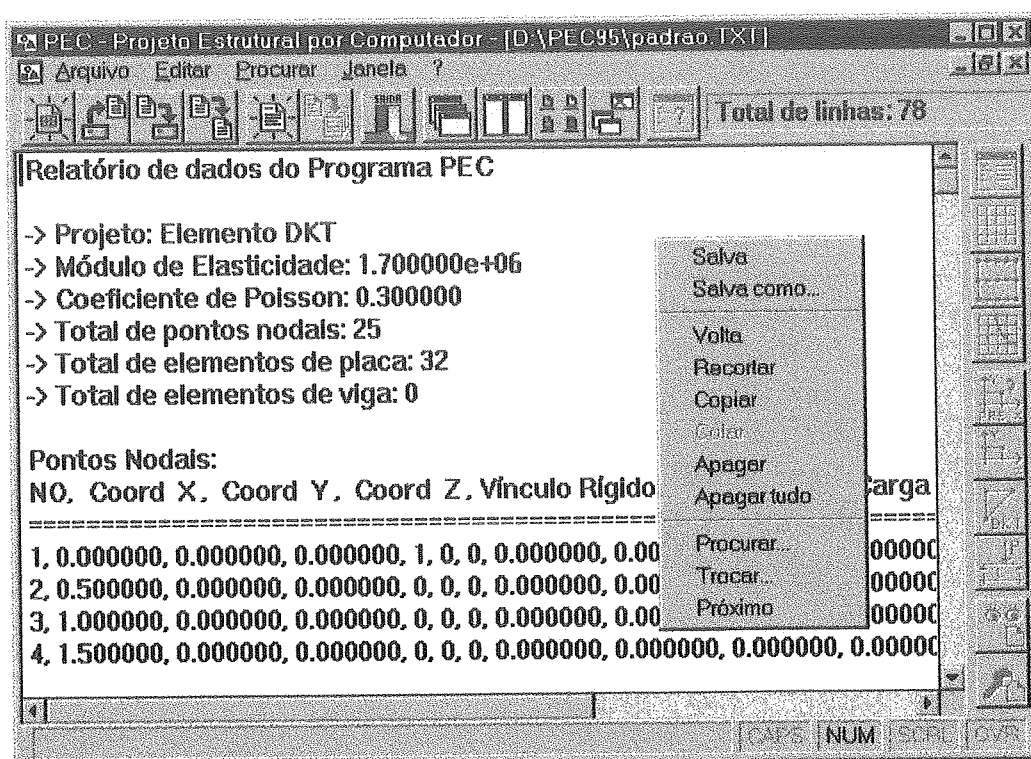


Figura 15 - Janela texto ativa

## 4.3 Janela gráfica

A janela gráfica serve para criar e editar uma malha de elementos finitos. Ao se tocar em uma janela gráfica os comandos de projeto tornam-se ativos. Quase todos os comandos de projeto estão na barra de botões vertical. Eles estão arrumados de tal forma que a seqüência de botões de cima para baixo na barra obedece à geração de uma malha, mas o usuário não é obrigado a seguir essa seqüência. Os comandos da janela gráfica que não estão na barra vertical e sim na barra horizontal são os de visão/informação e os de geração de relatório, pois estes são da categoria de comandos de manipulação de janelas. Para explicar os comandos de projeto será gerado uma malha simples chamada Exemplo 1. O primeiro passo é criar uma janela gráfica em branco com o comando Novo.

### 4.3.1 Dados Gerais

O comando **Dados** ativa a caixa de diálogo dos dados gerais (figura 16). Nela entra-se com o nome do projeto, módulo de Young e coeficiente

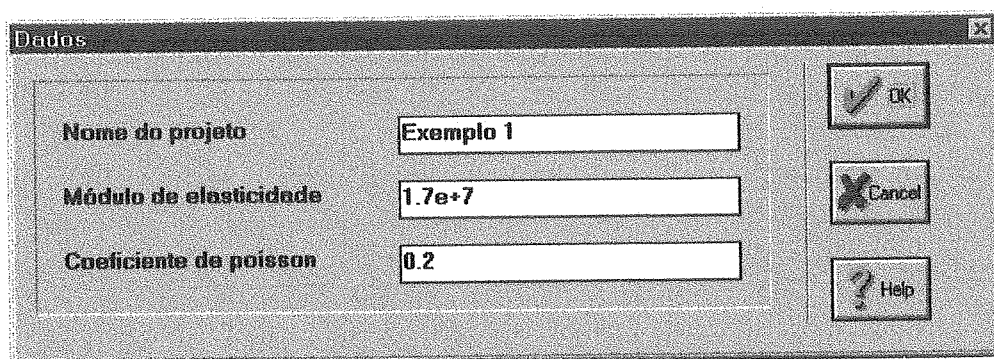


Figura 16 - Caixa de diálogo Dados



de Poisson. O nome do projeto é impresso na parte de cima da janela gráfica.

### 4.3.2 Poligonal de contorno

A caixa de diálogo **Poligonal de contorno** é uma lista que armazena poligonais (figura 17). Pode-se desenhar a planta baixa da edificação com estas poligonais para servir de orientação na geração da malha de elementos finitos. É preciso informar o número da poligonal, o número da linha e as coordenadas X e Y do ponto inicial e final da linha. Para que uma linha seja inserida na lista de poligonais, após os campos serem preenchidos, deve-se usar o botão **Add** ou teclar-se **ENTER**. Ao se fazer isto a linha passa a fazer parte da lista, o campo do número da poligonal

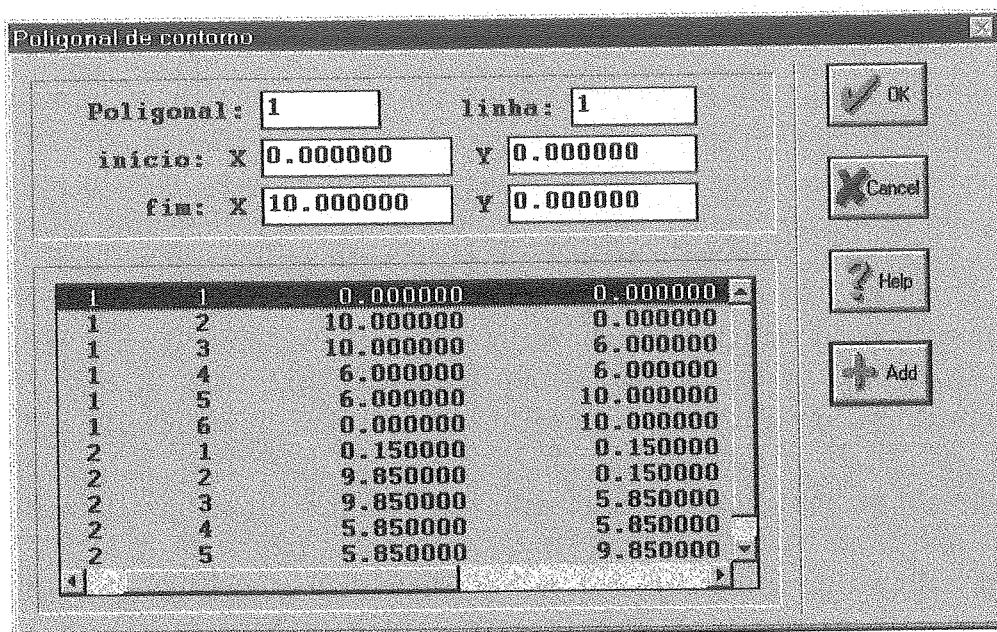


Figura 17 - Caixa de diálogo Poligonais de contorno

permanece igual, o do número da linha é incrementado de um e os das coordenadas são limpas a espera de uma nova linha. Para se tirar uma

linha da lista basta tocar com o *mouse* duas vezes na lista em cima dos dados da linha desejada e estes são automaticamente transferidos para os campos, onde podem ser editados e com o botão **Add**, se for o caso, podem voltar a fazer parte da lista. A lista de poligonais é provida de duas barras de rolamento, uma na vertical e outra na horizontal para que se possa visualizar os dados. Na figura 18 é mostrado o desenho do **Exemplo1** feito com as duas poligonais declaradas na figura 17.

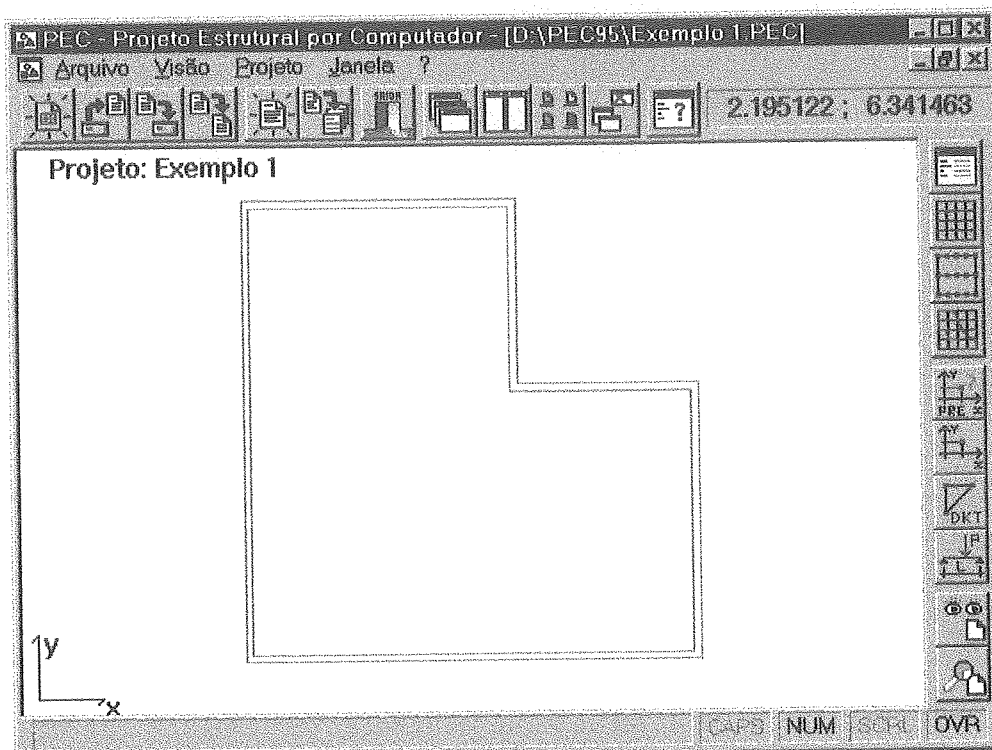


Figura 18 - Exemplo de uma planta feita com poligonais

### 4.3.3 Pontos pré-definidos

O comando **Pontos pré-definidos** ativa a caixa de diálogo **Lista de pontos pré-definidos**. O funcionamento desta caixa é semelhante ao do diálogo **Poligonal de contorno**. Antes de gerar a malha de elementos finitos o projetista pode definir alguns pontos nodais nesta lista.

Normalmente, são posições onde o projetista deseja que exista um nó, como por exemplo, no centro de uma laje ou de uma viga onde se queira saber os esforços e os deslocamentos ou ainda a localização de um pilar ou de pontos com carregamento aplicado.

Após os pontos nodais da malha serem gerados automaticamente pelo gerador de malha que será demonstrado no capítulo 4.3.4, o programa perguntará se o usuário deseja transformar os pontos que estão na lista de pontos pré-definidos em pontos nodais. Se a resposta do usuário for positiva, então para cada ponto pré-definido é identificado o nó que está mais próximo e este é substituído pelo ponto pré-definido, mantendo apenas a sua numeração. A numeração dos pontos pré-definidos existe apenas por uma questão de organização.

O uso desta lista é optativo. Ela pode ser preenchida antes ou depois da geração de uma malha. Sempre que se inserir um novo ponto nesta lista e se já existir algum nó, será perguntado se o usuário deseja converter os pontos da lista em nós. Um ponto nodal criado por ela possui todas as características de um ponto criado pelo gerador de malha ou pelo diálogo **Lista de pontos nodais** que será visto no capítulo 4.3.6. Os dados exigidos para cada nó são: (figura 19) os vínculos rígidos  $W$ ,  $W_x$  e  $W_y$  que são respectivamente o deslocamento na direção  $Z$ , a rotação em  $X$  e rotação em  $Y$  que podem ter o valor 1 para o caso de impedido ou 0 (zero) para livre; as coordenadas  $X$ ,  $Y$  e  $Z$ ; a carga  $P$  que é a carga vertical,  $M_x$  o momento em torno do eixo  $Y$  e  $M_y$  o momento em torno do eixo  $X$ ; e os vínculos elásticos  $K_1$ ,  $K_2$  e  $K_3$  que são os valores das constantes de mola. A figura 20 mostra os pontos pré-definidos do **Exemplo 1** que foram declarados na figura 19.

Lista de pontos pré-definidos

Nº:  Vinculos rígidos W:  W,x:  W,y:

Coord. X:  Y:  Z:

Carga P:  Mx:  My:

elást. K1:  K2:  K3:

1	1	0	0	0.000000	0.000000
2	1	0	0	10.000000	0.000000
3	1	0	0	10.000000	6.000000
4	1	0	0	6.000000	6.000000
5	1	0	0	6.000000	10.000000
6	1	0	0	0.000000	10.000000
7	1	0	0	3.500000	3.500000

OK  
Help  
Add

Figura 19 - Lista de pontos pré-definidos

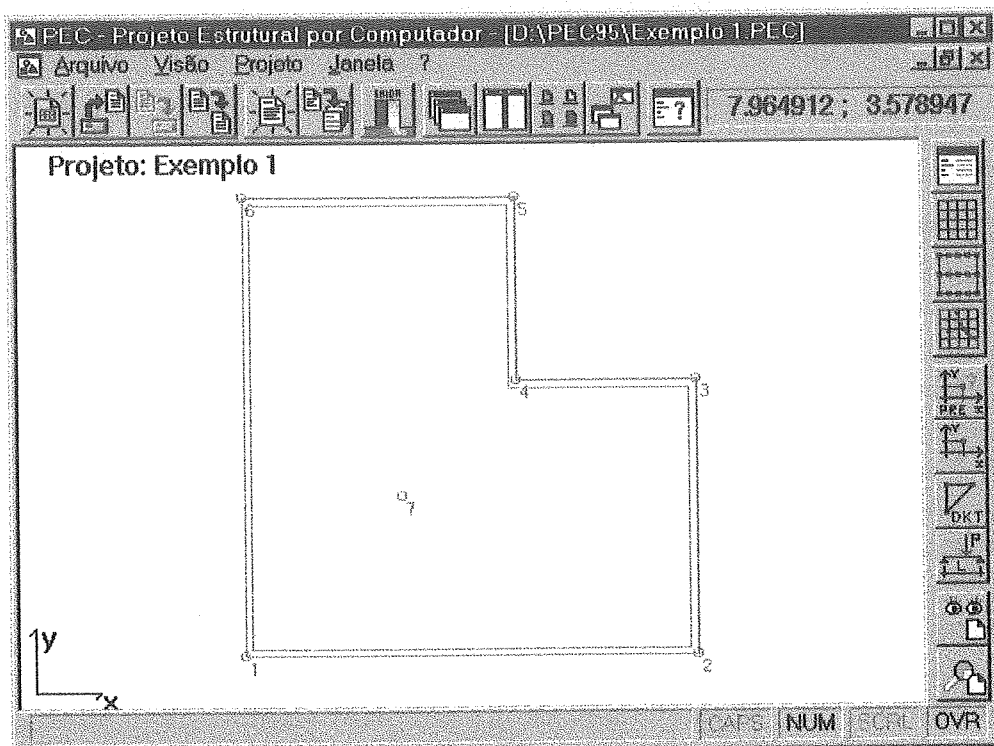


Figura 20 - Representação gráfica dos pontos pré-definidos

### 4.3.4 Gerador de malha

Na utilização do gerador de malhas, a estrutura pode ser dividida em um ou mais macro elementos, ou seja, o usuário pode chamar o comando **Malha**, o responsável pela geração da malha, mais de uma vez formando várias malhas quadrilaterais e o programa faz a junção das malhas automaticamente. Esse comando abre uma caixa de diálogo onde é preciso informar o tipo de elemento, quantidade de elementos na horizontal e vertical, espessura dos elementos, carregamento distribuído nos elementos e as coordenadas dos quatro cantos do macro elemento (figura 21). Após a entrada desses dados é ativada a janela do gerador (figura 22), na qual é gerada uma malha quadrilateral, não obrigatoriamente com os ângulos retos nos vértices (figura 23).

**Define Malha**

Formato dos elementos:

- Quadrilátero simples
- Quadrilátero com 2 triângulos
- Quadrilátero com 2 triângulos (2ª forma)

Quantidade de elementos na horizontal e vertical:

Horizontal:  (C3 to C4)

Vertical:  (C1 to C2)

	Coord. X	Coord. Y
C1	<input type="text" value="0.0"/>	<input type="text" value="0.0"/>
C2	<input type="text" value="10.0"/>	<input type="text" value="2.0"/>
C3	<input type="text" value="0.0"/>	<input type="text" value="15.0"/>
C4	<input type="text" value="0.0"/>	<input type="text" value="10.0"/>

Espessura:

Carga nos elementos:

Buttons: OK, Cancel, Help

Figura 21 - Caixa de diálogo Define Malha

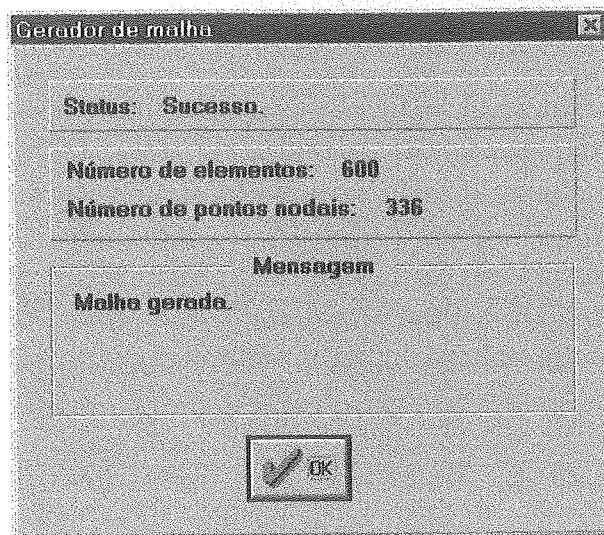


Figura 22 - O gerador de malha

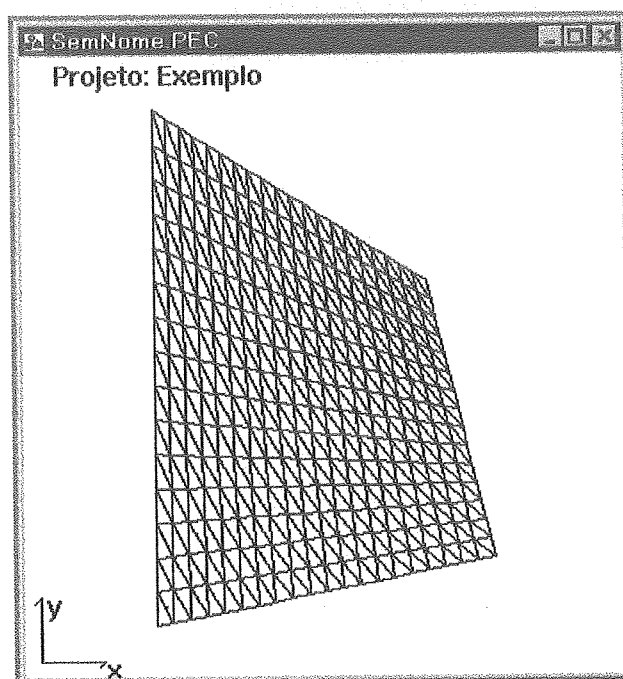


Figura 23 - Malha gerada com os dados da figura 21

As etapas do Gerador de Malhas são as seguintes:

- a) São gerados quatro pontos nodais localizados nos vértices do macro elemento:  $C1$ ,  $C2$ ,  $C3$  e  $C4$ .
- b) Os pontos nodais restantes são gerados. Sabendo-se o número de



elementos na horizontal e na vertical é possível calcular os espaçamentos entre os pontos nodais. Para cada linha de pontos é calculado um novo espaçamento horizontal. Para cada ponto nodal é calculado o seu espaçamento vertical.

- c) Tendo já calculado todos os pontos nodais necessários são gerados os elementos de placa de acordo com o tipo especificado pelo projetista.
- d) É feita a junção das malhas. Caso não tenha sido gerado nenhum ponto nodal antes da geração da malha que está sendo gerada esta etapa não é efetuada. É verificado na matriz de pontos nodais aqueles que possuem coordenadas iguais. Caso sejam encontrados nós com esta condição, os elementos de placa que possuírem estes nós são identificados e os nós em questão são substituídos pelo primeiro nó encontrado fazendo com que todos os elementos fiquem direcionados a um só dos pontos nodais que possuem coordenadas iguais. Depois os outros pontos nodais são eliminados.
- e) É verificado se existe algum ponto na matriz Pontos Pré-definidos. Em caso positivo é perguntado ao usuário se ele deseja transformar os pontos pré-definidos em pontos nodais. Caso queira, as propriedades (coordenadas, vínculos, carregamentos) dos pontos nodais mais próximos de cada ponto pré-definido são substituídas pelas propriedades destes.

Para que ocorra a junção correta de duas malhas é preciso que o comprimento da lateral dos elementos em que serão efetuadas as junções sejam iguais. Se isso não for verificado poderá ocorrer a geração de um nó sobre uma das hastes de um elemento, o que não é desejado. É simples evitar esse tipo de erro para o caso de malhas que possuem a mesma dimensão da lateral em que será feita a junção, basta que o número de elementos do lado da junção seja igual nas duas malhas. Se as malhas possuírem laterais com tamanhos distintos, em certos casos é possível prever com facilidade se os nós dos elementos das duas malhas irão

coincidir. Por exemplo, se a segunda malha a ser gerada possuir o comprimento da lateral exatamente a metade do comprimento da lateral da malha gerada anteriormente, o número de elementos também deve ser a metade do número de elementos da lateral da primeira malha (figura 24).

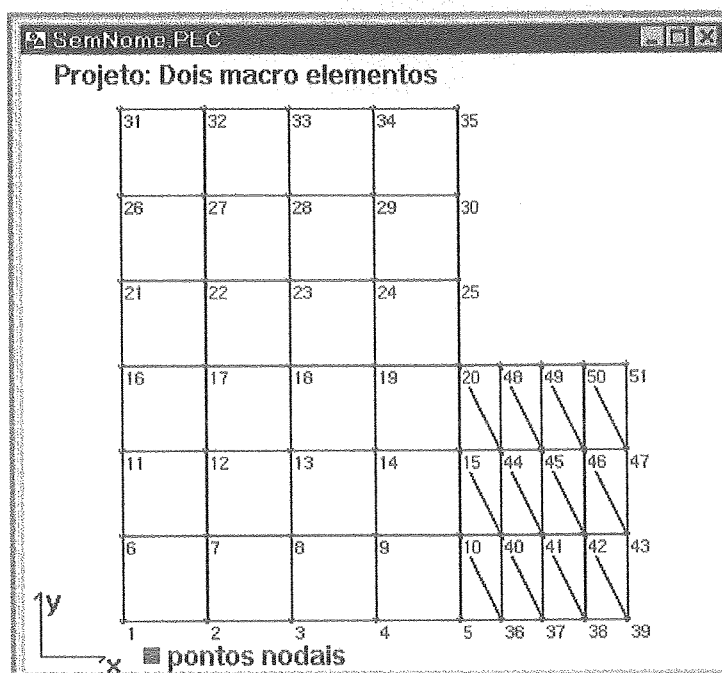


Figura 24 - Malha gerada por dois macro elementos sendo que um tem como altura a metade da altura do outro

Caso seja impossível definir elementos com os lados em que ocorrerá a junção do mesmo tamanho tem-se duas opções de geração. Como primeira opção pode-se gerar uma malha grande o suficiente para que englobe toda a área onde se deseja que exista malha. Depois define-se uma ou mais vigas (os comandos para gerar vigas serão explicados posteriormente no item 4.3.5) para que os pontos nodais se ajustem e fiquem na posição desejada pelo projetista. Pode-se então eliminar através do menu flutuante os elementos de viga (o menu flutuante será explicado no item 4.3.10). E também utilizando-se do menu flutuante pode-se eliminar os elementos de placa que não fazem parte do projeto (figura 25). A desvantagem dessa opção é que todos os elementos de placa serão do



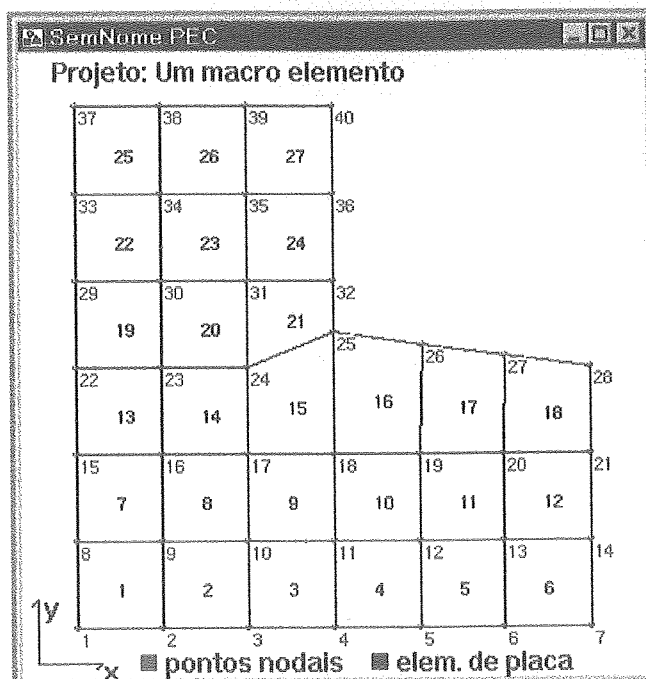


Figura 25 - Malha gerada por um macro elemento

mesmo tipo. Caso a análise exija elementos de tipos diferentes pode-se usar a segunda opção que é subdividir a malha final em tantos macro elementos o quanto forem necessários para que as laterais em que haverá junção sejam da mesma dimensão (figura 26).

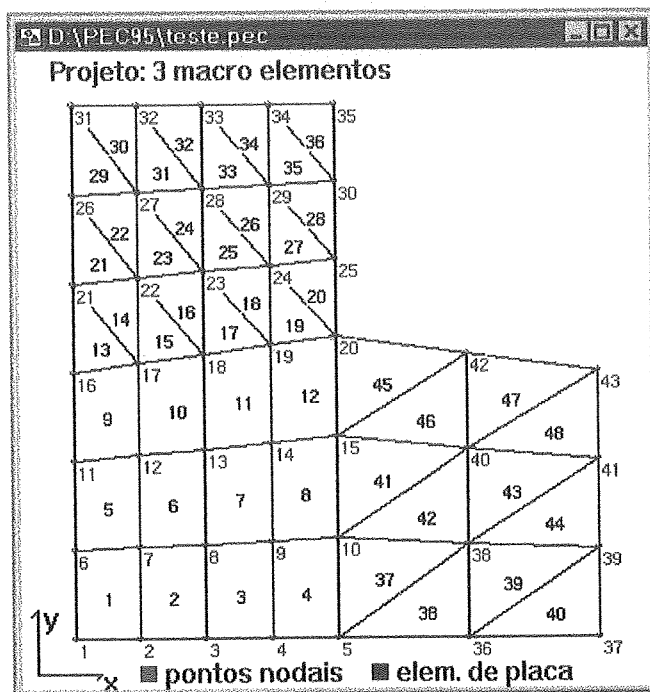


Figura 26 - Malha gerada por três macro elementos

A seguir tem-se os passos da geração da malha do Exemplo 1. Primeiramente com o comando **Malha** defini-se os dados da malha (figura 27).

**Define Malha**

Formato dos elementos:

- Quadrilátero simples
- Quadrilátero com 2 triângulos
- Quadrilátero com 2 triângulos (2ª forma)

Quantidade de elementos na horizontal e vertical:

8

8

C1 C2 C3 C4

	Coord. X	Coord. Y
C1	0	0
C2	10	0
C3	0	10
C4	10	10

Espessura: 1

Carga nos elementos: 50

OK

Cancel

Help

Figura 27 - Dados da malha do Exemplo 1

Após preencher os campos e apertar o botão **OK** surge o diálogo do gerador de malha (figura 28). Nota-se que neste ponto os nós *C1*, *C2*, *C3* e *C4* já foram gerados.

**Gerador de malha**

Status: Pronto para gerar.

Número de elementos: 0

Número de pontos nodais: 4

Mensagem

Tecle OK para gerar a malha.

OK

Figura 28 - Gerador de malha antes da malha ser gerada

Ao se pressionar o botão OK do gerador o campo **Status** passa a conter a mensagem "gerando..." e a proporção que os pontos nodais e os elementos de placa vão sendo gerados o valor dos campos **Número de elementos** e **Número de pontos nodais** vão se alterando e mostrando a quantidade de elementos e nós em tempo real de geração. No campo **Mensagem** é dito exatamente o que o gerador está fazendo a cada instante: gerando pontos nodais...; gerando elemento de placa...; juntando malhas... Se ocorrer algum erro durante a geração o campo **Status** se altera para "Erro na geração." e a mensagem de erro é mostrada no campo **Mensagem**. A figura 29 mostra a caixa de diálogo do gerador de malha após a malha do Exemplo 1 ter sido gerada.

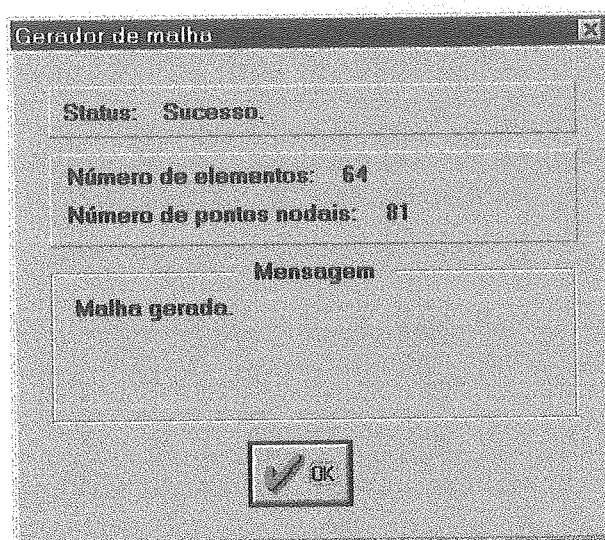


Figura 29 - Gerador de malha após a malha ter sido gerada

Como existem alguns pontos pré-definidos é feita a pergunta da figura 30. No caso do Exemplo 1 a resposta foi positiva.

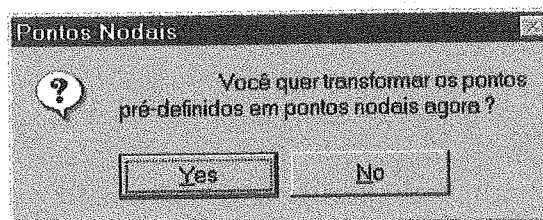


Figura 30 - Pergunta sobre os pontos pré-definidos

Na figura 31 tem-se a malha gerada. É bom observar que os pontos 1, 9, 31, 51, 54, 73 e 78 são os já extintos pontos pré-definidos. No item 4.3.5 que descreve o gerador de vigas será mostrado como ajustar os nós 52, 53, 60 e 69 para que coincidam com a posição das vigas. No item 4.3.10 que fala sobre o menu flutuante será mostrado como eliminar os elementos 46, 47, 48, 54, 55, 56, 62, 63 e 64.

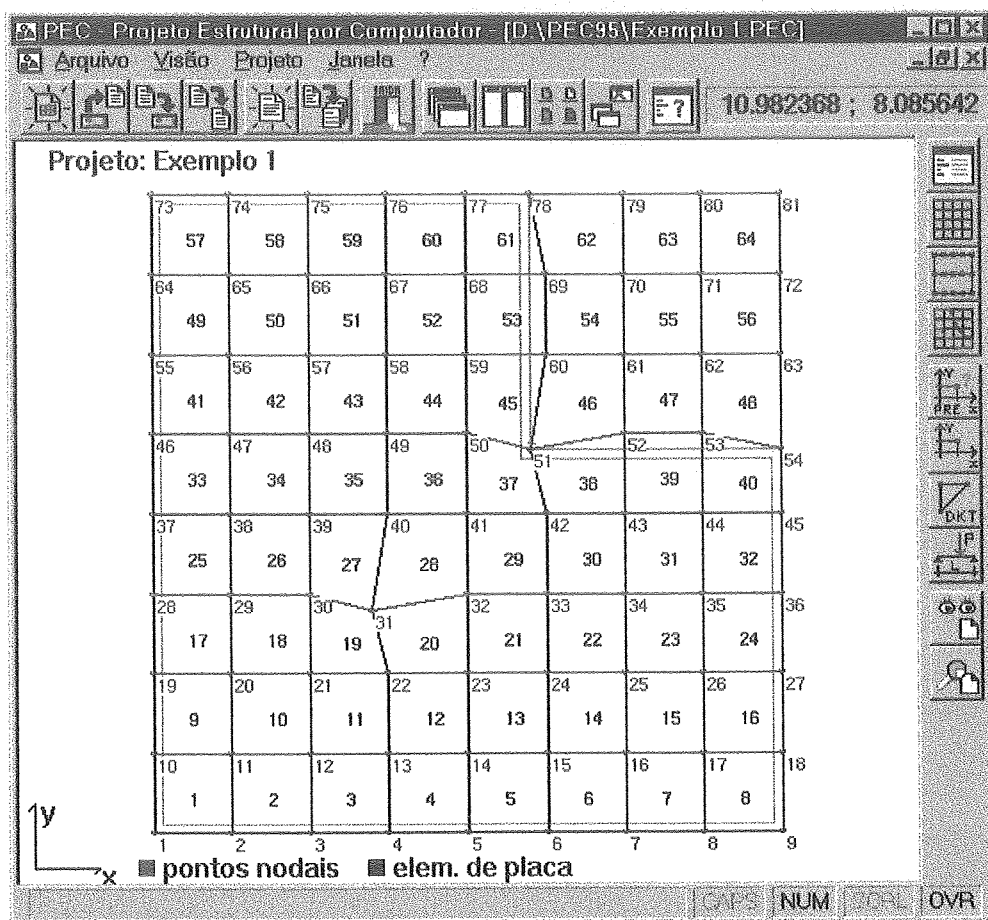


Figura 31 - Representação gráfica da Malha do Exemplo 1

### 4.3.5 Gerador de vigas

O Gerador de Vigas pode ser usado para definir vigas unitárias ou conjuntos de vigas. As vigas unitárias são definidas com a entrada das coordenadas de dois pontos, um como sendo o início e outro o fim da viga, As coordenadas dos pontos nodais que se encontram entre esses dois pontos são ajustadas para que coincidam com o eixo da viga. As vigas geradas em conjunto servem para gerar lajes nervuradas ou grelhas, dependendo se são definidas sobre elementos de placa ou não. Sua geração é similar a da viga unitária, sendo necessário primeiramente definir os dados da primeira viga e depois informar o espaçamento, ou passo, em X e em Y, que é a distância entre as vigas do conjunto, e ainda deve ser informado o número de vigas a serem geradas além daquela definida como sendo a primeira do conjunto. Na geração das vigas também devem ser informadas suas propriedades como área da seção transversal, momento de inércia a flexão e momento de inércia a torção e o carregamento distribuído (figura 32). As cargas concentradas devem ser definidas em pontos nodais.

Para que uma viga seja gerada é necessário primeiro que tenham sido definidos alguns pontos nodais, pois estes não são gerados pelo gerador de vigas e sim pelo gerador de elementos de placa, ou seja, o gerador de malha. Se for preciso gerar uma viga em um local que não terá elementos de placa, pode-se gerar uma malha provisória utilizando-se o gerador de malha e, após a viga ter sido gerada, os elementos de placa, assim como os pontos que não pertencerem aos elementos de viga, poderão ser facilmente removidos utilizando-se o *mouse* juntamente com o menu flutuante. Outra opção consiste em definir os pontos nodais que não fazem parte da malha existente e por onde deve passar a viga e depois definir os novos elementos da viga diretamente na **Lista de elementos de viga**. O número de elementos de viga a ser gerado depende da quantidade de pontos nodais existentes entre os pontos inicial e final da viga, ou seja,

depende da densidade da malha de elementos de placa sobre a qual são definidos os elementos de viga. O espaçamento entre os pontos nodais deve ser menor do que o espaçamento entre as vigas que forem geradas em conjunto, caso contrário, ao gerar uma das vigas secundárias os nós da viga gerada anteriormente podem ser deslocados e ocorrer um rearranjo não desejado dos elementos de viga.

	Coord. X	Coord. Y
C1	0.76	5.87
C2	5.48	8.98

C1 — C2	Área:	0.048
	Carga:	9.0
	I:	0.64E-3
	IT:	0

Número de repetições:	passo	
	X:	1.0
	Y:	-1.5

Buttons: OK, Cancel, Help

Figura 32 - Definição de vigas

Para gerar as vigas o primeiro passo do gerador de vigas é identificar o ponto nodal mais próximo do ponto *C1*, que é o ponto inicial da viga. Depois as coordenadas do ponto mais próximo são substituídas pelas de *C1*. O mesmo é feito para o ponto *C2*, que é o ponto final da viga. É importante salientar que os pontos *C1* e *C2* não são gerados e sim é feito um ajuste em pontos já existentes. Logo, devem existir pelo menos dois pontos antes de se gerar uma viga.

O segundo passo é ajustar os pontos nodais da malha entre os pontos *C1* e *C2* para que coincidam com o eixo da reta definida por esses dois pontos. O procedimento utilizado é o seguinte:

a) A inclinação  $m$  da reta que passa pelos pontos  $C1 (x_1, y_1)$  e  $C2 (x_2, y_2)$  (figura 33) é calculada pela expressão (1).

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (1)$$

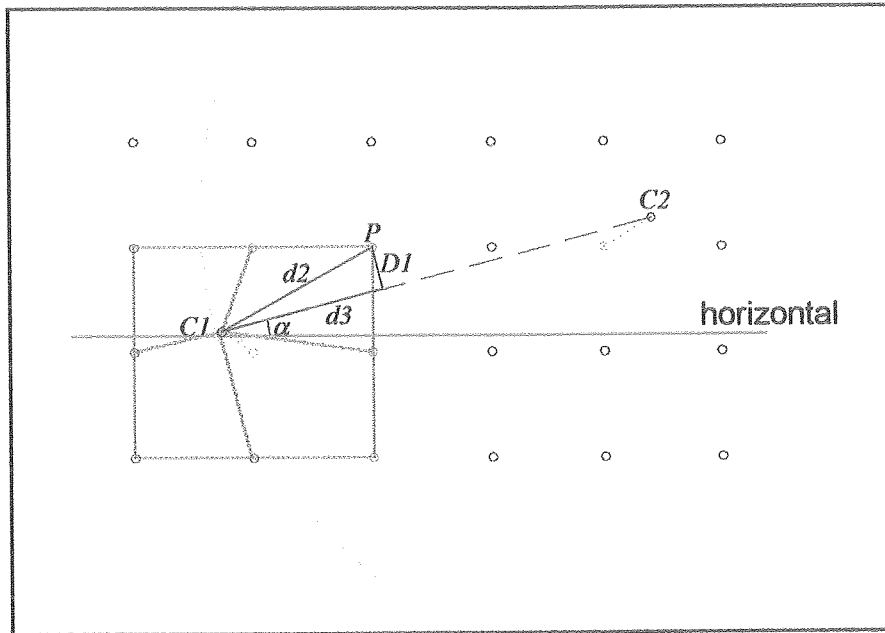


Figura 33 - Esquema representativo da geração de uma viga

b) O ângulo  $\alpha$  que a reta faz com a horizontal é obtido de (2).

$$\operatorname{tg} \alpha = m \quad (2)$$

c) Entre os pontos nodais pertencentes aos elementos de placa que convergem para o ponto  $C1$  são selecionados aqueles em que a distância a  $C2$  é menor que a distância de  $C1$  a  $C2$ . Dentre estes selecionados (os pontos de cor verde da figura 33) é escolhido o que estiver mais próximo da reta que passa por  $C1$  e  $C2$  e nomeado de ponto  $P$ . São utilizadas duas expressões de distância, a primeira (3) indica a distância  $d$  entre dois pontos  $P_1(x_{P1}, y_{P1})$  e  $P_2(x_{P2}, y_{P2})$  e a segunda (4), a distância  $D$  entre um ponto  $P(x_P, y_P)$  e uma reta  $Ax + By + C = 0$ . Estas expressões estão indicadas

a seguir:

$$d = \sqrt{(x_{p2} - x_{p1})^2 + (y_{p2} - y_{p1})^2} \quad (3)$$

$$D = \frac{A \cdot x_p + B \cdot y_p + C}{\pm \sqrt{A^2 + B^2}} \quad (4)$$

sendo que, os coeficientes  $A$ ,  $B$  e  $C$ , para a reta que passa por  $CI(x_1, y_1)$  e  $C2(x_2, y_2)$  (figura 33) são dados por:

$$A = y_2 - y_1 \quad (5)$$

$$B = x_1 - x_2 \quad (6)$$

$$C = (x_2 \cdot y_1) - (x_1 \cdot y_2) \quad (7)$$

d) Sendo  $D_1$  a distância entre  $P$  e a reta que passa por  $CI$  e  $C2$ ,  $d_2$  a distância entre os pontos  $P$  e  $CI$ , é calculado a distância  $d_3$  (figura 33).

$$d_3 = \sqrt{D_1^2 + d_2^2} \quad (8)$$

e) Para que o ponto  $P(x_p, y_p)$  seja movido para o eixo da reta que passa por  $CI$  e  $C2$  a sua coordenada  $X$  passa a ser a do ponto  $CI(x_1, y_1)$  mais a projeção de  $d_3$  no eixo  $X$  e a coordenada  $Y$  de  $P$  passa a ser a de  $CI$  mais a projeção de  $d_3$  no eixo  $Y$ , isto é:

$$x_p = x_1 + d_3 \cos \alpha \quad (9)$$

$$y_p = y_1 + d_3 \sen \alpha \quad (10)$$



- f) É então gerado um elemento de viga que inicia em  $C1$  e termina em  $P$ . Se o elemento de viga estiver passando sobre um elemento de placa, este é subdividido em dois elementos. O nó  $P$  passa a ser o novo ponto  $C1$  e o processo se repete para encontrar o novo ponto  $P$  e gerar o próximo elemento de viga. O processo vai se repetindo até que o ponto  $C1$  coincida com  $C2$ , neste ponto a viga já está gerada (figura 34). Se o procedimento escolhido for o de gerar vigas em conjunto, o valor do ponto  $C1$  inicial e o do ponto  $C2$  são incrementados com o valor do passo e o processo inteiro se repete para gerar a próxima viga.

A figura 35 mostra o resultado da geração dos dados da figura 32 sobre uma malha quadrada de dimensões 10 por 10 unidades de medida constituída por 100 elementos de placa.

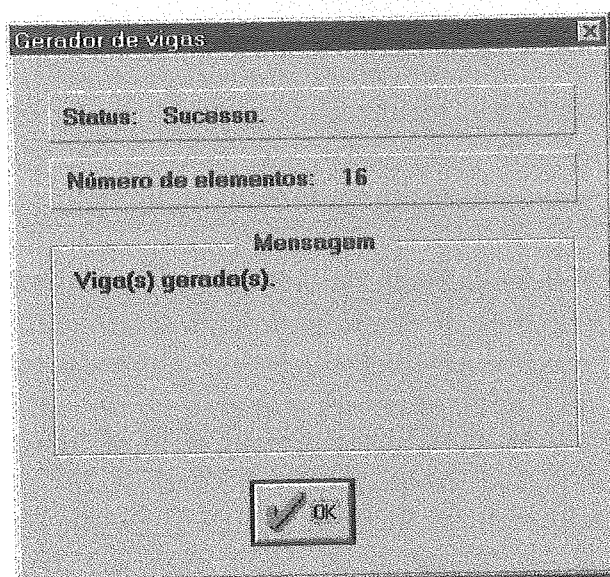


Figura 34 - Final da geração das vigas da figura 32

A caixa de diálogo do **Gerador de Vigas** (figura 32) é chamada para cada uma das seis vigas do **Exemplo 1** e seus campos são preenchidos com os dados da tabela 3. As vigas acompanham o contorno da estrutura. A figura 36 mostra o resultado da geração dos elementos de viga que estão numerados e de cor azul. Observar que os pontos 52, 53, 60 e 69 da figura 31, que estavam fora do eixo das vigas foram re-allocados sobre estas.

Tabela 3 - Dados para geração das vigas do Exemplo 1

C1		C2		Área	Carga	I	IT	Nº de repetições	passo	
Coord. X	Coord. Y	Coord. X	Coord. Y						X	Y
0.0	0.0	10.0	0.0	0.048	6.0	6.4e-4	1e-6	0	0	0
10.0	0.0	10.0	6.0	0.048	6.0	6.4e-4	1e-6	0	0	0
6.0	6.0	10.0	6.0	0.048	6.0	6.4e-4	1e-6	0	0	0
6.0	6.0	6.0	10.0	0.048	6.0	6.4e-4	1e-6	0	0	0
0.0	10.0	6.0	10.0	0.048	6.0	6.4e-4	1e-6	0	0	0
0.0	0.0	0.0	10.0	0.048	6.0	6.4e-4	1e-6	0	0	0

Este documento é propriedade intelectual de um dos membros do Grupo de Trabalho em Engenharia de Estruturas (GT-EE) do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). É permitida a reprodução parcial ou total deste documento sem fins comerciais, desde que seja dada a devida citação à fonte original.

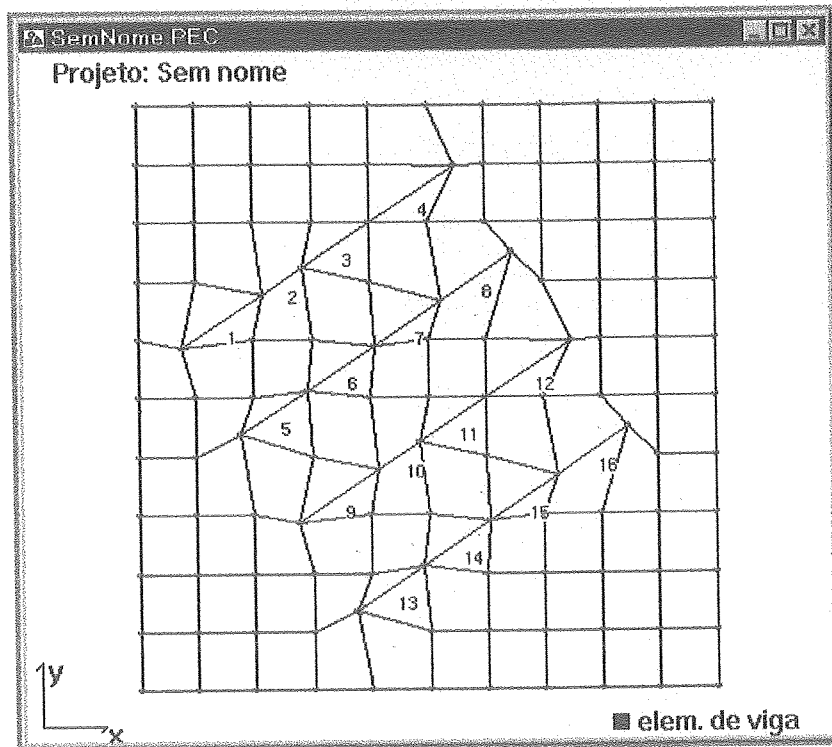


Figura 35 - Vigas em conjunto geradas com os dados da figura 32

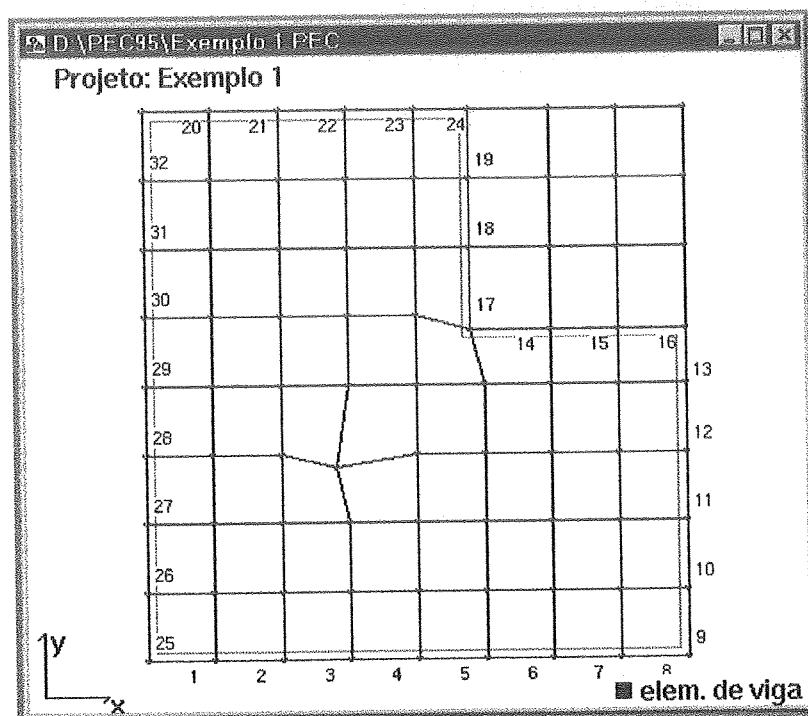


Figura 36 - Representação gráfica das vigas do Exemplo 1

### 4.3.6 Lista de pontos nodais

O funcionamento da Lista de Pontos Nodais é semelhante ao da Lista de pontos pré-definidos, a diferença é que aquela não permite eliminar um nó com um duplo toque do *mouse*. Isto porque a matriz de elementos de viga e a de elementos de placa estão ligadas diretamente à matriz de pontos nodais e se um nó for eliminado os elementos tanto de viga como de placa, que estiverem associados a este nó, devem ser eliminados automaticamente, caso contrário os dados ficam inconsistentes. Para que não se apague um nó e junto com ele outros elementos sem se perceber, é permitido apenas apagar um nó diretamente na tela gráfica, com o uso do menu flutuante e desta forma é possível observar a consequência da ação.

Os campos da caixa de diálogo Lista de Pontos Nodais (figura 37) são os mesmos da caixa Lista de Pontos Pré-definidos e já foram detalhados no capítulo 4.3.3.

Lista de pontos nodais

Nó:  Vinculos rígidos W:  W,x:  W,y:

Coord. X:  Y:  Z:

Carga P:  Mx:  My:

elást. K1:  K2:  K3:

1	1	0	0	0.000000	0.000000
2	0	0	0	1.250000	0.000000
3	0	0	0	2.500000	0.000000
4	0	0	0	3.750000	0.000000
5	0	0	0	5.000000	0.000000
6	0	0	0	6.250000	0.000000
7	0	0	0	7.500000	0.000000
8	0	0	0	8.750000	0.000000
9	1	0	0	10.000000	0.000000
10	0	0	0	0.000000	1.250000
11	0	0	0	1.250000	1.250000

Buttons: OK, Help, Add

Figura 37 - Lista de pontos nodais

### 4.3.7 Lista de elementos de placa

Os campos da lista de elementos de placa são (figura 38) :

- **número:** identifica o número do elemento;
- **tipo:** é um número inteiro que varia de 1 a 3 sendo que o número 1 indica um elemento quadrado e os números 2 ou 3, elementos triangulares, de diferentes orientações (ver figura 27);
- **pontos nodais:** são quatro campos que contém os nós que definem o elemento e quando for elemento triangular o ultimo campo é preenchido com zero;
- **espessura:** contém o valor da espessura do elemento;
- **carga:** a carga distribuída sobre o elemento.

Nesta lista é possível se fazer inclusão, alteração e exclusão de elementos de placa. Quando um novo elemento é gerado através desta janela, são feitas verificações de consistência de dados. O elemento é adicionado à lista através do comando **Add**. Os nós que definem os elementos devem obedecer uma seqüência anti-horária para evitar elementos com área negativa. Quando o usuário toca no botão **Add** a área do elemento é calculada. Caso seja negativa o usuário é informado e neste caso o elemento não é acrescentado na lista. Não são aceitos também elementos com espessura negativa.

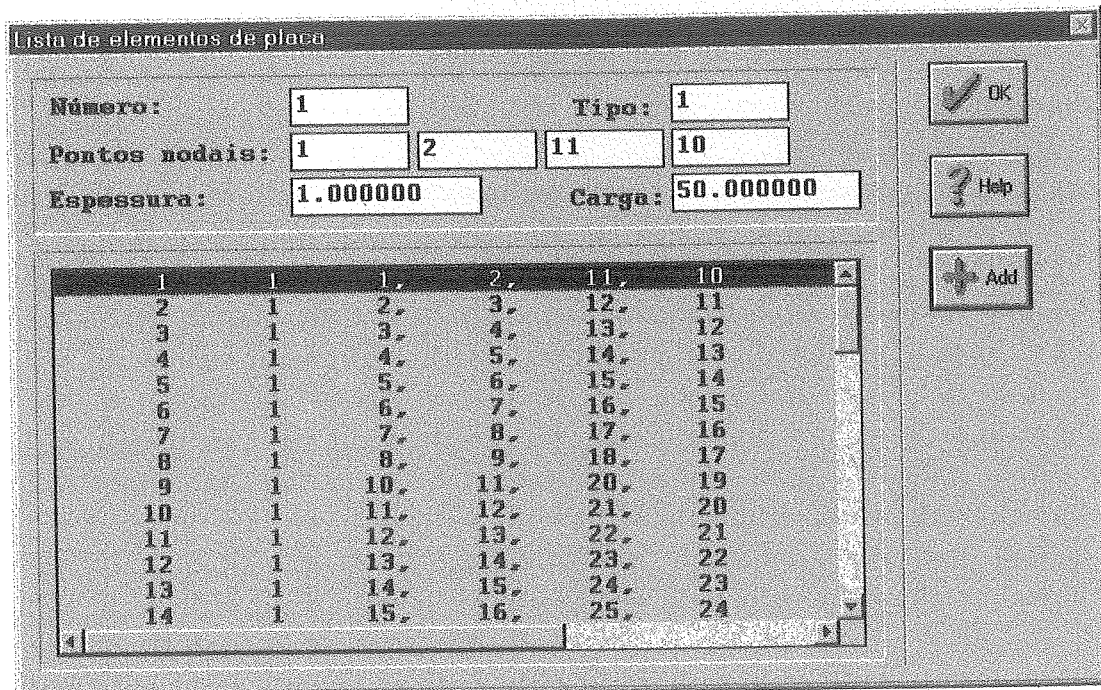


Figura 38 - Lista de elementos de placa

#### 4.3.8 Lista de elementos de viga

A Lista de Elementos de Viga permite que se tenha controle absoluto sobre a matriz de elementos de viga: inclusão, alteração e exclusão de elementos de viga. O seu funcionamento é igual ao das listas que já foram apresentadas. Os campos desta caixa de diálogo são (figura 39):

- Nº: o número do elemento de viga;
- nó inicial: número do ponto nodal que define o início do elemento de viga;

- final: número do ponto nodal que define o fim do elemento de viga;
- área: área da seção transversal do elemento;
- carga: carga distribuída sobre o elemento;
- I: inércia a flexão;
- $I_t$ : inércia a torção.

Lista de elementos de viga

Nº 1      Nó inicial: 1      final: 2

área: 0.048000      carga: 6.000000

I: 0.000640      It: 0.000001

Nº	Nó inicial	Nó final	área	carga
1	1	2	0.048000	6.000
2	2	3	0.048000	6.000
3	3	4	0.048000	6.000
4	4	5	0.048000	6.000
5	5	6	0.048000	6.000
6	6	7	0.048000	6.000
7	7	8	0.048000	6.000
8	8	9	0.048000	6.000
9	9	18	0.048000	6.000
10	18	27	0.048000	6.000
11	27	36	0.048000	6.000
12	36	45	0.048000	6.000

OK

? Help

+ Add

Figura 39 - Lista de elementos de viga



### 4.3.9 Diálogo visão/informação

Ativada pelo comando **Informação**, a caixa de diálogo **Visão / Informação** (figura 40) fornece o número de elementos de placa, o número de elementos de viga e o número de pontos nodais e ainda é nesta caixa que o usuário define o que deve ser visto na janela gráfica. É possível fazer qualquer combinação entre os campos da parte de baixo do diálogo. O campo que estiver marcado significa que está ativo e sua ação será executada na janela gráfica. O primeiro campo, **usar fonte pequeno**, está relacionado com o tamanho do caracter utilizado pelos quatro campos seguintes que indicam quais elementos devem ser numerados. Quando este campo estiver marcado será utilizado na numeração dos elementos o menor tamanho de fonte possível, quando não estiver marcado será utilizado o tamanho do fonte padrão que estiver definido no Windows. Os últimos cinco campos definem os elementos que devem ser desenhados na tela: os pontos nodais, os elementos de placa, os de viga, as poligonais de contorno e os pontos pré-definidos.

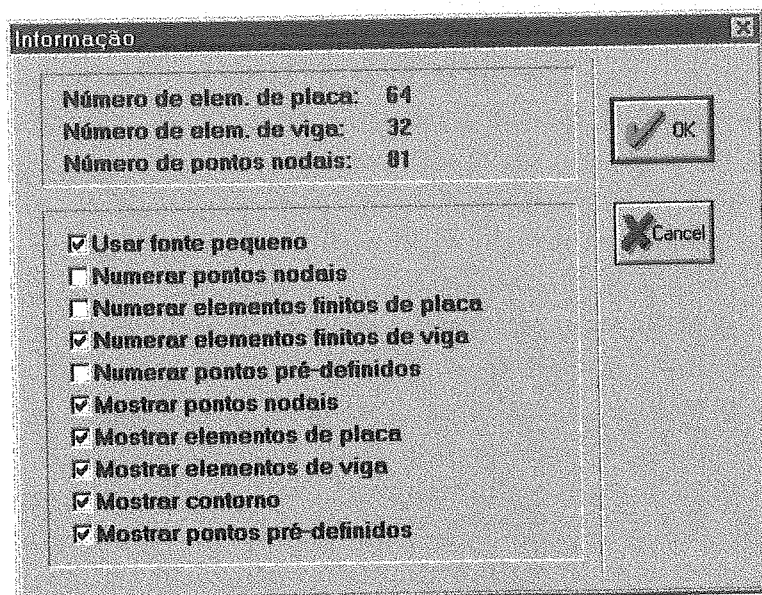


Figura 40 - Diálogo visão / informação



### 4.3.10 Menu flutuante

Quando se toca em uma janela gráfica com o botão direito do *mouse* é ativado o menu flutuante da janela gráfica. Este menu é dividido em cinco partes (figura 41): a primeira contém apenas o comando **Salva**; a segunda os comandos **Zoom total** e **Zoom janela**. Estes três comandos podem ser ativados também pelo menu principal ou pela barra de botões. Os demais comandos são exclusivos do menu flutuante. O terceiro grupo de comandos é o de edição, estes comandos servem para selecionar um ou mais elementos. Após escolhidos os elementos podem ser removidos da estrutura com o comando **Apaga selecionados** ou editados com o comando **Edita selecionados**. Este comando ativa as janelas de edição dependendo

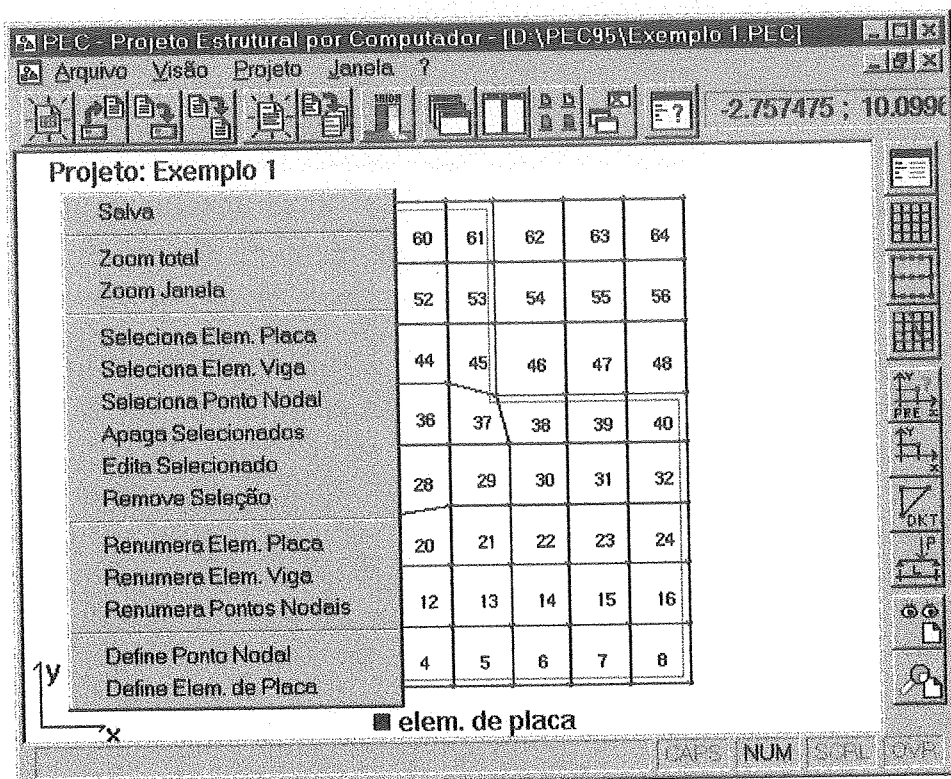


Figura 41 - Menu flutuante da janela gráfica

dos elementos que estiverem selecionados. Se houver algum nó selecionado a janela de edição de pontos nodais é ativada (figura 42), após esta, se houverem elementos de placa selecionados a janela de edição de elementos de placa é ativada (figura 43) e por último se houver algum elemento de viga selecionado a janela de edição de elementos de viga é ativada (figura 44). O último comando desta seção é o comando **Remove seleção** que serve para desmarcar os elementos selecionados anteriormente. A figura 45 mostra como exemplo alguns elementos sendo selecionados. Quando um elemento de placa ou de viga é selecionado ele fica da cor vermelha, quando um ponto nodal é selecionado a sua cor é alterada para verde. No caso da figura 45 os elementos 10 e 11 de viga estão selecionados, os pontos nodais 23, 24, 25 também e os elementos de placa 46, 47, 48, 54, 55, 56, 62, 63 e 64 estão sendo selecionados. Como estas janelas servem para editar grupos de elementos não é preciso preencher todos os campos mas, somente os campos preenchidos que provocam mudanças nos elementos. Por exemplo, se forem selecionados vários pontos nodais e o usuário for editá-los preenchendo somente o campo **Carga P** (figura 42), os demais campos dos pontos nodais não são alterados, unicamente o campo **Carga P** dos pontos nodais selecionados é alterado para o valor que o usuário digitou.

Nós		Vínculos rígidos		
		W: <input type="checkbox"/>	W.x: <input type="checkbox"/>	W.y: <input type="checkbox"/>
elást	K1: <input type="text"/>	K2: <input type="text"/>	K3: <input type="text"/>	
Carga	P: <input type="text"/>	Mx: <input type="text"/>	My: <input type="text"/>	

Figura 42 - Edição de pontos nodais

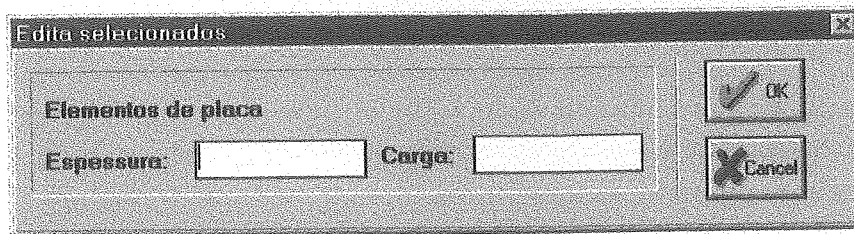


Figura 43 - Edição de elementos de placa

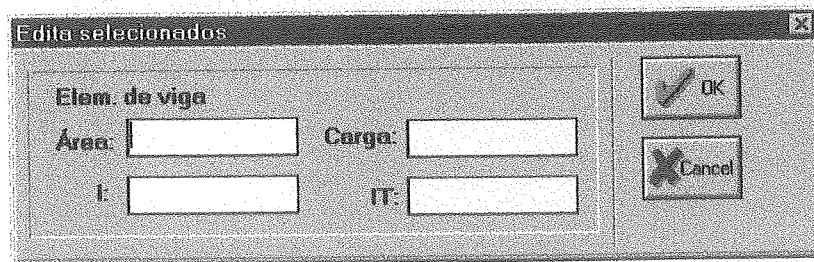


Figura 44 - Edição de elementos de viga

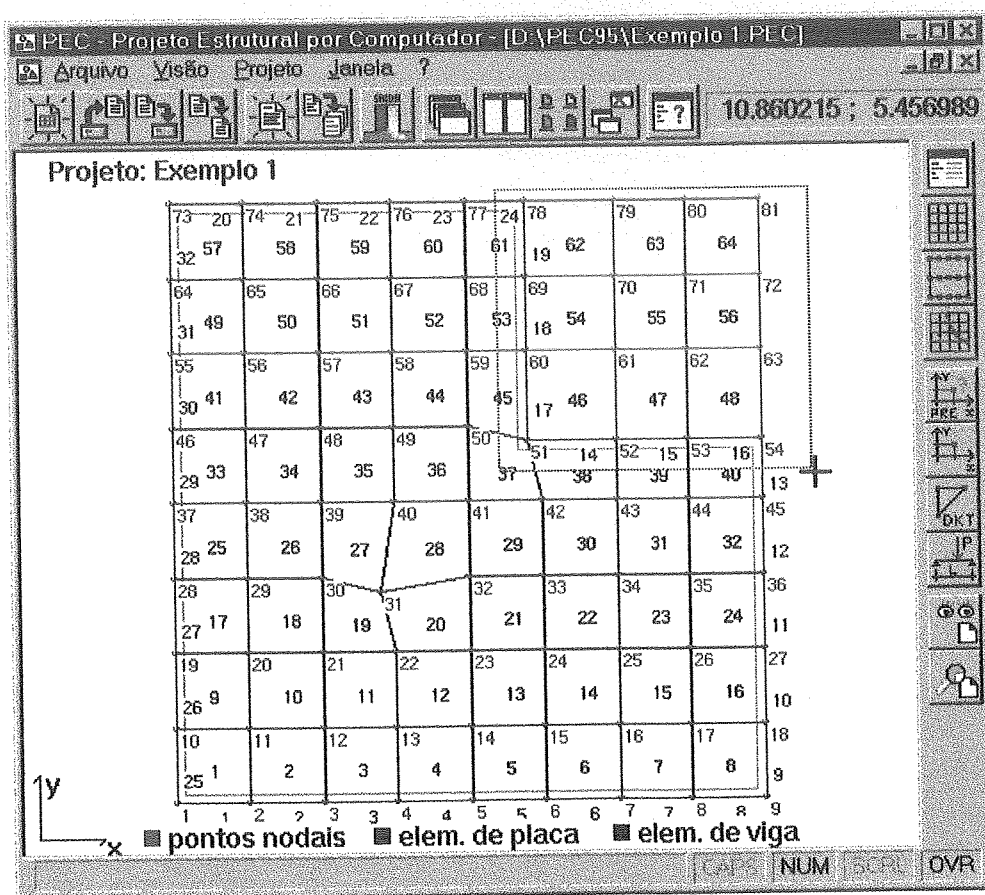


Figura 45 - Seleção de alguns elementos

O penúltimo grupo de comandos do menu flutuante serve para se fazer uma renumeração na matriz de elementos de placa, de viga ou de pontos nodais. Isto é necessário quando algum elemento é apagado. O último grupo de comandos serve para se criar um ponto nodal com o comando **Define ponto nodal** (figura 46) ou um elemento de placa com o comando **Define elemento de placa** (figura 47). Estas caixas de diálogo servem também para se editar um elemento. Por exemplo, se o usuário quiser alterar a coordenada X de um determinado ponto nodal ele pode utilizar o comando **Define ponto nodal**, a caixa de diálogo **Definição de ponto nodal** é acionada, o usuário deve marcar o campo **Atualizar ponto já existente** (figura 46), entrar com o número do ponto que deseja alterar e finalmente entrar apenas com o novo valor da coordenada X e não preencher mais nenhum campo. Se o usuário desejar criar um ponto nodal ele deve marcar o campo **Inserir novo ponto** e preencher os outros campos que definem o ponto nodal. Não é preciso preencher o campo **Nó**, que define o número do novo ponto, porque este número é gerado, automaticamente, de acordo com as coordenadas do ponto nodal que está sendo definido de tal forma que essa numeração provoque o menor aumento na largura de banda da matriz de rigidez da estrutura.

Definição de ponto nodal

Atualizar ponto já existente       Inserir novo ponto

Nó:       Vínculos rígidos    W:     W<sub>x</sub>:     W<sub>y</sub>:

Coord. X:     Y:     Z:

Carga P:     M<sub>c</sub>:     M<sub>y</sub>:

elást. K1:     K2:     K3:

Figura 46 - Define ponto nodal

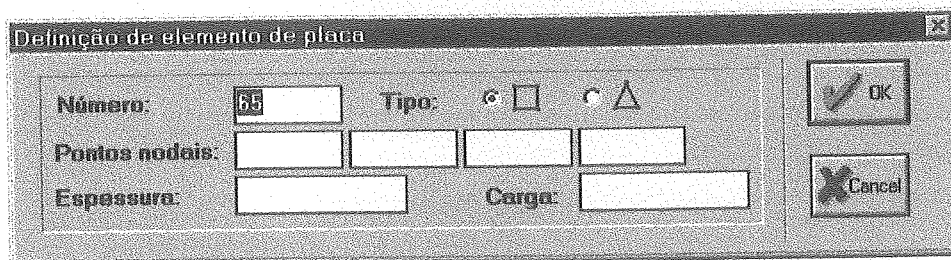


Figura 47 - Define elemento de placa

A caixa de diálogo **Definição de elemento de placa** (figura 47) também faz as mesmas verificações feitas na **Lista de elementos de placa** quando um elemento é gerado: verifica se a área e/ou a espessura do elemento são negativas e ainda se os pontos nodais que definem o elemento existem.

Para demonstrar a versatilidade do menu flutuante é exposto a seguir um exemplo de alteração de malha. Considera-se uma malha quadrada composta de dezesseis elementos de placa também quadrados. Se o usuário quiser acrescentar uma carga pontual sobre o elemento de placa de número dez será necessário que se tenha um ponto nodal exatamente no local do carregamento. A solução mais simples para o problema seria arrastar para o local do carregamento o ponto que estiver mais próximo. Mas, em certos casos, isso pode ocasionar a formação de elementos muito desproporcionais, o que pode produzir resultados não muito confiáveis. Uma outra solução seria eliminar o elemento de placa de número dez, criar um ponto nodal no local do carregamento e por último preencher o local vago pelo elemento de placa de número dez com novos elementos triangulares, por exemplo. Na figura 48 tem-se a malha descrita acima onde está sendo utilizado o comando **Seleciona elem. de placa** para que o elemento dez seja selecionado. Após isso deve-se utilizar o comando **Apaga selecionados** para se eliminar o elemento dez. Observar que na numeração de elementos não indica mais este elemento. Com o elemento dez apagado o usuário pode mover o cursor do *mouse* para a posição onde deve ser criado o ponto nodal (figura 49). Estando o *mouse* posicionado

pode-se ativar o menu flutuante com o botão direito do *mouse*, com isso as coordenadas do ponto a ser adicionado ficam gravadas na barra de botões horizontal. O usuário poderá observar e utilizar esses valores das coordenadas na barra horizontal mais tarde na definição do ponto nodal.

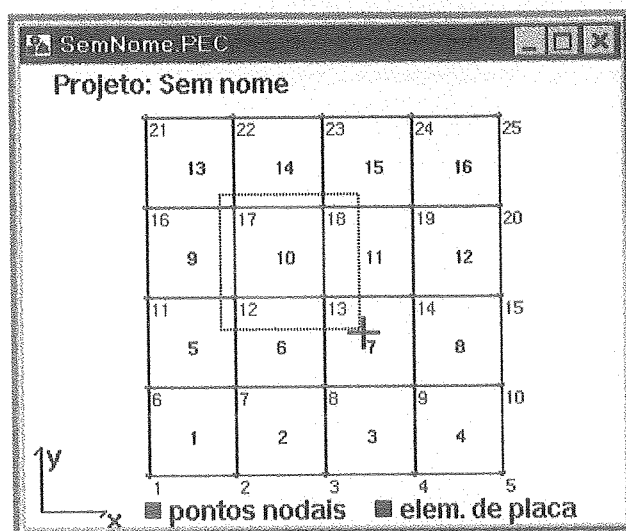


Figura 48 - Retirando-se um elemento da malha

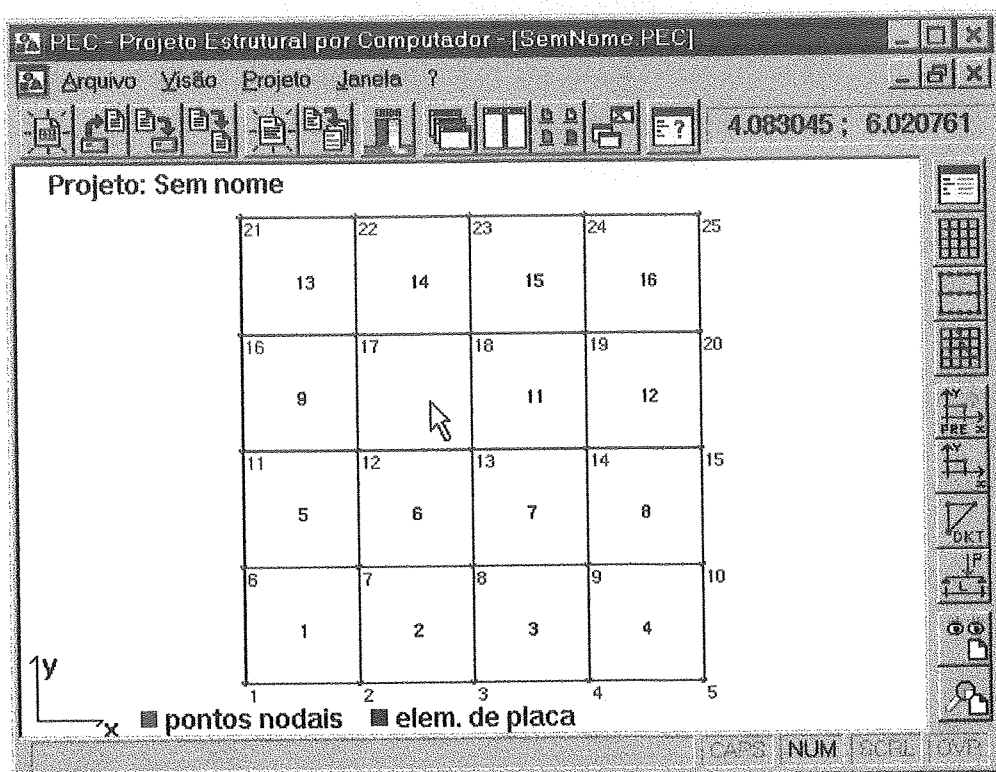


Figura 49 - Localização do ponto a ser gerado



O comando **Define ponto nodal** do menu flutuante deve então ser selecionado. A caixa de diálogo de definição de um ponto nodal irá aparecer e deverá ser preenchida como mostrado no figura 50. Na figura 51 tem-se o resultado da geração do ponto nodal. Nota-se que o número do novo nó foi gerado automaticamente e ele recebeu o número 14.

Figura 50 - Definição de um novo ponto nodal

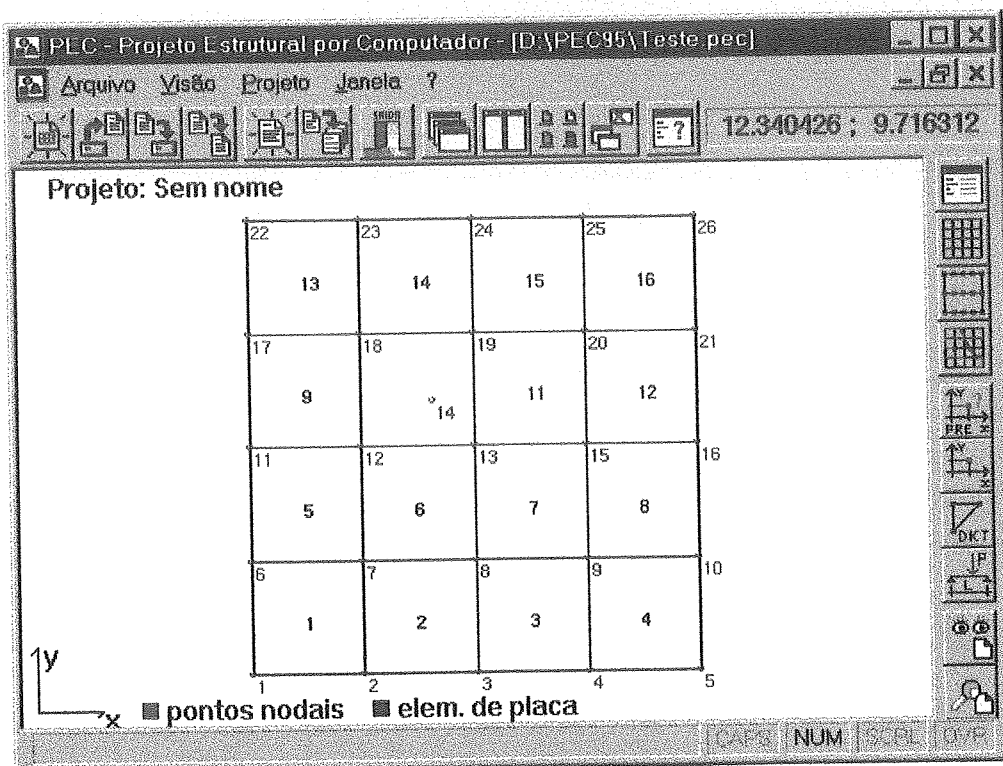


Figura 51 - Resultado da geração do ponto nodal

Após o ponto ter sido gerado os elementos de placa triangulares devem ser gerados através do comando **Define elemento de placa** do menu flutuante. As figuras de número 52 a 55 demonstram essa etapa e a figura 56 mostra os elementos gerados. Na figura 57 tem-se o resultado final onde já foi utilizado o comando de renumeração dos elementos de placa.

Definição de elemento de placa

Número: 17 Tipo:

Pontos nodais: 13 14 12

Espessura: 0.7 Carga: 1.0

OK Cancel

Figura 52 - Geração do primeiro elemento triangular

Definição de elemento de placa

Número: 18 Tipo:

Pontos nodais: 13 20 14

Espessura: 0.7 Carga: 1.0

OK Cancel

Figura 53 - Geração do segundo elemento triangular

Definição de elemento de placa

Número: 19 Tipo:

Pontos nodais: 14 20 19

Espessura: 0.7 Carga: 1.0

OK Cancel

Figura 54 - Geração do terceiro elemento triangular

Definição de elemento de placa

Número: 20 Tipo:

Pontos nodais: 12 14 19

Espessura: 0.7 Carga: 1.0

OK Cancel

Figura 55 - Geração do quarto elemento triangular



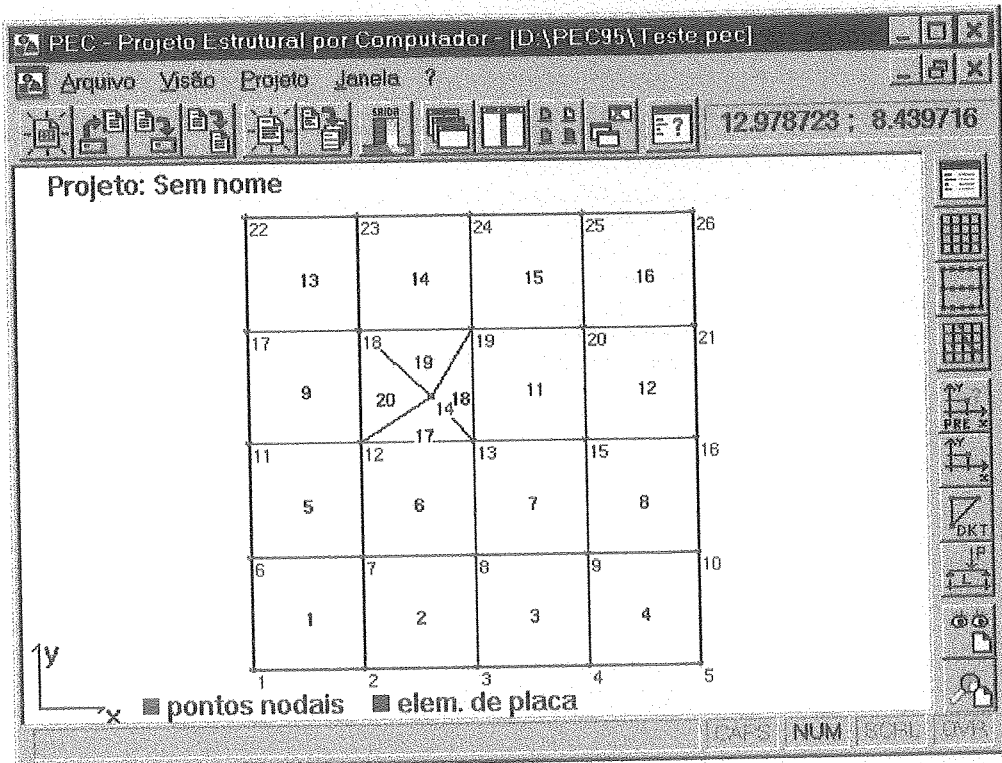


Figura 56 - Resultado da geração dos elementos triangulares

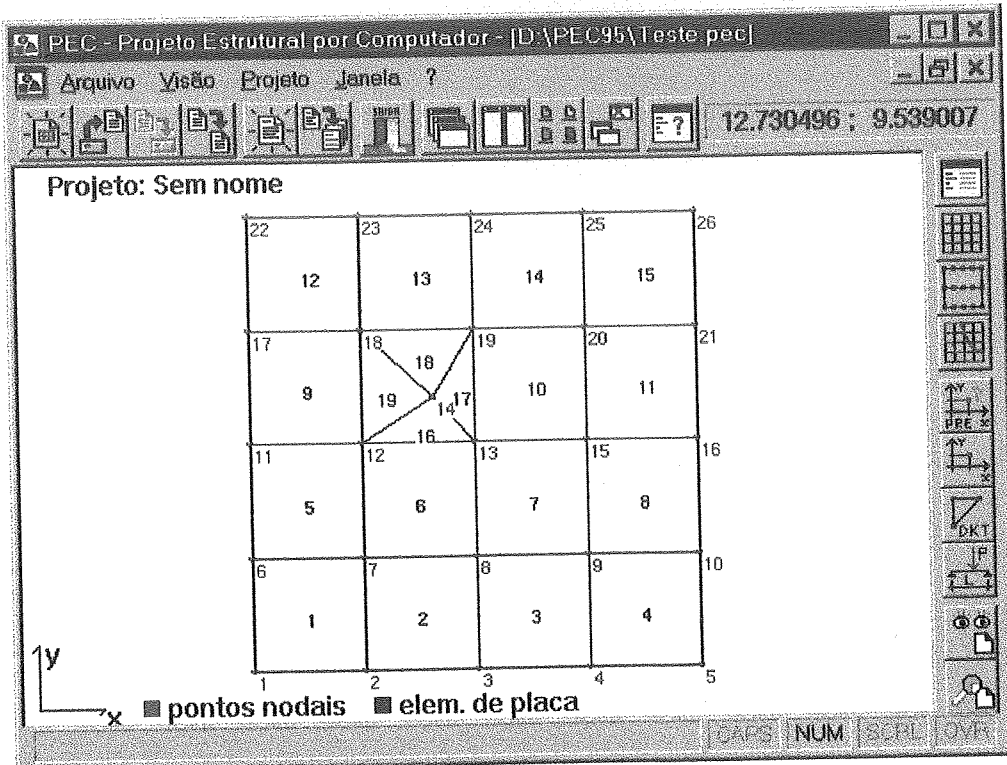


Figura 57 - Resultado final da alteração

A figura 58 mostra o **Exemplo 1** já finalizado e que recebeu os últimos retoques através dos comandos do menu flutuante da janela gráfica. Os pilares deste exemplo estão localizados sob os pontos 1, 9, 51, 54, 67 e 72.

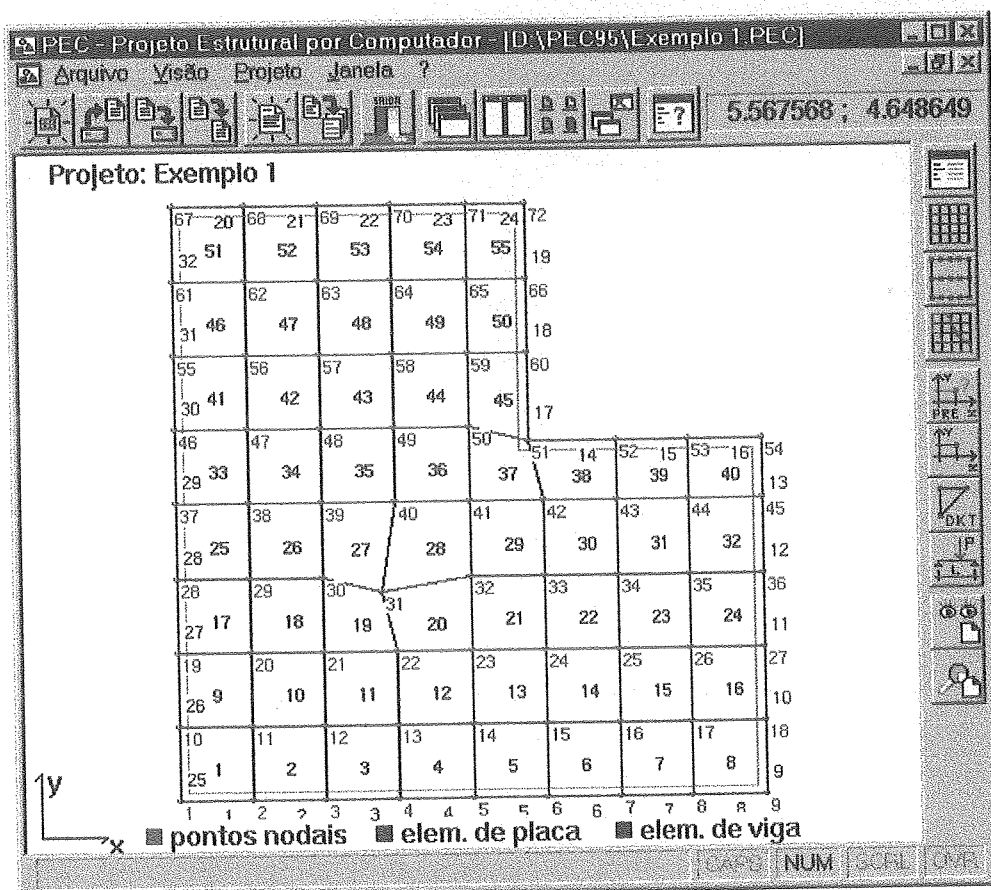


Figura 58 - Malha final do Exemplo 1

### 4.3.11 Gerador de relatórios

A função do **Gerador de Relatórios** é o usuário poder definir o formato do relatório de saída de dados que serão utilizados por um programa qualquer de cálculo. O usuário define o formato em um arquivo texto contendo comandos do interpretador. Esses comandos podem ser

escritos em qualquer coluna do arquivo e com qualquer combinação de letras maiúsculas e minúsculas. São comandos básicos como *Print* que imprime uma *string* e comandos de laço como *Do* e *EndDo*. Junto com esses comandos utiliza-se as variáveis do sistema como *TotBeam* que significa o número de elementos de viga que existem no projeto. A tabela 4 contém todos os comandos e as variáveis que podem ser utilizados na definição de um formato de relatório. É possível se definir quantos formatos de relatórios forem desejados. Como exemplo são definidos um formato denominado **Padrão** onde se utiliza quase todos os comandos e variáveis do sistema e um outro formato que serve para gerar arquivos do tipo DXF para serem lidos pelo AutoCAD.

Tabela 4 - Comandos do interpretador

Comando	Descrição
REM	define uma linha de comentários, o interpretador ignora a linha.
PRINT (string)	imprime a cadeia de caracteres que estiver entre os parênteses.
NEWLINE	imprime o caracter de retorno de linha.
DO ini, fim	comando de repetição, marca o inicio de um laço que será executado ( <b>fim-ini</b> ) vezes. <b>Ini</b> e <b>fim</b> devem ser números inteiros ou uma variável de sistema que contenha um número inteiro.
ENDDO	marca o final do laço.
TOTPOINT	total de pontos nodais.
TOTELEM	total de elementos de placa.
TOTBEAM	total de elementos de viga.
PROJECT	nome do projeto.
ELASTIC	módulo de elasticidade.
POISSON	coeficiente de Poisson.
POINTNUM	numero do ponto nodal.
COORDX	coordenada X do ponto nodal.
COORDY	coordenada Y do ponto nodal.
COORDZ	coordenada Z do ponto nodal.
VR1	vínculo rígido 1 do ponto nodal
VR2	vínculo rígido 2 do ponto nodal
VR3	vínculo rígido 3 do ponto nodal
LOADP	carregamento P do ponto nodal
LOADMX	momento Mx do ponto nodal
LOADMY	momento My do ponto nodal
VE1	vínculo elástico 1 do ponto nodal
VE2	vínculo elástico 2 do ponto nodal
VE3	vínculo elástico 3 do ponto nodal
BEAMNUM	número do elemento de viga
BEAMINI	número do ponto nodal onde inicia o elemento de viga
BEAMEND	número do ponto nodal onde termina o elemento de viga
BEAMIT	inércia a torção do elemento de viga
BEAMI	inércia a flexão do elemento de viga

Tabela 4 - Comandos do interpretador (continuação)

Comando	Descrição
BEAMAREA	área da seção de um elemento de viga
BEAMLOAD	carregamento distribuído em um elemento de viga
BEAMCX1	coordenada X do ponto nodal onde inicia o elemento de viga
BEAMCY1	coordenada Y do ponto nodal onde inicia o elemento de viga
BEAMCZ1	coordenada Z do ponto nodal onde inicia o elemento de viga
BEAMCX2	coordenada X do ponto nodal onde termina o elemento de viga
BEAMCY2	coordenada Y do ponto nodal onde termina o elemento de viga
BEAMCZ2	coordenada Z do ponto nodal onde termina o elemento de viga
ELEMNUM	número do elemento de placa
ELEMTYPE	tipo do elemento de placa
ELEMTHICK	espessura do elemento de placa
ELEMLoad	carregamento distribuído no elemento de placa
ELEMNO1	número do ponto nodal 1 do elemento de placa
ELEMNO2	número do ponto nodal 2 do elemento de placa
ELEMNO3	número do ponto nodal 3 do elemento de placa
ELEMNO4	número do ponto nodal 4 do elemento de placa
ELEMCX1	coordenada X do ponto nodal 1 do elemento de placa
ELEMCX2	coordenada X do ponto nodal 2 do elemento de placa
ELEMCX3	coordenada X do ponto nodal 3 do elemento de placa
ELEMCX4	coordenada X do ponto nodal 4 do elemento de placa
ELEMCY1	coordenada Y do ponto nodal 1 do elemento de placa
ELEMCY2	coordenada Y do ponto nodal 2 do elemento de placa
ELEMCY3	coordenada Y do ponto nodal 3 do elemento de placa
ELEMCY4	coordenada Y do ponto nodal 4 do elemento de placa
ELEMCZ1	coordenada Z do ponto nodal 1 do elemento de placa
ELEMCZ2	coordenada Z do ponto nodal 2 do elemento de placa
ELEMCZ3	coordenada Z do ponto nodal 3 do elemento de placa
ELEMCZ4	coordenada Z do ponto nodal 4 do elemento de placa

A seguir tem-se a listagem do arquivo de formato denominado **Padrão**. O símbolo ↵ indica que a linha do arquivo ainda não acabou e o limite de impressão foi alcançado, o restante da linha é impresso logo a seguir.

### Listagem 1 - Arquivo de formato **Padrão**

---

```

rem =====
rem Formato Padrão
rem =====

rem ----- CABEÇALHO -----

Print ( Relatório de dados do Programa PEC)
NewLine
NewLine
Print ( -> Projeto: )
    Project
    NewLine
Print ( -> Módulo de Elasticidade: )
    Elastic
    NewLine
Print ( -> Coeficiente de Poisson: )
    Poisson
    NewLine
Print ( -> Total de pontos nodais: )
    TotPoint
    NewLine
Print ( -> Total de elementos de placa: )
    TotElem
    NewLine
Print ( -> Total de elementos de viga: )
    TotBeam
    NewLine

rem ---- LISTA DE PONTOS NODAIS E COORDENADAS ----

NewLine
Print ( Pontos Nodais:)
NewLine
Print (NO,Coord X,Coord Y,Coord Z,Vínculo Rígido 1,V.R.2,V.R.3, ↵
Carga P, Mx , Mz, Vínculo Elástico 1, V.E.2, V.E.3)
NewLine
Print ( )
Do 1, 82
    Print (=)
EndDo
NewLine
Do 1, TotPoint
    Print ( )
    PointNum
    Print (, )
    CoordX
    Print (, )

```

## Listagem 1 - Arquivo de formato Padrão (continuação)

---

```
CoordY
Print (, )
CoordZ
Print (, )
VR1
Print (, )
VR2
Print (, )
VR3
Print (, )
LoadP
Print (, )
LoadMx
Print (, )
LoadMy
Print (, )
VE1
Print (, )
VE2
Print (, )
VE3
NewLine
EndDo

rem ----- LISTA DE ELEMENTOS FINITOS DE PLACA -----

NewLine
Print ( Elementos de Placa:)
NewLine
Print ( Elemento, Tipo, espessura, carga, pontos nodais)
NewLine
Print ( )
Do 1, 42
    Print (=)
EndDo
NewLine
Do 1, TotElem
    Print ( )
    ElemNum
    Print (, )
    ElemType
    Print (, )
    ElemThick
    Print (, )
    ElemLoad
    Print (, )
    ElemNo1
    Print (, )
    ElemNo2
    Print (, )
    ElemNo3
    Print (, )
    ElemNo4
    NewLine
EndDo
```

## Listagem 1 - Arquivo de formato **Padrão** (continuação)

---

```

rem ----- LISTA DE ELEMENTOS FINITOS DE VIGA -----

NewLine

Print ( Elementos de Viga:)
NewLine

Print ( Elemento, Nó inicial, Nó final, Inércia Flexão, Inércia ↵
Torção, Área, Carga)
NewLine
Print ( )
Do 1, 53
    Print (=)
EndDo
NewLine
Do 1, TotBeam
    Print ( )
    BeamNum
    Print (, )
    BeamIni
    Print (, )
    BeamEnd
    Print (, )
    BeamI
    Print (, )
    BeamIT
    Print (, )
    BeamArea
    Print (, )
    BeamLoad
    NewLine
EndDo

```

---

A listagem 2 é o exemplo de um arquivo de formato que serve para gerar um arquivo no formato DXF. Neste exemplo é gerado um *layer* chamado *VIGAS*, uma linha do tipo contínua e os elementos de viga utilizando-se a entidade *line*. Com pequenas modificações neste arquivo de formato é possível se definir um outro arquivo que gere no formato DXF os elementos de placa ou os pontos nodais. Para os pontos nodais a modificação deve ser feita na seção que imprime os elementos, o laço deve ir de 1 até **TotPoint**, ou seja, o total de pontos nodais, e a entidade utilizada deve ser *Point*. Para os elementos de placa o laço deve ir de 1 até **TotElem** e a entidade utilizada para desenhar o elemento deve ser *PoliLine*.

## Listagem 2 - Arquivo de formato DXF

---

```
rem =====
rem Formato DXF para os elementos de viga
rem =====

rem ----- TABLES section -----

rem =====
rem Define tipo de linha CONTINUOUS
rem
print(0)
newline
print(SECTION)
newline
print(2)
newline
print(TABLES)
newline
print(0)
newline
print(TABLE)
newline
print(2)
newline
print(LTYPE)
newline
print(70)
newline
print(1)
newline
print(0)
newline
print(LTYPE)
newline
print(2)
newline
print(CONTINUOUS)
newline
print(70)
newline
print(64)
newline
print(3)
newline
print(Solid line)
newline
print(72)
newline
print(65)
newline
print(73)
newline
print(0)
newline
print(40)
newline
print(0.0)
newline
print(0)
```



## Listagem 2 - Arquivo de formato DXF (continuação)

---

```
newline
print(ENDTAB)
newline

rem =====
rem Define o layer VIGAS
rem
print(0)
newline
print(TABLE)
newline
print(2)
newline
print(LAYER)
newline
print(70)
newline
rem O numero 1 da proxima linha é a quantidade de layer a ser declada
print(1)
newline
print(0)
newline
print(LAYER)
newline
print(2)
newline
rem A palavra VIGAS da proxima linha e' o nome do layer
print(VIGAS)
newline
print(70)
newline
print(64)
newline
print(62)
newline
rem O numero 5 da proxima linha define a cor azul para o layer
print(5)
newline
print(6)
newline
rem A proxima linha indica o tipo de linha do layer
print(CONTINUOUS)
newline
print(0)
newline
print(ENDTAB)
newline
print(0)
newline
print(ENDSEC)
newline

rem ----- ENTITIES section -----

rem =====
rem Define as vigas associadas ao Layer VIGAS
rem
```

## Listagem 2 - Arquivo de formato DXF (continuação)

---

```
print (0)
newline
print (SECTION)
newline
print (2)
newline
print (ENTITIES)
newline
do 1,totbeam
  print (0)
  newline
  print (LINE)
  newline
  print (8)
  newline
  print (VIGAS)
  newline
  print (10)
  newline
  beamCX1
  newline
  print (20)
  newline
  beamCY1
  newline
  print (30)
  newline
  beamCZ2
  newline
  print (11)
  newline
  beamCX2
  newline
  print (21)
  newline
  beamCY2
  newline
  print (31)
  newline
  beamCZ2
  newline
enddo
print (0)
newline
print (ENDSEC)
newline
print (0)
newline
print (EOF)
newline
```

---

Após o usuário ter definido pelo menos uma malha e um arquivo de formato ele pode utilizar o comando **Gerar relatório** que aciona a caixa de diálogo **Gerador de relatório** (figura 59). Primeiramente deve ser definido o nome do arquivo de formato pressionando-se o botão **Nome do arquivo de formato**. É aberta uma lista contendo o nome de todos os arquivos de formato definidos previamente. Deve ser apenas selecionado um dos formatos (figura 60). Depois, ao retornar à caixa de diálogo **Gerador de relatório** (figura 61) o botão **OK** deve ser pressionado, a proporção que o relatório vai sendo gerado o usuário vai monitorando o número de linhas geradas através do campo **Nº de linhas geradas**. Na figura 62 tem-se o status da caixa de diálogo do gerador de relatórios após uma geração. O relatório gerado recebe o mesmo nome do seu formato porém com a extensão **TXT**.

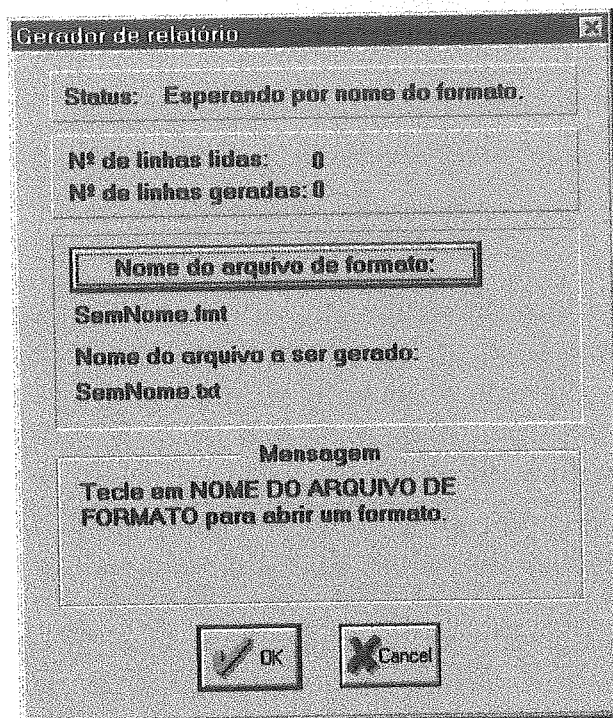


Figura 59 - Gerador de relatório

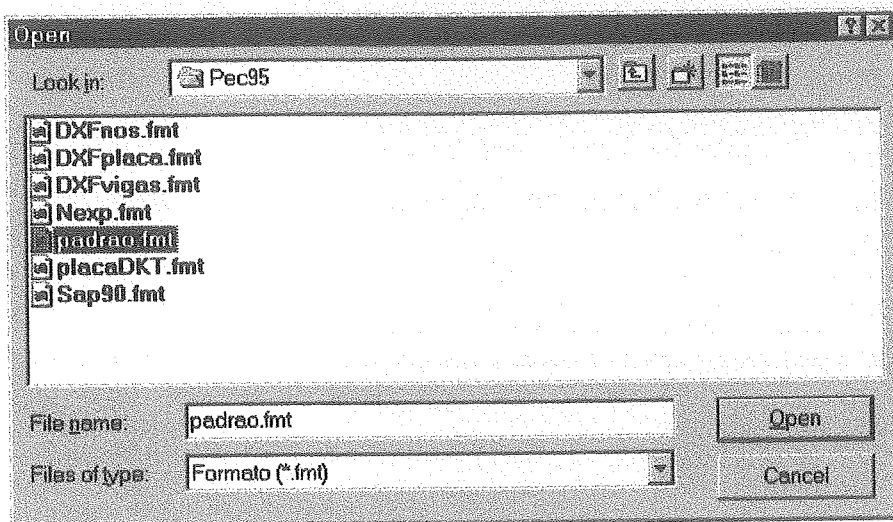


Figura 60 - Seleção do arquivo de formato

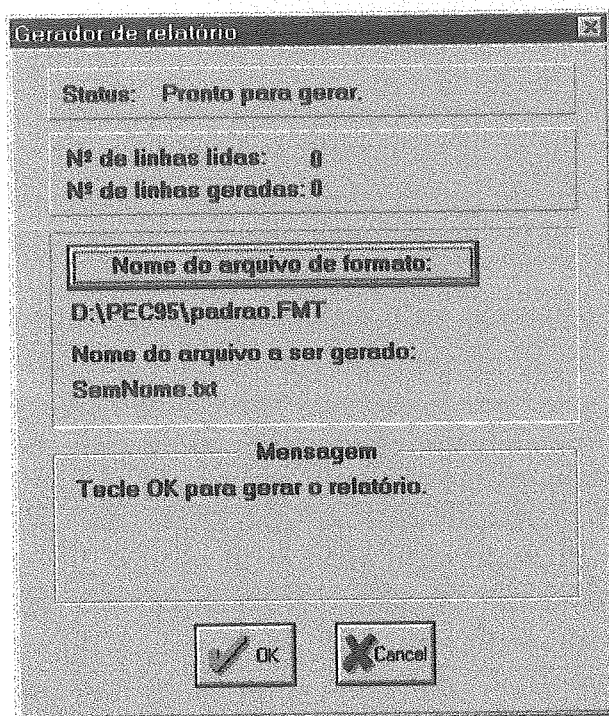


Figura 61 - Gerador de relatórios com um formato escolhido

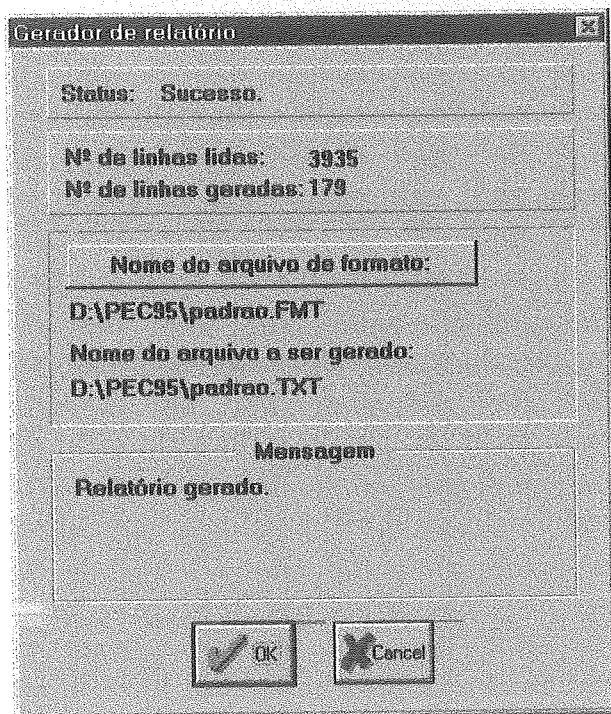


Figura 62 - Gerador de relatórios: final de uma geração

Na listagem 3 tem-se o resultado do relatório da malha do Exemplo 1 gerado com o formato Padrão. Como esta listagem é apenas ilustrativa e muito extensa, algumas linhas foram substituídas por reticências.

### Listagem 3 - Dados do Exemplo 1 gerados pelo formato Padrão

Relatório de dados do Programa PEC

-> Projeto: Exemplo 1  
 -> Módulo de Elasticidade: 1.700000e+07  
 -> Coeficiente de Poisson: 0.200000  
 -> Total de pontos nodais: 72  
 -> Total de elementos de placa: 55  
 -> Total de elementos de viga: 32

Pontos Nodais:

NO,Coord X,Coord Y,Coord Z,Vínculo Rígido 1,V.R.2,V.R.3,Carga P,Mx,Mz,Vínculo Elástico 1,V.E.2,V.E.3

```

=====
1, 0.000000, 0.000000, 0.000000, 1, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000
2, 1.250000, 0.000000, 0.000000, 0, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000
3, 2.500000, 0.000000, 0.000000, 0, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000
4, 3.750000, 0.000000, 0.000000, 0, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000
5, 5.000000, 0.000000, 0.000000, 0, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000
.
.

```

```

70,3.750000,10.000000, 0.000000, 0, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000
71,5.000000,10.000000, 0.000000, 0, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000
72,6.000000,10.000000, 0.000000, 1, 0, 0, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000

```

Elementos de Placa:

Elemento, Tipo, espessura, carga, pontos nodais

```

=====
1, 1, 1.000000, 50.000000, 1, 2, 11, 10
2, 1, 1.000000, 50.000000, 2, 3, 12, 11
3, 1, 1.000000, 50.000000, 3, 4, 13, 12
4, 1, 1.000000, 50.000000, 4, 5, 14, 13
5, 1, 1.000000, 50.000000, 5, 6, 15, 14
6, 1, 1.000000, 50.000000, 6, 7, 16, 15
.
.
.
52, 1, 1.000000, 50.000000, 62, 63, 69, 68
53, 1, 1.000000, 50.000000, 63, 64, 70, 69
54, 1, 1.000000, 50.000000, 64, 65, 71, 70
55, 1, 1.000000, 50.000000, 65, 66, 72, 71

```

Elementos de Viga:

Elemento, Nó inicial, Nó final, Inércia Flexão, Inércia Torção, Área, Carga

```

=====
1, 1, 2, 0.000640, 0.000001, 0.048000, 6.000000
2, 2, 3, 0.000640, 0.000001, 0.048000, 6.000000
3, 3, 4, 0.000640, 0.000001, 0.048000, 6.000000
.
.
.
27, 19, 28, 0.000640, 0.000001, 0.048000, 6.000000
28, 28, 37, 0.000640, 0.000001, 0.048000, 6.000000
29, 37, 46, 0.000640, 0.000001, 0.048000, 6.000000
30, 46, 55, 0.000640, 0.000001, 0.048000, 6.000000
31, 55, 61, 0.000640, 0.000001, 0.048000, 6.000000
32, 61, 67, 0.000640, 0.000001, 0.048000, 6.000000

```

Para o **Exemplo 1** também foi gerado um relatório no formato DXF e o resultado é mostrado na figura 63 onde se tem o arquivo DXF mostrado pelo AutoCAD.

Para o cálculo do **Exemplo 1** foram gerados dois relatórios: um para o sistema SAP 90 e outro para o sistema NEXP desenvolvido por REZENDE (1990). Os resultados dos cálculos estão nas tabelas 5 e 6 onde pode-se fazer uma comparação entre os sistemas. As figuras 64 e 65 mostram a malha do **Exemplo 1** vista pelos sistemas NEXP e SAP 90 respectivamente.

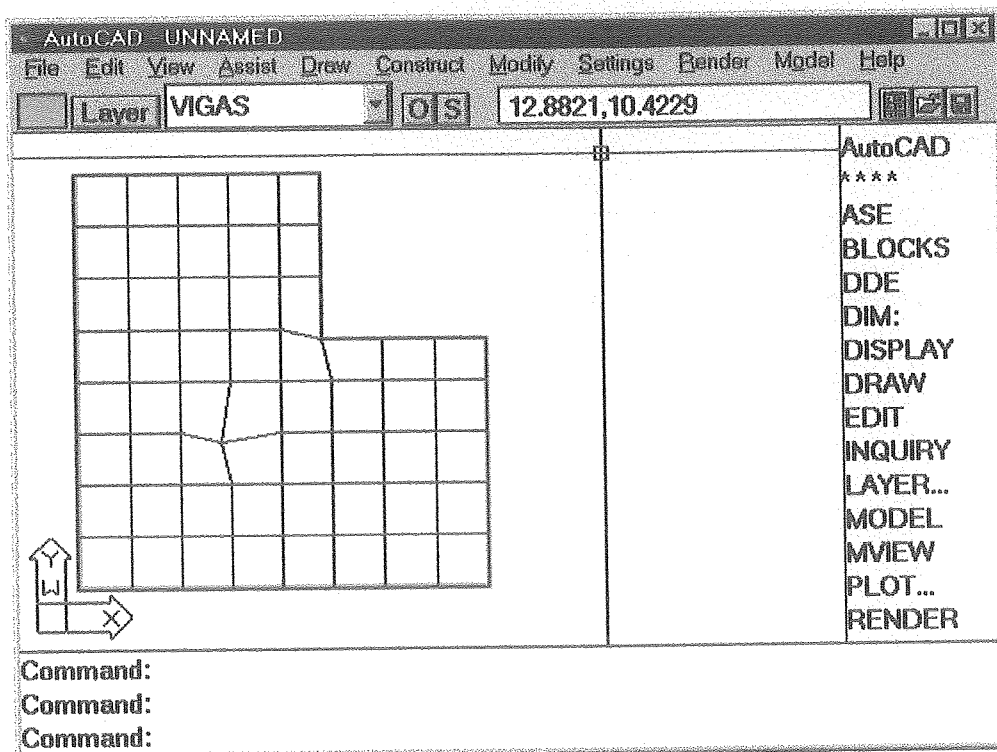


Figura 63 - Formato DXF do Exemplo 1 mostrado pelo AutoCAD

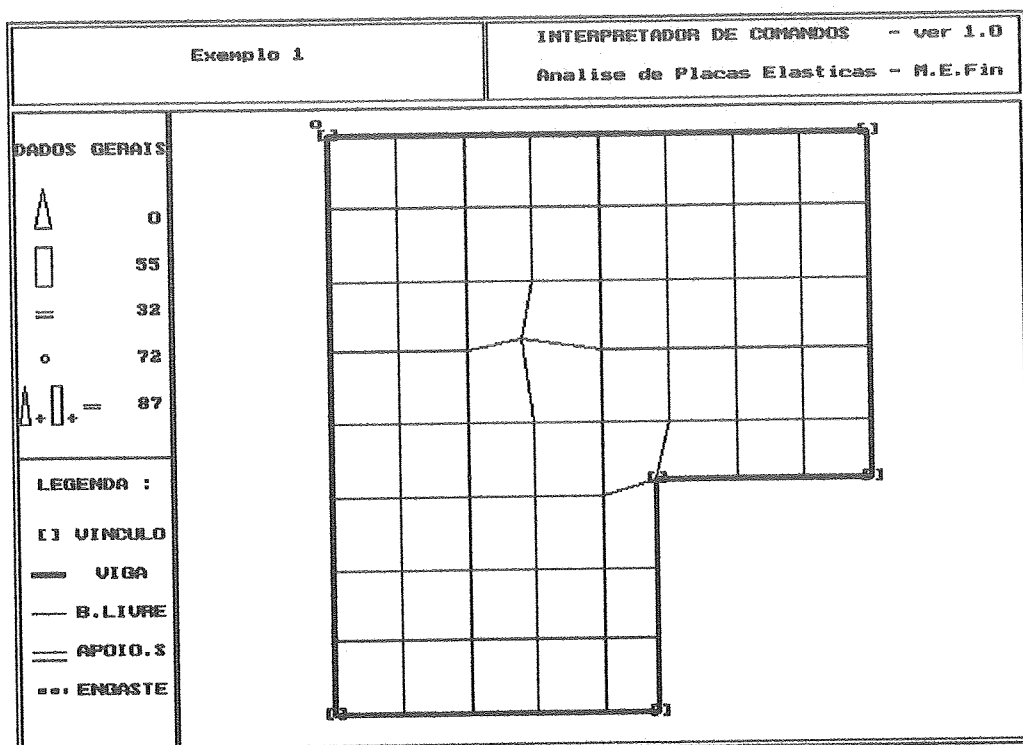


Figura 64 - Exemplo 1 visto pelo sistema NEXP

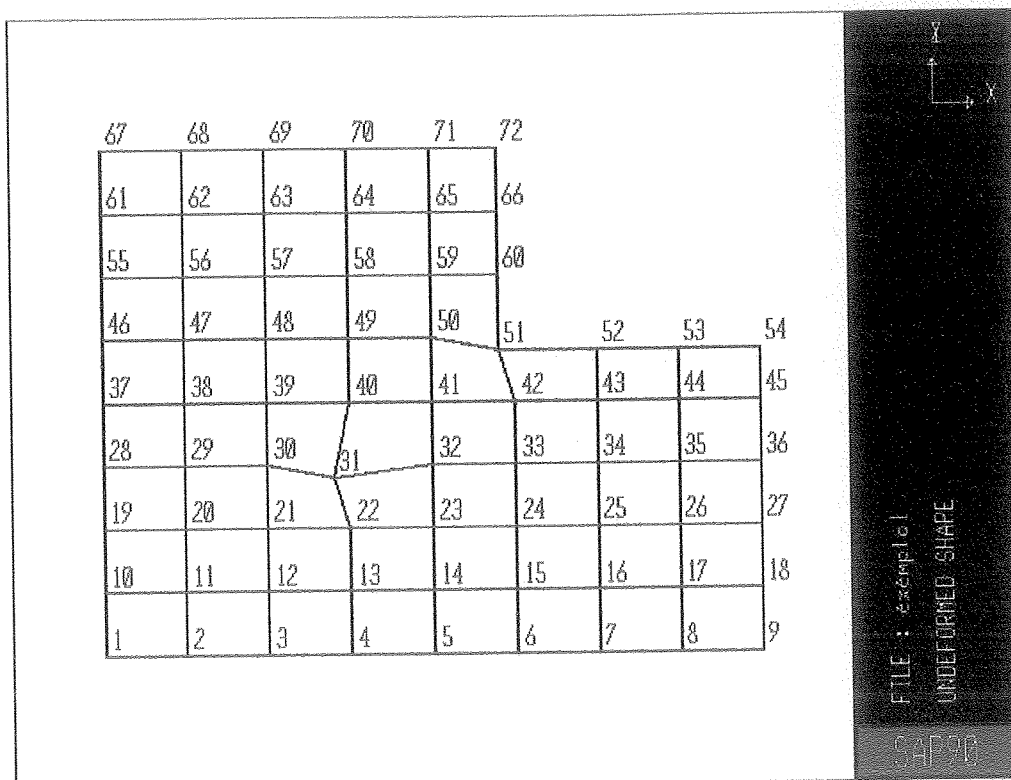


Figura 65 - Exemplo 1 visto pelo sistema SAP 90

Tabela 5 - Calculo do Exemplo 1 segundo o sistema NEXP

Nó	Deslocamentos ( $\times 10^{-4}$ )			Momentos			Força vertical
	U(z)	R(x)	R(y)	Mx	My	Mxy	
1	0	-8,08	8,08	-47,81	-47,81	344,24	875,64
9	0	-3,13	-7,32	-35,60	-39,99	-234,55	622,60
31	-19,45	1,92	-1,92	173,64	173,64	153,07	0
51	0	4,66	-4,66	-788,13	-788,13	146,90	2531,18
54	0	2,82	0,68	-8,99	-29,55	-94,94	-107,77
67	0	7,32	3,13	-39,99	-35,60	-234,55	622,60
72	0	-0,68	-2,82	-29,55	-8,99	-94,94	-107,77



Tabela 6 - Calculo do Exemplo 1 segundo o sistema SAP 90

Nó	Deslocamentos ( $\times 10^{-4}$ )			Momentos			Força vertical
	U(z)	R(x)	R(y)	Mx	My	Mxy	
1	0	-8,20	8,20	-6,92	-6,92	351,13	884,45
9	0	-3,16	-7,52	-6,40	-4,14	-239,48	628,85
31	-19,64	1,93	-1,93	151,13	151,13	164,98	0
51	0	4,80	-4,80	-589,70	-589,70	291,61	2483,63
54	0	2,93	0,57	-2,81	23,84	-82,04	-92,89
67	0	7,52	3,16	-4,14	-6,40	-239,48	628,85
72	0	-0,57	-2,93	23,84	-2,81	-82,04	-92,89

Para uma verificação mais apurada dos resultados foi gerada uma malha mais refinada para o **Exemplo 1**. Esta malha é demonstrada na figura 66 e o resultado do cálculo efetuado tanto pelo sistema SAP 90 como pelo NEXP desta malha são comparados nas tabelas 7 e 8.

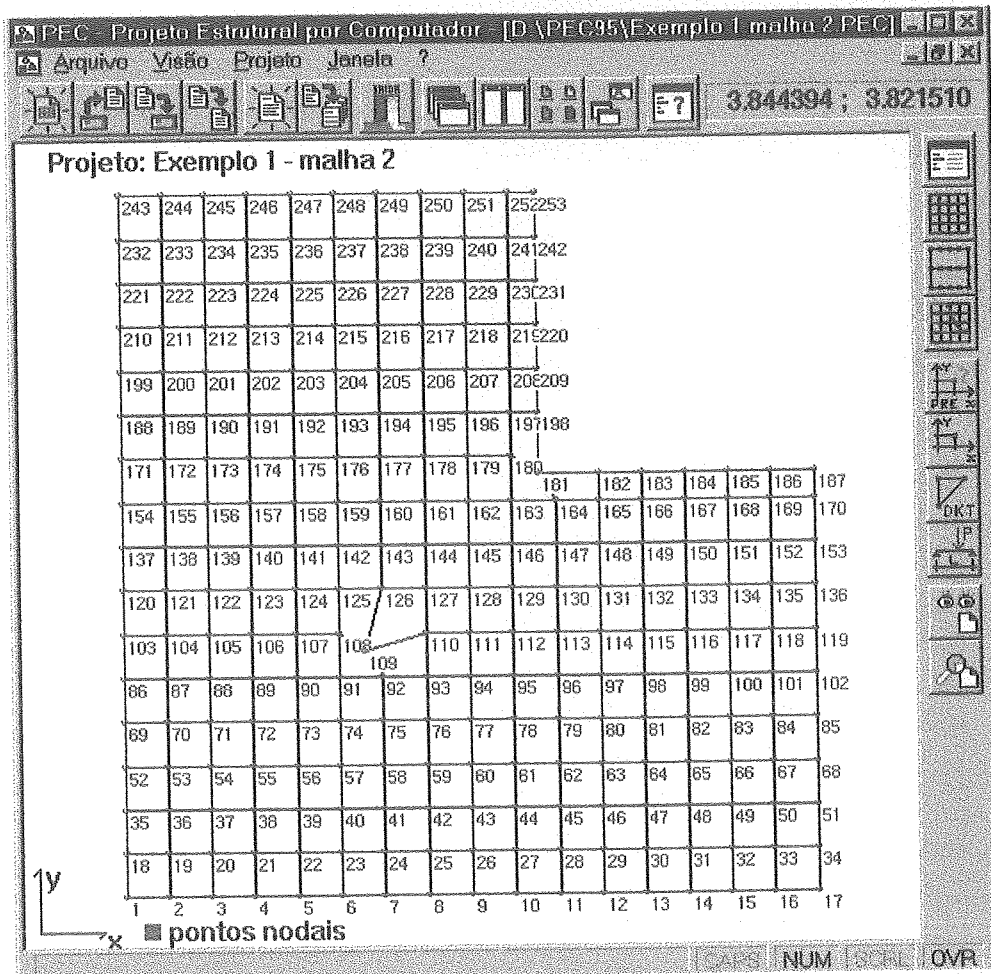


Figura 66 - Exemplo 1 com uma malha de 220 elementos de placa

Tabela 7 - Cálculo do Exemplo 1 com a malha mais refinada segundo o sistema Nexp

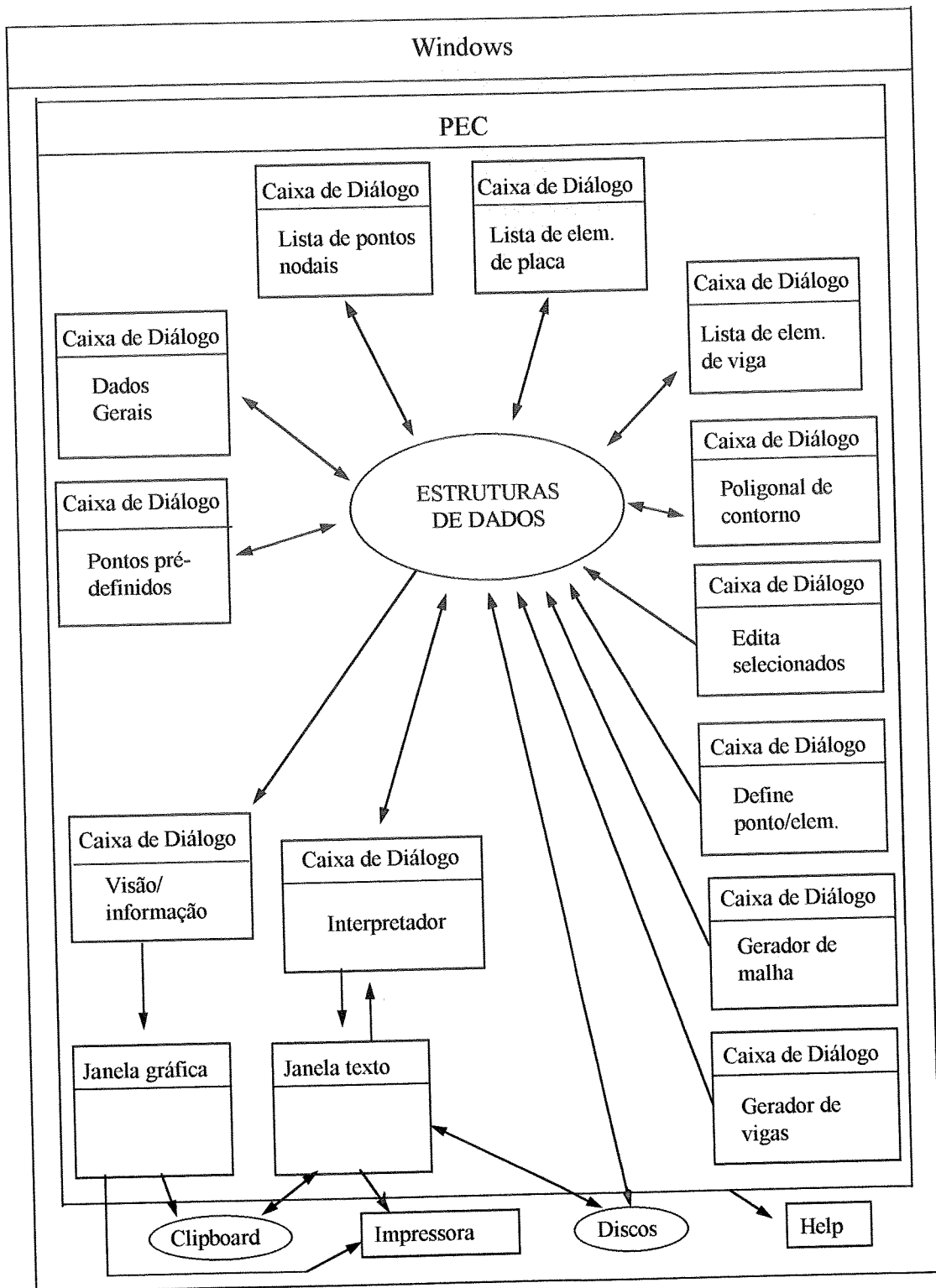
Nó	Deslocamentos ( $\times 10^{-4}$ )			Momentos			Força vertical
	U(z)	R(x)	R(y)	Mx	My	Mxy	
1	0	-8,22	8,22	-31,46	-31,46	381,10	878,05
17	0	-3,31	-7,55	-24,53	-26,05	-264,39	624,06
109	-19,83	1,92	-1,92	163,16	163,16	160,16	0
181	0	5,01	-5,01	-946,21	-946,21	483,29	2516,11
187	0	2,93	0,50	-8,39	-23,58	-76,20	-103,24
243	0	7,55	3,31	-26,39	-24,53	-264,39	624,06
253	0	-0,50	-2,93	-23,58	-8,39	-76,20	-103,24

Tabela 8 - Cálculo do Exemplo 1 com uma malha mais refinada segundo o sistema SAP 90

Nó	Deslocamentos ( $\times 10^{-4}$ )			Momentos			Força vertical
	U(z)	R(x)	R(y)	Mx	My	Mxy	
1	0	-8,38	8,38	-4,47	-4,47	384,17	881,40
17	0	-3,33	-7,62	-3,82	-3,15	-266,76	626,53
109	-19,94	1,92	-1,92	154,20	154,20	170,32	0
181	0	5,04	-5,04	-1244,70	-1244,70	331,85	2502,81
187	0	2,97	0,46	-3,19	-1,39	-73,60	-98,64
243	0	7,62	3,33	-3,15	-3,82	-266,76	626,53
253	0	-0,46	-2,97	-1,39	-3,19	-73,60	-98,64

As diferenças observadas entre os resultados do sistema NEXP e o sistema SAP 90 se devem aos diferentes elementos finitos utilizados pelos respectivos programas.

#### 4.4 Esquema gráfico representativo do aplicativo PEC



## 5 MANIPULAÇÃO INTERNA DOS DADOS

Normalmente quando se subdivide uma estrutura em elementos finitos as dimensões das matrizes utilizadas pelo aplicativo podem chegar a ser tão grandes que se torna inviável trabalhar com uma matriz desta como se fosse um só objeto. E também não seria conveniente que todas as matrizes já estivessem pré-dimensionadas, pois em alguns casos essas matrizes podem ser pequenas e desta forma seria um desperdício de recursos. O ideal é se fazer alocação dinâmica de memória, ou seja, as matrizes devem ser dimensionadas de acordo com as necessidades e em tempo de execução do aplicativo.

Essas matrizes podem chegar a ter dimensões tão grandes que precisem ser fracionadas e uma parte seja armazenada em disco, ou um dispositivo utilizado como expansão de memória, enquanto a outra parte fica sendo manipulada na memória convencional. Para isso deve-se utilizar uma técnica de dividir uma matriz em blocos. Como o aplicativo foi desenvolvido utilizando-se programação orientada a objetos, este problema de fracionar uma matriz foi simplificado e otimizado para que se possa tirar o máximo de aproveitamento dos recursos que a máquina em que será executado o aplicativo possa oferecer. Com a técnica utilizada, o programador passa a tratar uma matriz como sendo um só objeto e não vai mais se preocupar se há memória convencional suficiente para armazenar a matriz ou se ela está sendo armazenada uma parte em disco e outra em memória convencional. Para se explicar a técnica utilizada será mostrada apenas a matriz de pontos nodais, as demais foram tratadas da mesma forma. A matriz é

tratada como uma lista ligada. Desta forma é mais simples usufruir-se da alocação dinâmica de memória. Cada membro da matriz, no caso cada ponto nodal, também é um objeto. Então para se criar a matriz de pontos nodais são definidos dois objetos, um é o objeto lista de pontos nodais, que através deste se passa a ver os pontos nodais como se fosse uma só matriz, esse objeto serve para manipular o outro objeto denominado ponto nodal. A quantidade de objetos ponto nodal não é definida, a proporção que se precisa de mais um ponto nodal, um novo objeto ponto nodal é criado, ou seja, definido dinamicamente em tempo de execução pelo objeto lista de pontos nodais. Há mais vantagem em se tratar uma matriz em que cada elemento é um objeto do que uma matriz em que os objetos são blocos de elementos. Quanto menor for o objeto mais fácil fica de ser manipulado, menor será o espaço contínuo de memória necessária para que ele seja criado, maior será a quantidade de objetos possíveis de serem gerados. O programador não precisa se preocupar em enviar para o disco os objetos que não estão sendo utilizados, o próprio Windows faz essa tarefa, ele possui um arquivo de troca que serve como expansão da memória convencional e toda vez que ele precisa de memória e não há disponível ele envia para o arquivo de troca os objetos que não estão sendo utilizados no momento. O Windows procura não enviar para o arquivo de troca um objeto muito grande, ele prefere enviar objetos pequenos. Se ele tivesse preferência em enviar para o arquivo de troca objetos grandes ele iria perder mais tempo em transferência de dados, e ainda há mais probabilidade de que uma parte de um objeto grande seja solicitada pelo aplicativo do que uma parte de um objeto pequeno.

A declaração em C++ dos objetos ponto nodal e lista de pontos nodais é mostrada na listagem 4, os objetos foram denominados respectivamente de **ponto\_nodal** e **lista\_de\_nos**. A listagem 5 demonstra como se pode utilizar os objetos definidos na listagem 4 fazendo um *loop* que varre toda a matriz de pontos nodais. O objeto lista de pontos nodais possui as funções: **insere** que faz com que um objeto ponto nodal que foi

criado pelo comando **new** venha a fazer parte da matriz de pontos nodais, esta função insere o objeto classificando a lista de pontos nodais pelo número do ponto nodal; **ins\_ultimo** faz o mesmo que a função anterior, porém não faz a classificação, deve ser utilizada quando se tem certeza que o número do ponto nodal que será inserido é maior do que o número do último ponto que está na lista; e por último a função **remove** que serve para eliminar um ponto nodal da matriz. A lista possui ainda as variáveis de controle: **posicao** que é um ponteiro para o primeiro ponto nodal da lista; **ultimo** que é um ponteiro para o último elemento da lista; e **total** que indica o número de elementos da lista. Cada objeto ponto nodal tem uma estrutura **pontos** que contém as variáveis de um ponto nodal, um ponteiro **prox\_ligacao** que contém o endereço do próximo ponto nodal da lista, se esse endereço for zero significa que o ponto nodal corrente é o último elemento da lista; uma variável chamada **selec** que indica se o ponto nodal está ou não selecionado pelos comandos do menu flutuante e um ponteiro chamado **ponteiro** que é utilizado pela função **adiciona** na criação de um ponto nodal. A função **adiciona** do objeto ponto nodal é utilizada pelas funções **insere** e **ins\_ultimo** do objeto lista de pontos nodais na geração de um ponto nodal novo.

#### Listagem 4 - Declaração dos objetos ponto nodal e lista de pontos nodais

---

```
// =====
// A estrutura da lista ligada PONTOS NODAIS
// =====

struct _STRU pontos {
    unsigned long num;
    int VR1;
    int VR2;
    int VR3;
    double X;
    double Y;
    double Z;
    double carga_P;
};
```

```
double carga_Mx;
double carga_My;
double VE1;
double VE2;
double VE3;
};

// a classe dos pontos nodais
class _STRU ponto_nodal {

friend class _STRU lista_nos;

public:
    ponto_nodal (unsigned long num, double X, double Y, double Z=0,
                 int VR1=0, int VR2=0, int VR3=0,
                 double VE1=0, double VE2=0, double VE3=0,
                 double carga_P=0, double carga_Mx=0, double carga_My=0);

    void adiciona(){
        ponteiro = new ponto_nodal (No.num, No.X, No.Y, No.Z,
                                     No.VR1, No.VR2, No.VR3,
                                     No.VE1, No.VE2, No.VE3,
                                     No.carga_P, No.carga_Mx, No.carga_My);
    }

    pontos No;
    ponto_nodal* ponteiro;
    ponto_nodal* prox_ligacao;
    BOOL selec;
};

class _STRU lista_nos {

public:
    ponto_nodal* posicao; // inicio da lista
    ponto_nodal* ultimo; // ultimo elemento da lista
    unsigned long total; // quantidade de elementos da lista
    lista_nos()
    {
        posicao = 0;
        ultimo = 0;
        total = 0;
    }
    void insere(ponto_nodal* nodo);
    void ins_ultimo(ponto_nodal* nodo);
    void remove(unsigned long num);
};
```

---



## Listagem 5 - Exemplo de utilização dos objetos ponto nodal e lista de nós

---

```
lista_nos mat_nos; // matriz dos pontos nodais
ponto_nodal* no; // ponteiro utilizado para varrer a matriz dos pontos nodais
.
.
.
no = mat_nos.posicao;
while (no != 0){
    // neste ponto utiliza-se o nó
    .
    .
    // pega-se o próximo elemento da matriz
    no = no->prox_ligacao;
}
```

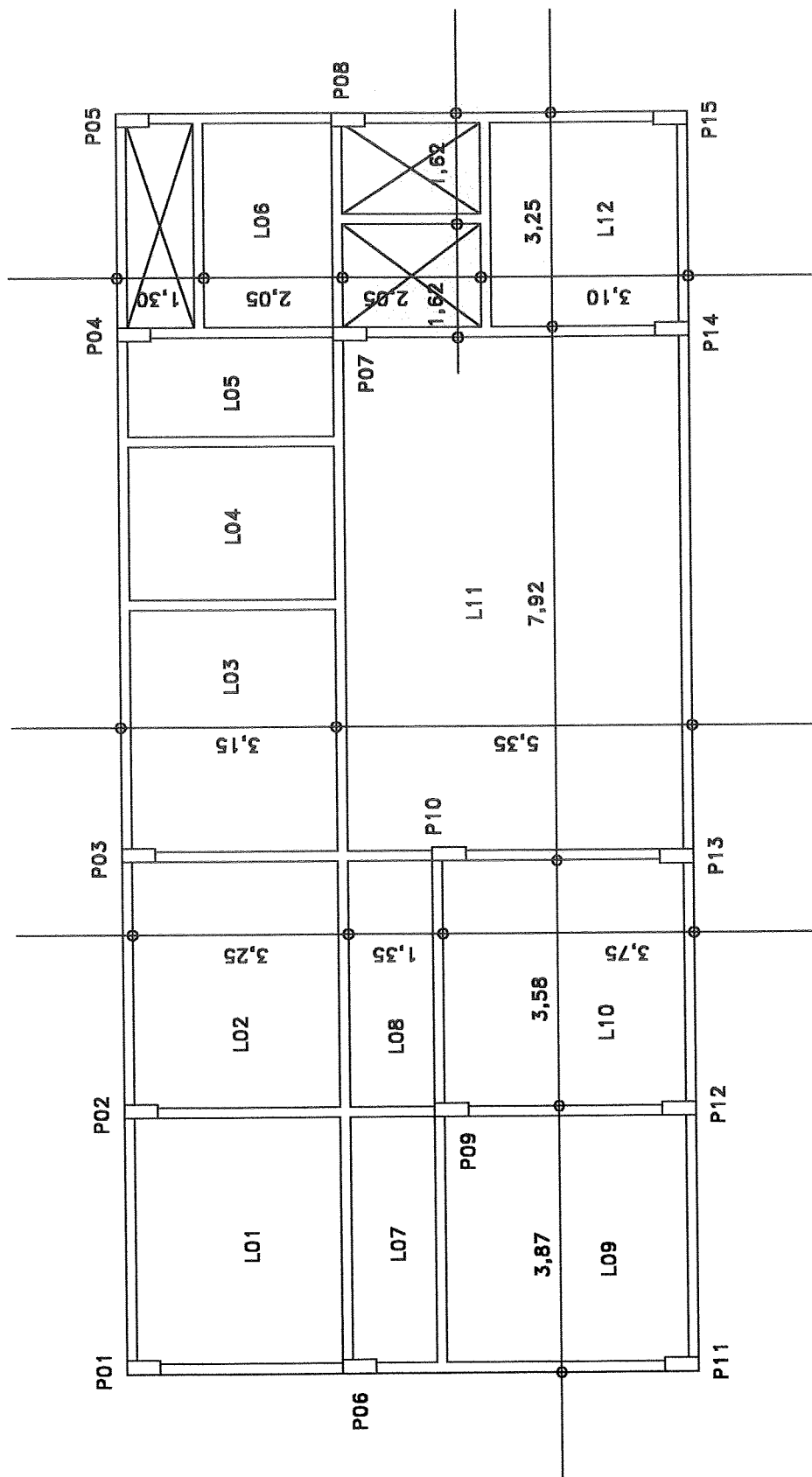
---

## 6 EXEMPLOS DE UTILIZAÇÃO DO APLICATIVO

A seguir são apresentadas algumas aplicações do sistema PEC. São gerados os dados dos pavimentos de três edifícios construídos ou em construção. Para cada um deles é apresentado a planta de fôrma e a malha gerada no sistema PEC.

### 6.1 Edifício Andrade

A figura 67 é a planta de fôrma do edifício encontrada em ANDRADE (1982). No corrente trabalho este edifício é chamado de edifício Andrade. A laje L11 deste projeto é uma laje nervurada. Para melhor visualização da malha de elementos finitos gerada pelo aplicativo PEC tem-se, na figura 68, uma visão somente dos elementos de viga, e na figura 69, a visão da malha completa. Esta malha é composta de 2.098 pontos nodais, 1.956 elementos finitos de placa e 918 elementos de viga.



Medidas em metro

Figura 67 - Planta de fôrma do edifício Andrade

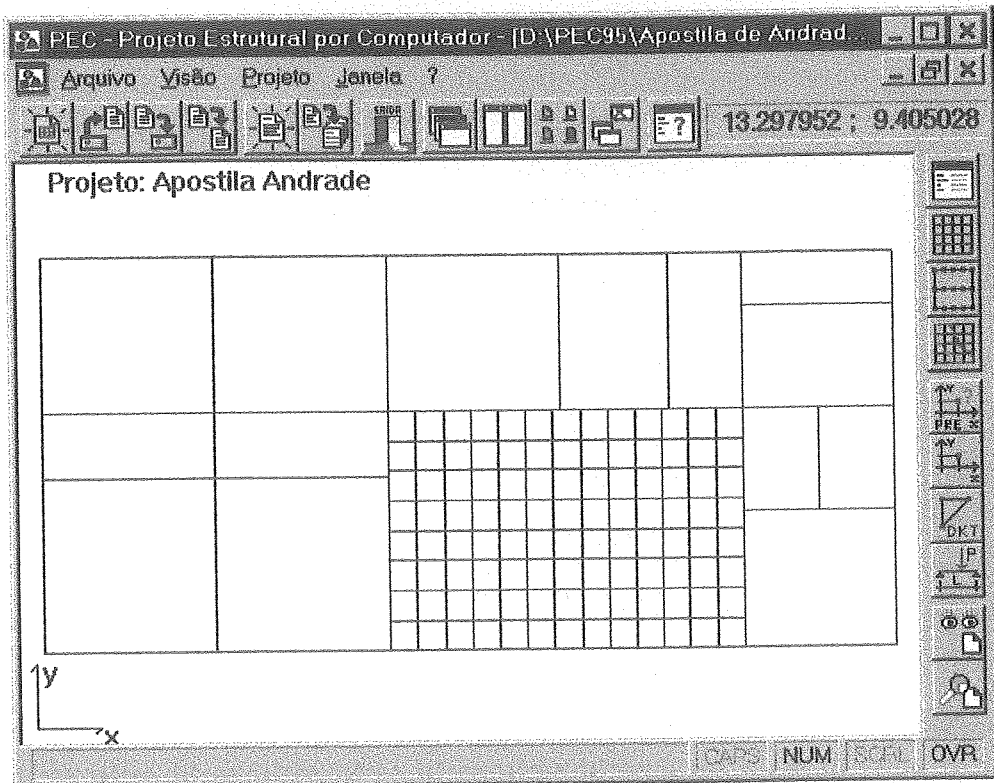


Figura 68 - Edifício Andrade, somente elementos de viga

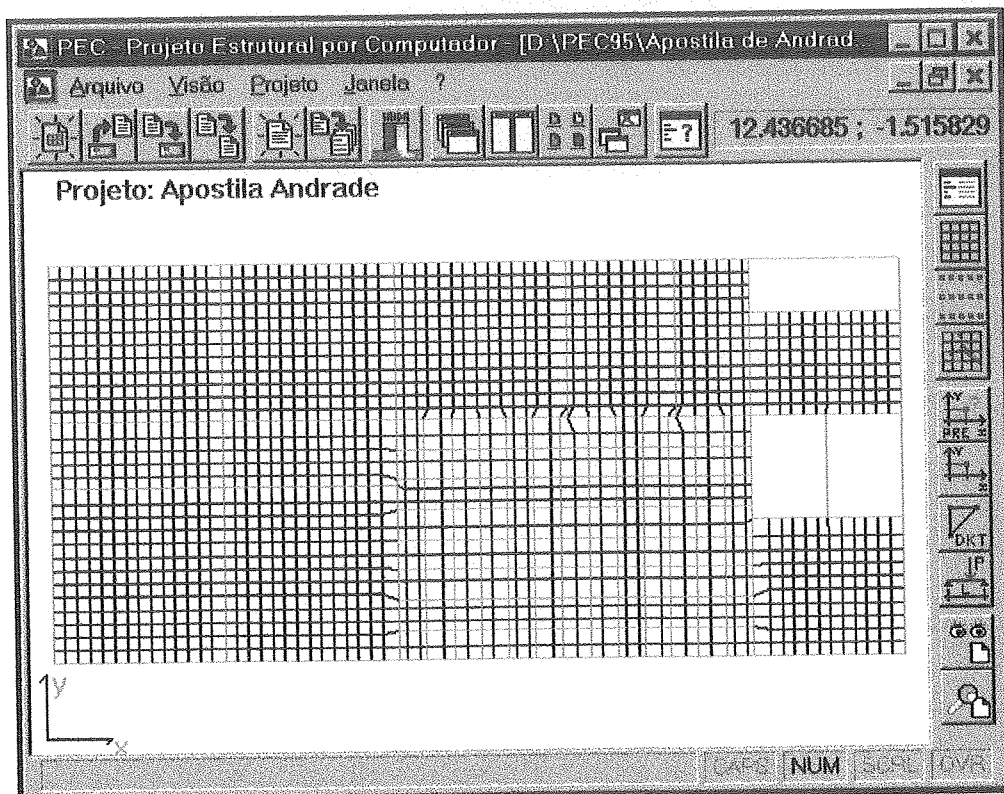


Figura 69 - Malha do edifício Andrade

## 6.2 Edifício Firenze

A figura 71 é a planta de forma do Edifício Firenze. Na figura 70 tem-se a malha gerada pelo aplicativo PEC. Esta malha é composta de 989 pontos nodais, 889 elementos finitos de placa e 269 elementos de viga.

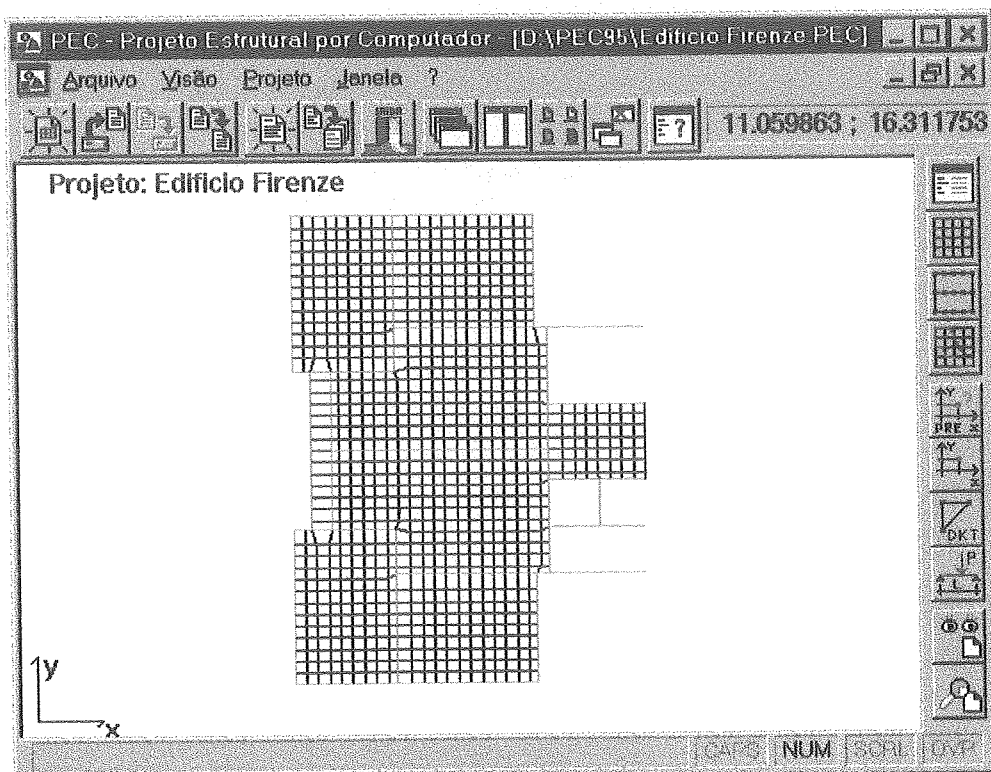


Figura 70 - Malha do Edifício Firenze

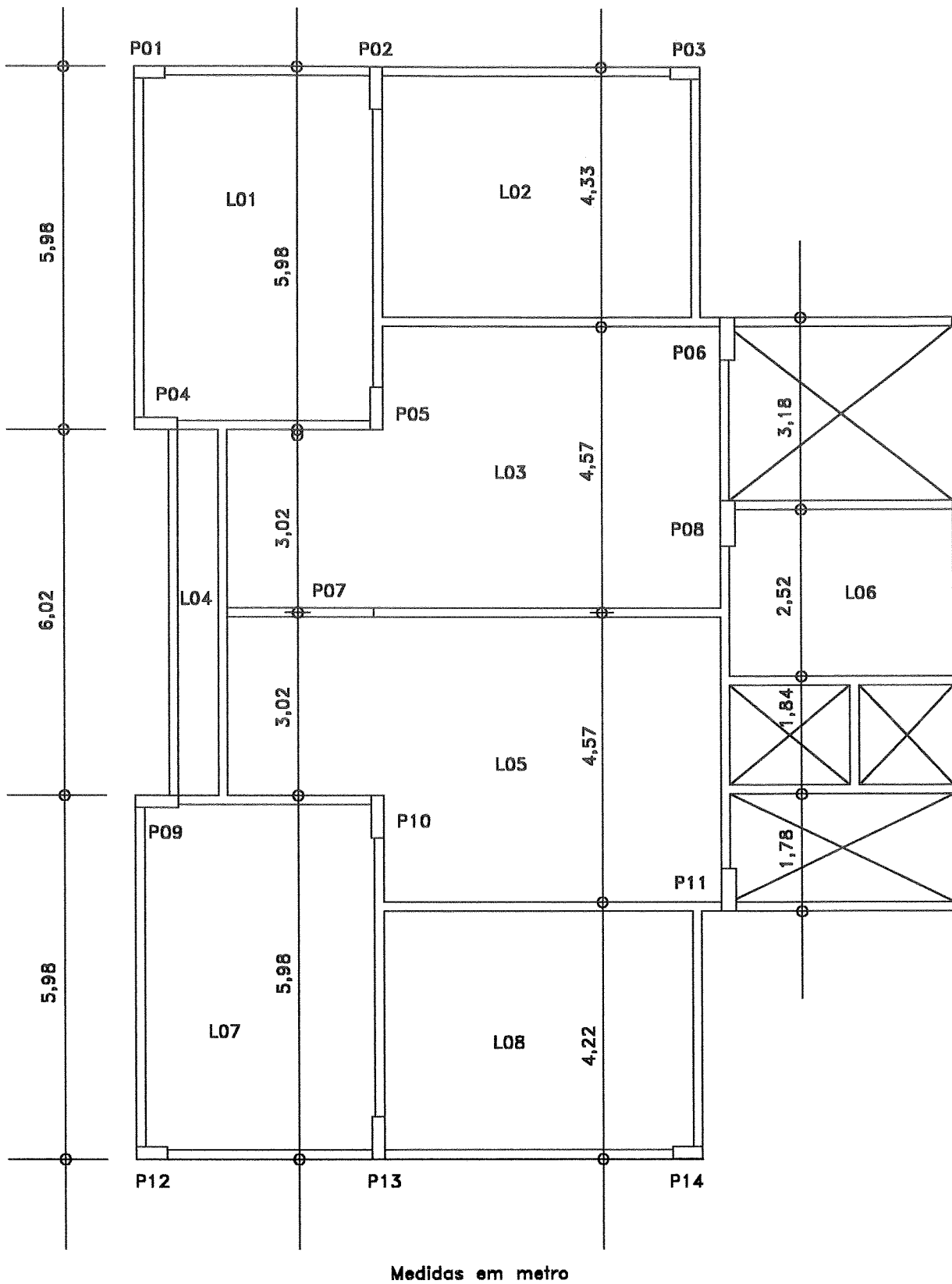


Figura 71 - Planta de fôrma do Edifício Firenze

### 6.3 Edifício Casa Blanca

A figura 73 é a planta de fôrma do Edifício Casa Blanca. Na figura 72 tem-se a malha gerada pelo PEC. Esta malha é formada por 937 pontos nodais, 859 elementos de placa e 291 elementos de viga. Com o gerador de relatórios foi gerado o arquivo de entrada de dados para o sistema NEXP. A figura 74 mostra o sistema NEXP após ter lido este arquivo. A listagem 6 é o arquivo de saída de dados do sistema NEXP. Algumas linhas desta listagem foram removidas porque o arquivo é muito extenso e é apresentado apenas como caráter ilustrativo.

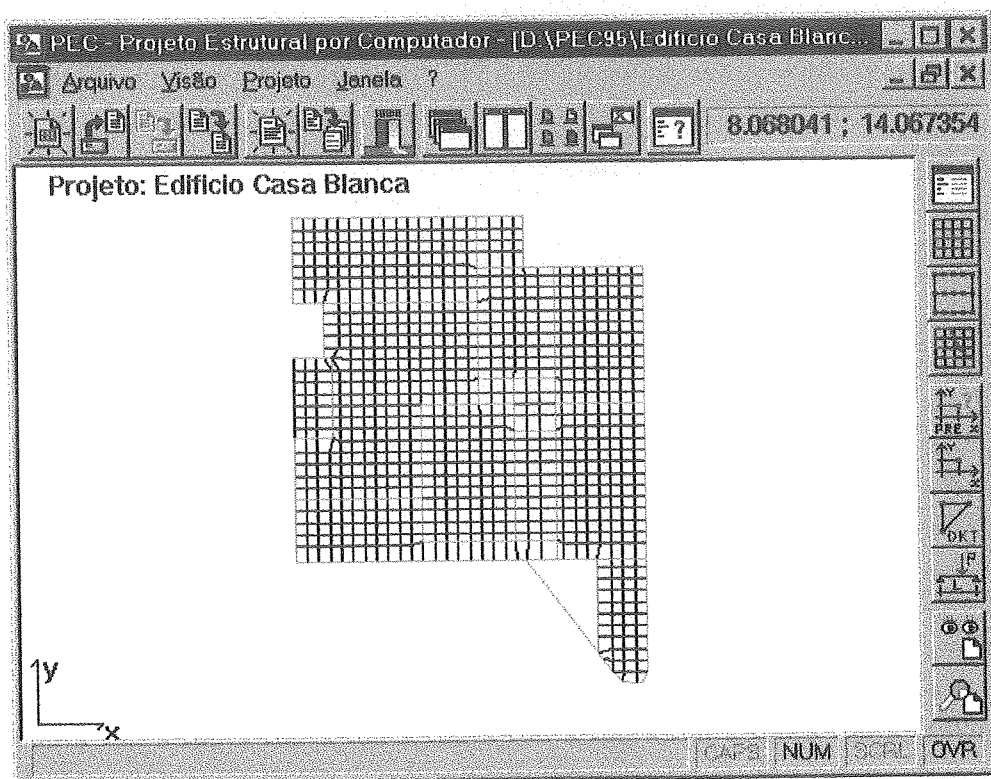


Figura 72 - Malha do Edifício Casa Blanca

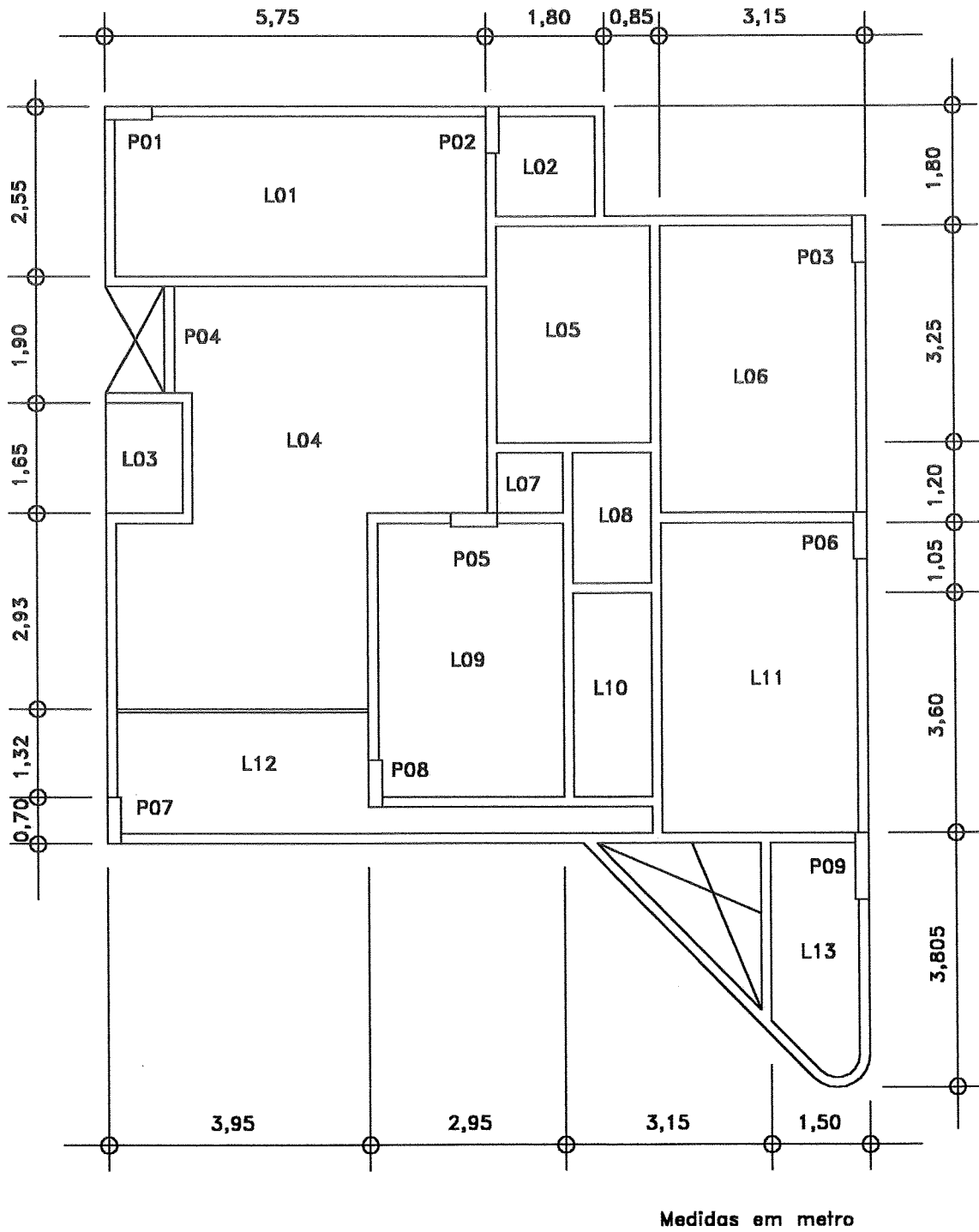


Figura 73 - Planta de fôrma do Edifício Casa Blanca



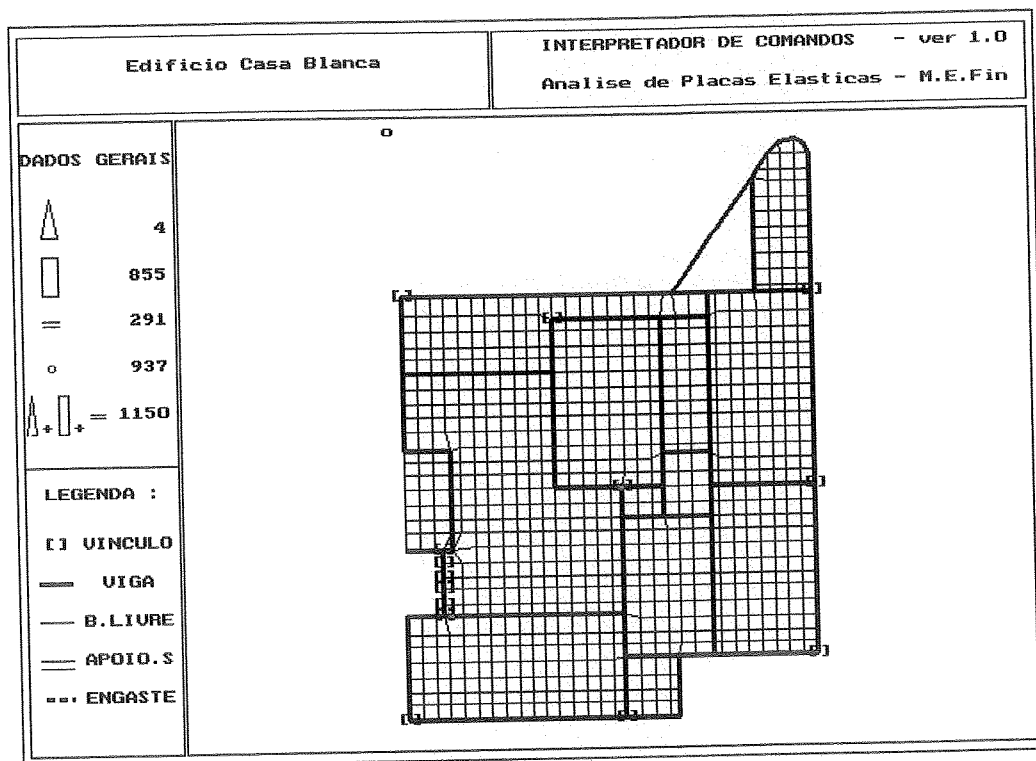


Figura 74 - Malha do Edifício Casa Blanca vista pelo sistema NEXP

### Listagem 6 - Arquivo de resultados do sistema NEXP - Edifício Casa Blanca

SISTEMA NEXP  
SOFTWARE PARA USO DIDÁTICO

Autor : Eng. Marcelo Novaes de Rezende  
Orientador : Prof. Dr. Joao Batista de paiva  
Departamento de Estruturas EESC USP 1993  
Arquivo processado : Edifício Casa Blanca

#### DESLOCAMENTOS DOS PONTOS

PONTO	W	TETAx	TETAy
1	4.4388380345E-02	-1.5235557546E-02	1.1879631980E-02
2	4.0591381015E-02	-1.2855522311E-02	1.1533920975E-02
3	3.6651668300E-02	-1.1510825214E-02	1.2121632989E-02
4	4.3157430578E-02	-1.3176678879E-02	9.4818677585E-03
5	3.9942739904E-02	-1.2428092554E-02	1.0271802433E-02
6	3.6003451167E-02	-1.1740974326E-02	1.0517534011E-02
7	3.1661945922E-02	-1.0886930546E-02	1.3131998716E-02
8	4.0643661574E-02	-1.1545011213E-02	6.9232151789E-03
9	3.9093112187E-02	-1.1546201857E-02	8.6650876096E-03
10	3.5530376258E-02	-1.1238653979E-02	9.7953363054E-03
.			
.			
.			
310	1.7262714214E-02	3.5047458256E-03	4.7591178170E-03
311	1.5316111884E-02	2.5628343164E-03	5.4331361418E-03

312 1.3158575034E-02 1.5575230337E-03 5.8316411876E-03  
 313 1.0927601000E-02 6.9878153545E-04 5.8605330461E-03  
 314 8.7336880845E-03 7.7644679099E-05 5.6971737484E-03  
 315 6.6136172814E-03 -3.7521691918E-04 5.4854882184E-03  
 316 4.5813281596E-03 -6.9417555151E-04 5.2187033647E-03  
 317 2.6779766532E-03 -8.4985993100E-04 4.7649290459E-03  
 318 1.0177220493E-03 -8.1726195910E-04 3.8655079384E-03  
 319 -9.7908183311E-05 -7.4329357555E-04 2.3486413799E-03  
 320 -6.2264231851E-04 -8.6072592997E-04 2.8591287665E-04

929 1.6654307859E-03 -3.7381097151E-03 1.1499691716E-03  
 930 1.2106501898E-03 -4.0003850955E-03 1.2291545687E-03  
 931 7.4319884847E-04 -4.2266577762E-03 1.2141514916E-03  
 932 3.0166561713E-04 -4.4332154415E-03 1.0904170165E-03  
 933 0.0000000000E+00 -4.5962329358E-03 9.0755287566E-04  
 934 -3.4927850509E-04 -4.9293971294E-03 6.3312492088E-04  
 935 -5.6195748630E-04 -5.1007642805E-03 4.9881206238E-04  
 936 -7.3696680182E-04 -5.1282114008E-03 4.3242148159E-04  
 937 -8.5554566243E-04 -5.0838882062E-03 4.1906042444E-04

## MOMENTOS NA PLACA

PONTO	MX	MY	MX Y
1	-7.8444258756E+00	-3.4675607413E+01	-1.3798061370E+01
2	6.0068609583E-01	-1.0291379703E+01	-7.3687538722E+00
3	3.2288769472E+00	3.7703332541E+00	-3.1552260119E+00
4	8.9552238105E+00	-1.4029762520E+01	-4.5235831983E+00
5	-2.5876463850E+00	-9.1598826970E+00	-4.5878065457E+00
6	1.2311614642E+00	-3.1756890535E+00	-3.3076474124E+00
7	3.0568646759E+01	4.4326398551E+00	-6.5107465717E+00
8	1.0655176539E+01	-1.6003449395E+01	-6.3827147885E+00
9	9.5545507990E+00	-6.6134190536E+00	-1.5050474171E+00
10	1.8233698310E+00	-3.3529090796E+00	-1.0977115613E+00
310	5.9713508102E+00	8.1406928025E+00	4.3479805235E+00
311	5.4519294280E+00	9.7111525759E+00	4.8631975734E+00
312	2.8921246419E+00	9.9496591599E+00	4.6580370335E+00
313	8.9966506117E-01	9.1795194950E+00	3.5704937575E+00
314	2.8932064976E-01	7.6021064513E+00	2.5837731673E+00
315	-1.2783629523E-01	5.7124172244E+00	1.8847178047E+00
316	-9.9874111925E-01	3.8099928496E+00	1.1869131156E+00
317	-2.9481033000E+00	2.0157855433E+00	3.1074148916E-01
318	-6.8561213094E+00	4.3939780108E-01	-3.6641159827E-01
319	-1.2898041991E+01	-9.7385818377E-01	-1.2608047135E-02
320	-1.0411668279E+01	-4.9505072308E-01	1.1595048195E+00
929	7.0443084431E-01	-1.0494728809E-01	1.3872318517E+00
930	1.9133552535E-01	-1.0466745390E-01	1.1936850979E+00
931	-4.1525632011E-01	-1.2591335201E-01	1.0567141510E+00
932	-1.1048129097E+00	-1.4591092800E-01	1.0276356855E+00
933	-1.7351295756E+00	-3.4253002230E-01	1.2162433228E+00
934	-1.0102342255E+00	4.1332154671E-02	1.1350501016E+00
935	-5.3487513203E-01	1.8811751031E-01	5.0877387829E-01
936	-1.9006404278E-01	1.2151894636E-01	-8.9776226352E-02
937	1.0880932960E-03	1.0579247848E-03	-2.8268460760E-01

## ESFORÇOS VERTICAIS RESULTANTES NOS PONTOS

PONTO	ESFORÇO
1	5.9604644775E-08
2	-1.9073486328E-06

3 -1.9073486328E-06  
 4 -2.0861625671E-07  
 5 -4.6507921070E-08  
 6 -4.9039954320E-09  
 7 -1.9073486328E-06  
 8 -1.4454126358E-06  
 9 4.4728949433E-08  
 10 -5.2350515034E-08

310 -4.1691237129E-08  
 311 -4.5838532969E-09  
 312 -3.2494426705E-08  
 313 -5.0931703299E-09  
 314 4.4237822294E-09  
 315 -3.1111994758E-08  
 316 -2.3719621822E-09  
 317 -1.5854311641E-08  
 318 3.6634446587E-09  
 319 2.9802322388E-08  
 320 -2.4811015464E-09

929 -1.4901161194E-07  
 930 2.0489096642E-07  
 931 6.3329935074E-08  
 932 -8.1956386566E-08  
 933 -1.8918784857E+02  
 934 2.0489096642E-08  
 935 7.6368451118E-08  
 936 3.4272670746E-07  
 937 7.4505805969E-08

VIGA ==>V1

Pi	Pf	ESF1	ESF2	ESF3	ESF4	ESF5	ESF6
60	61	0.0161	0.0182	-0.0073	-0.0161	-0.0182	0.0012

VIGA ==>V12

Pi	Pf	ESF1	ESF2	ESF3	ESF4	ESF5	ESF6
71	72	-57.0442	-0.0501	90.5553	57.0442	0.0501	-68.8785

---

## 6.4 Folha poliédrica

É gerado no PEC uma malha de elementos finitos de uma folha poliédrica e mostrado na figura 75. Como o sistema PEC ainda não dispõe de recursos para visualização tridimensional, foi gerado, através do gerador de relatórios, o arquivo DXF e este é visto na figura 76 obtida com o AutoCAD.

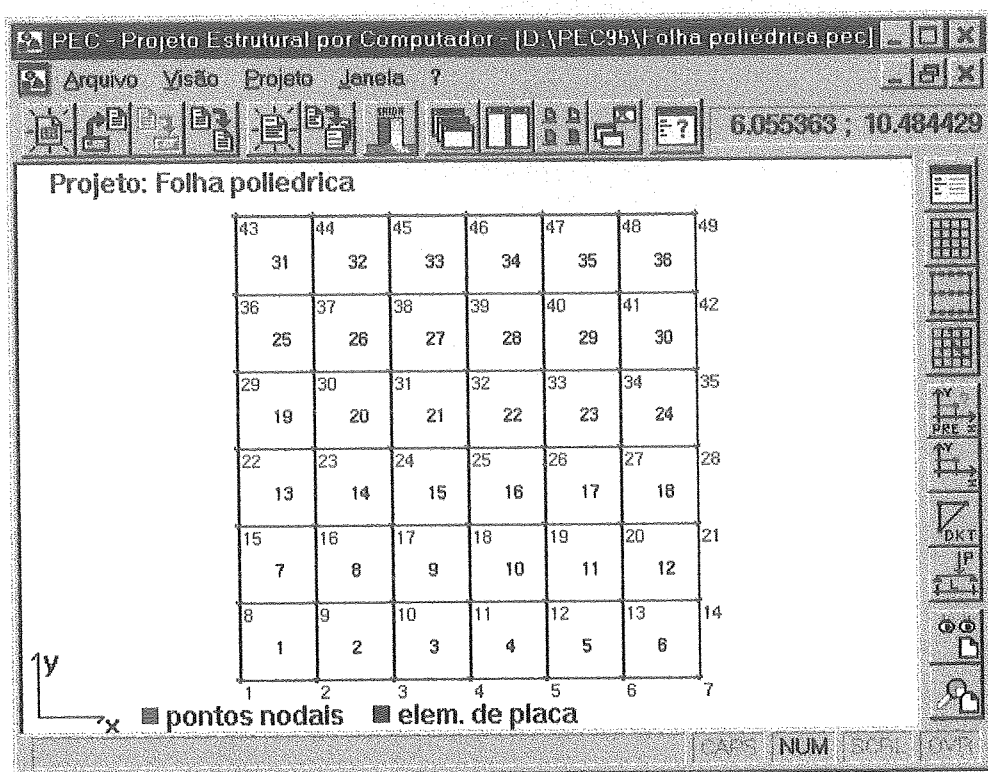


Figura 75 - Malha da folha poliédrica

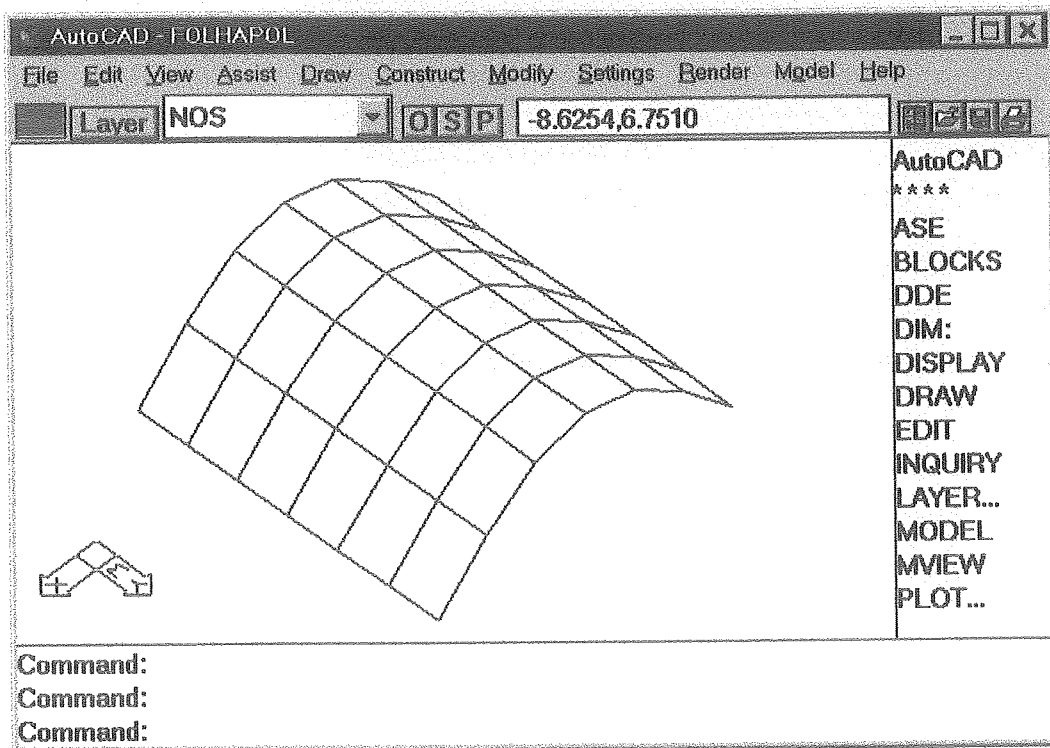


Figura 76 - Malha da folha poliédrica vista no AutoCAD

## 7 CONCLUSÕES

Neste trabalho foi apresentado o aplicativo PEC, um pré-processador de dados para a aquisição de dados para a análise de pavimentos de edifícios via Método dos Elementos Finitos. Os diversos exemplos apresentados mostraram a versatilidade do aplicativo na modelagem de pavimentos de edifícios de diferentes concepções arquitetônicas.

A interface do aplicativo PEC com o usuário demonstrou ser de fácil utilização além de oferecer as ferramentas necessárias para definir e manipular malhas de elementos finitos. A eficiência do **Gerador de Malha** foi comprovada com a geração da malha de vários pavimentos de formas variadas que continham curvas, vigas em diagonal e lajes nervuradas.

O desempenho do aplicativo foi considerado satisfatório tanto no tempo de processamento quanto no volume de dados que o aplicativo suporta trabalhar. Em um computador pessoal Pentium 100 MHz com 16 MB de RAM uma malha retangular composta por 10.000 elementos finitos de placa quadrilaterais e 10.201 pontos nodais demora 1 minuto e 15 segundos para ser gerada. No mesmo equipamento, uma malha retangular composta por 20.000 elementos finitos de placa triangulares e 10.201 pontos nodais demora 1 minuto e 44 segundos para ser gerada. Isto porque o aplicativo pode utilizar toda a memória convencional disponível no equipamento em que estiver sendo executado e o limite do tamanho das matrizes que forem utilizadas no aplicativo estará relacionado à quantidade de memória RAM disponível e ao tamanho do arquivo de troca que o Windows estiver

utilizando, ou seja, o limite estará relacionado ao *hardware* e não ao *software*.

A eficiência e versatilidade do **Gerador de Relatórios** foram comprovadas através da geração de vários relatórios de entrada de dados para programas de análise estrutural e editores gráficos.

Os comandos do **Gerador de Relatórios** na definição dos arquivos de formatos são de fácil utilização e não foi encontrada dificuldade na definição dos arquivos de formato dos exemplos testados neste trabalho. Normalmente o usuário irá defrontar-se com os comandos do **Gerador de Relatórios** apenas uma vez para cada programa de análise estrutural ou programa gráfico, pois uma vez definido um arquivo de formato ele poderá ser utilizado para inúmeras malhas quantas vezes forem preciso.

Como continuidade deste trabalho, pode-se sugerir o desenvolvimento de um assistente para o **Gerador de Relatórios**, ou seja, um programa ou uma janela que possa gerar um arquivo de formato contendo comandos do **Gerador de Relatórios** em que o usuário não precise saber o nome das variáveis do sistema e utilize apenas o *mouse* para definir a ordem e a posição dos campos na folha do relatório. Uma outra sugestão, seria aperfeiçoar o **Gerador de Relatórios** de tal forma que ele pudesse gerar arquivos de formato de leitura de dados, para serem utilizados pelo aplicativo PEC, para a leitura de arquivos de dados gerados por outros *softwares*. Isso permitiria um aproveitamento de dados de pavimentos já analisados. E que fossem acrescentados ao PEC recursos de rotação do objeto na tela gráfica para facilitar sua visualização em três dimensões, outros tipos de geradores de malha e ainda implementá-lo para a aquisição de dados para a análise de chapas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ANDRADE, J. R. L. (1982). *Estruturas correntes de concreto armado - parte 1*. São Carlos. Apostila. EESC, USP.
- AUTOCAD REFERENCE MANUAL, release 10. Autodesk, Inc., Feb. 1989.
- BAEHMANN, P. L.; WITTCHEN, S. L.; SHEPHARD, M. S.; GRICE, K.R.; YERRY, M. A. (1987). Robust, geometrically based, automatic two-dimensional mesh generation. *International Journal for Numerical Methods in Engineering*, v.24, n.6, p.1043-1078, June.
- BAPTISTA, S. M. (1994). *Análise de pavimentos de edifícios com a utilização do Método dos Elementos Finitos*. São Carlos. Tese (Mestrado). EESC, USP.
- BORLAND OBJECT WINDOWS PROGRAMMERS GUIDE, version 2.5. Borland International, Inc., Abr. 1995.
- CAVENDISH, J. C. (1974). Automatic triangulation of arbitrary planar domains for the Finite Element Method. *International Journal for Numerical Methods in Engineering*, v.8, p.679-696.
- 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.
- 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.
- GHASSEMI, F. (1982). Automatic mesh generation scheme for a two- or three-dimensional triangular curved surface. *Computers & Structures*, v.15, n.6, p.613-626.
- GRANERO, A. F. (1995). *Programação orientada para o objeto em C++ no ambiente Windows*. São Paulo, Atlas.
- GUREWICH, O. (1995). *Programação multimídia com Borland C++*. São Paulo, Makron Books.
- HABER, R.; SHEPHARD, M. S.; ABEL, J. F.; GALLAGHER, R. H.; GREENBERG, D. P. (1981). A general two-dimensional graphical finite element preprocessor utilizing Discrete Transfinite Mappings.



- International Journal for Numerical Methods in Engineering*, v.17, p.1015-1044.
- HEINY, L. (1993). *Programação gráfica para Windows com Borland C++*. Rio de Janeiro, Livros Técnicos e Científicos.
- KULLAA, J; KLINGE, P. (1995). A geometry based generation of a finite element model for stiffened shell structures. *Computers & Structures*, v.54, n.5, p.979-987.
- LIU, Y.; CHEN, K. (1987). An improved technique for eliminating duplicate nodes in a two-dimensional mesh generator. *Computers & Structures*, v.27, n.2, p.317-319.
- LIU, Y.; CHEN, K. (1988). Modifications of the program "A versatile two-dimensional mesh generator with automatic bandwidth reduction" for superelements sharing more than one boundary. *Computers & Structures*, v.28, n.2, p.305-307.
- NORTON, P.; YAO, P. (1992). *Programando em Borland C++ para Windows*. Rio de Janeiro, Berkeley.
- PAIVA, J. B. (1980). *Instabilidade de folhas planas com vinculações elásticas*. São Carlos. Tese (Mestrado). EESC, USP.
- PAPPAS, C. H.; MURRAY, W. H. (1994). *Borland C++ 4.0*. São Paulo, Makron Books.
- REZENDE, M. N. (1990). *Análise de pavimentos de edifícios pelo Método dos Elementos Finitos em microcomputador*. São Carlos. Tese (Mestrado). EESC, USP.
- 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.
- SANDER, R.; PITTMAN, J. F. T. (1995). Finite element analysis on PCs with an MS-Windows graphical user interface. *Engineering Computations*, v.12, p.35-60.
- SAP90 USERS MANUAL. Computers & Structures, Inc.
- SHEPHARD, M. S.; BAEHMANN, P. L.; GRICE, K. R. (1987). Automatic finite element modeling: geometry control for direct models. *Engineering Computations*, v.4, n.2, p.119-125, June.

- STEWART, M. G.; MELCHERS, R. E. (1989). Checking models in structural design. *Journal of Engineering Mechanics*, v.115, n.6, p.1309-1324.
- SWAN, T. (1994). *Programação avançada em Borland C++ 4 para Windows*. São Paulo, Berkeley.
- WALNUM, C. (1994). *Borland C++ 4.0.x dicas segredos & armadilhas*. Rio de Janeiro, Axcel Books.
- WU, E. (1980). Some findings in using the program, "A versatile two-dimensional mesh generator with automatic bandwich reduction". *Computers & Structures*, v.12, n.2, p.181-183.
- WU, E. (1982). Techniques to avoid duplicate nodes end relax restrictions on the superelement numbering in a mesh generator. *Computers & Structures*, v.15, p.419-422.
- YERRY, M. A.; SHEPHARD, M. S. (1983). A Modified Quadtree approach to finite element mash generation. *IEEE Computer Graphics & Applications*, v.3, n.1, p.39-46.
- ZIENKIEWICZ, O. C.; PHILLIPS, D. V. (1971). An automatic mash generation scheme for plane and curved surfaces by "Isoparametric" Coordinates. *International Journal for Numerical Methods in Engineering*, v.3, p.519-528.