

**UNIVERSIDADE FEDERAL DE MINAS GERAIS**  
**ESCOLA DE ENGENHARIA**  
**DEPARTAMENTO DE ENGENHARIA DE ESTRUTURAS**

**Automação do Processo de Modelamento e Detalhamento de Estruturas Pré-  
Moldadas de Concreto Armado Via Tecnologia CAD  
e Programação Orientada a Objeto**

Regina Célia Guedes Leite

Dissertação apresentada ao Curso de Pós Graduação em Engenharia de Estruturas da Escola de Engenharia da Universidade Federal de Minas, como parte dos requisitos necessários à obtenção do título de "Mestre em Engenharia de Estruturas".

Comissão Examinadora:

---

Prof. Dr. José Ricardo Queiroz Franco.  
DEES/UFMG – (Orientador)

---

Prof. Dr. Ney Amorim Silva.  
DEES/UFMG

---

Prof. Dr. Philippe R. B. Devloo  
UNICAMP

Belo Horizonte, 30 de Abril de 2002.

## DEDICATÓRIA

Ao meu Pai e Eterno Deus, que tantas bênçãos tem me concedido.

Ao meu marido, Edmilson, que me ensinou a acreditar em mim mesma.

Aos meus pais, Dorgival e Rosélis, que sempre acreditaram que a maior herança que se pode dar a um filho é a instrução.

## AGRADECIMENTOS

A Deus que me deu o CONHECIMENTO para ser alcançado,  
a SABEDORIA para discernir o melhor caminho e  
a FÉ para continuar apesar das dificuldades,

Ao Professor José Ricardo Queiroz Franco, pela orientação, pela exigência, pela  
compreensão e pelo apoio.

Ao meu marido, Edmilson, pelo seu AMOR, pelo grande apoio e pela enorme paciência.

Ao meu pai pelas longas revisões, discussões, questionamentos e orientações.

À minha mãe pelo apoio SEMPRE.

Ao meu irmão pelo seu exemplo e interesse.

À Renata e ao Marcelo pela valiosa ajuda, pela seriedade e pelo empenho nos trabalhos.

A Rosália pela companhia, pelo apoio e troca de experiências.

A todos do CADTEC pelo espírito de equipe sempre presente, por suas contribuições para  
este trabalho e para o crescimento do conhecimento do grupo como um todo.

Ao Professor Mounir Khalil El Debs por gentilmente ceder várias figuras de seu livro para  
inserção neste trabalho.

Aos demais colegas, professores e funcionários do DEES/UFMG, que, de alguma forma,  
colaboraram para o desenvolvimento desse trabalho.

## RESUMO

Esta Dissertação de Mestrado apresenta o projeto e o desenvolvimento de um sistema CAD (*Computer Aided Design*) para automatizar algumas etapas dos processos de modelamento e detalhamento de estruturas de concreto pré-moldado, desenvolvido segundo o modelamento sólido.

Este aplicativo é implementado através dos paradigmas da Programação Orientada e Objetos (POO) e com o uso da linguagem C++, utilizando o AutoCAD 2002 como plataforma gráfica. São aplicados os conceitos de objetos, entidades, banco de dados e eventos dentro do AutoCAD, utilizando a biblioteca ObjectARX. O sistema gerado foi denominado PREMOLD, que funciona como pré e pós-processador para a análise e detalhamento de estruturas de concreto pré-moldado a partir do lançamento dos elementos constituintes da estrutura. O aplicativo foi desenvolvido usando os fundamentos do Modelador 3D, um modelador geométrico unifilar desenvolvido para o lançamento de estruturas formadas por nós, barras e superfícies.

Este trabalho foi desenvolvido em duas etapas básicas, que são relatadas nessa dissertação. A primeira etapa consistiu de estudos e das reformulações do Modelador 3D. Nesta etapa são apresentados os estudos realizados sobre diretrizes e fundamentos de projetos de estruturas de concreto pré-moldado, sobre a linguagem C++ e sobre os paradigmas da POO. Na seqüência são relatadas as alterações e reestruturações, que foram implementadas no Modelador 3D para adequá-lo a esses fundamentos, diretrizes e paradigmas, possibilitando seu aproveitamento no presente trabalho. A segunda etapa consistiu no desenvolvimento conceitual e computacional do PREMOLD como um modelador sólido para estruturas de concreto pré-moldado, concebidas segundo o sistema reticulado com elementos de eixo reto, tendo como subproduto o modelo unifilar gerado inicialmente pelo Modelador 3D. É, então, apresentada a estrutura de classes desenvolvida para o PREMOLD. Essa estrutura de classes foi concebida objetivando-se gerar uma base para o desenvolvimento de novos aplicativos para estruturas que possam ser representadas por elementos sólidos, segundo sistemas estruturais formados por nós, barras e placas. O PREMOLD foi conceitualmente desenvolvido, pensando numa estrutura que permitisse seu aprimoramento, com a inserção de novas funcionalidades para as classes criadas e de novas classes, sem prejuízo para aquelas implementadas. Novos comandos e novas funções também poderão ser inseridos. O sistema CAD PREMOLD pretende ser flexível e independente de tal forma que interfaces simples permitam sua comunicação com sistemas CAE (*Computer Aided Engineering*) como um pré-processador e com sistemas CAM (*Computer Aided Manufacturing*) como um pós-processador. Estas são as fases do processo produtivo para estruturas pré-moldadas de concreto, as quais este projeto de pesquisa se propõe a automatizar.

Finalmente são abordadas as contribuições deste trabalho em diversas áreas, as limitações do aplicativo gerado, assim como sugestões e propostas para a continuidade do mesmo.

CONCRETO PRÉ-MOLDADO, MODELAMENTO, DETALHAMENTO, AUTOMAÇÃO, TECNOLOGIA CAD, POO, OBJECTARX.

## ABSTRACT

This Master Dissertation presents the design and implementation of a CAD (Computer Aided Design) system for automation of some steps of the process of modeling and detailing of pre-molded concrete structures, based on solid modeling.

This system is implemented using the Object Oriented Programming paradigm (OOP) with the C++ language, using AutoCAD 2002 as the graphical platform. The concepts of objects, entities, databases and events are applied in AutoCAD, using the ObjectARX library. The resulting system was named "PREMOLD" which is intended to work as a pre- and post-processor for the analysis and detailing of pre-molded concrete structures starting from the launching of the structural elements. The application was constructed using the fundamental features of another modeler named "Modelador 3D" for unifilar structural geometrical modeling based on elements such as nodes, bars and surfaces.

This work was developed in two basic steps, which are described in this dissertation. The first step consisted of the study and reformulation of the "Modelador 3D" so that it could be used for this work. Studies about design directives and fundamental requirements for pre-molded concrete structures, the C++ language and OOP paradigms are then presented. Next, the necessary changes and reorganization of the application "Modelador 3D" to satisfy those fundamental, directives and paradigms are described. The second step consisted of the computational and conceptual definition of the application "PREMOL" as a solid modeler for pre-molded structures, designed for skeleton structural systems or for unifilar models. The structure of classes developed for the "PREMOLD" is then presented. Such a class structure was constructed to be the basis for new developments on structural modeling using solid elements defined by nodes, bars and plates. The application "PREMOLD" was constructed to allow new implementations, such as the inclusion of new functionalities for the existent classes and creation of new classes. The insertion of new commands and functions will also be permitted. The CAD system "PREMOLD" is intended to be flexible and independent so that simple interfaces will allow the communication with CAE system (Computer Aided Engineering) as a pre-processor and with CAM systems (Computer Aided Manufacturing) as a post-processor. Those are the phases of the productive process for pre-molded concrete structure, which this research work propose to automate.

Finally, the contributions of this work to other areas, the limitations of the resulting application at this point are discussed, with suggestions for future work.

PRE-MOLDED CONCRETE, MODELLING, DETAILING, AUTOMATION, CAD TECNOLOGY, OOP, OBJECTARX.

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>16</b>
1.1 OBJETIVOS DO PROJETO.....	16
1.2 A INDÚSTRIA DE PRÉ-MOLDADOS NO PANORAMA DA CONSTRUÇÃO CIVIL .....	19
1.3 A COMPUTAÇÃO APLICADA À ENGENHARIA DE ESTRUTURAS: SISTEMAS CAD, CAE E CAM ( A AUTOMAÇÃO DO PROCESSO PRODUTIVO DE ESTRUTURAS DE CONCRETO PRÉ-MOLDADO) .....	22
1.4 ETAPAS DO PROCESSO DE PRODUÇÃO COM O USO DAS TECNOLOGIAS CAD/CAE .....	27
1.5 MODELAMENTO GEOMÉTRICO X MODELAMENTO ESTRUTURAL .....	28
1.6 ETAPAS DE DESENVOLVIMENTO DO PROJETO DE PESQUISA .....	33
1.7 CONTEÚDO POR CAPÍTULOS.....	33
<b>2 FUNDAMENTOS PARA PROJETOS DE ESTRUTURAS DE CONCRETO PRÉ-MOLDADO .....</b>	<b>35</b>
2.1 O CONCRETO PRÉ-MOLDADO .....	37
2.2 Os COMPONENTES DE SISTEMAS ESTRUTURAIS DE EDIFICAÇÕES EM CONCRETO PRÉ-MOLDADO .....	39
2.2.1 Componentes dos Sistemas Reticulados.....	39
2.2.2 Componentes do Sistema de Pavimentos .....	41
2.2.3 Componentes de Sistemas de Paredes.....	43
2.2.4 Outros Componentes .....	44
2.3 SISTEMAS ESTRUTURAIS PARA EDIFÍCIOS DE UM PAVIMENTO .....	45
2.3.1 Sistema Estrutural Reticulado com Elemento de Eixo Reto .....	45
2.3.2 Sistemas Estruturais com Elemento de Trechos de Eixo Reto ou Curvo.....	46
2.3.3 Sistemas Estruturais de Parede Portante .....	48
2.4 SISTEMAS ESTRUTURAIS PARA EDIFÍCIOS DE MÚLTIPLOS PAVIMENTOS .....	48
2.4.1 Sistema Estrutural Reticulado com Elemento de Eixo Reto .....	49
2.4.2 Sistema Estrutural Reticulado com Elemento Composto por Trecho de Eixo Reto.....	51
2.4.3 Sistemas Estruturais Reticulados em Pavimentos sem Vigas.....	53
2.4.4 Sistemas Estruturais de Parede Portante com Grandes Painéis de Fachada....	54
2.4.5 Sistemas Estruturais de Parede Portante com Painéis da Altura do Andar .....	55

2.4.6 <i>Sistemas Estruturais de Parede Portante com Elementos Tridimensionais</i> .....	56
2.5 DIRETRIZES E PARÂMETROS PARA PROJETO DE ESTRUTURAS EM CONCRETO PRÉ-MOLDADO .....	56
2.6 ASPECTOS A SEREM CONSIDERADOS NO PROJETO E NA ANÁLISE DE ESTRUTURAS DE CONCRETO PRÉ-MOLDADO .....	58
2.7 CONSIDERAÇÕES SOBRE AS LIGAÇÕES ENTRE ELEMENTOS DE UMA ESTRUTURA DE CONCRETO PRÉ-MOLDADO .....	60
2.7.1 <i>Ligações entre Pilar e Fundação</i> .....	66
2.7.2 <i>Ligações entre Pilares</i> .....	68
2.7.3 <i>Ligações entre Viga e Pilar e entre duas Vigas</i> .....	69
2.7.4 <i>Ligações entre Elementos de Laje e Parede</i> .....	74
<b>3 FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO DO PROJETO</b>	
"MODELADOR PREMOLD" .....	<b>76</b>
3.1 AUTOCAD 2002: BANCO DE DADOS E SISTEMA GRÁFICO .....	77
3.2 A LINGUAGEM C++ E A PROGRAMAÇÃO ORIENTADA A OBJETOS - POO.....	79
3.3 O AMBIENTE DE PROGRAMAÇÃO OBJECTARX .....	87
3.3.1 <i>Criando um Aplicativo ARX</i> .....	88
3.3.2 <i>Banco de Dados</i> .....	88
3.3.3 <i>Manipulação de Objetos do Banco de Dados</i> .....	89
3.3.4 <i>Objetos e Entidades Personalizados</i> .....	90
<b>4 MODELADOR 3D: O PONTO DE PARTIDA</b> .....	<b>93</b>
4.1 DESCRIÇÃO E FUNCIONALIDADE DO MODELADOR 3D .....	94
4.2 ESTRUTURA GLOBAL DO APLICATIVO .....	97
4.2.1 <i>Classes Básicas: Node, Bar, Face</i> .....	97
a) <i>Classe Node</i> .....	100
b) <i>Classe Bar</i> .....	102
c) <i>Classe Face</i> .....	105
4.2.2 <i>Classes De Controle: Config1, Casos, CurrCaso, ReactroNode, ReactorBar, ReactorFace, CommandReactor</i> .....	107
a) <i>Numeração da Estrutura: Classe Config1</i> .....	108
b) <i>Casos de Carregamento: Classe Casos</i> .....	109
c) <i>Carregamento Corrente: Classe Curr caso</i> .....	110

d) Reator Da Entidade Node: Classe Reactornode.....	110
e) Reator Da Entidade Bar: Classe Reactorbar .....	111
f) Reator Da Entidade Face: Classe Reactorface.....	111
g) Monitoramento De Comandos Do AutoCAD: Classe CommandReactor .....	111
4.2.3 Funções Globais .....	112
4.2.4 Comandos do Aplicativo .....	116
a) Comandos Globais.....	116
b) Comandos Relativos a Entidade Node:.....	117
c) Comandos Relativos a Entidade Bar: .....	117
d) Comandos Relativos a Entidade Face .....	117
4.3 MODULARIDADE ORIGINAL X NOVA ESTRUTURAÇÃO DO CÓDIGO .....	118
4.4 ESTRUTURA DE CLASSES ORIGINAL E ALTERAÇÕES INICIAIS .....	121
4.5 NOVA ESTRUTURA DE CLASSES .....	126
4.5.1 Classe CdTcCNode.....	126
4.5.2 Classe CdTcCBar.....	128
4.5.3 Classe CdTcCPlaca.....	131
4.5.4 Classe CdTcCDictEnum.....	132
4.5.5 Classe CdTcCCasos .....	133
4.5.6 Classes CdTcCReactorNode, CdTcCReactorBar, CdTcCReactorPlaca.....	133
4.5.7 Classe CdTcCCommandReactor .....	134
4.6 REESTRUTURAÇÃO DAS FUNÇÕES GLOBAIS .....	135
<b>5 O MODELADOR PREMOLD .....</b>	<b>136</b>
5.1 DESENVOLVIMENTO CONCEITUAL DO PREMOLD: CONCEPÇÃO E DEFINIÇÕES GERAIS DO APLICATIVO .....	137
5.1.1 Escolha do Sistema Estrutural a Ser Modelado .....	137
5.1.2 Delimitação do Produto Pretendido .....	138
5.1.3 Tipos de Elementos a Serem Modelados com Levantamento dos Dados Essenciais para sua Representação.....	139
5.1.4 Critérios para o Lançamento da Estrutura.....	141
5.1.5 Tipos de Ligações Modeladas e seu Tratamento Inicial.....	142
5.2 DESENVOLVIMENTO COMPUTACIONAL DO PREMOLD .....	142
5.2.1 Estrutura de Classes do PREMOLD .....	143
5.2.2 Descrição das Classes do PREMOLD .....	147

a) Classe CdTcCElem_Estrutural.....	148
b) Classe CdTcCElem_Linear.....	151
c) Classe CdTcCElem_Plano.....	151
d) Classe CdTcCBloco.....	152
e) Classe CdTcCConsolo.....	154
f) Classe CdTcCPilar.....	154
g) Classe CdTcCViga.....	156
h) Classe CdTcCPainel.....	158
i) Classe CdTcCDictElemt .....	160
j) Classe CdTcCDictNode .....	160
k) Classe CdTcCDictBar.....	161
l) Classe CdTcCFormato.....	162
m) Classe CdTcCLegenda.....	163
n) Casse CdTcCNota .....	164
5.2.3 Adição de funcionalidade às classes reestruturadas do Modelador 3D.....	165
a) Classe CdTcCReactorNode.....	165
b) Classe CdTcCReactorBar.....	165
c) Classe CdTcCReactorPlaca.....	166
d) Classe CdTcCCommandReactor.....	166
5.2.4 Funções Globais do PREMOLD .....	167
a) Funções declaradas e definidas nos arquivos DictUtils do projeto DBX. ....	167
b) Funções declaradas e definidas nos arquivos DictUtils do projeto ARX.....	168
c) Funções declaradas e definidas nos arquivos Utils dos projetos DBX e ARX. ....	168
d) Funções declaradas e definidas nos arquivos GeraSecao do projeto DBX. ....	169
5.2.5 Os Comandos Do PREMOLD .....	170
a) Comando "GeraMalha":.....	171
b) Comando "GeraPilar":.....	173
c) Comando "GeraViga":.....	177
d) Comando "GeraPainel":.....	177
e) Comando "GeraBloco":.....	188
f) Comando "GeraRelatorio":.....	190
g) Comando "GeraFormato":.....	192
h) Comandos "GeraLegenda" e "EditaLegenda":.....	193

i) Comandos "GeraNota" e "EditaNota": .....	195
j) Comando "GeraDesenho": .....	197
k) Comando GeraCarimbo:.....	198
<b>6 CONSIDERAÇÕES FINAIS .....</b>	<b>200</b>
6.1 O TRABALHO DESENVOLVIDO E AS CONTRIBUIÇÕES GERADAS .....	200
6.1.1 Contribuições Gerais para o grupo CADTEC .....	201
6.1.2 Contribuições Específicas para a Automação de Processo Produtivos .....	202
6.1.3 Contribuições Específicas para a Indústria de Concreto Pré-Moldado.....	203
6.2 LIMITAÇÕES DO APLICATIVO E PROPOSTAS PARA TRABALHOS FUTUROS.....	204

## ÍNDICE DE FIGURAS

FIGURA 1.1 - AUTOMAÇÃO DO PROCESSO PRODUTIVO DE ESTRUTURAS EM CONCRETO PRÉ-MOLDADO .....	26
FIGURA 2.1 - SEÇÕES TRANSVERSAIS UTILIZADAS NOS PILARES [EL DEBS, 2000] .....	40
FIGURA 2.2 - SEÇÕES TRANSVERSAIS MAIS UTILIZADAS NAS VIGAS [EL DEBS, 2000] .....	41
FIGURA 2.3 - OUTRAS FORMAS DE SEÇÕES TRANSVERSAIS UTILIZADAS NAS VIGAS (EL DEBS, 2000).....	41
FIGURA 2.4 - SEÇÕES MAIS COMUNS PARA PAINÉIS DE LAJE E DE PAREDE (EL DEBS, 2000) .....	42
FIGURA 2.5 - SISTEMA DE PAVIMENTOS COM VIGA E PAINÉIS ALVEOLARES OU DUPLO T (TT) (EL DEBS, 2000) .....	43
FIGURA 2.6 - SISTEMAS DE CONTRAVENTAMENTO COM PAREDES E COM NÚCLEO (EL DEBS, 2000).....	43
FIGURA 2.7 - BLOCO DE FUNDAÇÃO COM COLARINHO E SAPATA (EL DEBS, 2000) .....	44
FIGURA 2.8 - FORMAS BÁSICAS DOS SISTEMAS ESTRUTURAIS COM ELEMENTOS DE EIXO RETO (EL DEBS, 2000) .....	46
FIGURA 2.9 - FORMAS BÁSICAS DOS SISTEMAS ESTRUTURAIS COM ELEMENTOS COMPOSTOS DE TRECHOS DE EIXO RETO (EL DEBS, 2000) .....	47
FIGURA 2.10 - FORMAS BÁSICAS DOS SISTEMAS ESTRUTURAIS COM ELEMENTOS DE EIXO RETO (EL DEBS, 2000) .....	50
FIGURA 2.11 – ESQUEMAS CONSTRUTIVOS COM ELEMENTOS DE EIXO RETO (EL DEBS, 2000) .....	51
FIGURA 2.12 - FORMAS BÁSICAS DOS SISTEMAS ESTRUTURAIS COM ELEMENTOS COMPOSTOS DE TRECHOS DE EIXO RETO (EL DEBS, 2000) .....	52
FIGURA 2.13 - ESQUEMAS CONSTRUTIVOS COM ELEMENTOS COMPOSTOS POR TRECHOS DE EIXO RETO (EL DEBS, 2000) .....	52
FIGURA 2.14 - FORMAS BÁSICAS DOS SISTEMAS ESTRUTURAIS EM PAVIMENTOS SEM VIGAS (EL DEBS, 2000) .....	53
FIGURA 2.15 - FORMAS BÁSICAS E ESQUEMA CONSTRUTIVO COM GRANDES PAINÉIS DE FACHADA (EL DEBS, 2000).....	54
FIGURA 2.16 - ESQUEMAS CONSTRUTIVOS COM GRANDES PAINÉIS DA ALTURA DO PAVIMENTO (EL DEBS, 2000) .....	55

FIGURA 2.17 - ESQUEMAS CONSTRUTIVOS COM ELEMENTOS TRIDIMENSIONAIS (EL DEBS, 2000) .....	56
FIGURA 2.18 - COMPORTAMENTO DA LAJE COMO DIAFRAGMA (EL DEBS, 2000) .....	60
FIGURA 2.19 - PARÂMETROS PARA DEFINIÇÃO DO TIPO DE DIMENSIONAMENTO DOS CONSOLOS SEGUNDO A NBR 9062/85 (EL DEBS, 2000) .....	63
FIGURA 2.20 - CLASSIFICAÇÃO DAS LIGAÇÕES EM ELEMENTOS TIPO BARRA (EL DEBS, 2000) .....	65
FIGURA 2.21 - LIGAÇÃO PILAR X FUNDAÇÃO POR MEIO DE CHAPA DE BASE (EL DEBS, 2000) .....	66
FIGURA 2.22 - LIGAÇÃO PILAR X FUNDAÇÃO COM EMENDA DA ARMADURA COM GRAUTE E BAINHA E VARIANTE COM CONCRETO OU GRAUTE (EL DEBS, 2000) .....	66
FIGURA 2.23 - LIGAÇÕES PILAR X PILAR (EL DEBS, 2000) .....	69
FIGURA 2.24 - LIGAÇÕES VIGA X PILAR RÍGIDAS COM SOLDA (EL DEBS, 2000) .....	70
FIGURA 2.25 - LIGAÇÕES VIGA X PILAR RÍGIDAS COM EMENDA DA ARMADURA E CONCRETO MOLDADO NO LOCAL (EL DEBS, 2000) .....	70
FIGURA 2.26 - LIGAÇÕES VIGA X PILAR RÍGIDAS COM CABOS DE PROTENSÃO (EL DEBS, 2000) .....	71
FIGURA 2.27 - FORMAS DE ESTRANGULAMENTO DOS PILARES TENDO EM VISTA A LIGAÇÃO COM AS VIGAS OU LAJES (EL DEBS, 2000) .....	71
FIGURA 2.28 - LIGAÇÕES VIGA X PILAR ARTICULADAS (EL DEBS, 2000).....	72
FIGURA 2.29 - LIGAÇÕES VIGA X VIGA FORA DO PILAR (EL DEBS, 2000) .....	73
FIGURA 2.30 - LIGAÇÕES VIGA SECUNDÁRIA X VIGA PRINCIPAL (EL DEBS,2000) .....	73
FIGURA 2.31- CLASSIFICAÇÃO DAS LIGAÇÕES ENTRE ELEMENTOS DO TIPO FOLHA (EL DEBS, 2000).....	74
FIGURA 3.1 – ABSTRAÇÃO DE UMA ENTIDADE REAL ATRAVÉS DE SUAS CARACTERÍSTICAS	81
FIGURA 3.2 - ENCAPSULAMENTO: ATRIBUTOS + MÉTODOS .....	81
FIGURA 3.3 - CLASSE E OBJETO .....	82
FIGURA 3.4 - HERANÇA DE CLASSES .....	83
FIGURA 3.5 - ESTRUTURAS: GENERALIZAÇÃO/ESPECIALIZAÇÃO X TODO/PARTE .....	84
FIGURA 3.6 - POLIMORFISMO .....	85
FIGURA 4.1 - RELACIONAMENTO ENTRE CLASSES <i>FRIEND</i> E SUAS CLASSES DERIVADAS.....	99
FIGURA 4.2 - ESTRUTURA ORIGINAL DE CLASSES DO MODELADOR 3D.....	122
FIGURA 4.3 - NOVA ESTRUTURA DE CLASSES DO MODELADOR 3D .....	124

FIGURA 4.4 - ALTERAÇÕES REALIZADAS SOBRE AS CLASSES ORIGINAIS DO MODELADOR 3D .....	125
FIGURA 5.1 - ESTRUTURA DE CLASSES DO PREMOLD .....	147
FIGURA 5.2 – QUADRO DE DIÁLOGO DO COMANDO GERAMALHA. ....	172
FIGURA 5.3 – MALHA GERADA COM DIMENSÕES ESPECIFICADAS SEGUNDO A FIGURA 5.2. .....	173
FIGURA 5.4 – QUADRO DE DIÁLOGO DO COMANDO GERAPILAR PARA INSERÇÃO DE NOVE PILARES PELO SEU CENTRO DE GRAVIDADE, COM SEÇÃO PADRÃO DE 40CMX60CM E ALTURA DE 12M .....	174
FIGURA 5.5 – RELATÓRIO COM INDICAÇÃO DAS RESTRIÇÕES NODAIS DE QUATRO PILARES CRIADOS PELO COMANDO GERAPILAR. ....	176
FIGURA 5.6 – PILARES DEFINIDOS COMO MOSTRADO NA FIG. 5.4, INSERIDOS NO BANCO DE DADOS DO AUTOCAD E DO PREMOLD. ....	176
FIGURA 5.7 – QUADRO DE DIÁLOGO DO COMANDO GERAVIGA PARA INSERÇÃO DE VIGAS DO TIPO VTS SOBRE OS PILARES INTERNOS DA ESTRUTURA DA FIG. 5.5. ....	178
FIGURA 5.8 – RELATÓRIO COM DADOS SOBRE AS RESTRIÇÕES NODAIS DE NÓS INICIAL E FINAL DE PILARES E VIGAS CRIADOS PELO PREMOLD. ....	180
FIGURA 5.9 – VIGAS INSERIDAS NO PRIMEIRO PAVIMENTO DA ESTRUTURA DA FIG. 5.6... 181	
FIGURA 5.10 - VIGAS INSERIDAS NO SEGUNDO PAVIMENTO DA ESTRUTURA MOSTRADA NA FIG. 5.9. ....	181
FIGURA 5.11 - VIGAS INSERIDAS NO TERCEIRO PAVIMENTO DA ESTRUTURA MOSTRADA NA FIG. 5.10.....	182
FIGURA 5.12 – INSERÇÃO DE VIGAS DE FECHAMENTO DA ESTRUTURA DA FIG. 5.11. ....	182
FIGURA 5.13 – QUADRO DE DIÁLOGO DO COMANDO GERAPAINEL .....	184
FIGURA 5.14 – INSERÇÃO DE PAINÉIS DO TIPO ALVEOLAR NO PRIMEIRO, SEGUNDO E TERCEIRO PAVIMENTOS DA ESTRUTURA MOSTRADA NA FIG. 5.13. ....	186
FIGURA 5.15 – DETALHE DO ENCAIXE DE VIGAS COM PILARES E DO APOIO DE PAINÉIS SOBRE VIGAS DA ESTRUTURA DA FIG. 5.14.....	186
FIGURA 5.16 – OUTRO ÂNGULO DE VISÃO DA ESTRUTURA DA FIG. 5.14.....	187
FIGURA 5.17 – VISTA MAIS APROXIMADA DA ESTRUTURA MOSTRADA NA FIG.5.16.....	187
FIGURA 5.18 – QUADRO DE DIÁLOGO DO COMANDO GERABLOCO PARA INSERÇÃO DE NOVE BLOCOS SOB OS PILARES DA ESTRUTURA MOSTRADA NA FIG. 5.11. ....	189

FIGURA 5.19 – BLOCOS DE FUNDAÇÃO INSERIDOS SOB OS PILARES DA ESTRUTURA DA FIG. 5.11.....	190
FIGURA 5.20 – RELATÓRIO QUANTITATIVO TOTAL GERADO PELO PREMOLD.....	191
FIGURA 5.21 – FORMATOS GERADOS PELO COMANDO GERAFORMATO. ....	193
FIGURA 5.22 – QUADRO DE DIÁLOGO DO COMANDO GERALLEGENDA. ....	194
FIGURA 5.23 – LEGENDA CRIADA ATRAVÉS DO COMANDO GERALLEGENDA. ....	195
FIGURA 5.24 – QUADRO DE DIÁLOGO DO COMANDO GERANOTA. ....	196
FIGURA 5.25 – NOTA GERADA PELO COMANDO GERANOTA .....	196
FIGURA 5.26 – DESENHO DE FORMA DE UMA VIGA VJ, INSERIDA EM UMA ESTRUTURA GERADA PELO PREMOLD, CRIADO PELO COMANDO GERADESENHO. ....	198
FIGURA 5.27 – CARIMBO GERADO PELO COMANDO GERACARIMBO. ....	199
FIGURA 5.28 – DESENHO COMPLETO GERADO PELO COMANDO GERADESENHO, ACIONAR DENTRO DESSE OS COMANDO GERAFORMATO, GERALLEGENDA, GERACARIMBO E GERANOTA.....	199

## ÍNDICE DE TABELAS

TABELA 2.1 - ELEMENTOS PRÉ-MOLDADOS DE USO MAIS COMUM [EL DEBS, 2000] .....	38
TABELA 2.2 - VALORES DAS TOLERÂNCIAS INDICADOS PELA NBR 9062/85 (EL DEBS, 2000) .....	59
TABELA 2.3 - LIGAÇÕES EM ELEMENTOS TIPO BARRA (EL DEBS, 2000) .....	65
TABELA 2.4 - VALORES MÍNIMOS DO COMPRIMENTO DE EMBUTIMENTO DO PILAR SEGUNDO A NBR 9062/85 (EL DEBS, 2000) .....	67
TABELA 4.1 - ATRIBUTOS E MÉTODOS DA CLASSE NODE.....	101
TABELA 4.2 - ATRIBUTOS E MÉTODOS DA CLASSE BAR .....	103
TABELA 4.3 - ATRIBUTOS E MÉTODOS DA CLASSE FACE.....	105
TABELA 4.4 - ATRIBUTOS DA CLASSE CONFIG1 .....	109
TABELA 4.5 - ATRIBUTOS E MÉTODOS DA CLASSE CASOS .....	109
TABELA 4.6 - ATRIBUTOS E MÉTODOS DA CLASSE CURRCASO.....	110
TABELA 4.7 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCNODE.....	126
TABELA 4.8 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCBAR .....	129
TABELA 4.9 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCPLACA .....	131
TABELA 5.1 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCCELEM_ESTRUTURAL .....	148
TABELA 5.2 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCCELEM_LINEAR.....	151
TABELA 5.3 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCCELEM_PLANO .....	152
TABELA 5.4 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCBLOCO .....	153
TABELA 5.5 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCCONSOLO.....	154
TABELA 5.6 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCPILAR.....	155
TABELA 5.7 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCVIGA .....	157
TABELA 5.8 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCPAINEL.....	159
TABELA 5.9 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCDICTELEMENT.....	160
TABELA 5.10 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCDICTNODE.....	161
TABELA 5.11 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCDICTBAR .....	161
TABELA 5.12 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCFORMATO .....	162
TABELA 5.13 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCLEGENDA.....	163
TABELA 5.14 - ATRIBUTOS E MÉTODOS DA CLASSE CdTCCNOTA .....	164

# 1

## INTRODUÇÃO

### 1.1 Objetivos do Projeto

A presente pesquisa teve como meta inicial o desenvolvimento de um aplicativo CAD para o modelamento de estruturas de concreto pré-moldado, concebidas segundo o sistema reticulado para edificações de múltiplos pavimentos. Esse aplicativo foi desenvolvido com base no modelamento sólido, utilizando o AutoCAD como plataforma gráfica. Esse modelador foi desenvolvido para servir de base a um pré-processador e a um pós-processador capazes de integrar as várias etapas do processo produtivo de estrutura de concreto pré-moldado. Na busca pelo desenvolvimento desse aplicativo a presente pesquisa teve como principal contribuição a concepção de uma base computacional, com lançamento de uma estrutura de classes bem definida, capaz de suportar o aplicativo inicialmente desenvolvido e, ainda, novas implementações. Essas novas implementações

podem ser desenvolvidas com a inserção de novas classes, de outros sistemas estruturais, de novos elementos e de outros tratamentos para as ligações entre os elementos, além daqueles implementados no presente trabalho. Não foram contemplados, nesse aplicativo, o modelamento via método dos elementos finitos nem o dimensionamento da estrutura. O aplicativo resultante dos trabalhos até aqui desenvolvidos foi intitulado Modelador PREMOLD, ou apenas PREMOLD.

Foi estabelecido o tratamento de alguns tipos de elementos e de alguns tipos de ligações entre os mesmos. Propôs-se, ainda, a geração de um banco de dados contendo, além dos dados do modelo estrutural, outras características dos elementos que compõem a estrutura, relevantes para a integração do modelamento com as demais etapas do processo produtivo.

O desenvolvimento de um sistema global de automação do processo de produção de estruturas de concreto pré-moldado foi iniciado pelo lançamento de bases capazes de suportar o desenvolvimento de sistemas integrados contemplando desde a modelagem, análise, dimensionamento, detalhamento e fabricação dos componentes dessa estrutura. A estrutura de classes desenvolvida nesta etapa de trabalho foi idealizada para permitir novas implementações necessárias ao desenvolvimento completo de pré-processadores e pós-processadores capazes de fazer essa integração. Foi implementada a base de um banco de dados que será o responsável pela manutenção, pela transmissão e pelo gerenciamento de todos os dados necessários às demais etapas do processo produtivo. O PREMOLD vem ser a base para esses processadores possuindo a estrutura necessária para dar suporte às futuras implementações para o completo desenvolvimento dos mesmos.

No lançamento dos elementos estruturais automatizado pelo PREMOLD, são adotados os critérios estabelecidos pela NBR 9062/85, "Projeto e Execução de Estruturas de Concreto Pré-moldado" (ABNT, 1985). Permite-se, ainda, o lançamento de ligações entre elementos padronizados segundo essa norma e, ainda, segundo os critérios predefinidos pela empresa produtora de tais estruturas através da configuração de dados gerais a serem adotados nos projetos.

O modelo lançado através do PREMOLD contém os dados geométricos básicos da estrutura. A partir desse, é possível obter-se relatórios gerados automaticamente pelo

PREMOLD com todos os dados essenciais da estrutura gerada, possibilitando-se, ainda, a verificação dos dados não gráficos gerenciados pelo aplicativo.

Os princípios desta nova tecnologia CAD aplicada ao modelamento de estruturas foi iniciado pelos mestres Fernando Poinho Malard (MALARD, 1998) e Alexandra Hütner (HÜNTNER, 1998) sob orientação do Professor José Ricardo Queiroz Franco do Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais - DEES/UFMG. Como resultado daqueles trabalhos foi desenvolvido o Modelador 3D, para modelamento unifilar tridimensional de estruturas metálicas prediais. A presente pesquisa teve como base de trabalho esse modelador.

Para o desenvolvimento do PREMOLD, foi realizada uma reestruturação do Modelador 3D, com o lançamento de uma nova herança de classes para ele. Após a implementação dessa nova herança, o presente trabalho de pesquisa ampliou a estrutura de classes com a geração de novas classes capazes de representar uma estrutura de concreto pré-moldado através de seus elementos e de um sistema estrutural predefinido. As classes reestruturadas do Modelador 3D foram utilizadas para representar esse sistema estrutural através de nós, barras e superfícies ou placas. A representação dos elementos estruturais foi obtida através da modelagem sólida onde esses elementos são representados pelas classes desenvolvidas no presente trabalho de pesquisa. A partir de então, foi desenvolvido um modelador sólido, o PREMOLD, através da implementação de comandos que manipulam as classes criadas e aquelas reestruturadas. Esses comandos possibilitam o lançamento dos elementos que representam pilares, vigas, painéis de piso e blocos de fundação para estruturas de concreto pré-moldado. Também as ligações entre esses elementos são tratadas pelos comandos implementados, possibilitando a automatização da solução para interações e interferências entre os elementos. O lançamento do modelo estrutural, segundo um sistema previamente definido, é realizado pelo aplicativo ao mesmo em tempo que os elementos são gerados. Assim, o modelo unifilar surge como um subproduto do modelamento sólido. Esse possui, além dos dados do modelo estrutural, outros dados dos elementos necessários não só à análise e ao dimensionamento da estrutura, como ao seu detalhamento, tratamento de ligações, e ainda úteis para o orçamento e planejamento de produção da mesma.

O Modelador 3D foi implementado como um modelador unifilar lançado no espaço tridimensional, para estruturas compostas por barras, nós e superfícies planas horizontais representadas por classes criadas para ele. Nele os elementos da estrutura não possuíam seções transversais e as ligações resumiam-se ao encontro de barras em um ponto formando um nó da estrutura. O gerenciamento da estrutura era realizado por outras classes também criadas para esse aplicativo. Foi desenvolvido um sistema de numeração automática de nós, barras e faces com sistemas de controle de conectividade e de compatibilidade entre os elementos, além do lançamento de alguns tipos de cargas para os três elementos, permitindo-se o agrupamento das mesmas em casos de carregamentos distintos. Um estudo mais detalhado do Modelador 3D e das alterações realizadas sobre o mesmo será desenvolvido no capítulo 4.

O PREMOLD representa um grande avanço em relação ao Modelador 3D, proporcionando o lançamento da estrutura através de elementos sólidos com o tratamento geométrico automático de ligações, o correto posicionamento dos componentes da estrutura, o estudo preliminar de interferências entre elementos e uma visualização imediata da estrutura real a ser gerada. A partir do modelo sólido, o detalhamento automático da estrutura pode também ser facilmente obtido, agilizando a execução de desenhos de forma, de ferragens, etc..

## **1.2 A Indústria de Pré-Moldados no Panorama da Construção Civil**

A construção civil vem experimentando um constante aprimoramento de técnicas construtivas, avanços tecnológicos na produção de materiais e progressos consideráveis nas técnicas de análise e de dimensionamento de estruturas. Apesar disso, a construção civil apresenta, de uma maneira geral, problemas de baixa produtividade, grande desperdício de materiais, morosidade e baixo controle de qualidade.

Na área de estruturas, a melhoria desse quadro tem sido buscada através do uso de estruturas pré-fabricadas ou pré-moldadas. Segundo EL DEBS (2000), elementos pré-moldados são aqueles moldados ou produzidos fora do seu lugar de uso definitivo na

estrutura, sendo que a pré-fabricação é a pré-moldagem aplicada à produção em grande escala. Assim, o uso de elementos pré-fabricados é uma forma de se buscar a industrialização da construção e a pré-moldagem é uma forma de racionalizar a construção. As estruturas de concreto armado ou de concreto protendido que se adequam a esses preceitos são conhecidas como estruturas de concreto pré-moldado.

Segundo o mesmo autor, a pré-moldagem esteve sempre presente no desenvolvimento do concreto armado e do concreto protendido, tendo acompanhado o desenvolvimento desses até a Segunda Guerra Mundial. A partir de então esse tipo de estrutura surgiu como solução para os problemas de produção em grande escala, de escassez de mão-de-obra e para a diminuição de custos das estruturas para a construção civil.

EL DEBS (2000), cita que o uso do concreto pré-moldado está diretamente relacionado ao grau de desenvolvimento tecnológico e social do país. Esse desenvolvimento reflete-se na disponibilidade de ferramentas tecnológicas para o projeto e para a produção dessas estruturas, assim como na disponibilidade de mão-de-obra qualificada e num maior controle da qualidade dos produtos. O emprego de concreto pré-moldado no Brasil é ainda limitado devido a fatores macroeconômicos como o sistema tributário e a instabilidade econômica do país, além dos aspectos culturais. Entre os últimos pode-se citar: o conservadorismo e a falta de conhecimento das alternativas em concreto pré-moldado na construção civil, a escassez de oferta dos equipamentos necessários e a falta de dispositivos para as ligações entre os elementos pré-moldados e para o manuseio das peças. Um fator notável no mercado de pré-moldados é a existência de um ciclo vicioso onde o pouco uso do concreto pré-moldado não desperta o desenvolvimento de insumos tecnológicos, e a falta de insumos tecnológicos não estimula a produção de estruturas de concreto pré-moldado.

A falta de tais insumos é sentida não só em equipamentos para fabricação, transporte, manuseio e montagem de elementos, mas também na ausência de sistemas CAD (*Computer Aided Design*), CAE (*Computer Aided Engineering*) e CAM (*Computer Aided Manufacturing*) que permitam uma maior automação do seu processo produtivo, englobando modelamento, análise estrutural, dimensionamento, detalhamento e fabricação de elementos.

A automação dos processos de modelamento e detalhamento de estruturas com o uso de sistemas CAD é especialmente importante para a indústria de estruturas pré-moldadas ou pré-fabricadas, cujos projetos têm uma vocação natural para a automação, devido à padronização e modularidade dessas estruturas. Nesse sentido, o desenvolvimento de aplicativos compatíveis com os procedimentos de operação e com os modos de trabalho próprios dessa indústria fica mais fácil e intuitivo.

Até hoje, os processos de modelamento e de detalhamento para a fabricação de estruturas de concreto pré-moldado são feitos manualmente. Normalmente o computador é usado apenas como prancheta eletrônica, sem preocupação com a automação do processo. Como resultado tem-se um processo composto somente por procedimentos artesanais que constituem uma fonte de erros e conseqüentes prejuízos, provenientes da necessidade de correções e dos atrasos na fabricação e na montagem das peças.

Apesar de, no Brasil, o concreto armado ainda ser um material mais atrativo do ponto de vista financeiro, a indústria de estruturas de concreto pré-moldado tem encontrado grandes desafios na competição pelo mercado da construção civil. A indústria de estruturas de aço tem investido na melhoria do seu processo produtivo, com a automação dos processos de modelamento e de detalhamento, tornando-as assim competitiva, pela redução de prazos e de custos, melhoria da qualidade dos projetos e minimização de erros. A parceria Universidade-Empresa no desenvolvimento de aplicativos CAD/CAE/CAM para automação desses processos, na área de estruturas de aço, já mostrou resultados animadores em termos da melhoria de eficiência, melhoria de produtividade e melhoria de qualidade, entre outros fatores.

A importância e a potencialidade das estruturas de concreto pré-moldado nos processos de industrialização e racionalização da construção civil e a necessidade de sua adequação às demandas mercadológicas atuais são fatos inquestionáveis. Para responder às tendências atuais do mercado da construção civil e colocar a indústria de pré-moldados no contexto presente de modernização no mundo globalizado, é necessário investir na automação do seu processo produtivo.

### **1.3 A Computação Aplicada à Engenharia de Estruturas: Sistemas CAD, CAE e CAM ( A Automação do Processo Produtivo de Estruturas de Concreto Pré-Moldado)**

O processo produtivo de uma estrutura pré-moldada pode ser dividido nas seguintes etapas:

- Lançamento ou modelamento da estrutura;
- Análise estrutural;
- Dimensionamento dos elementos;
- Detalhamento;
- Fabricação dos elementos;
- Transporte e armazenamento das peças;
- Montagem da estrutura.

No projeto de estruturas de concreto pré-moldado o conhecimento de todas as etapas envolvidas na produção é fundamental para o cálculo estrutural e para uma boa concepção da estrutura através de sua divisão em elementos e definição de sua seção transversal básica. Na concepção dos sistemas estruturais pré-moldados os aspectos construtivos devem ser considerados juntamente com os aspectos estruturais. Os primeiros podem, até mesmo, ter preponderância sobre os últimos, devido às restrições impostas pelos equipamentos disponíveis ao transporte e à montagem das peças. Cuidados especiais devem ser dispensados aos detalhamentos e especificações. Alterações durante a montagem nos elementos já fabricados, para solucionar problemas não previstos pelo projeto, são incompatíveis com a pré-fabricação. Também a análise estrutural merece uma atenção especial devendo ser consideradas, além da situação final de cálculo, as situações transitórias de desmoldagem, transporte, armazenamento e montagem. A análise, o dimensionamento e o detalhamento das ligações entre os elementos pré-moldados constituem também aspectos fundamentais dessas estruturas. As ligações representam uma das principais diferenças entre as estruturas de concreto moldado "in loco" e as estruturas de concreto pré-moldado. (EL DEBS, 2000)

Segundo o mesmo autor, tantas particularidades e interdependências entre as fases de produção tornam o projeto de estruturas de concreto pré-moldado mais elaborado do que o projeto de estruturas correspondentes em concreto armado moldado “in loco”. Por isso o uso de estruturas de concreto pré-moldado exige padronização das construções, com o uso de soluções estruturais mais simples devido às limitações de ferramentas e aplicativos que agilizem os processos de modelamento, de análise, de dimensionamento e de detalhamento. Opções de elementos, de tipos de ligações e de soluções construtivas são ainda limitadas. Para se tornar competitiva no mercado atual, a indústria de pré-moldados precisa responder de forma mais personalizada às necessidades do cliente e às suas próprias necessidades. Isso implica em diversificação de soluções, obtida pela automação do processo produtivo com a integração entre projeto, fabricação e produto. Segundo SHAH e MÃNTILA (1995), a proposta dos chamados “produtos orientados”, resultaria em um decréscimo no tempo de confecção do produto, numa otimização do processo e num resultado eficiente, com uma maior exatidão e sensível economia.

Em projetos de estruturas, o desenvolvimento de sistemas CAD específicos para o modelamento e detalhamento é ainda restrito. Trabalhos mais elaborados têm sido realizados principalmente na área de estruturas de concreto moldado "in loco" ou para estruturas metálicas. Para aquelas estruturas são encontrados alguns aplicativos específicos como CYPECAD e TQS, e Metálicas 3D. Para estruturas metálicas encontra-se o TecnoMETAL, TecnoSTEEL e XSTEEL, ROBOT, além de outros. Para estruturas de concreto pré-moldado existem poucos aplicativos nacionais de automação desenvolvidos, alguns ainda apenas agilizam o desenho da estrutura a partir de uma biblioteca básica como, por exemplo, o PremoCAD desenvolvido no DEES (Departamento de Engenharia de Estruturas) da UFMG, pelo CADTEC (Centro Apoio, Desenvolvimento Tecnológico e Ensino da Computação Gráfica) e em parceria com uma empresa de concreto pré-moldado de Minas Gerais, ou o PreMOLDAR, um aplicativo desenvolvido pela MULTIPLUS que também executa o lançamento e algum dimensionamento de sistemas estruturais em concreto pré-moldado. O PremoCAD realiza apenas o desenho bidimensional de seções transversais, vistas superior e lateral de elementos isolados de sua biblioteca, não gerando um modelo completo tridimensional com tratamento de interferências ou ligações.

Em Minas Gerais existem duas grandes empresas de concreto pré-moldado de reconhecimento e atuação nacional. Nessas empresas o projeto estrutural ainda é desenvolvido em uma plataforma gráfica genérica. Utiliza-se o AutoCAD como simples prancheta eletrônica, onde o projeto é concebido pelo desenho de modelos tridimensionais com a geração de maquetes eletrônicas. A restrita utilização dos aplicativos existentes pode ser entendida pela generalidade dos aplicativos comerciais disponíveis. A generalização do aplicativo, com a finalidade de atender a todos os usuários, acaba por não responder de forma adequada às necessidades específicas de cada empresa, nem aos modos e procedimentos de trabalho próprios de cada uma. Por outro lado, aplicativos, que disponibilizam recursos apenas para o desenho da estrutura, não oferecem tantas vantagens para a necessária automação do processo.

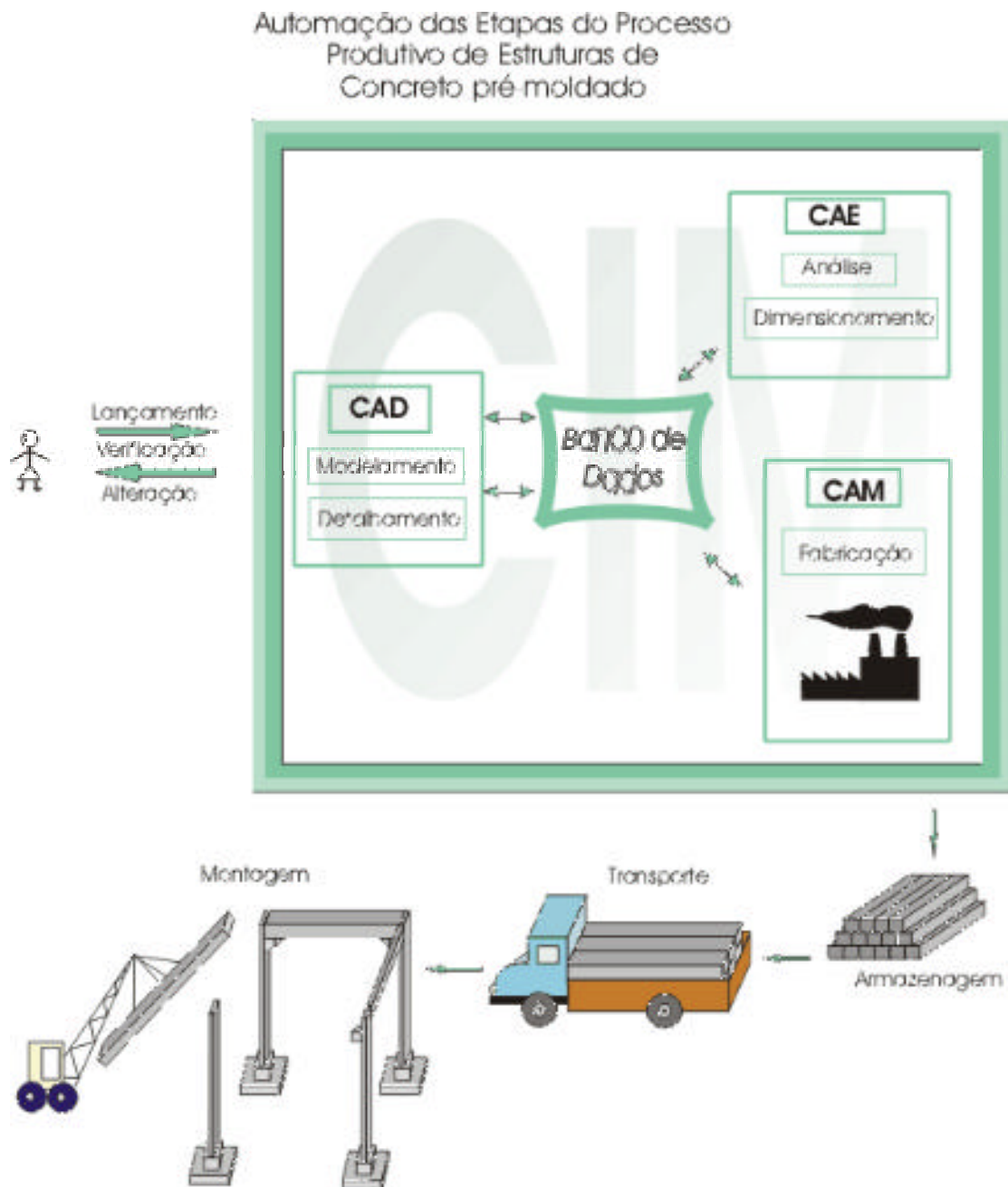
O desenvolvimento de sistemas CAE para a etapa de análise de estruturas genéricas encontra-se mais evoluído e já existem no mercado programas consagrados, baseados no Método de Elementos Finitos como o ANSYS, o SAP2000, o ROBOT e o ABAQUS. Encontra-se também, no mercado, aplicativos para o dimensionamento de estruturas de concreto armado moldado "in loco", como CYPECAD e TQS, para o modelamento e detalhamento da estrutura. Em contrapartida, aplicativos comerciais, específicos para o dimensionamento de estruturas pré-moldadas, ainda são pouco divulgados e pouco conhecidos.

Os sistemas CAM são responsáveis pela automação da produção durante o processo de fabricação. Para estruturas de concreto pré-moldado a automação da produção pode ser implementada no corte, na dobra e na soldagem de ferragem através de máquinas digitais, no uso de formas deslizantes e ainda na produção e usinagem do concreto sob um controle rigoroso de qualidade.

Apesar da existência de alguns sistemas genéricos CAD/CAE/CAM, o processo de produção de estruturas de concreto pré-moldado precisa de uma integração, ainda não disponível, entre esses sistemas. A integração dos sistemas pode ser obtida pelo desenvolvimento de pré e pós-processadores capazes de interagir com programas de análise e de dimensionamento existentes, com a finalidade de automatizar o processo. O pré-processador, além de automatizar o modelamento da estrutura, executará a integração

de seus dados com a fase de análise e de dimensionamento. O pós-processador, cuja principal função é a automação do processo de detalhamento dos elementos estruturais para a fabricação, deve ter, ainda, recursos para estabelecer interfaces de comunicação entre o detalhamento e as máquinas digitais de corte e dobra dentro do processo de manufatura assistida por computador (CAM). O intercâmbio de informações entre os vários sistemas envolvidos no processo deve ser consistente, completo e correto para garantir a qualidade do produto, a agilidade do processo de produção e a eficiência das ações a serem aplicadas. O desenvolvimento de sistemas CAD que possibilitem, através do modelamento da estrutura, o preenchimento automático de um banco de dados com as informações necessárias às diversas etapas do processo, garante um alto grau de confiabilidade ao sistema global. Interfaces compatíveis com sistemas CAE e CAM permitem o gerenciamento dos dados e dos resultados obtidos no processo, sem perda de informações nas interações entre as diversas etapas do mesmo.

Este trabalho foi desenvolvido buscando-se o preenchimento dessas lacunas, dando respostas à necessidade de um modelador que considere as especificidades dos projetos de estruturas de concreto pré-moldado, não apenas agilizando desenhos, mas possuindo um banco de dados capaz de dar suporte às outras etapas do processo de produção dessas estruturas. O PREMOLD foi desenvolvido para ser a base para os processadores gráficos tridimensionais responsáveis pela integração das diversas etapas desse processo. Ele é responsável pelo preenchimento automático de um banco de dados, armazenando as informações necessárias às etapas seguintes. A proposta inicial não é gerar um aplicativo auto-suficiente para o modelamento, análise, dimensionamento e detalhamento dessas estruturas, mas sim lançar as bases para um sistema capaz de interagir com sistemas CAE e CAM já desenvolvidos promovendo a integração e automação de todo o processo (FIG. 1.1). Assim, o objetivo do presente trabalho é o desenvolvimento de um aplicativo para a automação dos processos de modelamento e de detalhamento da estrutura com o auxílio da computação gráfica, da Programação Orientada a Objetos (POO) e do uso da tecnologia CAD.



**FIGURA 1.1** - Automação do processo produtivo de estruturas em concreto pré-moldado

## **1.4 Etapas do Processo de Produção com o Uso das Tecnologias CAD/CAE**

O uso das tecnologias CAD/CAE no processo de produção pode ser dividido nas seguintes etapas básicas:

- Concepção do projeto: modelagem inicial do objeto;
- Dimensionamento do objeto pelo sistema CAE;
- Detalhamento do projeto refinado;
- Informações complementares;
- Interface com os sistemas CAM.

Na etapa de concepção do projeto, o objeto real é modelado. Suas características principais são traduzidas em atributos e funções que são organizadas e armazenadas em um banco de dados. Cabe aqui fazer a distinção entre modelamento geométrico e maquete eletrônica. O modelamento geométrico, discutido mais detalhadamente no item 1.5, é um sistema de modelagem utilizado pelos sistemas CAD, onde o modelo gerado é capaz de produzir não somente a visualização geométrica do objeto, mas também é responsável pela organização e armazenagem dos dados representativos do objeto para futuros tratamentos. Por outro lado, a maquete eletrônica é um sistema de modelagem capaz de representar visualmente um objeto, sem armazenar ou tratar dados não geométricos do mesmo. É um conjunto de linhas e símbolos simplesmente organizados em uma plataforma gráfica, segundo as técnicas de representação utilizadas.

Bibliotecas gráficas, ferramentas de edição gráfica e comandos específicos gerados por rotinas de automação da modelagem de objetos são ferramentas utilizadas pelo sistema CAD.

Nessa etapa pode-se obter uma visualização do objeto modelado, facilitando a verificação de consistência entre o modelo virtual e o objeto real a ser gerado.

No dimensionamento do objeto pelo sistema CAE, o banco de dados, organizado e preenchido durante a modelagem do objeto, é transmitido a esse sistema onde os dados são

tratados de acordo com o tipo de análise a ser realizada. São gerados novos dados resultantes dessa análise. Esses novos dados são repassados para o sistema CAD que promoverá as alterações, ou refinamentos necessários no modelo inicial para que o mesmo esteja compatível com o dimensionamento realizado.

Na etapa de detalhamento, estando projeto refinado pronto e adequado ao dimensionamento realizado, podem ser gerados projetos específicos para a fabricação do objeto. O tipo de detalhamento dependerá do tipo de objeto tratado. Para estruturas, de um modo geral, gera-se o detalhamento das peças com informações precisas sobre dimensões, tipos de ligações, tipos de materiais a serem utilizados, etc. Os dados referentes ao detalhamento também são organizados e armazenados em um banco de dados que pode ser utilizado para a obtenção de informações detalhadas sobre o objeto e ainda ser transmitido aos sistemas CAM.

Com base no banco de dados gerado, podem ser obtidos diversos documentos complementares com informações detalhadas sobre o objeto modelado. Essas informações, organizadas em listas de material, relatórios qualitativos e quantitativos ou memórias de cálculo são úteis em outras etapas da produção como, por exemplo, no orçamento e no planejamento da obra.

Na etapa de interface com os sistemas CAM, os dados gerados pelo modelador são transcritos para esses sistemas, sendo utilizados na programação dos equipamentos responsáveis pela fabricação do objeto; máquinas de corte, dobra, montagem, soldagem, etc..

## **1.5 Modelamento Geométrico X Modelamento Estrutural**

Modelo é uma abstração de um objeto real, de um processo do mundo real ou o projeto de um produto a ser construído (MAGALHÃES, 1998). É um objeto construído artificialmente com a finalidade de tornar mais fácil a observação de outro objeto (MÄNTYLÄ, 1988) Nesse modelo as características ou dados essenciais do objeto real,

capazes de representá-lo e de distingui-lo do seu entorno, são separadas do todo (SHAH e MANTILA, 1995), armazenadas num banco de dados, passando a constituir o objeto virtual.

Um modelo geométrico busca definir a forma e outras características geométricas de um objeto. A geometria é fundamental para a reprodução do objeto dentro de uma plataforma gráfica e para a fabricação do produto final. A modelagem geométrica ficou conhecida como a coleção de métodos usados para definir a forma e outras características de um objeto, a partir do início dos anos 70 (OLIVEIRA, 1991). Ela estuda a representação computacional da geometria de um objeto.

HÜNTER (1998) apresenta uma classificação simplificada do modelamento geométrico baseada nos ramos básicos que caracterizam o tipo de modelagem e seus resultados:

- *Modelagem gráfica*: representa os objetos através de arestas e vértices, sem fazer referência a faces ou superfícies; são os modelos fio-de-aramé ou *wireframe*;
- *Modelagem de superfície*: armazena e trata informações detalhadas sobre superfícies planas ou curvas completas, mas não reconhece sólidos;
- *Modelagem de sólidos*: descreve a geometria tridimensional de um objeto reconhecido como um sólido.

Segundo MAGALHÃES (1998), para representar computacionalmente objetos sólidos deve-se basear sua representação na geometria do objeto e no uso de esquemas de representação. Esses esquemas são estruturas de dados especiais que permitem codificar o conjunto infinito de pontos definindo o objeto em uma porção finita da memória do computador. Esses esquemas de representações podem ser divididos em três grandes grupos:

- *Modelos de Decomposição*: Esses modelos representam os sólidos através de uma lista de componentes básicos, combinados entre si através da colagem, sendo os modelos mais importantes criados por esse esquema: a Decomposição Celular, a Enumeração da Ocupação Espacial e a representação *Octree*.

- Modelos Construtivos: Atualmente é o grupo mais utilizado para aplicações em CAD. Esses modelos partem de um conjunto de pontos denominados primitivas para modelar outros conjuntos através de combinações dessas primitivas. Além das operações de colagem incluem operações booleanas (união, interseção e diferença) e as transformações geométricas (rotação, translação e escalamento). A definição dessas primitivas é definida basicamente pelo esquema CSG (*Constructive Solid Geometry*).
- Modelos de Fronteira: Esses modelos representam um conjunto de pontos em termos de seus limites espaciais, geralmente uma superfície fechada com alguma convenção para indicar em que lado da superfície o sólido está. Os sólidos são representados por uma coleção de faces limitadas por arestas, que sua vez são limitadas por vértices.

Os principais esquemas de representação de representação de sólidos são:

- Fio-de-aramé ou Wireframe: representa os objetos através de arestas, consistindo inteiramente de pontos, linhas e curvas. Não possui informação geométrica sobre faces ou sólidos;
- Decomposição Celular: São usadas células primitivas parametrizadas e combinadas pela colagem para decompor o sólido. Esse modelo é usualmente utilizados como representação básica para métodos dos elementos finitos. É o mais preciso dos modelos de decomposição.
- Enumeração da Ocupação Espacial: O sólido é visto como um conjunto de cubos de tamanho fixo, localizados em uma malha espacial fixa. Cada célula é especificada como ocupada por um sólido ou não.
- *Octrees*: Realiza a divisão recursiva de uma região cúbica em oito octantes parcialmente preenchidos até obter octantes homogêneos (totalmente vazios ou cheios).
- Instanciamento de Primitivas: Tendo como base uma família de formatos sólidos cujos membros variam em relação a alguns parâmetros, são especificados os valores para a geração de um componente específico da família.

- Varredura (*Sweep*): Representação baseada na noção de mover uma região (gerador) por um caminho (diretor), utilizando duas formas básicas: a rotacional ou a translacional, que podem ser combinadas entre si e com escalamento.
- Geometria Sólida Construtiva: Sólidos descritos em termos de primitivas, combinada por operadores booleanos regularizados e movimentos rígidos.
- Representação de Fronteira (B-rep): Os sólidos são descritos em termos de suas superfícies limitantes. A topologia é definida pelo conjunto de faces, arestas e vértices que compõem o sólido; a geometria é definida pelas coordenadas dos vértices e pelas equações das curvas e superfícies.

Os modelos sólidos permitem a caracterização completa de um objeto, podendo gerar, quando necessário, os modelos de superfície e de fio-de-aramé como subprodutos. A inserção de texturas e de luz aos modelos sólidos possibilita a representação mais realista do objeto.

Modelamento estrutural é a representação de uma estrutura através de um modelo. Esse modelo deve ser capaz de representar sua geometria e suas condições de contorno, simulando o seu comportamento físico e respondendo de forma adequada às solicitações impostas, com indicação de deformações, deslocamentos e reações. O modelamento estrutural deve representar, de forma simples e organizada, diversos tipos de estruturas, armazenando os dados necessários à sua análise e ao seu dimensionamento (MALARD, 1998).

No modelo estrutural gerado sem a utilização da tecnologia CAD automatizada, as características geométricas da estrutura e outros dados essenciais à análise são transformados em dados numéricos e a alimentação do sistema de modelagem geralmente é feita pelo preenchimento de um banco de dados diretamente pelo usuário. Com o desenvolvimento da computação gráfica alguns programas de análise incorporaram recursos dessa tecnologia, gerando pré-processadores para a concepção e definição da geometria do objeto estudado. Desta maneira, a entrada de dados via processador gráfico possibilita a visualização do modelo. Um maior controle sobre a inserção de dados pelo usuário é possibilitado pela introdução de restrições ou de avaliações de consistência, o que reduz a probabilidade de erros durante o preenchimento do banco de dados. No entanto,

por serem aplicativos genéricos, aplicáveis a qualquer tipo de objetos, a entrada de dados dos sistemas de análise pode ser muito complexa. O desenvolvimento de um pré-processador específico que possibilite um tratamento diferenciado das estruturas de concreto pré-moldado é altamente recomendável, devido às diversas particularidades das mesmas e, ainda, devido às interdependências entre as fases de produção, que devem ser consideradas já na concepção da estrutura.

A maioria das técnicas numéricas para análise estrutural utiliza o Método de Elementos Finitos, que discretiza a estrutura em elementos finitos uni, bi ou tridimensionais. Na análise estrutural de edificações, o modelo mais usado é geralmente unifilar, composto basicamente por barras e nós, os quais, em conjunto, podem definir limites de superfícies. As barras representam os elementos lineares, geralmente vigas e pilares. Os nós representam as ligações entre elementos lineares, e as superfícies representam elementos planos como lajes, vigas parede, painéis verticais, etc.. Os elementos podem receber carregamentos e nos nós, além dos carregamentos são ainda inseridas restrições ou liberações ao deslocamento ou à rotação.

Assim, para que os processos de modelamento e de análise sejam interligados automaticamente é necessário que o modelador para estruturas prediais seja capaz de representar o modelo da análise estrutural, sendo ainda compatível com o modelamento utilizado pelo método numérico aplicado na etapa de análise.

## **1.6 Etapas de Desenvolvimento do Projeto de Pesquisa**

Para a realização deste projeto de pesquisa foi necessário um estudo maior sobre os fundamentos e pré-requisitos de projeto e execução de estruturas de concreto pré-moldado. Além desse conhecimento sobre o concreto pré-moldado, foi necessário passar por um programa de capacitação e de treinamento para adquirir a qualificação necessária ao desenvolvimento do aplicativo. O aprendizado sobre o uso de ferramentas como o AutoCAD 2002, a linguagem C++, o ObjectARX 2002, Visual C++ 6.0, entre outros, e os conhecimentos fundamentais para o desenvolvimento do aplicativo foram adquiridos nesse programa.

Após a etapa inicial de qualificação, identificou-se a necessidade de estudar e de implementar alterações na estrutura do código computacional do Modelador 3D para reaproveitamento das classes comuns. As alterações e atualizações do código para reaproveitamento foram devidas às mudanças da versão 2.0 da biblioteca ObjectARX do AutoCAD 14, utilizado no desenvolvimento do Modelador 3D, para a versão utilizada pelo AutoCAD 2002, utilizado no desenvolvimento do PREMOLD. Nova estrutura de classes também foi concebida para um melhor aproveitamento dos paradigmas da Programação Orientada a Objetos e do conceito de derivação de classes. Nessa reestruturação utilizou-se os conceitos de Herança, das estruturas Generalização/Especialização e Todo-Parte, descritas no item 3.2.

## **1.7 Conteúdo por Capítulos**

No capítulo 1 foram apresentados os objetivos desse trabalho de pesquisa, o panorama atual das estruturas de concreto pré-moldado, a possibilidade de utilização de técnicas CAD, CAE e CAM no processo produtivo dessas estruturas além de uma abordagem genérica do que seja modelamento geométrico e modelamento estrutural.

No capítulo 2 será apresentado um estudo realizado sobre os fundamentos para projetos de estruturas de concreto pré-moldado que nortearam o desenvolvimento do PREMOLD.

No capítulo 3 são apresentadas as ferramentas utilizadas no desenvolvimento deste trabalho com uma breve explicação sobre essas.

O capítulo 4 apresenta a estrutura original do Modelador 3D e as alterações aplicadas sobre ele através deste trabalho de pesquisa.

O capítulo 5 retrata o desenvolvimento conceitual e computacional do PREMOLD com geração de classes e de comandos para o modelamento da estrutura de concreto pré-moldado.

A conclusão, com apresentação das contribuições geradas por este trabalho, limitações do aplicativo e propostas para futuros trabalhos, é apresentada no capítulo 6.

# 2

## FUNDAMENTOS PARA PROJETOS DE ESTRUTURAS DE CONCRETO PRÉ-MOLDADO

Neste capítulo é apresentada uma breve revisão do estudo realizado sobre estruturas de concreto pré-moldado. Essa pesquisa foi feita com base nos fundamentos apresentados no livro "CONCRETO PRÉ-MOLDADO: Fundamentos e Aplicações" de Mounir Khalil El Debs (EL DEBS, 2000), professor do Departamento de Engenharia de Estrutura da Escola de Engenharia de São Carlos, USP, e na NBR 9062/85, " Projeto e Execução de Estruturas de Concreto Pré-moldado" (ABNT, 1985). As figuras aqui inseridas foram retiradas do citado livro, tendo sido gentilmente cedidas pelo professor El Debs.

Durante esse período de pesquisa a NBR 9062/85 foi reformulada. A nova norma foi lançada em dezembro de 2001, quando os trabalhos de pesquisa e de desenvolvimento do aplicativo já estavam praticamente concluídos. Devido ao prazo disponível para a

conclusão dos trabalhos e a não publicação da norma em tempo hábil para sua adoção, a nova norma não foi considerada neste trabalho.

Foi realizada uma revisão da norma NBR 6118/78 (ABNT, 1978) e de seu Projeto de Revisão (ABNT, 2000) anteriormente ao estudo relatado neste capítulo com a finalidade de levantar dados necessários a um possível dimensionamento de alguns elementos tratados pelo aplicativo. Porém, no decorrer dos trabalhos, optou-se por não abordar, nesta etapa de desenvolvimento do aplicativo, o dimensionamento das armaduras e dos componentes das estruturas, devendo este assunto ser abordado em trabalho futuros.

O tipo de sistema estrutural modelado nesta primeira etapa de desenvolvimento do PREMOLD, os tipos de componentes utilizados, os tipos de seções transversais tratados e o tratamento dados às ligações foram desenvolvidos com base nos estudos aqui apresentados.

A concepção das classes do PREMOLD e a concepção de sua estrutura hierárquica foram influenciadas por este estudo, assim como a reestruturação das classes do Modelador 3D. Passa-se então a apresentação daqueles dados que nortearam o desenvolvimento desta etapa de trabalho e aqueles que deverão ser abordados em etapas futuras como: outros sistemas estruturais e outros tipos de ligações. Este estudo vem justificar e fundamentar diversas decisões tomadas no lançamento das bases do PREMOLD.

Nessa primeira etapa de desenvolvimento do PREMOLD, não serão tratados os carregamentos ou solicitações aplicadas sobre os elementos da estrutura. Assim, apenas as prescrições construtivas estritamente geométricas, que não dependam dessas solicitações para o dimensionamento de elementos e de ligações, serão adotadas no PREMOLD, sendo aqui relacionadas quando for o caso.

## 2.1 O Concreto Pré-Moldado

Segundo a NBR 9062/85 (ABNT,1985), "os elementos produzidos em usina ou instalações analogamente adequadas aos recursos para produção e que disponham de pessoal, organização de laboratórios e demais instalações permanentes para o controle de qualidade, devidamente inspecionado pela fiscalização do proprietário, recebem a classificação de pré-fabricados." A mesma norma define que os elementos produzidos em condições menos rigorosas de controle de qualidade são classificados como pré-moldados.

Como foi mencionado no item 1.2, EL DEBS (2000) considera elementos pré-moldados aqueles que são moldados ou produzidos fora do seu lugar de uso definitivo na estrutura, sendo a pré-fabricação a pré-moldagem aplicada à produção em grande escala.

Entende-se, então, o elemento pré-moldado como aquele produzido fora de seu local definitivo, segundo um controle de qualidade, que envolve desde o projeto à montagem da estrutura.

O concreto pré-moldado tem um amplo campo de aplicação, desde edificações, construção pesada, infra-estrutura urbana, mobiliário urbano, etc. Nas edificações o concreto pré-moldado pode ser aplicado tanto como elemento estrutural quanto como vedação e fechamento.

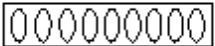

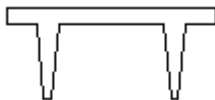

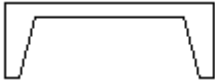
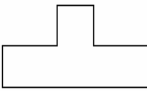


O elemento de concreto pré-moldado pode ser classificado, quanto a sua concepção em:

- Pré-moldado de fábrica ou de canteiro: com relação à produção dos elementos em instalações permanentes fora do local da obra ou em instalações temporárias no canteiro de obras, respectivamente;
- Pré-moldado de seção parcial ou de seção completa: com relação à incorporação ou não de material moldado "in loco" para aumentar sua seção resistente, respectivamente;
- Pré-moldado pesado ou leve: com relação à ordem de grandeza de seu peso, sendo considerado pesado aquele elemento que exige equipamentos especiais para seu transporte e sua montagem;

- Pré-moldado arquitetônico ou normal: com relação a exploração ou não de sua aparência na estética final da edificação, respectivamente.

Os elementos pré-moldados mais comuns em edificações prediais são: pilares, vigas, lajes e paredes. Esses elementos podem apresentar uma grande variedade de formas para suas seções transversais, porém, algumas formas são mais utilizadas do que outras. As formas mais utilizadas para os elementos principais de uma estrutura pré-moldada, segundo EL DEBS (2000), são aquelas mostradas na TAB. 2.1.

**TABELA 2.1** - Elementos pré-moldados de uso mais comum [EL DEBS, 2000]

Lajes e paredes		Vigas e pilares	
	Painel alveolar		Seção retangular
	Painel TT ou $\pi$		Seção I
	Painel U		Seção T invertido
	Painel maciço		Seção quadrada vazada

O uso da pré-moldagem em uma edificação apresenta vantagens e desvantagens que devem ser avaliadas na escolha do tipo de estrutura a ser utilizada em determinado

empreendimento. Como vantagens da pré-moldagem EL DEBS (2000) destaca a possibilidade de desmonte da edificação e a facilidade de produção dos elementos com redução ou eliminação do cimbramento. Quando os elementos são produzidos em escala o autor cita, ainda, a:

- Grande reutilização de formas;
- Possibilidade de uso de protensão com armadura pré-tracionada;
- Produção de seções com maior aproveitamento dos materiais;
- Maior produtividade da mão-de-obra;
- Maior controle de qualidade;

Como desvantagens da pré-moldagem o autor lista:

- Possíveis dificuldades para a colocação dos elementos nos locais definitivos de utilização relacionados a limitações quanto ao transporte e à montagem dos elementos;
- Ligações entre elementos caras e de difícil execução quando é necessário gerar uma estrutura que se comporte de forma monolítica com redistribuição de momentos fletores.

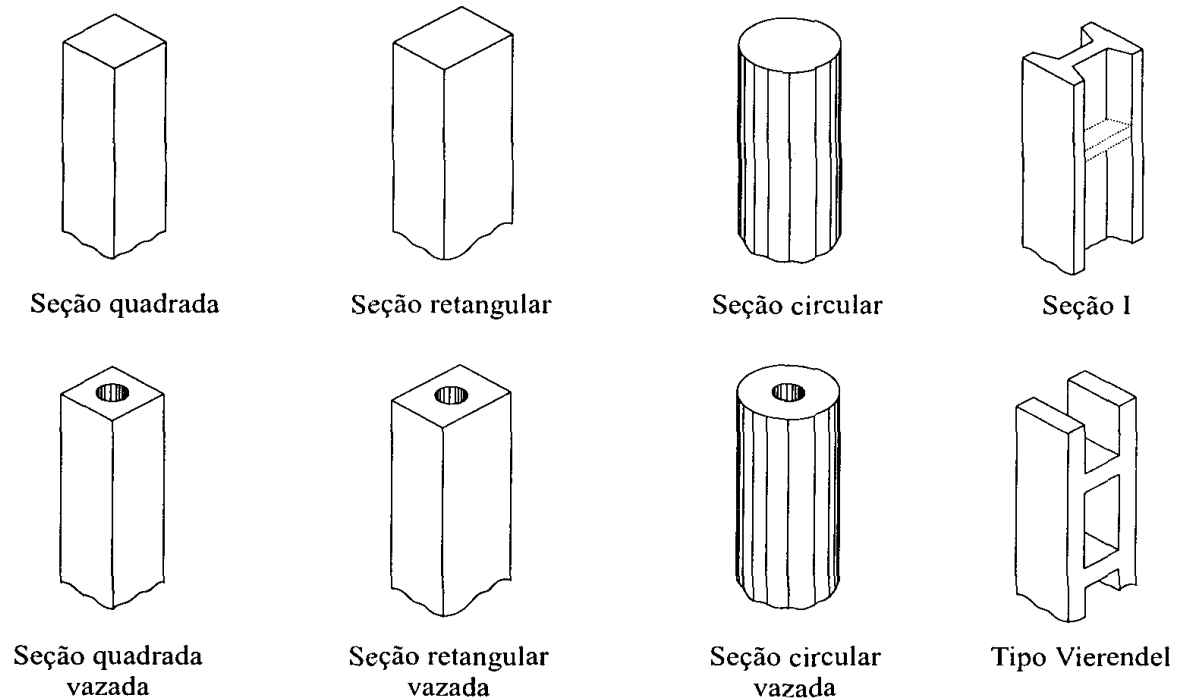
## **2.2 Os Componentes de Sistemas Estruturais de Edificações em Concreto Pré-Moldado**

### **2.2.1 Componentes dos Sistemas Reticulados**

Os componentes básicos nas edificações com estruturas segundo o sistema reticulado são os pilares e as vigas.

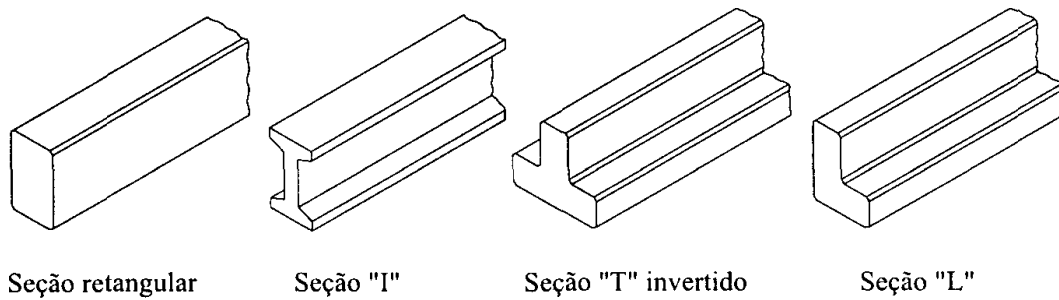
As seções transversais mais empregadas para pilares são as seções retangulares ou quadradas, vazadas ou não. Para galpões as seções "I" ou do tipo Vierendel são também bastante utilizadas (FIG. 2.1). Para edificações, os pilares podem vencer mais de um andar. Caso seja possível, é interessante que o pilar cubra toda a altura da edificação reduzindo, assim, ligações intermediárias. Os pilares podem alcançar um comprimento da ordem de 30 metros, apresentando, geralmente, seções transversais constantes. São usualmente de

concreto armado, podendo ser de concreto protendido, caso estejam sujeitos a elevados momentos fletores. Os pilares podem possuir no seu interior um duto para funcionar como um condutor de águas pluviais. Nesse caso a redução da área de concreto deve ser considerada no dimensionamento do pilar e a espessura mínima para a parede do pilar resultante deve ser de 10cm. (ABNT, 1985).

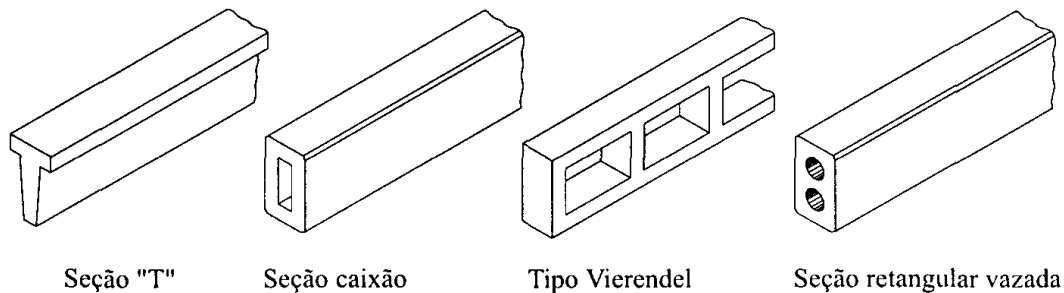


**FIGURA 2.1** - Seções transversais utilizadas nos pilares [EL DEBS, 2000]

Para as vigas as seções transversais mais empregadas são: seção retangular, seção "I", "T" invertido e seção "L" (FIG. 2.2). Podem ser utilizadas, ainda, as seções: "T", seção caixão, tipo Vierendel, retangular vazada, etc (FIG. 2.3). Geralmente vigas retangulares podem vencer vão de até 15 metros e vigas "I" vencem vão de até 35 metros. O uso de concreto protendido é indicado para vigas, exceto no caso de pequenos vãos.



**FIGURA 2.2** - Seções transversais mais utilizadas nas vigas [EL DEBS, 2000]



**FIGURA 2.3** - Outras formas de seções transversais utilizadas nas vigas (EL DEBS, 2000)

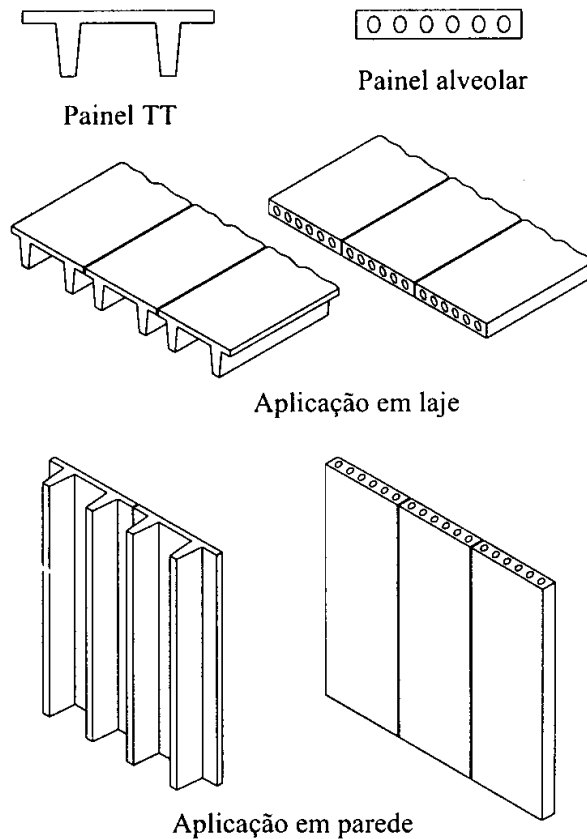
As vigas podem ter altura constante ou variável conforme o uso. Para pisos de edifício as vigas geralmente têm seção constante, porém, em coberturas as vigas podem apresentar variação de seção ao longo do comprimento possibilitando o caimento mínimo para os elementos de cobertura apoiados sobre elas.

### 2.2.2 Componentes do Sistema de Pavimentos

Os componentes dos sistemas de pavimentos são basicamente lajes e vigas.

Os tipos mais comuns de seções transversais para lajes são as seções duplo "T", ou  $\pi$ , e a seção alveolar (FIG. 2.4). Essas duas seções apresentam-se em forma de painéis. Pode-se, ainda, listar as nervuras pré-moldadas ou vigotas pré-moldadas utilizadas na confecção de

lajes pré-moldadas ou pré-fabricadas e os elementos de pré-laje que são painéis pré-moldados completados com concreto “in loco”.

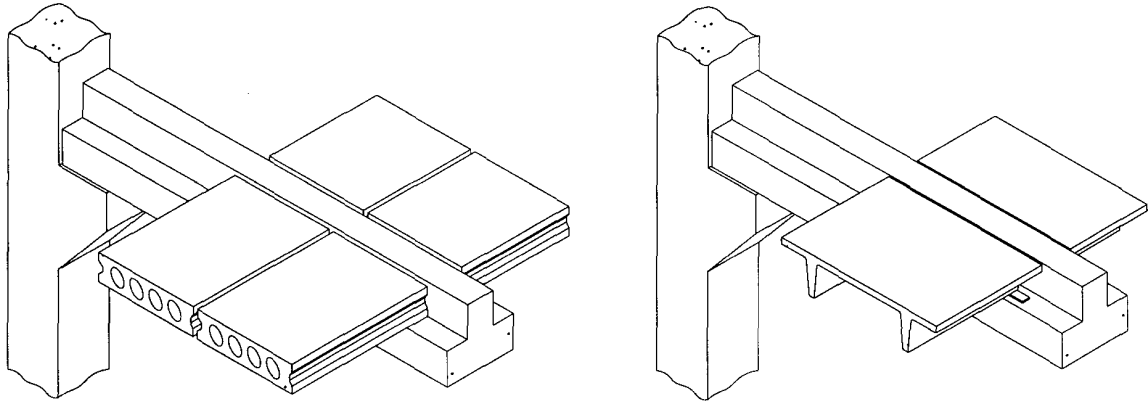


**FIGURA 2.4** - Seções mais comuns para painéis de laje e de parede (EL DEBS, 2000)

Outras seções também utilizadas são: seção "T", múltiplos de "T", seção "U", seção duplo "T" invertido ou painéis nervurados.

As seções duplo "T" e as seções alveolares podem receber ou não cobertura de concreto moldado "in loco" (FIG. 2.5). Podem ser produzidas em pistas de rolamento com o uso de concreto protendido, podendo vencer grandes vãos. Os painéis alveolares apresentam orifícios longitudinais com seção circular, pseudo-elíptica, oval ou ainda retangular. Para os painéis duplo "T" a relação vão x altura é da ordem de 30, sendo vencidos vãos de 5 a

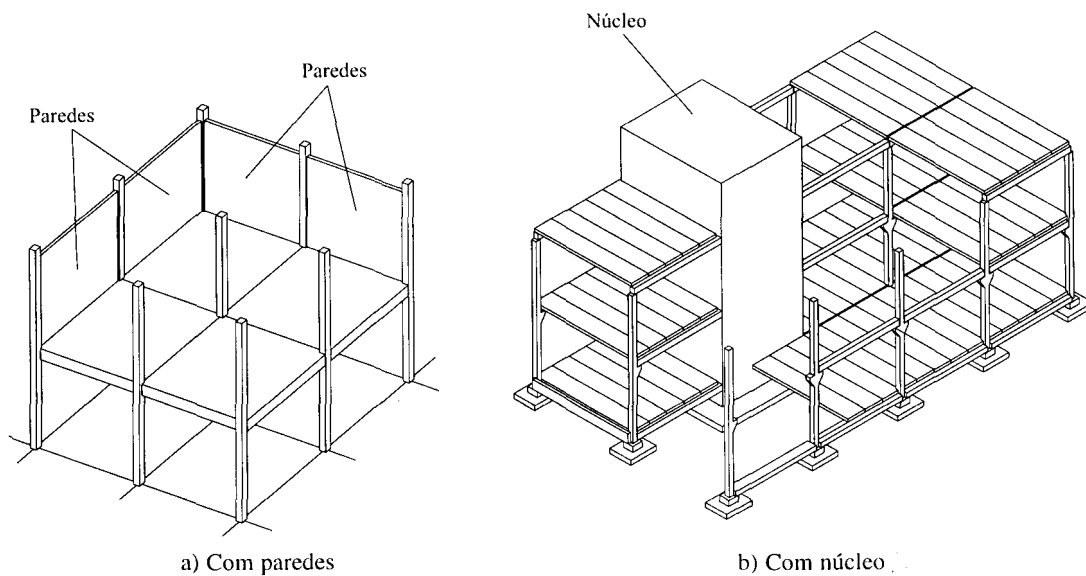
30 metros. Para os painéis alveolares essa relação é da ordem de 50 e os vãos variam de 5 a 15 metros.



**FIGURA 2.5** - Sistema de pavimentos com viga e painéis alveolares ou duplo T (TT) (EL DEBS, 2000)

### 2.2.3 Componentes de Sistemas de Paredes

Segundo EL DEBS (2000), os elementos do sistema de paredes podem fazer parte dos sistemas estruturais de parede portante, dos sistemas de contraventamento com núcleos de paredes ou como elementos de fechamento. (FIG. 2.6)



**FIGURA 2.6** - Sistemas de contraventamento com paredes e com núcleo (EL DEBS, 2000)

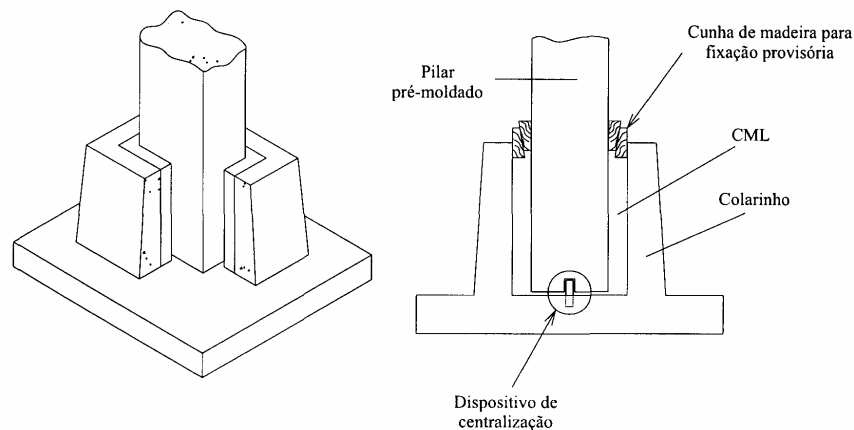
As seções transversais mais utilizadas são: seções maciças (em concreto simples armado ou protendido), seções vazadas, nervuradas ou sanduíche. Esse último tipo de seção é composto por duas camadas de concreto separadas por algum tipo de material de enchimento. É possível, também, o uso de concreto arquitetônico em painéis de fachada.

Os painéis podem ser usados na posição vertical ou horizontal. Eles transmitem seu peso próprio e a ação do vento para a estrutura principal. A escolha das posições das ligações e dos movimentos, que serão permitidos aos painéis, pode proporcionar um enrijecimento da estrutura principal, resultando em economia para essa.

#### 2.2.4 Outros Componentes

Os componentes de cobertura podem ser formados por elementos que cobrem os vãos principais da estrutura como telhas de concreto pré-moldado ou pelo vigamento secundário, quando são usadas terças de concreto para suportar telhas de pequenas dimensões. Entre os componentes de coberturas estão também as vigas calhas, que coletam as águas pluviais dos elementos da cobertura e vigas de altura variável.

Para as fundações podem ser executados, em concreto pré-moldado, blocos de fundação e vigas baldrames. Os blocos de fundação podem ter cálice completo com colarinho e sapata (FIG. 2.7) ou somente colarinho.



**FIGURA 2.7** - Bloco de fundação com colarinho e sapata (EL DEBS, 2000)

Além dos componentes de cobertura e de fundação, pode ser executado em concreto pré-moldado: escadas, sacadas, brise-soleil e outros elementos de acabamento.

## **2.3 Sistemas Estruturais para Edifícios de um Pavimento**

Esses sistemas estruturais apresentam vãos relativamente grandes para uso como galpões na indústria, comércio, como depósitos, oficinas, etc..

Para essas edificações podem ser utilizadas soluções mistas com uso de pilares em concreto pré-moldado e cobertura em estrutura metálica ou de madeira.

Esses sistemas podem ser divididos em:

- Sistema estrutural reticulado:
  - Com elemento de eixo reto;
  - Com elemento composto por trecho de eixo reto ou curvo;
- Sistema estrutural de parede portante.

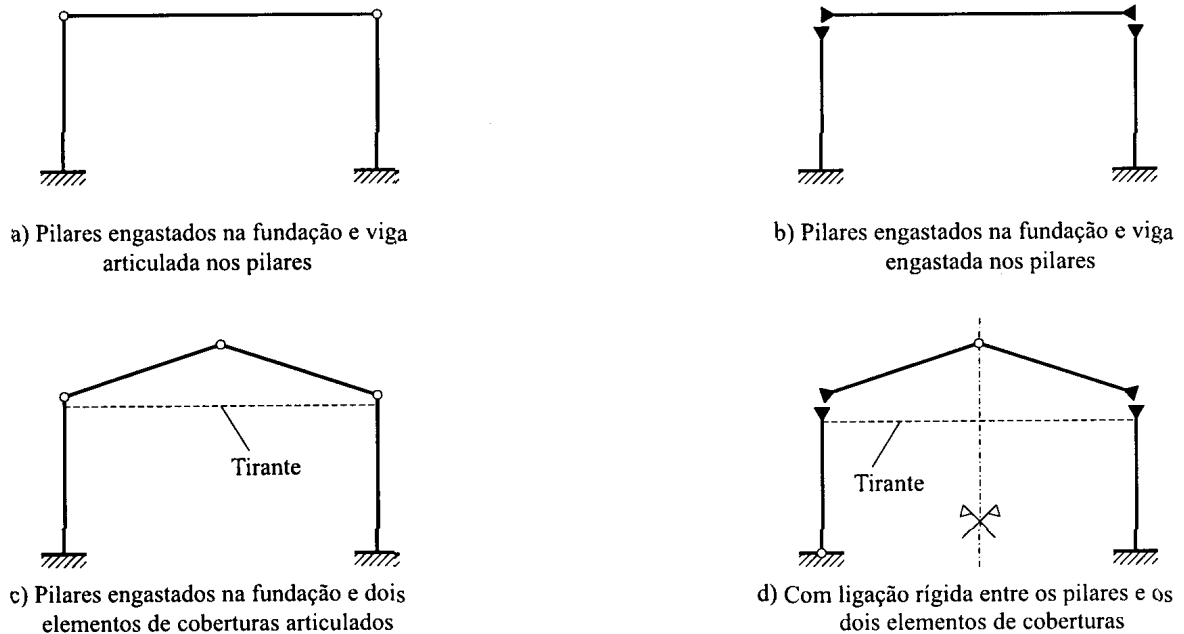
### **2.3.1 Sistema Estrutural Reticulado com Elemento de Eixo Reto**

Segundo EL DEBS (2000), esse sistema estrutural apresenta todas as facilidades de execução de estruturas de concreto pré-moldado, podendo, ainda, receber protensão com aderência inicial. Assim como um sistema com o uso de ligações articuladas, ou semi-articuladas, são pouco favoráveis à distribuição de esforços solicitantes.

Para esse sistema podem ser listadas as seguintes formas básicas:

- Pilares engastados na fundação com viga articulada nos pilares (FIG. 2.8a): essa forma apresenta facilidade de montagem e de execução das ligações;
- Pilares engastados na fundação com viga engastada nos pilares (FIG. 2.8b): forma indicada para pilares com momentos fletores elevados;

- Pilares engastados na fundação com dois elementos de cobertura articulados (FIG. 2.8c): forma para cobertura inclinada, exigindo a existência de tirante no topo dos pilares;
- Ligação rígida entre pilares e os dois elementos de cobertura (FIG. 2.8d): para coberturas inclinadas com ou sem tirante no topo dos pilares.



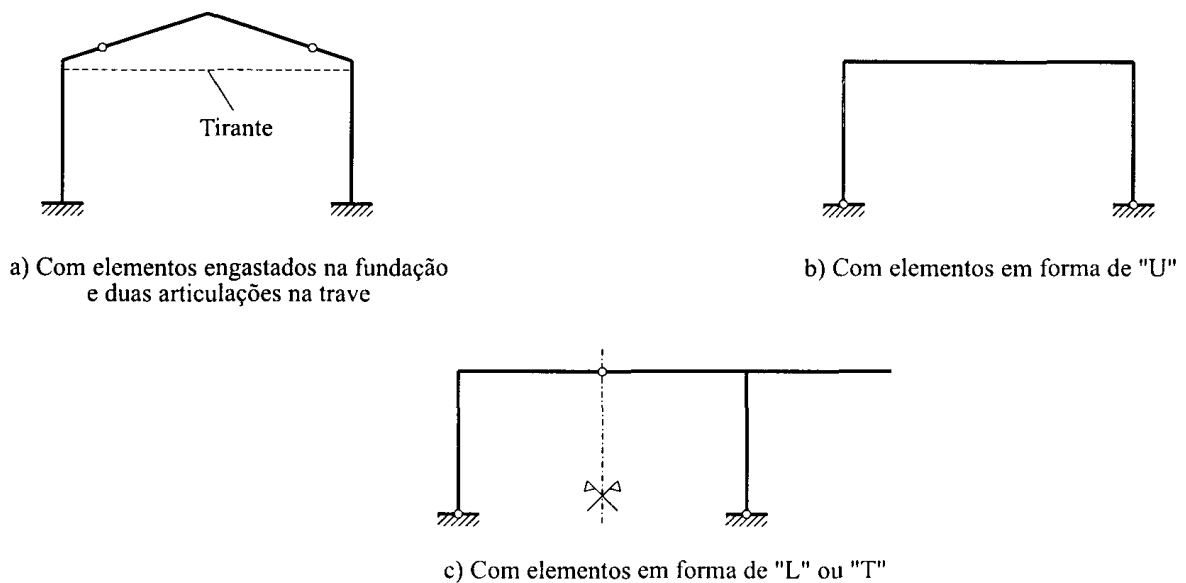
**FIGURA 2.8** - Formas básicas dos sistemas estruturais com elementos de eixo reto (EL DEBS, 2000)

### 2.3.2 Sistemas Estruturais com Elemento de Trechos de Eixo Reto ou Curvo

Esse sistema estrutural apresenta uma melhor distribuição dos esforços solicitantes que o sistema descrito no item 2.3.1. Porém o uso de ligações mais rígidas torna a pré-tração praticamente inviável e a execução, o transporte e a montagem mais trabalhosos.

Para esse sistema podem ser listadas as seguintes formas básicas:

- Elementos engastados na fundação com duas articulações na trave e tirante no topo dos pilares para reduzir o peso dos elementos (FIG. 2.9a).
- Elementos em forma de "U" ou de duplo "T", englobando pilar e trave, restritos a pré-moldado de canteiro (FIG. 2.9b);
- Elementos formados pela metade dos elementos em forma de "U" ou de duplo "T", para galpões altos e estreitos de um só vão (FIG. 2.9c);



**FIGURA 2.9** - Formas básicas dos sistemas estruturais com elementos compostos de trechos de eixo reto (EL DEBS, 2000)

O uso de elementos de eixo curvo se restringe à cobertura, possibilitando a redução nas solicitações de flexão, proporcionando uma redução no peso e no consumo de materiais desses elementos. Os apoios são formados por pilares do mesmo modo que para elementos de eixo reto.

Há, ainda, o sistema estrutural com abertura entre os banzos com o uso de elementos em treliça, viga Vierendel, ou viga armada, podendo ser empregado em: vigas, pilares ou em elementos compostos por trechos de eixo reto dos sistemas estruturais listados anteriormente. Apresenta a vantagem de redução de consumo de materiais para os elementos.

### **2.3.3 Sistemas Estruturais de Parede Portante**

Nesse sistema os painéis verticais promovem a sustentação da cobertura além do fechamento do galpão. Quando os vãos são muito grandes é utilizado o sistema reticulado no interior da edificação e apenas os painéis externos possuem função estrutural.

Há duas formas básicas para esse sistema:

- Paredes engastadas na fundação com cobertura apoiada sobre elas;
- Paredes apenas apoiadas na fundação com paredes e cobertura comportando com se formassem uma caixa.

### **2.4 Sistemas Estruturais para Edifícios de Múltiplos Pavimentos**

Os sistemas estruturais para edifícios de múltiplos pavimentos, comparados aos sistemas estruturais para edifícios de um pavimento, apresentam um maior número de elementos e, conseqüentemente, maior número de ligações, elementos de menor peso e vários elementos concorrendo em um mesmo nó.

Esses sistemas são classificados em:

- Sistemas estruturais reticulados:
  - Com elemento de eixo reto: formados basicamente por pilares e vigas;
  - Com elemento composto por trechos de eixo reto: os elementos incluem parte do pilar e parte da viga;
  - Em pavimentos sem viga: com elementos, do tipo laje, apoiados diretamente sobre elementos do tipo pilar;
- Sistemas estruturais de parede portante:
  - Com grandes painéis de fachada;
  - Com painéis da altura de um pavimento;
  - Com elementos tridimensionais.

Combinação entre esses sistemas pode gerar outros tipos de sistemas com paredes externas estruturais e parte interna da edificação com estrutura segundo em sistema reticulado.

Além desses sistemas estruturais, cabe destaque para os sistemas de pavimentos que englobam as vigas e lajes responsáveis por receber as cargas verticais e transmiti-las aos pilares. Os sistemas de pavimentos mais utilizados em todo o mundo são formados por vigas de seção "T" invertido, ou "L", recebendo painéis alveolares ou duplo "T", com ou sem capa de concreto moldado "in loco" (FIG. 2.5).

Para a estabilidade das estruturas podem ser utilizados sistemas de contraventamento que são compostos por barras cruzadas, paredes de contraventamento ou núcleos de contraventamento. Esses últimos são utilizados geralmente para edifícios de doze metros de altura ou mais.

#### **2.4.1 Sistema Estrutural Reticulado com Elemento de Eixo Reto**

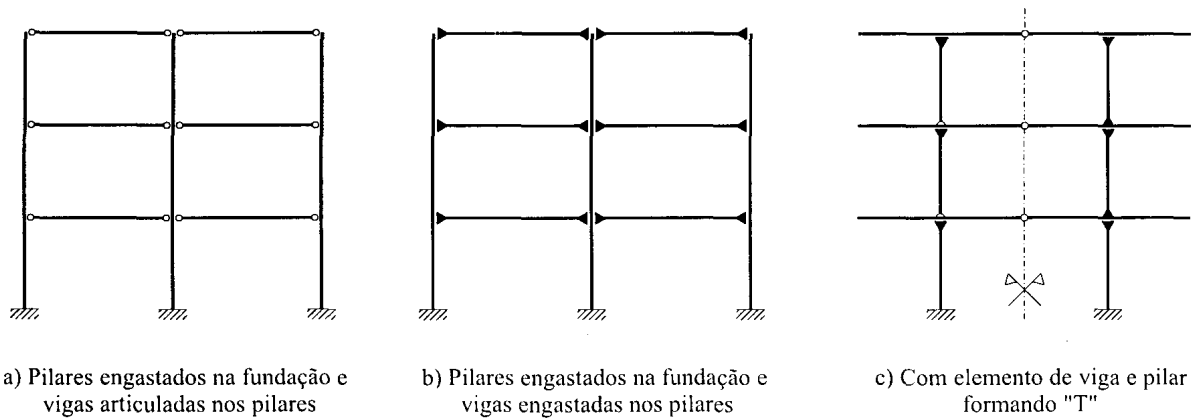
EL DEBS (2000) avalia que esse sistema estrutural é indicado para edifícios onde são necessários: grandes ou médios vãos, flexibilidade de *layout*, possibilidade de ampliações e cargas de utilização elevadas ou médias. Com essas características estão incluídos: edifícios comerciais, industriais ou de estacionamento, edifícios de escritório, escolas e hospitais.

Compostos basicamente por pilares e vigas, esse sistema apresenta elementos apropriados para a pré-fabricação com facilidade de manuseio e transporte, porém com a desvantagem da baixa redistribuição de esforços.

Para esse sistema pode-se listar as seguintes formas básicas:

- Pilares engastados na fundação com viga articulada nos pilares. (FIG. 2.10a);
- Pilares engastados na fundação com viga engastada nos pilares (FIG. 2.10b): forma indicada para edifícios altos;

- Elemento de viga e pilar formando "T", com pilares apresentando altura igual à altura do pavimento, recebendo as vigas sobre eles com ligações rígidas (FIG. 2.10c).

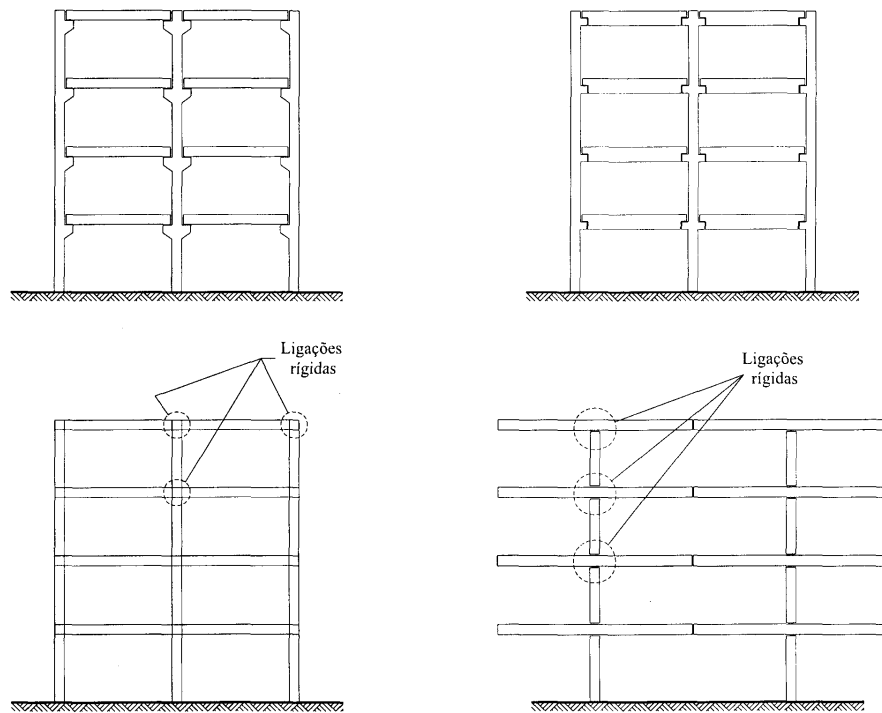


**FIGURA 2.10** - Formas básicas dos sistemas estruturais com elementos de eixo reto (EL DEBS, 2000)

Os esquemas construtivos resultantes desse sistema estrutural são mostrados na FIG. 2.11.

EL DEBS (2000) menciona que as duas primeiras formas básicas geralmente apresentam pilares na altura da edificação, sem emendas. Nesse caso a edificação apresenta até 20 metros, no máximo 30 metros. No caso de edificações com alturas maiores que 30 metros os pilares são emendados geralmente no terço médio entre dois pavimentos, defasando-se, ainda, as ligações dos diversos pilares entre os pavimentos. A forma com elementos tipo "T", por outro lado, possibilita a estabilidade de edificações relativamente altas.

Elementos com aberturas entre banzos são menos utilizados em edifícios de múltiplos pavimentos, podendo ser utilizados em coberturas ou para a passagem de instalações complementares da edificação.



**FIGURA 2.11** – Esquemas construtivos com elementos de eixo reto (EL DEBS, 2000)

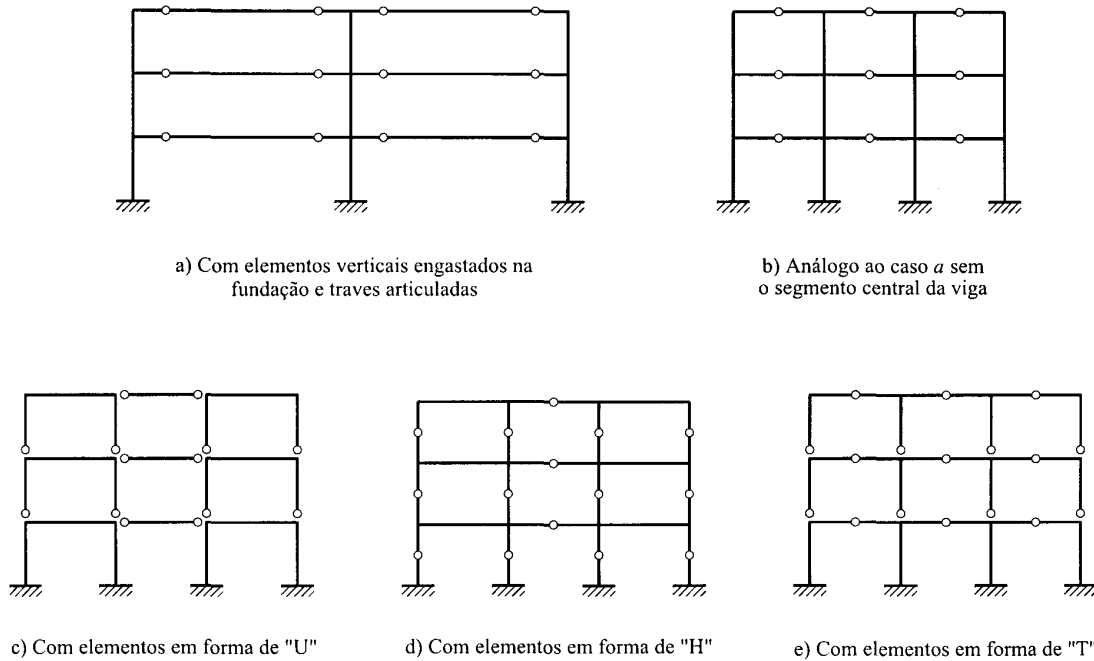
### 2.4.2 Sistema Estrutural Reticulado com Elemento Composto por Trecho de Eixo Reto

Esse sistema é formado por elementos que incluem em sua forma parte do pilar e parte das vigas, podendo-se listar as seguintes formas básicas:

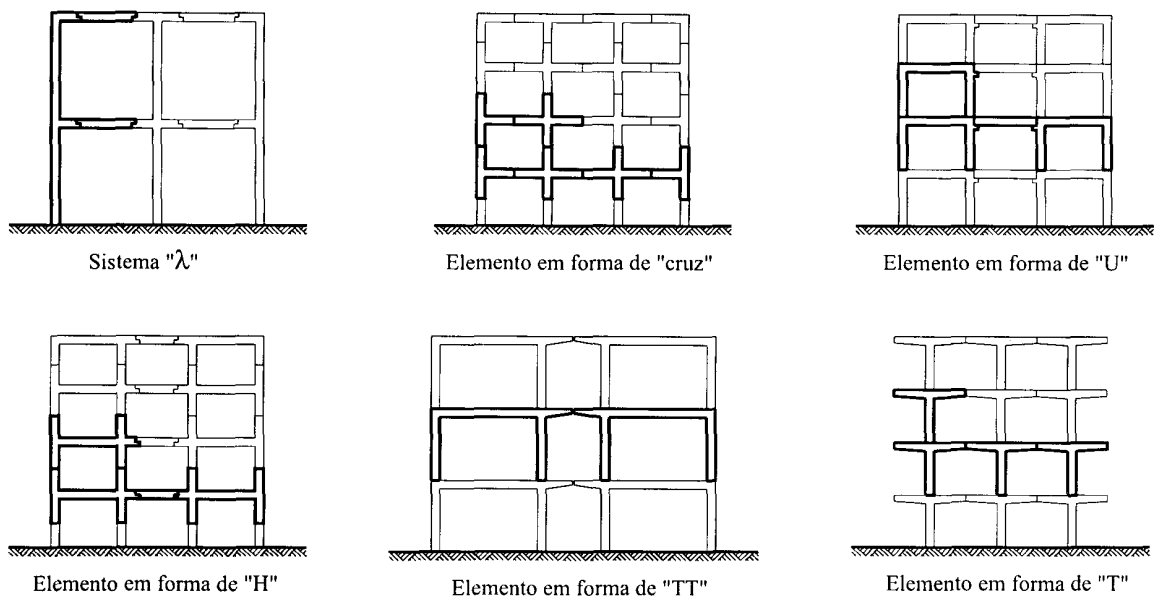
- Elementos verticais engastados na fundação e articulados nas travessas (FIG. 2.12a e FIG. 2.12b).
- Elementos em forma de "U" (FIG. 2.12c), de "H" (FIG. 2.12d), de "T" (FIG. 2.12e) ou similares.

Para a primeira forma as articulações geralmente estão localizadas próximas à posição de momento fletor nulo, considerando-se as cargas permanentes para estruturas monolíticas equivalentes. Essa forma equivale à primeira forma dos elementos de eixo reto, enquanto os elementos da segunda forma garantem a estabilização da estrutura para edifícios

relativamente altos. A FIG. 2.13 mostra os esquemas construtivos com elementos compostos por trechos de eixo reto.



**FIGURA 2.12** - Formas básicas dos sistemas estruturais com elementos compostos de trechos de eixo reto (EL DEBS, 2000)

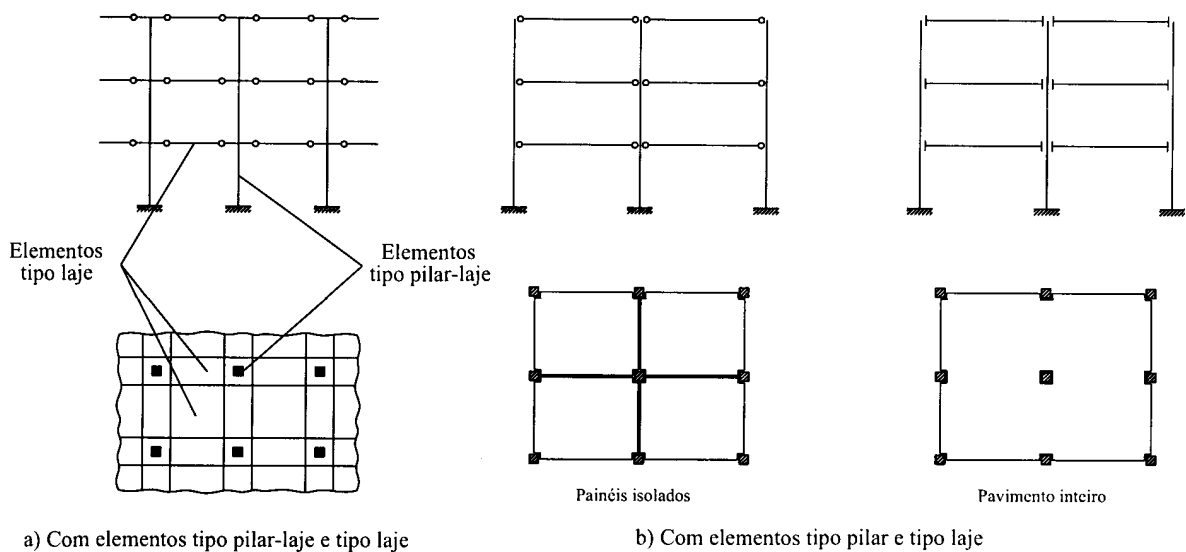


**FIGURA 2.13** - Esquemas construtivos com elementos compostos por trechos de eixo reto (EL DEBS, 2000)

### 2.4.3 Sistemas Estruturais Reticulados em Pavimentos sem Vigas

Esse tipo de sistema é indicado quando se deseja uma maior flexibilidade no *layout* da edificação. Os elementos estruturais desse sistema são pilares e lajes e suas formas básicas são:

- Com elementos dos tipos pilar-laje e laje (FIG. 2.14a);
- Com elementos dos tipos pilar e laje (FIG. 2.14b).



a) Com elementos tipo pilar-laje e tipo laje

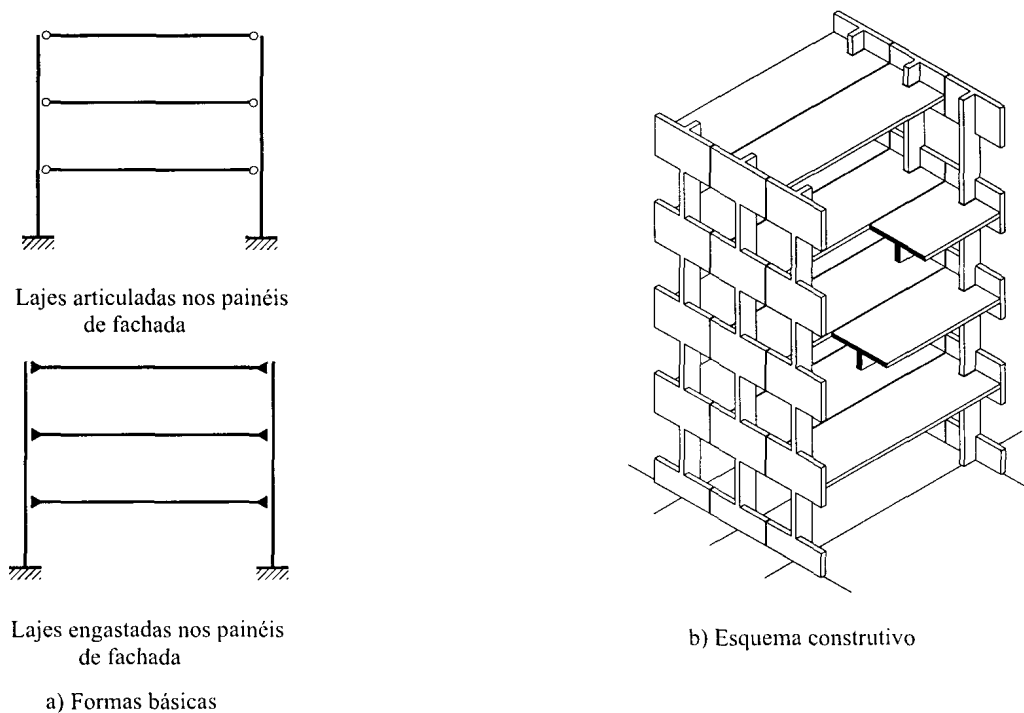
b) Com elementos tipo pilar e tipo laje

**FIGURA 2.14** - Formas básicas dos sistemas estruturais em pavimentos sem vigas (EL DEBS, 2000)

No primeiro caso os elementos pilar-laje correspondem a pilares com parte da laje acoplada. Devido a questões executivas, quando esse elemento apresenta laje acoplada pequena, quase como um capitel, os pilares podem ter comprimento superior ao do pavimento, porém, se essa laje for muito pronunciada, os pilares terão a altura do pavimento exigindo emendas em todos os pavimentos. No segundo caso as lajes podem ser apoiadas em quatro pilares ou apresentar as dimensões do pavimento. A estabilidade desse sistema é obtida geralmente pelo uso de contraventamentos.

## 2.4.4 Sistemas Estruturais de Parede Portante com Grandes Painéis de Fachada

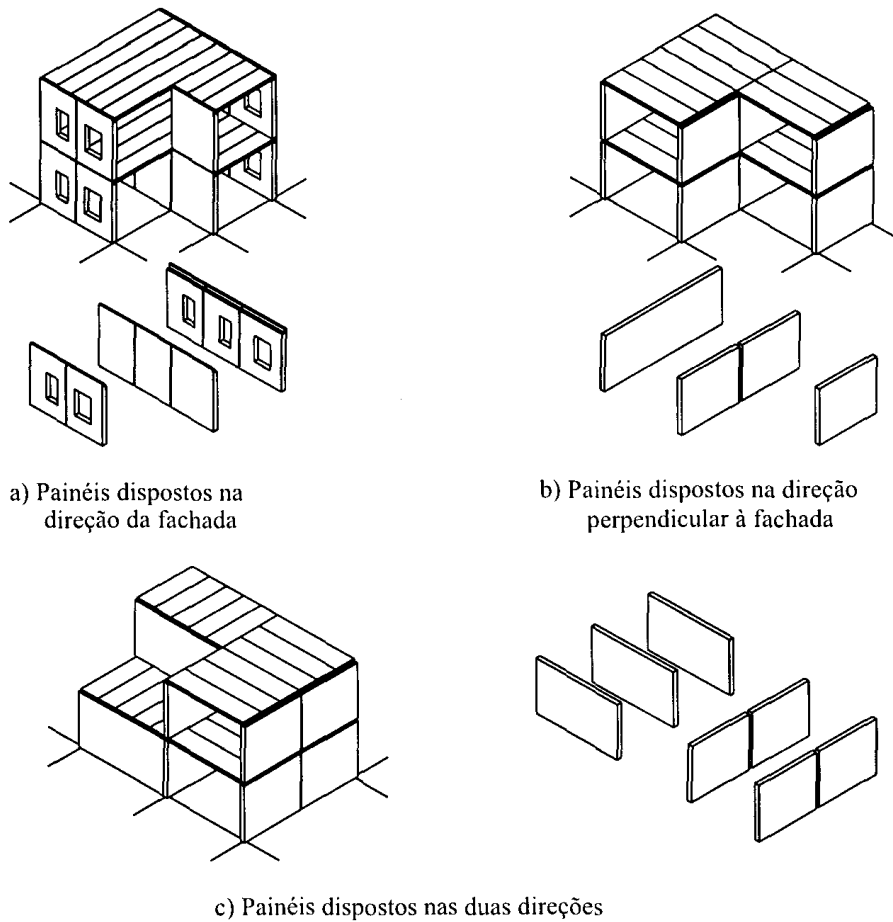
Nesses sistemas os painéis de fachada, englobando toda a altura da edificação, possuem função estrutural e recebem os elementos dos pavimentos através de ligações articuladas ou rígidas, essas últimas são de uso menos comum, ocorrendo em casos especiais (FIG. 2.15).



**FIGURA 2.15** - Formas básicas e esquema construtivo com grandes painéis de fachada  
(EL DEBS, 2000)

### 2.4.5 Sistemas Estruturais de Parede Portante com Painéis da Altura do Andar

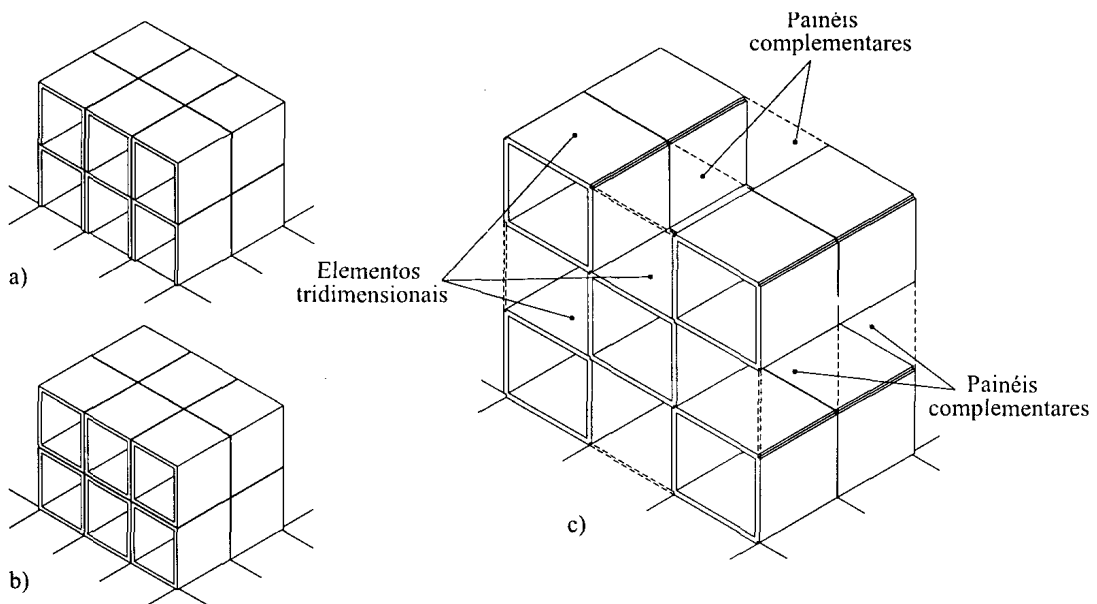
Esse sistema é formado por painéis de altura igual à altura do pavimento podendo ter largura igual às dimensões dos ambientes, ou serem compostos por vários painéis interligados (FIG. 2.16). Nesse caso a estrutura apresenta um elevado número de ligações, porém apresenta a vantagem de painéis com menor peso por unidade, facilitando o transporte e a montagem.



**FIGURA 2.16** - Esquemas construtivos com grandes painéis da altura do pavimento (EL DEBS, 2000)

### 2.4.6 Sistemas Estruturais de Parede Portante com Elementos Tridimensionais

Segundo EL DEBS (2000), os elementos desse sistema são células tridimensionais ou elementos volumétricos formados por elementos dispostos em dois ou mais planos formando parede e laje ou parede e parede (FIG. 2.17). Esses elementos, devido ao seu formato, possuem elevado peso porém, apresentam a vantagem de, muitas vezes, já apresentarem os acabamentos finais das superfícies.



**FIGURA 2.17** - Esquemas construtivos com elementos tridimensionais (EL DEBS, 2000)

### 2.5 Diretrizes e Parâmetros para Projeto de Estruturas em Concreto Pré-Moldado

Como princípios gerais para o projeto de estruturas de concreto pré-moldado o Professor EL DEBS (2000) destaca:

- O projeto já deve ser concebido para o uso de concreto pré-moldado. Esse uso deve ser considerado no dimensionamento de vãos e das cargas de utilização. A forma de

produção (execução dos elementos, transporte e montagem das peças e realização das ligações) deve ser avaliada para a concepção da estrutura.

- É primordial resolver as interações da estrutura com outras partes da construção já na fase de projeto. Através desse procedimento a pré-moldagem pode se utilizada para racionalizar os serviços de outras partes da construção. Devem ser consideradas as instalações hidráulica, sanitária, elétrica, captação de águas pluviais e de ar condicionado, além da consideração de impermeabilização, isolamento térmico, tipos de esquadrias, etc.
- O item ligações é um dos principais dificultadores do uso de estruturas de concreto pré-moldado. Por isso é primordial um projeto que busque uma minimização do número de ligações na estrutura, reduzindo o número de elementos da mesma. Porém, nessa redução, deve-se avaliar o tamanho e o peso dos elementos resultantes, considerando-se o transporte e os equipamentos disponíveis para a montagem da estrutura.
- A padronização da construção, com utilização de um número reduzido de tipos de elementos em uma estrutura, possibilita a produção seriada com utilização das mesmas formas para diversos elementos. Também o uso de elementos que desempenham mais de uma função pode favorecer a redução de custos de uma estrutura. Porém a redução de custos obtida pela padronização deve ser avaliada em relação às necessidades do cliente e expectativas do mercado.
- A previsão de utilização de elementos de mesma faixa de peso proporciona a racionalização do uso dos equipamentos de montagem da estrutura.

No projeto de estruturas de concreto pré-moldado é interessante o uso de coordenação modular, entendida como o uso de uma dimensão básica que relacione as dimensões de uma edificação com as dimensões dos elementos por ele utilizados. Ela consiste na criação de uma ordem dimensional para a padronização da edificação através do desenvolvimento do projeto com o uso de uma malha modular. As dimensões dos elementos devem ajustar-se a essa malha. O uso de coordenação modular permite a padronização dos componentes da estrutura com redução, ou até mesmo a eliminação, de adaptações nesses componentes com a escolha daqueles mais apropriados para a edificação.

## **2.6 Aspectos a Serem Considerados no Projeto e na Análise de Estruturas de Concreto Pré-Moldado**

O arranjo e a interação entre os elementos e os detalhes construtivos de uma estrutura de concreto pré-moldado são importantes aspectos a serem observados para garantir a rigidez e a estabilidade dessas estruturas.

Devido ao uso de ligações articuladas e de elementos geralmente mais esbeltos que os elementos de uma estrutura de concreto moldado "in loco" equivalente, as estruturas de concreto pré-moldado estão mais sujeitas a vibrações excessivas e ao colapso progressivo (ou ruína em cadeia). Assim, no projeto e na análise dessas estruturas EL DEBS (2000) lista os seguintes aspectos a serem considerados:

- Comportamento dos elementos isoladamente;
- Possíveis mudanças do esquema estático;
- Análise do comportamento da estrutura pronta;
- Incertezas na transmissão de forças nas ligações;
- Ajuste na introdução de coeficientes de segurança;
- Disposições construtivas específicas.

Entre as disposições construtivas, as tolerâncias e folgas devem ser consideradas na definição do comprimento dos elementos, prevendo-se possíveis desvios entre as medidas de projeto e as medidas reais. A tolerância é então definida como o valor máximo aceito para esse desvio entre as dimensões de projeto e as dimensões executadas, considerando-se execução, montagem e locação dos elementos.

As tolerâncias são estabelecidas para: garantir uma adequada montagem, possibilitar a consideração normatizada das variações da posição das forças nas ligações, estabelecer critérios de aceitação, definir responsabilidades e garantir um conforto visual para o usuário, evitando grandes distorções.

As tolerâncias indicadas pela NBR 9062/85 (ABNT, 1985) são apresentadas na TAB. 2.2.

**TABELA 2.2** - Valores das tolerâncias indicados pela NBR 9062/85 (EL DEBS, 2000)

<b>Execução</b>	Comprimento	até 5m	10mm
		de 5 a 10m	15mm
		acima de 10m	20mm
	Dimensão transversal		5mm
	Linearidade da peça		//1000
	Medida em planta entre apoios consecutivos		<sup>1</sup> 10mm
<b>Montagem</b>	Verticalidade		10mm a cada 3,0m, com 25 mm no máximo
	Nível dos apoios		<sup>2</sup> 10mm
	Em planta e em elevação dos pilares		10mm
<b>Locação</b>	Em planta para blocos pré-moldados sobre a fundação		50mm
1. Não pode exceder o valor acumulado de 0,1% do comprimento da estrutura.			
2. Não pode exceder o valor acumulado de 30mm, exceto para caminhos de rolamento, para os quais o valor é 20mm.			

Na definição do comprimento nominal da viga e comprimento mínimo do consolo devem ser consideradas as tolerâncias do pilar, da viga e do consolo.

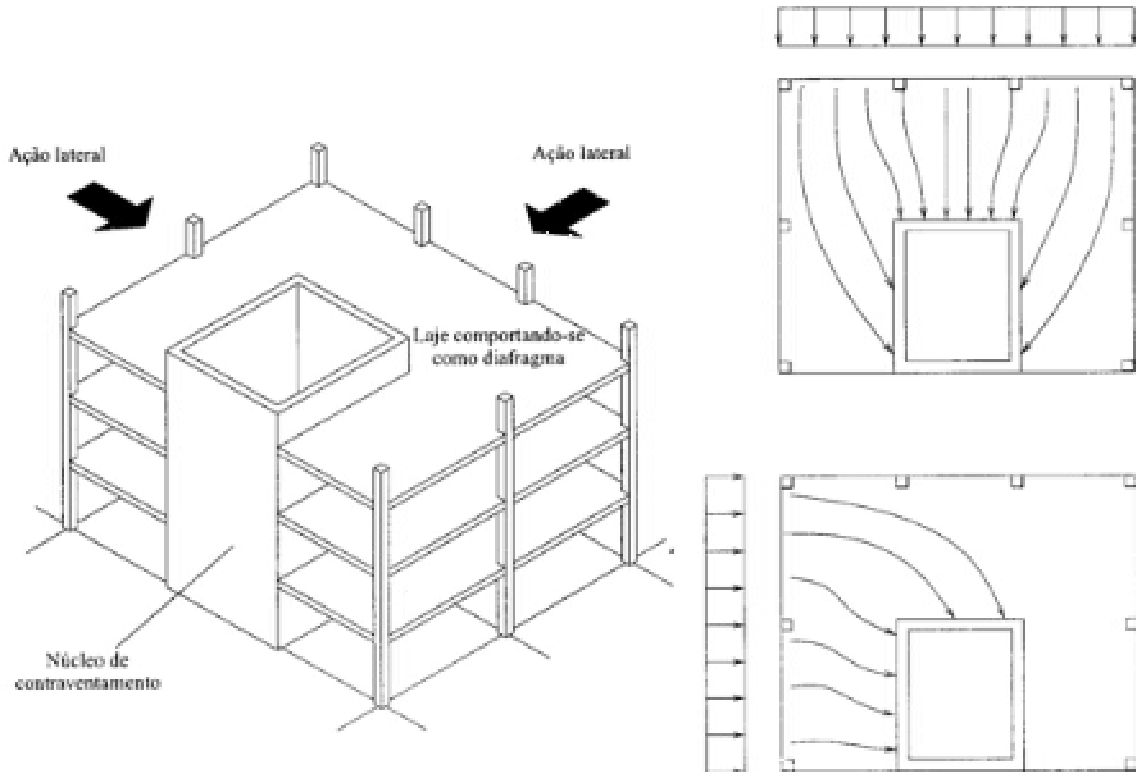
O professor EL DEBS (2000) define que "a estabilidade global da estrutura é associada à sua capacidade de transmitir com segurança, incluindo os efeitos de segunda ordem, as ações laterais, como vento e desaprumo, para a fundação e apresentar rigidez suficiente para limitar os movimentos devido a essas mesmas ações".

Com relação à estabilidade global, as estruturas de concreto pré-moldado, segundo o sistema reticulado, podem ser formadas por:

- Pilar engastado na base, comportando-se como vigas em balanço com relação às ações laterais, e vigas articuladas.
- Pilares e vigas formando pórticos onde, ou as ligações entre esses elementos são capazes de transmitir momento fletor, ou são utilizados elementos compostos por trechos retos.
- Paredes de contraventamento, ou núcleos que formam a estrutura principal, garantem a estabilidade global, contraventando os demais pilares.

As lajes podem transmitir os esforços em seu plano para os elementos de contraventamento através do efeito diafragma (FIG. 2.18). Para isso lajes formadas por elementos pré-

moldados devem ser capazes de transmitir os esforços entre os elementos que as constituem e entre esses e os elementos de contraventamento, exigindo cuidados especiais com a ligação entre tais elementos.



**FIGURA 2.18** - Comportamento da laje como diafragma (EL DEBS, 2000)

## 2.7 Considerações sobre as Ligações entre Elementos de uma Estrutura de Concreto Pré-Moldado

As ligações entre os elementos constituem um dos pontos principais de estruturas de concreto pré-moldado. Uma vez que nessas estruturas é usual o emprego de vigas em tramos simplesmente apoiados, com o uso de ligações articuladas, elas apresentam elementos mais solicitados à flexão com pouca capacidade de redistribuição de momentos fletores. Se por um lado, as ligações articuladas facilitam a ligação entre os elementos e a montagem da estrutura, por outro lado, o uso de ligações rígidas possibilita uma melhor redistribuição dos momentos fletores. Essas últimas, porém, trazem como consequência

ligações mais difíceis de executar, de custo mais elevado e com redução de rapidez na execução, reduzindo, portanto, as vantagens de uma estrutura de concreto pré-moldado.

Segundo EL DEBS (2000), a correta definição das ligações é uma das etapas mais importantes no projeto de estruturas de concreto pré-moldado. Ela é fundamental para a correta e adequada produção da estrutura, assim como para a definição do comportamento dessa, quando montada.

O mesmo autor apresenta uma classificação das ligações segundo o tipo de vinculação, o emprego de concreto e argamassa "in loco", o tipo de esforço principal transmitido, o uso de material de amortecimento.

A ligação, segundo o tipo de vinculação, pode ser articulada, quando não há transmissão de momento fletor, rígida, quando ocorre a transmissão de momento fletor, e semi-rígida, quando há transmissão parcial de momento fletor.

Quanto ao emprego de concreto e argamassa "in loco" a ligação será seca quando não há o uso de argamassa "in loco", caso contrário ela é definida como úmida.

Quanto ao esforço principal transmitido, a ligação pode ser solicitada: por compressão, por tração, por cisalhamento, por momento fletor ou por momento de torção.

Caso haja o uso de material de amortecimento entre as peças, a ligação é definida como macia. Caso contrário ela é definida como dura.

EL DEBS (2000) cita, ainda, alguns recursos utilizados nas ligações entre elementos de concreto pré-moldado:

- Armadura saliente e concreto moldado "in loco";
- Conformação por encaixes, recortes e chaves para disfarçar uma ligação, impedir deslocamentos relativos e proporcionar engastamento à torção durante a montagem;
- Cabos de protensão;
- Conectores metálicos, soldas e parafusos;

- Apoios de elastômero para possibilitar deslocamentos horizontais e rotações nos apoios e para melhor distribuir as tensões de contato;
- Dispositivos metálicos para fixação de outros elementos metálicos;
- Argamassa de concreto de granulometria fina para uniformizar tensões entre os elementos ou para preencher espaços.

Quando a transmissão de forças entre os elementos é feita por compressão, podem ser usados: argamassa de assentamento ou de enchimento, elastômeros ou elementos metálicos. Quando são transmitidos esforços de tração são usados emenda de armadura ou elementos mergulhados no concreto (dispositivos metálicos de içamento ou fixadores). Os esforços de cisalhamento podem ser transmitidos pelo concreto ou pela armadura quando são utilizadas barras cruzando a ligação ou conectores metálicos.

O autor coloca como princípios e recomendações gerais para o projeto de ligações:

- Assegurar rigidez e estabilidade global da estrutura;
- Considerar tolerâncias de fabricação e de montagem;
- Estender a análise das ligações às extremidades dos elementos que nelas concorrem;
- Previsão das acomodações das ligações até que seja atingida sua carga de trabalho.

Como diretrizes para o projeto e análise das ligações são sugeridas:

- Padronização de tipos de ligações e de dispositivos com o uso de simetria dos detalhes;
- Evitar acúmulo de armadura e de dispositivos metálicos;
- Reduzir os trabalhos após a desforma;
- Atenção com limitações dos materiais e dimensões reais dos elementos;
- Considerar folgas e tolerâncias, evitando tolerâncias de execução não padronizadas;
- Evitar elementos salientes sujeitos a danos durante transporte e montagem.

São componentes das ligações: as juntas de argamassa, os aparelhos de apoio de elastômero, os chumbadores, os consolos de concreto ou metálicos e os dentes de concreto ou dentes metálicos.

As juntas de argamassa são utilizadas para nivelamento e distribuição das tensões de contato, devendo ter a menor dimensão possível dentro de limites de execução e de

tolerância, sendo, de preferência menor que 10% da menor dimensão transversal dos elementos conectados.

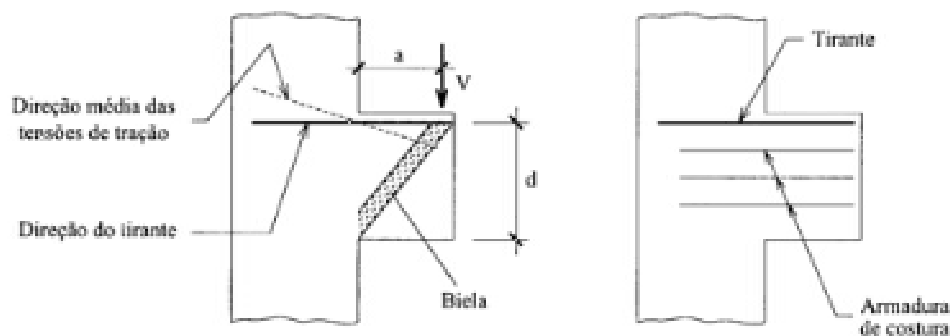
Para os aparelhos de apoio de elastômero o autor recomenda a espessura mínima de 6mm para apoio de nervuras de painéis duplo "T" e de 10mm para vigas em geral.

Os consolos são definidos como elementos estruturais que se projetam dos pilares ou paredes servindo de apoio para outros elementos da estrutura. Constituem balanços muito curtos para os quais em geral não se aplica a teoria da flexão. A NBR 9062/85 (ABNT, 1985) classifica os consolos em consolos curtos e muito curtos. Sendo:

- Consolos curtos, com  $0,5 < \frac{a}{d} \leq 1,0$ , são calculados pelo modelo matemático de uma treliça de duas barras, uma tracionada ou tirante e outra comprimida ou biela.
- Consolos muito curtos, com  $\frac{a}{d} < 0,5$ , o dimensionamento se faz supondo a ruptura ao longo do plano de ligação do consolo com seu suporte.

Essa norma sugere o tratamento dos consolos com  $1,0 < \frac{a}{d} \leq 2,0$  como viga em balanço, aplicando-se o disposto na NBR 6118/78, "Projeto e execução de obras de Concreto Armado" (ABNT, 1978), para flexão e força cortante.

Os parâmetros  $a$  e  $d$  são definidos conforme a FIG. 2.19



**FIGURA 2.19** - Parâmetros para definição do tipo de dimensionamento dos consolos segundo a NBR 9062/85 (EL DEBS, 2000)

A NBR 9062/85 (ABNT, 1985) especifica outras disposições construtivas e verificações adicionais para os consolos baseados no tipo de tirante do consolo, nos parâmetros definidos acima, no cobrimento e no diâmetro do tirante. Para consolos trapezoidais essa norma estabelece que "a altura da face externa do consolo não deve ser menor que metade da altura do consolo no engastamento, deduzido o afastamento da almofada de apoio à borda externa". Essa distância de afastamento da face externa da almofada de apoio à face externa do consolo é definida em função do diâmetro do tirante, do tipo de ancoragem e do cobrimento desse.






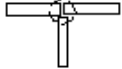
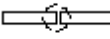

Segundo a NBR 9062/85 (ABNT, 1985) os dentes de concreto, ou dentes de apoio, são elementos de apoio na extremidade de vigas com recorte, placas ou painéis. Eles possuem altura menor que a altura do elemento a ser apoiado e podem ser assemelhados a consolos. Esses dentes podem ser também chamados de dentes Gerber (EL DEBS, 2000).

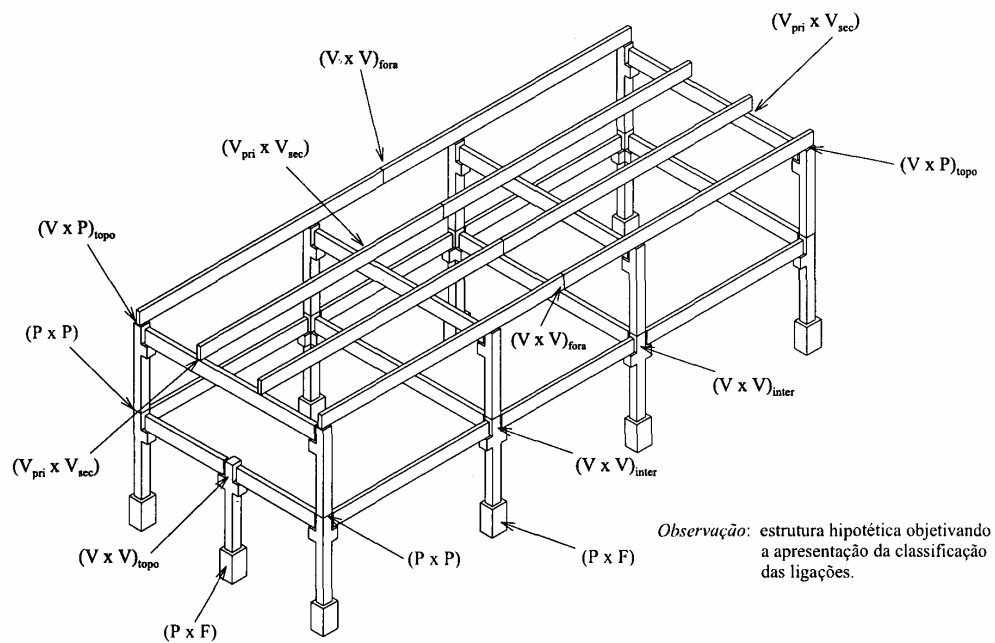
As principais formas de ligações entre elementos de concreto pré-moldado são as ligações entre elementos do tipo barra (pilares e vigas) e as ligações entre elementos de laje e parede.

Há ainda as ligações entre elementos não estruturais e a estrutura principal da edificação. Essas ligações são geralmente realizadas através de dispositivos metálicos, podendo transmitir esforços verticais relativos ao peso próprio dos elementos, esforços horizontais devido ao vento ou esforços de cisalhamento ao impedir o deslocamento relativo entre os painéis.

As ligações entre elementos do tipo barra são ligações entre: pilar e fundação, pilar e pilar, viga e pilar, viga e viga. No caso de ligações entre duas vigas essa pode ser junto ao pilar, fora do pilar ou entre viga principal e viga secundária. (TAB. 2.3 e FIG. 2.20).

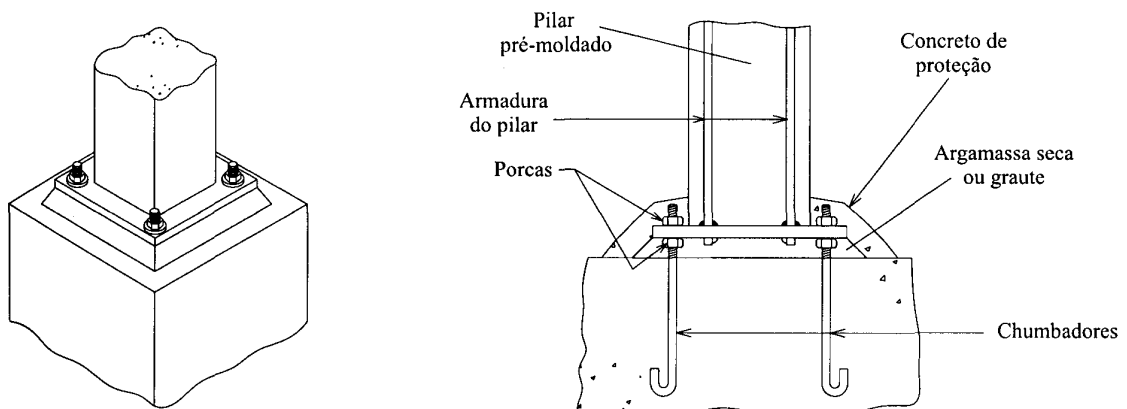
**TABELA 2.3 - Ligações em elementos tipo barra (EL DEBS, 2000)**

Grupo 1	pilar x fundação (P x F)	
	Pilar x pilar (P x P)	
Grupo 2	viga x pilar em ponto intermediário do pilar ( $V \times P$ ) <sub>inter</sub>	
	viga x pilar no topo do pilar ( $V \times P$ ) <sub>topo</sub>	
	viga x viga em ponto intermediário do pilar ( $V \times V$ ) <sub>inter</sub>	
	viga x viga sobre o topo do pilar ( $V \times V$ ) <sub>topo</sub>	
Grupo 3	viga x viga fora do pilar ( $V \times V$ ) <sub>fora</sub>	
	viga principal x viga secundária ( $V_{pri} \times V_{sec}$ )	

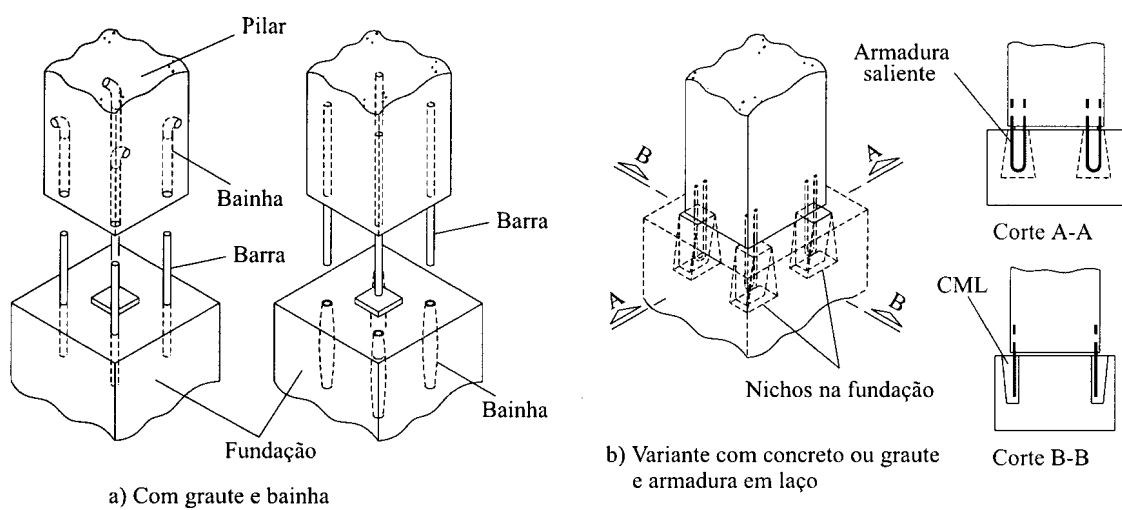
**FIGURA 2.20 - Classificação das ligações em elementos tipo barra (EL DEBS, 2000)**

### 2.7.1 Ligações entre Pilar e Fundação

As ligações entre pilar e fundação podem ser feitas por meio de cálice de fundação (FIG. 2.7), de chapa de base (FIG. 2.21) ou ainda por emenda da armadura, quer seja com graute e bainha (FIG. 2.22), quer seja com armadura saliente. Geralmente os espaços entre os elementos são preenchidos por graute ou concreto.



**FIGURA 2.21** - Ligação pilar x fundação por meio de chapa de base (EL DEBS, 2000)



**FIGURA 2.22** - Ligação pilar x fundação com emenda da armadura com graute e bainha e variante com concreto ou graute (EL DEBS, 2000)

Segundo EL DEBS (2000), as ligações por meio de cálice de fundação são de fácil montagem proporcionando ajustes aos desvios de execução. São realizadas pelo encaixe do pilar no elemento de fundação. Possibilitam uma boa transmissão de momentos fletores, porém geram uma ligação com a fundação mais onerosa. A TAB. 2.4. indica os valores mínimos para a profundidade do cálice, indicados pela NBR 9062/85 (ABNT, 1985).

**TABELA 2.4** - Valores mínimos do comprimento de embutimento do pilar segundo a NBR 9062/85 (EL DEBS, 2000)

<b>Paredes</b>	$\frac{M_d}{N_d h} \leq 0,15$	$\frac{M_d}{N_d h} \geq 2$
<b>Lisas</b>	1,5h	2,0h
<b>Rugosas</b>	1,2h	1,6h

Nota: Interpolar valores intermediários.

As paredes do pilar, no comprimento de engastamento, e as paredes do colarinho podem ser lisas ou apresentar dentes para melhorar a aderência entre esses elementos, nesse caso admiti-se que as solicitações sejam transmitidas em conjunto pelo pilar e pelo colarinho e o dimensionamento da fundação pode ser realizado considerando as dimensões do pilar na base iguais às dimensões externas do colarinho.

Segundo a NBR 9062/85 (ABNT, 1985), o comprimento de engastamento do pilar deve ser função do momento fletor e da força normal atuantes nele e da sua dimensão paralela ao plano de ação do momento, não sendo inferior a 40cm (quarenta centímetros), devendo, ainda, ser compatível com o comprimento de ancoragem da armadura do pilar. Quando a rugosidade do pilar e a rugosidade do colarinho são consideradas no dimensionamento dos elementos, essas devem ser obtidas por dentes no concreto os quais devem apresentar profundidade mínima de 1cm a cada 10cm. A porção do elemento de fundação situada abaixo do plano da superfície inferior do pilar não deve ser menor que 20 cm. As paredes do encaixe em pedestal ou colarinho devem ser armadas e suportar os esforços de montagem, momento fletor e força normal e devem ter espessura não inferior a 10cm.

EL DEBS (2000) sugere que, no detalhamento do cálice de fundação, devem ser observadas também as seguintes disposições construtivas:

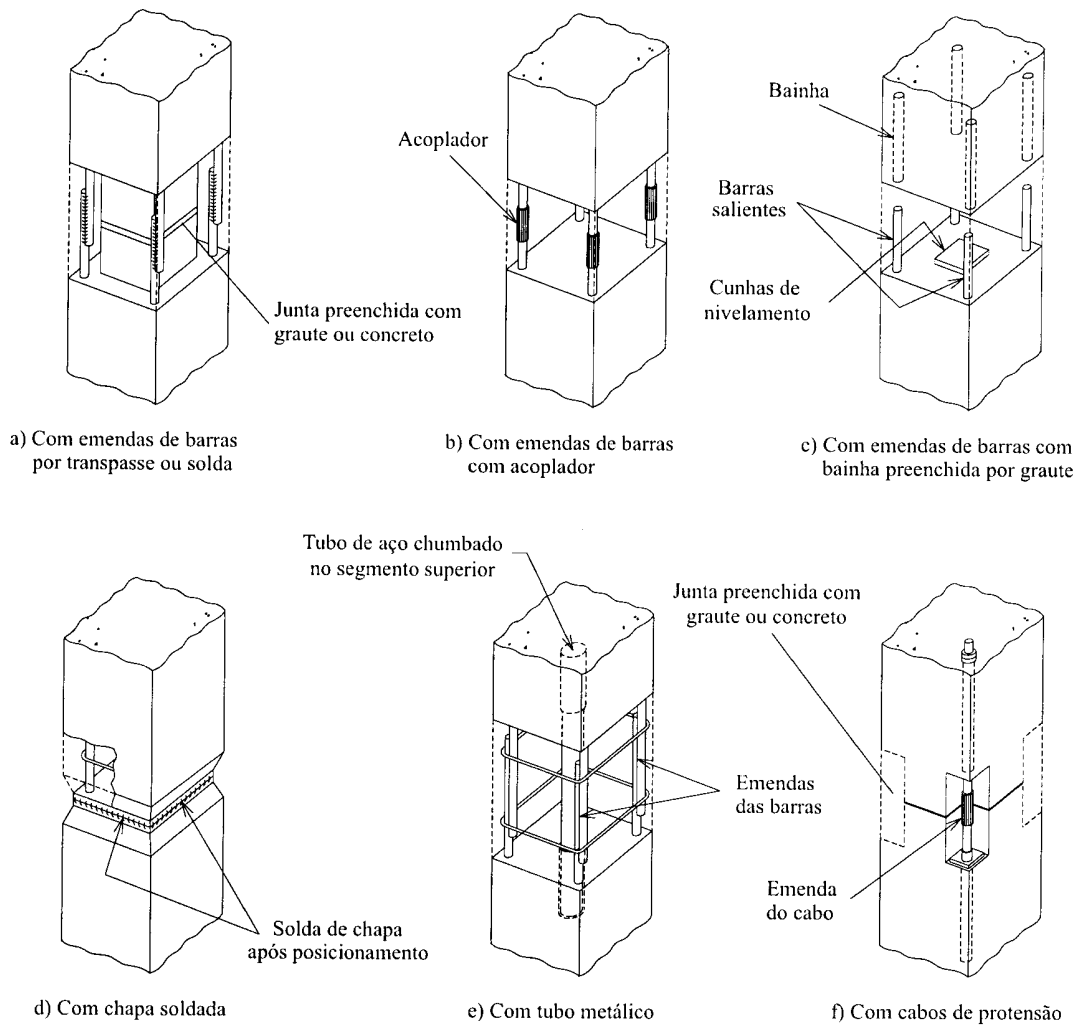
- Espessura da parede do colarinho não inferior a  $1/3$  da menor dimensão interna do colarinho;
- Um mínimo de 5cm de espaço entre o pilar e o colarinho para a entrada do vibrador de agulha, com exceção feita quando for usado graute auto-adensável.

As ligações com chapa de base unidas à armadura principal do pilar, chumbadores e porcas apresentam facilidade de montagem e de ajustes. Para chapas com dimensões maiores que as dimensões da seção transversal do pilar a transmissão de momentos fletores é boa, porém para chapas com dimensões iguais às do pilar essa transmissão é limitada.

As ligações por meio de armadura com graute e bainha também possuem boa capacidade de transmissão de momentos fletores, porém não possibilitam um ajuste fácil aos desvios de execução e de locação e exigem escoramento provisório. Já as ligações por meio de emenda, geralmente através de solda ou acopladores, das armaduras salientes do pilar e da fundação, apresentam dificuldade de montagem, de execução das soldas em campo e de concretagem adequada na área da emenda.

### **2.7.2 Ligações entre Pilares**

As ligações entre pilares podem ser executadas com emenda das barras da armadura dos pilares ou com cabos de protensão. As ligações com conectores metálicos e solda apresentam as mesmas características que as ligações com emenda de armadura entre os pilares e a fundação. Pode-se utilizar na ligação um tubo metálico que é concretado junto com um dos pilares para facilitar o posicionamento e prumo do pilar. Nesse caso os dois segmentos de pilar a serem unidos devem ser moldados em conjunto, na mesma posição em que serão montados. (FIG. 2.23)

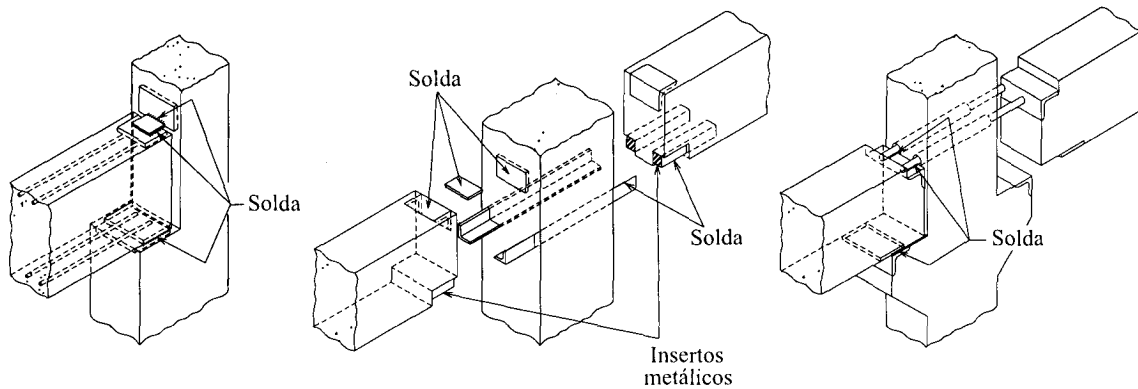


**FIGURA 2.23 - Ligações pilar x pilar (EL DEBS, 2000)**

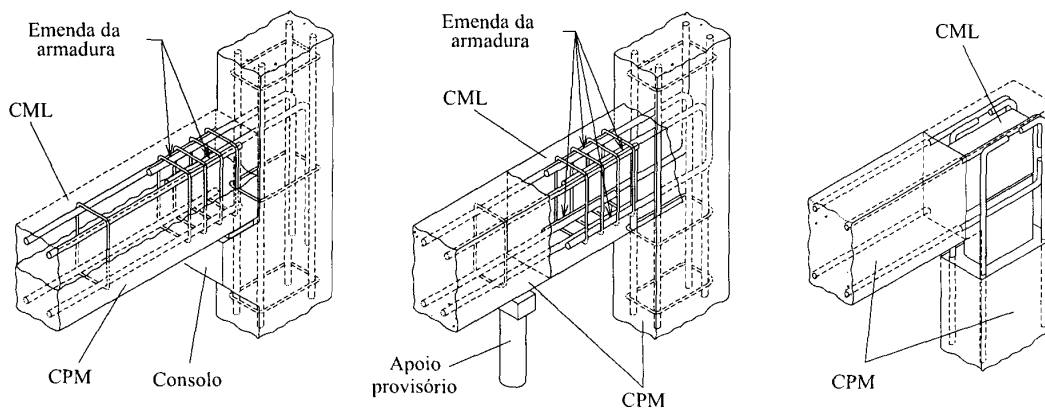
### 2.7.3 Ligações entre Viga e Pilar e entre duas Vigas

Segundo EL DEBS (2000), as ligações entre viga e pilar e entre duas vigas junto ao pilar podem ser rígidas ou articuladas. As ligações rígidas podem ser obtidas pelo uso de conectores metálicos e solda (FIG. 2.24), emenda das armaduras da viga e do pilar (FIG. 2.25) ou com cabos de protensão (FIG. 2.26). Nesses casos há a transmissão de momentos fletores. Para algumas ligações rígidas entre vigas e pilares as dimensões transversais do pilar são reduzidas junto à ligação para encaixe das vigas (FIG. 2.27). Porém as seções resultantes, nas regiões de reduções devem garantir a resistência do pilar em situações

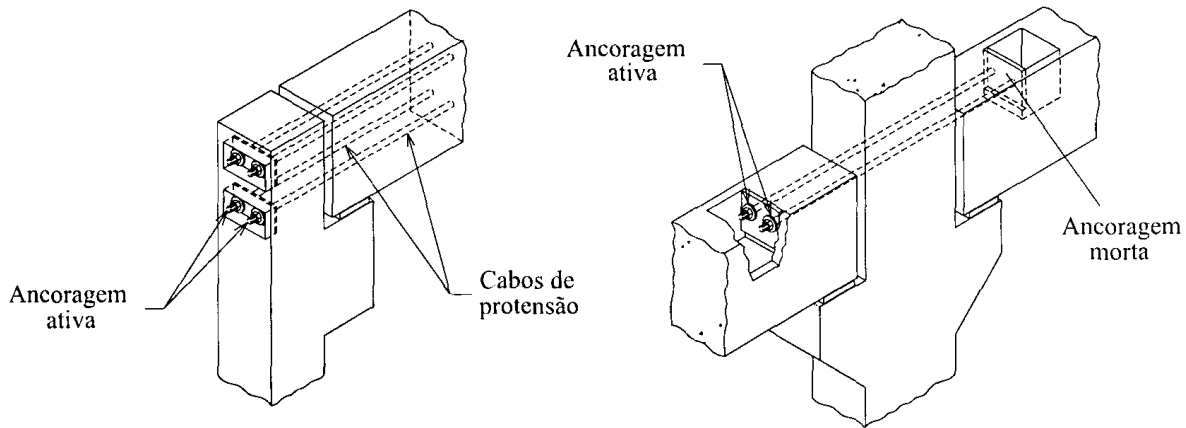
transitórias. Para as ligações articuladas podem ser utilizados chumbadores e chapas metálicas soldadas no topo das vigas para proporcionar a estabilidade lateral dessas (FIG. 2.28).



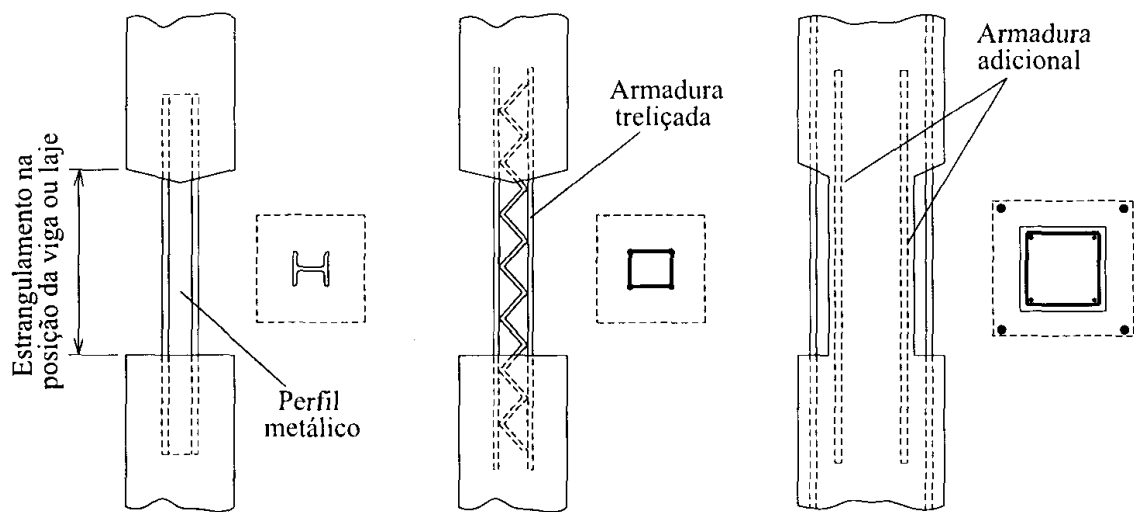
**FIGURA 2.24** - Ligações viga x pilar rígidas com solda (EL DEBS, 2000)



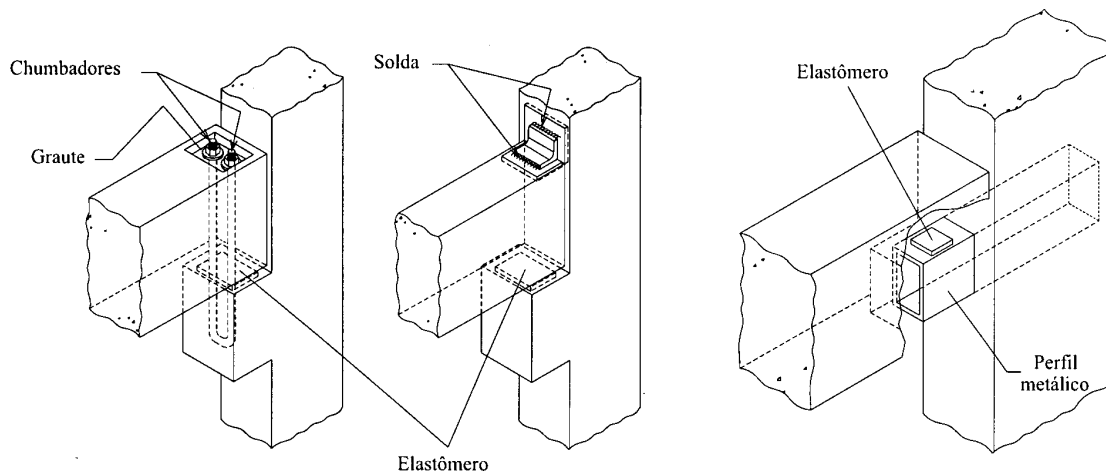
**FIGURA 2.25** - Ligações viga x pilar rígidas com emenda da armadura e concreto moldado no local (EL DEBS, 2000)



**FIGURA 2.26** - Ligações viga x pilar rígidas com cabos de protensão (EL DEBS, 2000)



**FIGURA 2.27** - Formas de estrangulamento dos pilares tendo em vista a ligação com as vigas ou lajes (EL DEBS, 2000)



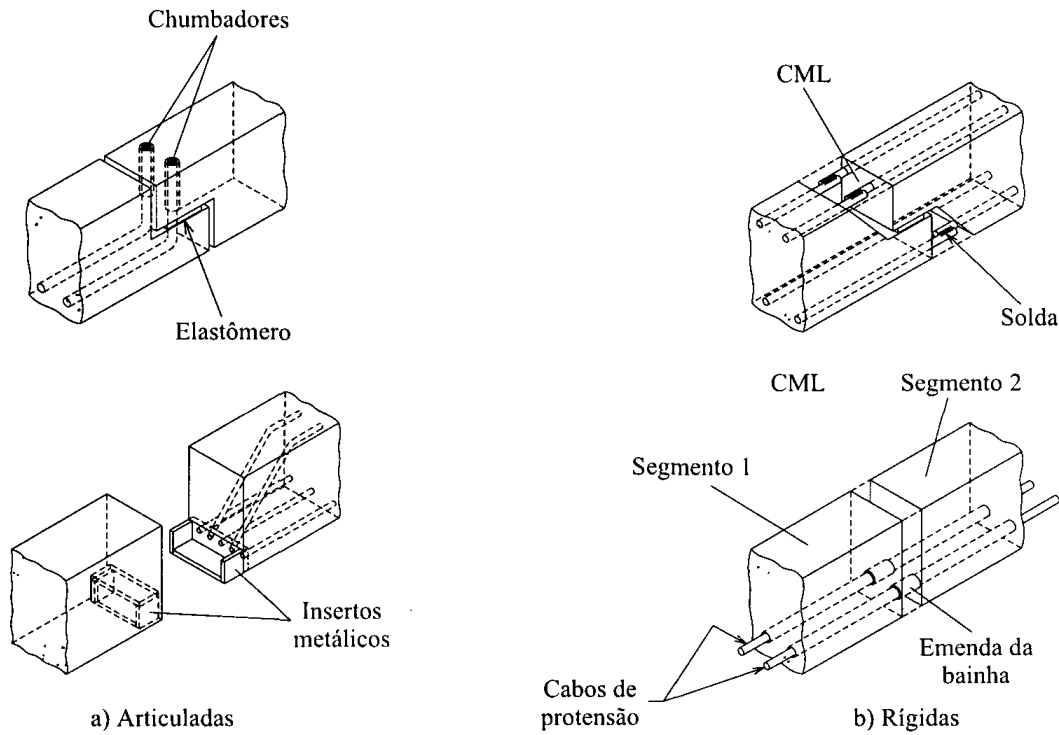
**FIGURA 2.28** - Ligações viga x pilar articuladas (EL DEBS, 2000)

As ligações entre duas vigas fora do pilar são geralmente articuladas e posicionadas próximo ao ponto de momento nulo de uma estrutura monolítica correspondente. As ligações rígidas são utilizadas geralmente em pontes com balanços sucessivos com o uso de juntas conjugadas coladas e protensão das armaduras. (FIG. 2.29)

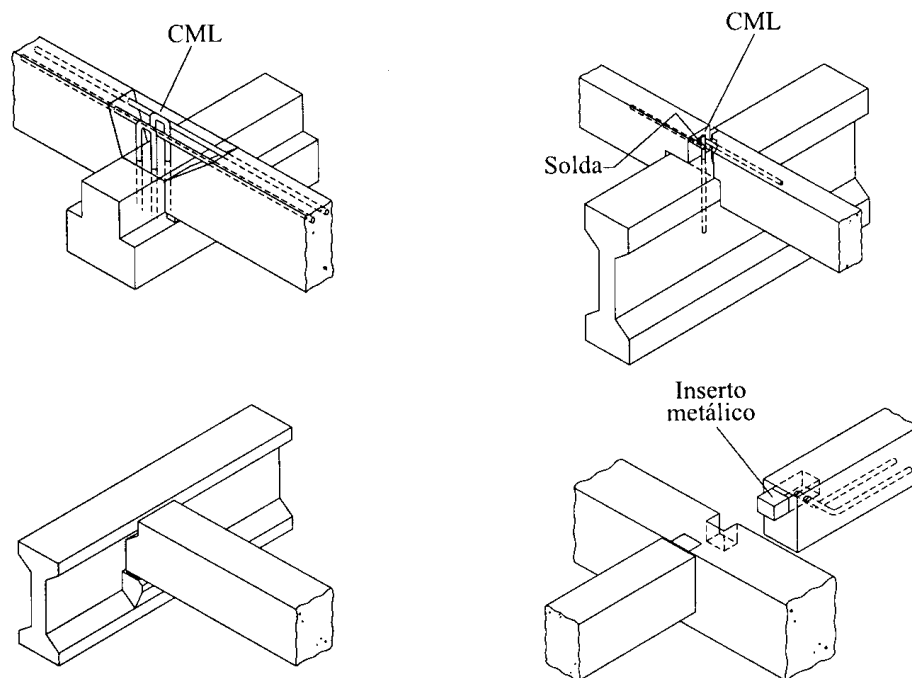
As vigas também podem servir de apoio para outras vigas. Nesse caso, as primeiras são chamadas vigas principais e as segundas são chamadas vigas secundárias.

As ligações entre viga principal e viga secundária ocorrem em pisos e coberturas e são geralmente articuladas. Nesses casos podem ser utilizados recortes nas vigas, quando necessário, para evitar o aumento da altura do piso ou da cobertura. (FIG. 2.30)

Quando as ligações entre vigas, ou entre laje e viga, são feitas por meio de apoios em abas das vigas (por exemplo, com o uso de vigas de apoio em forma de "L" ou de "T" invertido) a NBR 9062/85 (ABNT, 1985) estabelece que a largura da aba não deve ser inferior a 15 cm, a menos que a região de contato do elemento apoiado na base seja protegida por cantoneira metálica de largura maior ou igual à largura do elemento apoiado na aba.



**FIGURA 2.29 - Ligações viga x viga fora do pilar (EL DEBS, 2000)**

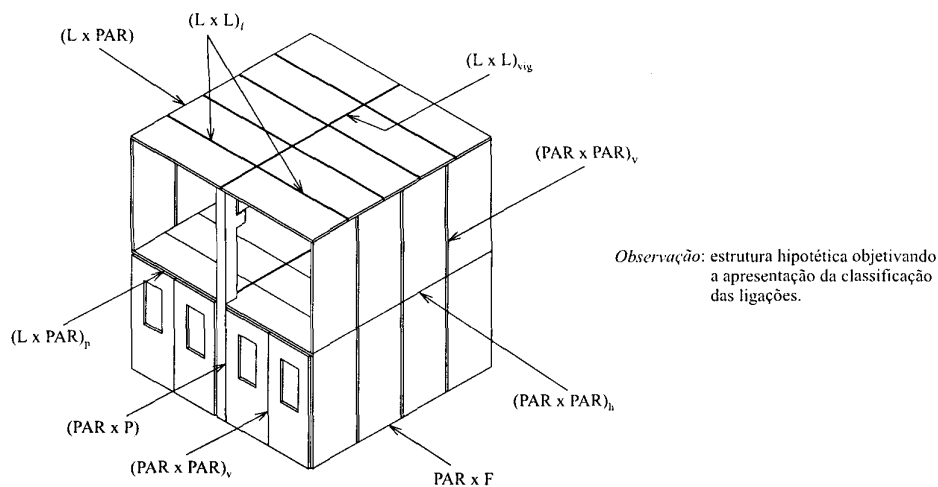


**FIGURA 2.30 - Ligações viga secundária x viga principal (EL DEBS,2000)**

## 2.7.4 Ligações entre Elementos de Laje e Parede

Através da FIG. 2.31, EL DEBS (2000) mostra a classificação dessas ligações para o caso de lajes formadas por elementos dispostos em uma direção.

	Principais tipos de ligações	Principais tensões ou esforços transmitidos
Grupo 1	laje x laje na direção longitudinal dos elementos $(L \times L)_l$	cisalhamento
	parede x parede na direção vertical $(PAR \times PAR)_v$	cisalhamento
Grupo 2	laje x parede $(L \times PAR)$	reação de apoio e eventualmente momento fletor
	laje x laje sobre viga $(L \times L)_{vig}$ ou sobre parede $(L \times L)_{par}$	reação de apoio e eventualmente momento fletor
Grupo 3	parede x parede na direção horizontal $(PAR \times PAR)_h$	força normal e cisalhamento
	parede x fundação $(PAR \times F)$	força normal e cisalhamento
<b>Outros tipos de ligações</b>		
	laje x parede $(L \times PAR)_p$ ou laje x viga $(L \times V)_p$ na direção paralela ao eixo dos elementos de laje	cisalhamento
	parede x pilar em estrutura de contraventamento $(PAR \times P)$	tensões normais e cisalhamento



**FIGURA 2.31-** Classificação das ligações entre elementos do tipo folha (EL DEBS, 2000)

Nessas ligações são transmitidas tensões de cisalhamento e tensões devidas à força normal. A transmissão de momento fletor pode ocorrer em ligações entre lajes sobre parede ou viga. No caso da ligação entre laje e viga essa deve merecer atenção quando há a necessidade de garantir o efeito diafragma. Na ligação entre lajes na direção longitudinal dos elementos pode-se usar uma capa de concreto moldado "in loco".

Assim, com base no estudo realizado, o PREMOLD foi desenvolvido para sistema estrutural reticulado, para edifícios de múltiplos pavimentos, com elemento de eixo reto, com pilares engastados na fundação e vigas articuladas nos pilares. As ligações articuladas entre pilares e vigas consideram: uma folga definida pelo aplicativo que pode ser alterada pelo usuário e uso de consolos de concreto com dimensões mínimas previstas pela NBR 9062/85 (ABNT, 1985). Nas ligações dos pilares com a fundação por meio de cálice, o dimensionamento dos blocos de fundação também considera as dimensões mínimas previstas pela norma NBR 9062/85 (ABNT, 1985), inseridas nas rotinas de dimensionamento automático do PREMOLD. As seções de pilares, vigas e painéis de piso são aquelas adotadas por uma empresa de pré-moldados de Minas Gerais (PREMO, 1999) e estão de acordo com os tipos de elementos indicados pelo professor El Debs em seu livro "CONCRETO PRÉ-MOLDADO: Fundamentos e Aplicações" (EL DEBS, 2000).

O sistema de pavimentos adotado neste trabalho foi o de uso mais comum com lajes e vigas, trabalhando juntamente com o sistema estrutural reticulado com elementos de eixo reto.

No presente trabalho não foram abordados os sistemas de contraventamento e não foram consideradas vigas com altura variável ao longo de seu eixo ou com alterações de seção transversal junto aos apoios, a não ser quando são formados dentes de apoio pelo uso de consolos embutidos nas vigas. Também não foram usados elementos com aberturas entre banzos.

O tratamento de ligações é automaticamente realizado pelo aplicativo, evitando erros de interferência entre elementos, proporcionando a correta locação dos mesmos. As ligações realizadas pelo PREMOLD não recebem elementos de ligação como elastômeros, chumbadores ou conectores metálicos. Em trabalhos futuros será possível inserir acessórios nas ligações pelo desenvolvimento de classes apropriadas para os mesmos e pelo acréscimo de funcionalidade aos comandos que lançam os elementos da estrutura.

O PREMOLD também permite o uso de uma malha modular tridimensional para auxiliar o lançamento dos elementos da estrutura.

# 3

## **FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO DO PROJETO "MODELADOR PREMOLD"**

Para o desenvolvimento deste projeto de pesquisa foram utilizados como ferramentas de desenvolvimento o AutoCAD 2002, o ObjectARX 2002 e o Visual C++ 6.0, conforme já mencionado no item 1.6.

O AutoCAD foi a plataforma gráfica escolhida para o desenvolvimento do aplicativo PREMOLD, por ser uma plataforma de grande aceitação no mercado, amplamente utilizada pelos escritórios e empresas de arquitetura e engenharia civil. Além de ser uma ferramenta genérica, aplicável em todas as áreas de desenvolvimento de projetos gráficos, o AutoCAD possui uma arquitetura aberta para o desenvolvimento de aplicativos específicos. Essa arquitetura aberta permite o desenvolvimento de aplicativos para áreas específicas através uma interface de programação (API - *Application Program Interface*),

que utiliza o ObjectARX (*AutoCAD Runtime Extention*). O ObjectARX é uma biblioteca de vinculação dinâmica (DLL - *dynamic link library*) onde as classes e funções utilizadas pelo AutoCAD estão disponíveis para implementações específicas. Através do ObjectARX podem ser realizadas derivações por Herança de suas classes nativas, acesso às estruturas de banco de dados e ao sistema gráfico do AutoCAD, além da criação de novos comandos que operam do mesmo modo que os comandos nativos do AutoCAD.

O AutoCAD e o ObjectARX foram desenvolvidos na linguagem C++. O Visual C++ é o compilador predefinido pela Autodesk, para o desenvolvimento de aplicativos para o AutoCAD com o uso do ObjectARX. O Visual C++ é uma ferramenta de programação desenvolvida pela Microsoft com suporte das ferramentas MFC (*Microsoft Foundation Classes*), um ambiente de desenvolvimento integrado baseado em Windows, chamado *Developer Studio* (HORTON, 1997), para criação de interfaces com o usuário. Com esse ambiente integrado de projeto, o Visual C++ permite uma programação mais rápida e eficiente devido às suas ferramentas internas de programação ((CULLENS et. al. ,1997) e (DEITEIL e DEITEIL,1997)).

Os aplicativos ARX podem ser criados com uma vinculação dinâmica à biblioteca MFC. Essa biblioteca é compartilhada com o AutoCAD, possibilitando a criação de aplicativos com interface gráfica do Windows (GUI - *Graphical User Interface*) com o uso de quadros de diálogo, barras de ferramentas, menus, etc..

### **3.1 AutoCAD 2002: Banco de Dados e Sistema Gráfico**

O AutoCAD, desenvolvido pela Autodesk Inc., é uma plataforma gráfica genérica. Ele possui uma série de recursos e comandos para a geração de entidades gráficas com elevada precisão geométrica (AUTODESK, 1999b, 1999c, 1999d, 1999f). Um desenho do AutoCAD é formado por um conjunto de objetos armazenados em um banco de dados. Esses objetos podem ser entidades gráficas, assim como outras formas de dados ou construções necessárias ao desenho como tabelas de símbolos e dicionários (CADTEC, 2000).

O AutoCAD é capaz de gerar e reconhecer objetos formados por curvas, superfícies ou sólidos. Entre seus recursos mais comuns estão a criação e edição de formas geométricas, recursos de modificação do desenho (apagar, mover, copiar), recursos de visualização, criação e inserção de blocos ou arquivos, criação, formatação e edição de textos e de dimensionamentos. Os recursos para o tratamento tridimensional e para o modelamento por sólidos utilizados pelo AutoCAD foram explorados no desenvolvimento do PREMOLD.

A partir das versões 12 e 13 do AutoCAD foram oferecidos os primeiros ambientes de programação para essa plataforma gráfica. Até a versão 12 a possibilidade de programação do AutoCAD era restrita a implementações menos complexas com o uso da linguagem AutoLISP. Essa linguagem fornece uma maneira simples de adição de comandos ao AutoCAD. Ela é uma linguagem interpretada e não compilada, ou seja, o código fonte é convertido em código de máquina pelo interpretador durante a execução do programa. Numa linguagem compilada o compilador converte o código fonte em linguagem de máquina antes da execução do programa, gerando um arquivo executável.

A versão 12 do AutoCAD, tornou disponível o ADS (*AutoCAD Development System*) que viabilizou a programação através da linguagem C, formando um conjunto de funções externas ao AutoCAD. Essas funções continuavam necessitando de um interpretador AutoLISP, o que mantinha a programação externa ao AutoCAD e a restrição gerada por aquela linguagem. O Auto LISP e o ADS comunicam-se com o AutoCAD através de "*interprocess communication*" (IPC) ou comunicação interprocessual.

O AutoCAD versão 14 foi o primeiro a apresentar um verdadeira arquitetura aberta, com a possibilidade de desenvolvimento de aplicativos na mesma linguagem do AutoCAD, a linguagem C++. Essa abertura foi proporcionada por uma biblioteca de vínculo dinâmico (DLL) denominada ObjectARX, que possui funções chamadas diretamente pelo AutoCAD. O ObjectARX, além de proporcionar o uso das classes e funções do próprio AutoCAD para programação, permite a aplicação dos paradigmas da Programação Orientada a Objetos (POO) para o desenvolvimento de aplicativos mais complexos pelo usuário.

A arquitetura flexível do AutoCAD permite que sejam desenvolvidos aplicativos complexos com módulos desenvolvidos em linguagem diferentes, devido a aplicação dos conceitos de programação COM (*Component Object Model*) que permite a comunicação entre módulos de programação utilizando os conceitos de programação orientada a objetos (KRUGLINSKI, 1997). Esta estruturação torna possível a construção no AutoCAD de um aplicativo base e a criação de extensões, utilizando outras linguagens de programação, que se utilizem daquele aplicativo através da interface COM. Torna-se possível, então, a utilização de módulos gráficos em VBA (*Visual Basic for Applications*), rotinas complexas em ObjectARX e funções simples de personalização utilizando-se Active-X (MALARD, 1998).

Os trabalhos de pesquisa para a geração do Modelador 3D, iniciaram-se com a versão 14 do AutoCAD que acabara de ser lançada, contando com os recursos do ObjectARX 2.0, que foram utilizados no desenvolvimento daquele aplicativo. O PREMOLD foi iniciado para a versão 2000i do AutoCAD, utilizando o ObjectARX 2000i. Devido a alterações consideráveis entre as versões 2.0 e 2000i do ObjectARX, foram necessárias adaptações na estrutura de código do Modelador 3D para que esse se adequasse às versões mais recentes e pudesse ser reutilizado no PREMOLD.

No decorrer do presente trabalho foi lançado o AutoCAD 2002 e o ObjectARX 2002. As diferenças entre as versões 2000i e 2002 não foram tão acentuadas, e a utilização do aplicativo para as versões 2002 (AutoCAD e ObjectARX) foi possível sem alterações na estrutura do código fonte até então gerado.

### **3.2 A Linguagem C++ e a Programação Orientada a Objetos - POO**

Segundo MONTENEGRO e PACHECO (1994), a linguagem C++ foi desenvolvida durante a primeira metade dos anos 80, nos laboratórios da AT&T, por Bjarne Stroustrup (ELLIS e STROUSTRUP, 1993). Seu desenvolvimento deu-se pela aplicação dos paradigmas da POO à linguagem de programação C. Essa linguagem fornecia um maior

poder de programação que as linguagens até então desenvolvidas e apresentava grande portabilidade do sistema, tanto pela sua padronização, quanto pela existência de vários compiladores para diferentes plataformas.

No desenvolvimento da linguagem C++, Stroustrup manteve a compatibilidade com os códigos já escritos em C, assim como com as bibliotecas C existentes.

A formalização da versão 1.0 da linguagem foi feita em 1986 com o lançamento do livro de Stroustrup, "*The C++ Programming Language*", quando foram acrescentados à linguagem C os elementos de POO, como herança simples e polimorfismo, além de elementos de sintaxe (MONTENEGRO e PACHECO, 1994). Desde então a linguagem vem sendo aprimorada com adição de novos preceitos da POO como: ponteiros e tipo de acesso a membros de classe, herança múltipla, templates, etc.

A POO constitui um padrão de programação, que possibilita um alto reaproveitamento de códigos, com sistemas formados a partir de sistemas menores ou projetos. Esses são constituídos por classes e objetos, onde dados e procedimentos constituem uma só entidade, estando "encapsulados" em um só elemento. O programa gerado segundo a POO é caracterizado pela intercomunicação entre objetos, contrastando com a programação estruturada, onde a execução do programa é caracterizada principalmente pelo acionamento de procedimentos que tratam e repassam dados.

Um sistema orientado a objetos deve possuir uma série de mecanismos básicos e conceitos que o definam como tal. Entre os mecanismos pode-se citar: objeto, classe, membros de classe, tipos de acesso, mensagens, herança, etc. São conceitos básicos associados a um sistema orientado a objetos: Abstração, Encapsulamento, Polimorfismo, Modularidade, Persistência e Tipificação.

MONTENEGRO e PACHECO (1994) definem a Abstração como o "processo de retirar do domínio do problema os detalhes relevantes e representá-los não mais na linguagem do domínio, e sim na linguagem da solução". Esse conceito está ligado à definição dos mecanismos de um sistema orientado a objetos (classes, objetos, atributos, etc.) na busca da formulação computacional do problema.

Um objeto é o modelo ou a representação de uma entidade real ou abstrata do domínio de um problema (FIG. 3.1). Essa representação deve incluir características ou dados do objeto abstraído (Atributos) e o modo de manipular tais dados (Métodos). O conceito de Encapsulamento baseia-se na implementação, em um mesmo objeto, de dados e procedimentos correlacionados (FIG. 3.2). Aos objetos estão associados: seu estado, comportamento e identidade. O estado de um objeto é caracterizado pelos valores assumidos por seus atributos. Seu comportamento é definido pelo seu modo de agir e reagir a mudanças em seu estado e sua identidade é o que o distingue dos demais objetos num determinado domínio.



**FIGURA 3.1** – Abstração de uma entidade real através de suas características



**FIGURA 3.2** - Encapsulamento: Atributos + Métodos

Uma classe é a abstração de um conjunto de objetos semelhantes, com atributos e métodos comuns, enquanto o objeto é a concretização de uma entidade no domínio do problema, assumindo um estado e uma identidade definida (FIG. 3.3). Os atributos e métodos de uma classe são chamados membros da classe. O tipo de acesso permitido a esses membros deve ser sempre definido, sendo responsável pelo grau de proteção do sistema contra

intervenções indesejáveis do usuário. Esse acesso pode ser público (acesso permitido a todas as classes do sistema), protegido (acesso permitido somente às classes derivadas), ou privado (membros acessados somente pela própria classe). O acesso privado é aquele que possibilita a maior proteção ao sistema.



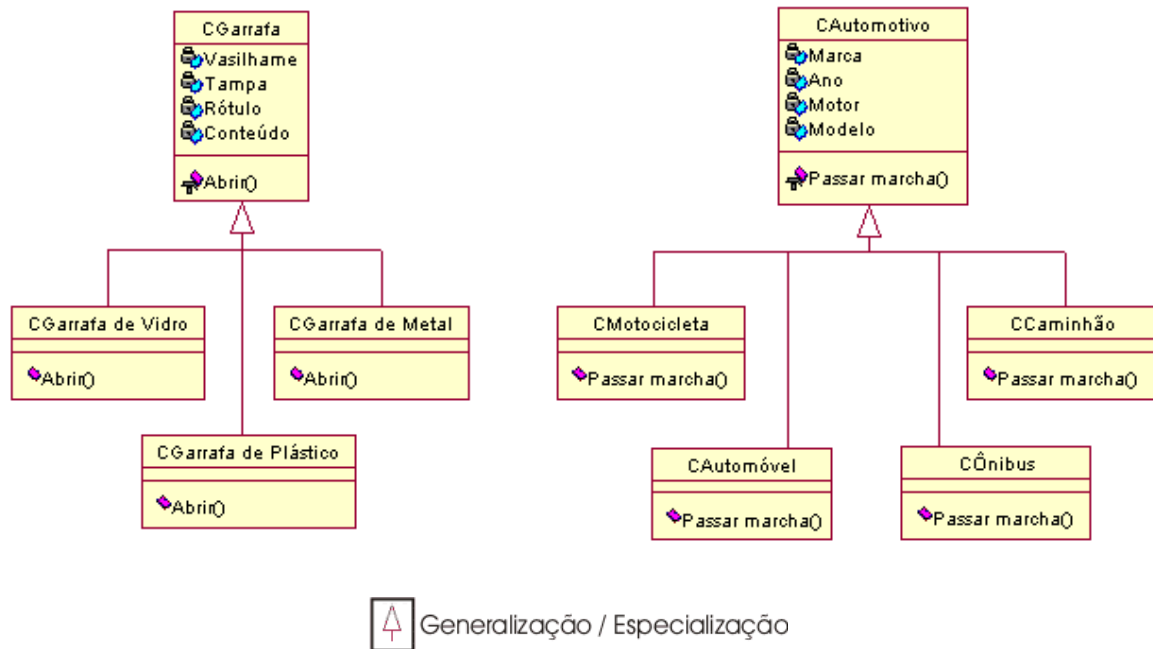
**FIGURA 3.3** - Classe e Objeto

A comunicação entre objetos caracterizando a execução de um programa orientado a objetos é feita pelo envio de mensagens, pela interpretação das mesmas e pelas respostas formuladas pelos objetos acionados. As mensagens podem ser entendidas como solicitações feitas aos objetos. Esses devem alterar ou não seu estado e retornar ao objeto solicitante algum valor que expresse o resultado dessa alteração. Assim, as mensagens são o acionamento, por um objeto, de procedimentos de um outro objeto.

A Herança na POO é o mecanismo que permite o compartilhamento de atributos e métodos entre classes. Pela hierarquia de classes uma classe pode ser constituída pela derivação de outra. A classe derivada, mais especializada, é chamada classe filha e a classe original, mais geral, é chamada classe base (FIG. 3.4). A classe filha herda as propriedades da classe base, sendo definidas para a primeira apenas suas diferenças ou particularidades em relação à segunda. A Herança pode ser simples quando uma classe filha herda as funções de apenas uma classe base ou múltipla quando herda características de mais de uma classe base.

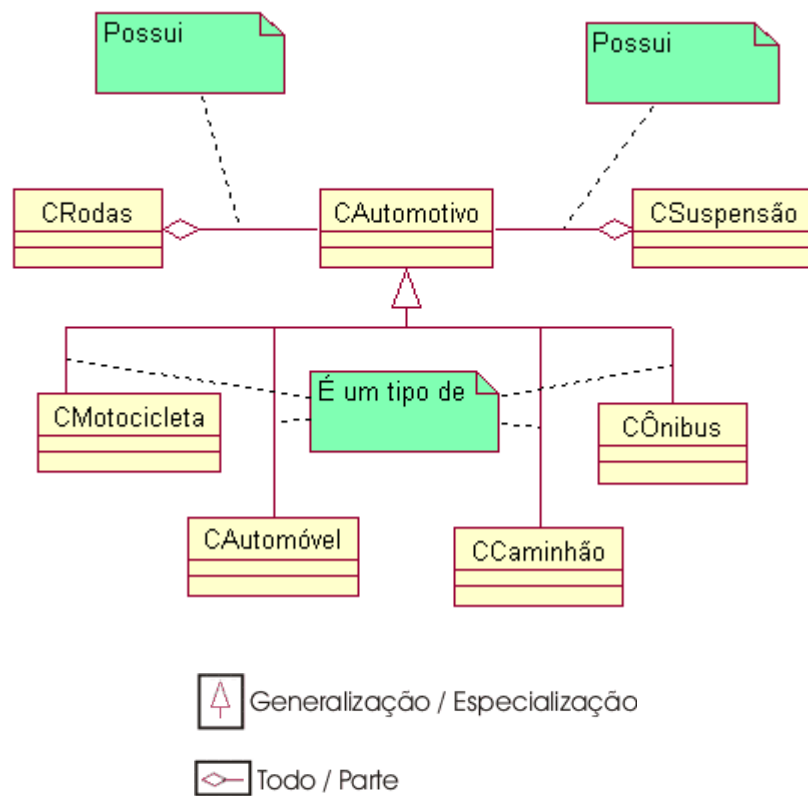
No presente trabalho, a análise orientada a objetos foi desenvolvida seguindo a abordagem sugerida pela engenharia de *software* utilizando Unified Modeling Language (UML). Os diagramas de classes e as relações de dependência serão sempre apresentados através dos

recursos de engenharia reversa disponível no Rational Rose (QUATRANI, 1998). Assim a FIG. 3.4 apresenta uma estrutura de Herança de classes segundo esses recursos.



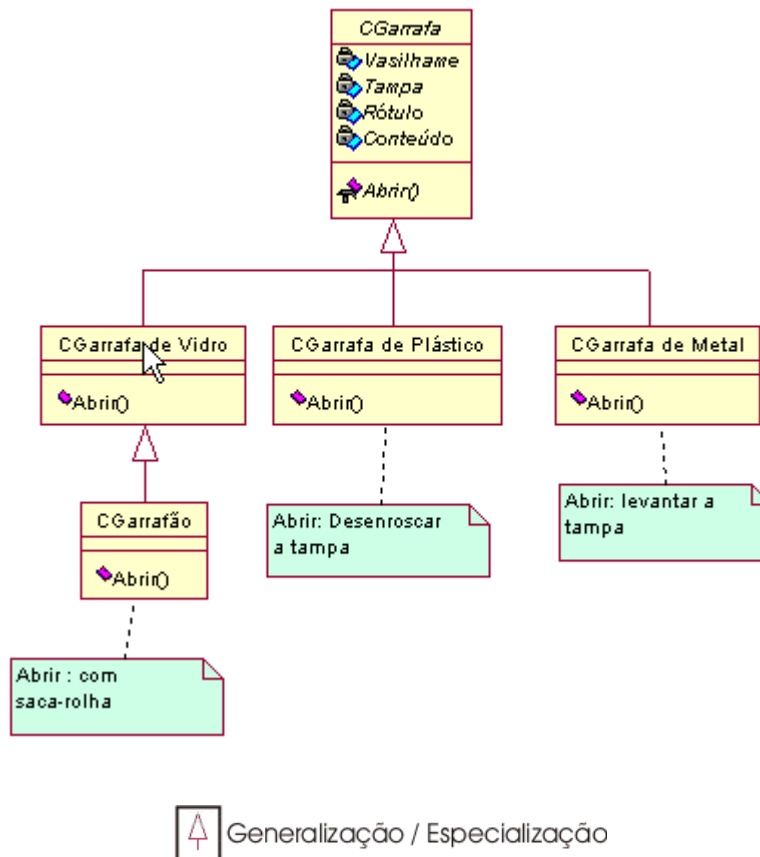
**FIGURA 3.4** - Herança de classes

A Herança caracteriza uma ligação entre classes do tipo Especialização/Generalização em que uma classe filha pode ser concebida como uma estrutura "é um tipo de" em relação à classe base. Outro tipo de ligação entre classes é o Todo/Parte ou Agregação/Decomposição que representa a estrutura "contém" em que uma classe é formada pelo agrupamento de subclasses, ou seja, possui como atributos instâncias de outras classes (FIG 3.5).



**FIGURA 3.5** - Estruturas: Generalização/Especialização X Todo/Parte

Na Herança de classes, muitas vezes é necessário definir novas funcionalidades para a classe derivada. Isso pode ser feito através da implementação funções membro da classe base com comportamento diferente para as classes derivadas. O termo Polimorfismo é utilizado para indicar a propriedade de se usar o mesmo nome para métodos diferentes, implementados em diferentes níveis de uma hierarquia de classes (MONTENEGRO e PACHECO, 1994). Para cada classe tem-se um comportamento específico para o método. Para se efetivar uma extensão de um programa que utiliza Polimorfismo, basta derivar novas subclasses de uma classe base, programando as modificações na nova classe (FIG. 3.6). A função a ser utilizada é definida pelo objeto sobre o qual ela é acionada. Para que haja Polimorfismo a ligação entre o objeto e a função é feita em tempo de execução. A implementação do Polimorfismo em C++ pressupõe a utilização de funções virtuais na classe base redefinidas nas classes derivadas. Os protótipos das funções nas classes derivadas devem ser idênticos aos protótipos da classe base.



**FIGURA 3.6 - Polimorfismo**

É possível também utilizar funções com o mesmo nome com ligação entre o objeto e a função em tempo de compilação, nesse caso não há Polimorfismo e sim sobrecarga de função. A sobrecarga sempre ocorre quando uma função já existente é declarada. Na sobrecarga de funções os protótipos das funções declaradas devem ser diferentes, ou seja, o número ou o tipo de parâmetros deve ser único para cada função homônima (MONTENEGRO e PACHECO, 1994).

A Modularidade é uma dos conceitos básicos de um sistema orientado a objetos. Um sistema complexo pode ser dividido em módulos. Cada módulo deve conter classes do sistema e manter sua independência de funcionamento. Os módulos independentes podem ser compilados separadamente. Assim, a modularidade é o agrupamento, em módulos independentes, de classes e objetos da estrutura lógica de um aplicativo.

São também conceitos básicos de um sistema orientado a objetos a Persistência e a Tipificação. Segundo MONTENEGRO e PACHECO (1994), a persistência está relacionada ao tempo em que um objeto permanece na memória do computador. MESQUITA (2001) define objetos persistentes como "aqueles que continuam existindo mesmo após o término de um programa". O mesmo autor cita que o gerenciamento dinâmico de memória e o armazenamento dos objetos em bancos de dados estão relacionados à persistência e que a Tipificação é a capacidade de um sistema orientado a objetos de distinguir diferentes tipos de classes.

Vê-se, então, a diferença básica da programação estruturada para a programação orientada a objetos. Na primeira é impossível identificar objetos devido à separação entre dados e procedimentos. Tal limitação impossibilita a existência de classes e, portanto, as vantagens de reaproveitamento de código proporcionado pela POO devido à herança e às estruturas Especialização/Generalização ou Todo-Parte.

Na linguagem C++ as definições das classes são feitas em arquivos fonte (terminação cpp). Esses possuem as implementações dos membros das classes, definindo atributos e implementando procedimentos ou métodos. As declarações das classes, ou seja, seus protótipos com declaração de atributos e funções, são armazenados em arquivos cabeçalho (terminação h). Os arquivos cpp ao serem compilados geram os arquivos codificados (terminação obj) e a montagem desses arquivos gera a biblioteca de vínculo dinâmico que recebe a extensão ARX no caso do ObjectARX. Na POO o reaproveitamento de código se dá pela utilização dos arquivos de cabeçalho, através de informações sobre a interface de uma classe, sem haver a necessidade de acessar o código de implementação dessa classe. Quando classes e funções são alteradas, desde que sua interface não sofra modificações, não há a necessidade de atualizações nos programas que utilizam essas classes. Esta é outra característica da POO: novas implementações podem ser efetuadas nas classes básicas como o melhoramento de códigos ou de desempenho, sem alterar os códigos gerados que utilizem tais classes.

### 3.3 O Ambiente de Programação ObjectARX

O ambiente de programação ObjectARX disponibiliza uma interface de programação orientada a objetos para o desenvolvimento de aplicativos em C++. Através do ObjectARX pode-se personalizar o uso do AutoCAD. As bibliotecas do ObjectARX compõem um conjunto versátil de ferramentas que possibilitam o acesso direto à estrutura do banco de dados do AutoCAD, ao seu sistema gráfico e à definição dos comandos nativos do AutoCAD (AUTODESK, 1999a, 1999e).

O ObjectARX pode ser utilizado para construir aplicativos complexos com a criação de interfaces com o usuário usando o MFC, além de possibilitar a interação com outros ambientes de programação. Para a criação de novos aplicativos pela derivação de novas classes (classes personalizadas), através da adição de funcionalidade às classes existentes e criação de comandos, são oferecidas macros que agilizam e facilitam o trabalho do programador.

O ObjectARX é constituído por cinco grupos de classes básicas: AcRx, AcDb, AcEd, AcGi e AcGe, onde Ac identifica classes do AutoCAD.

O grupo AcRX engloba classes do tipo estendidas, "*runtime extended*". São classes para a construção de aplicativos e para o registro e identificação de classes personalizadas em tempo de execução. O grupo AcEd (Ed = "*editor*") identifica as classes para registro de comandos nativos e eventos de notificação do AutoCAD. AcDb (Db = "*database*") é o conjunto de classes de banco de dados. O grupo AcGe (Ge = "*geometric*") engloba classes utilitárias para álgebra linear e objetos geométricos, enquanto as classes do grupo AcGi (Gi = "*graphics*") são classes com recursos gráficos para representação de entidades do AutoCAD.

### 3.3.1 Criando um Aplicativo ARX

O primeiro passo para a criação de um aplicativo ARX é a construção de dois projetos com todas as configurações exigidas pelo ObjectARX. O primeiro projeto, denominado ARX, armazena as funções, os comandos e as classes derivadas de classes do MFC, necessárias à interface do aplicativo. São também armazenadas funções de inicialização, funções para carregar e descarregar o aplicativo. É gerado um arquivo "*commands*" onde são definidos e implementados os comandos próprios do aplicativo criado. O segundo projeto deve ser criado como dependente do primeiro e recebe a terminação DBX. Esse armazena as classes definidas e utilizadas pelo aplicativo.

Uma das diferenças básicas do ObjectARX 2.0, utilizado no desenvolvimento do Modelador 3D e do ObjectARX 2002 é a estruturação do aplicativo. Até a versão 2000 havia apenas o projeto ARX, não havendo separação entre os arquivos de classes e demais arquivos. A partir da versão 2000 as funções de inicialização, as classes de interface e os comandos foram separados no projeto ARX, enquanto as classes próprias do aplicativo, derivadas de classes do AutoCAD, foram colocadas no projeto DBX. Além disso os comandos do Modelador 3D eram implementados nos arquivos de definição de classes derivadas do MFC, não existindo um arquivo próprio que reunisse todos os comandos. Essa diferença na estruturação dos aplicativos entre a versão 2.0 e a versão 2000, e posteriores a essa, exigiu uma reformulação do Modelador 3D para que o mesmo pudesse ser utilizado segundo a nova estruturação definida.

### 3.3.2 Banco de Dados

Um arquivo do AutoCAD é formado por um conjunto de objetos gráficos e não gráficos armazenados em um banco de dados. Os objetos do banco de dados, que possuem representação gráfica, são chamados de *entidades*, enquanto os objetos sem representação gráfica são chamados simplesmente de *objetos*.

As classes do AutoCAD para objetos e entidades são respectivamente *AcDbObject* e *AcDbEntity*. Assim, as classes personalizadas que utilizam objetos ou entidades devem ser derivadas de uma dessas duas classes. Na hierarquia de classes do ObjectARX a classe *AcDbEntity* é derivada da classe *AcDbObject*.

Os objetos do banco de dados são armazenados em objetos de armazenamento (*containers objects*). O AutoCAD possui dois tipos de objetos de armazenamento: as tabelas de símbolos (*symble tables*) e os dicionários. As tabelas de símbolo e os dicionários contêm entradas que são objetos do banco de dados, os quais podem ser procurados nos objetos de armazenamento através de *iterators*, objetos responsáveis por percorrer as tabelas de símbolos e os dicionários pesquisando seu conteúdo.

No banco de dados do AutoCAD existem nove tabelas de símbolo e um dicionário principal, o dicionário de objetos nomeados (*Named Object Dictionary*), referenciado apenas por NOD. As tabelas de símbolos não podem ser criadas nem destruídas por um aplicativo, porém pode-se adicionar ou alterar as entradas em uma tabela de símbolos. Essas entradas são chamadas de registros (*records*). Cada tabela de símbolos armazena apenas um tipo específico de objeto do banco de dados como: entidades, camadas (*layers*), tipos de linhas ou estilos de dimensionamento, etc. O *Block Table* é uma tabela de símbolos que armazena entidades. O NOD pode ser entendido como a tabela principal para o armazenamento de objetos não gráficos em um desenho. O NOD, segundo seu padrão inicial básico, possui quatro dicionários específicos. É permitida a criação de novos dicionários dentro do NOD, que armazenarão tipos específicos de objetos criados pelo aplicativo.

### **3.3.3 Manipulação de Objetos do Banco de Dados**

A comunicação entre objetos, com troca de mensagens e informações, caracteriza a execução de um programa orientado a objetos (MONTENEGRO, 1994). Assim, é indispensável para um aplicativo a individualização de um determinado objeto do banco de dados.

Quando uma entidade ou objeto é inserido em um objeto de armazenamento ele recebe um identificador único através do qual poderá ser acessado dentro do banco de dados. Esse acesso é necessário para que os objetos do banco de dados possam ser manipulados pelos comandos do AutoCAD ou pelo aplicativo criado. Esse identificador recebe o nome de *object ID*. O *object ID* identifica um objeto mesmo que vários bancos de dados estejam abertos ao mesmo tempo. Porém, o *object ID* não perdura de uma seção de edição para outra, ou seja, o *object ID* de um objeto não será necessariamente o mesmo em várias edições de um mesmo arquivo. Um outro identificador associado ao objeto do banco de dados é chamado *handle*. O *handle*, ao contrário do *object ID* é armazenado juntamente com o objeto quando o arquivo de edição é gravado, perdurando de uma sessão de edição para outra. Por outro lado, o *handle* é um identificador único para um objeto apenas dentro de um banco de dados, não garantindo sua identificação única caso outros bancos de dados estejam ativos dentro de uma sessão do AutoCAD.

Neste trabalho o *object ID* de um objeto ou entidade será referido apenas por identificador. O *handle* dos objetos e entidades não será usado.

### 3.3.4 Objetos e Entidades Personalizados

Os aplicativos desenvolvidos através do ObjectARX podem criar e gerenciar entidades nativas do AutoCAD. Entende-se por entidades nativas aquelas derivadas de classes do ObjectARX, como linhas, círculos, poligonais, arcos, textos, dimensionamentos, etc..

O desenvolvimento de aplicativos exige, muitas vezes, o gerenciamento de outros tipos de dados além de dados geométricos, todos incluídos em uma única entidade. Através de dados estendidos (*xrecords*), ou de protocolo de extensão (*protocol extension*) é possível armazenar alguns dados juntamente com as entidades nativas do AutoCAD. Porém, tais procedimentos apresentam certa limitação quanto ao armazenamento de informações e quanto à sua utilização. Essa limitação pode inviabilizar o desenvolvimento de aplicativos mais complexos.

Através do ObjectARX é possível criar também entidades ou objetos personalizados (*Custom Entities* e *Custom Objects*). As entidades e os objetos personalizados possuem características específicas definidas pelo programador, podendo herdar as características de entidades ou objetos do AutoCAD pela herança de classes. Seu comportamento é gerenciado pelo aplicativo juntamente com o AutoCAD. A criação desses objetos através da Herança de Classes, com a definição através de polimorfismo de funções específicas, permite ao AutoCAD um maior ou menor controle sobre os mesmos, tornando-os compatíveis com seu ambiente de trabalho. Esse tipo de entidade ou objeto não possui limites tão restritivos para o armazenamento de informações ou para sua utilização quanto os dados estendidos ou os protocolos de extensão. Porém, o controle completo dessas entidades exige uma implementação mais elaborada, com definição através de Polimorfismo de diversas funções herdadas da classe base para que todos os seus dados possam ser criados e gerenciados corretamente, conforme as peculiaridades do problema tratado pelo aplicativo.

Para garantir o controle do AutoCAD sobre as entidades ou objetos personalizados é necessário definir através de Polimorfismo um conjunto mínimo de funções. Esse conjunto é definido de acordo com a classe do ObjectARX, que será usada como classe base para a classe personalizada. Para a classe *AcDbObject* devem ser definidas, no mínimo, a função destrutora da classe, as funções que controlam a gravação dos objetos em arquivos de terminação dwg ou dxf, *dwgOutFields* e *dxfOutFields*, assim como as funções que lêem os mesmos dados quando tais arquivos são abertos, *dwgInFields* e *dxfInFields*. Para a classe *AcDbEntity* devem ser definidas, no mínimo, as funções obrigatórias da classe *AcDbObject* descritas acima, sendo sugerida a definição das funções que controlam alterações básicas nas entidades (AUTODESK,1999a) como:

- A função *worldDraw*, que controla a exibição das entidades na tela gráfica;
- As funções *transformBy* e *getTransformedCopy*, que gerenciam a movimentação, rotação e cópia de entidades;
- A função *getGeomExtents*, que retorna sua geometria;
- A função *getGripPoints*, que retorna os pontos de controle da entidade;
- A função *moveGripPointsAt*, que movimenta a entidade através dos pontos de controle da entidade.

A definição através de polimorfismo de outras funções além das mínimas exigidas para a herança de classes dependerá da funcionalidade que a classe personalizada deverá apresentar.

# 4

## **MODELADOR 3D: O PONTO DE PARTIDA**

O Modelador 3D, desenvolvido durante os projetos de mestrado de Fernando Poinho Malard (MALARD, 1998) e Alexandra Hütner (HÜNTNER, 1998) sob orientação do Professor José Ricardo Queiroz Franco do Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais - DEES/UFMG, foi implementado como um modelador unifilar tridimensional para estruturas compostas por barras, nós e superfícies planas horizontais.

A partir de um estudo prévio desse aplicativo, foram feitas algumas adequações necessárias para as versões 2000i e 2002 do ObjetARX, além de modificações consideradas importantes na sua herança de classes, para a geração do PREMOLD. A seguir apresenta-se a descrição do Modelador 3D original com o detalhamento da sua funcionalidade, da sua estrutura de classes, da aplicação de carregamentos, das suas classes

de controle e dos seus comandos. Em seguida são descritas as alterações iniciais aplicadas sobre o mesmo, para sua adequação às ferramentas atuais disponíveis. São explicitadas aquelas classes e funções aproveitadas, com alterações ou não, para o desenvolvimento do PREMOLD, objeto do presente trabalho.

#### 4.1 Descrição e Funcionalidade do Modelador 3D

No Modelador 3D, as entidades da classe **Node** representam os nós ou pontos de encontro de barras. Eles estão sujeitos a restrições ou liberações à translação ou à rotação em relação aos três eixos principais. Os nós podem também receber cargas concentradas nas direções dos mesmos eixos. As entidades da classe **Bar** representam barras, que são elementos lineares delimitados por um nó inicial e um nó final. Elas podem receber cargas concentradas, cargas uniformemente distribuídas em toda sua extensão ou parcialmente distribuídas, todas na direção z. As barras podem ser principais, secundárias ou de contraventamento. As entidades da classe **Face**, denominadas faces, representam as superfícies planas horizontais, que são planos delimitados por barras formando polígonos de três ou mais vértices, com uma direção de trabalho determinada. As faces correspondem a placas bi-apoiadas e recebem cargas uniformemente distribuídas em toda sua extensão. Existem três tipos de representação das faces: face plana, vão vazio ou vão de escada.

Todos os elementos gerados recebem uma numeração progressiva que corresponde à numeração de nós, barras e superfícies do modelo de análise estrutural. Para controle dessa numeração foi gerada uma classe chamada **Config1**, que gerencia a numeração independente de barras, nós e faces. Através dessa classe é possível numerar os elementos em ordem crescente quando esses são inseridos no banco de dados do AutoCAD. O número dado aos elementos passa a fazer parte de seus atributos e é um número inteiro único no desenho, que identifica o elemento. A classe **Config1** também é responsável pela captura da numeração dos elementos excluídos do banco de dados, para que os números utilizados por eles possam ser reaproveitados na numeração de outros elementos criados posteriormente.

O lançamento do modelo unifilar é feito através da criação de barras e de faces, uma vez que a existência de nós independentes não tem sentido estrutural. Ao serem lançadas as barras seu nó inicial e nó final são criados e numerados automaticamente. Dos dois pontos que definem a criação da barra, o nó inicial, por definição, é aquele que recebe a menor numeração, independentemente do fato de ter sido escolhido como ponto inicial ou final da barra. O eixo X local da barra é coincidente com sua direção e tem como origem o nó inicial da mesma. Caso uma barra seja criada com extremidade coincidente com a extremidade de outra barra já existente, não é criado outro nó naquele ponto, mas o nó existente passa a fazer a ligação entre as duas barras. As entidades da classe **Node** possuem como atributo uma lista, que armazena o número das barras ligadas ao nó. Essa lista é utilizada para o controle de conectividade e para o controle de compatibilidade entre nós e barras. Se uma barra é excluída do banco de dados, mas um de seus nós possui outras barras conectadas ao mesmo, esse não é excluído. Caso não existam outras barras conectadas ao nó, a exclusão da barra implica na exclusão automática do nó e a numeração dos elementos volta para o objeto de controle da classe **Config1** para serem reaproveitados posteriormente. É permitida ainda a renumeração automática de barras, nós e faces. A renumeração automática permite o estabelecimento de critérios de renumeração, como número inicial e precedência de um conjunto de barras em relação a outro, segundo sua posição espacial.

O lançamento de faces pode ser feito automaticamente em conjunto, quando são criadas entidades face em todas as regiões planas fechadas delimitadas por três ou mais barras, ou separadamente pela escolha de uma região particular. As faces criadas recebem, além de sua numeração, uma direção de trabalho predefinida conforme o tipo de geometria básica da face, se triangular, quadrada, retangular ou poligonal (com mais de quatro faces). Apesar de ser possível a criação de faces triangulares, não foi definida uma codificação para as mesmas. São determinados: seu vértice de origem e demais vértices. Por definição o vértice de origem é aquele de menor coordenada global. Os vértices coincidem com os nós das barras, que limitam a região de criação da face. Os eixos locais da face são definidos da seguinte forma: eixo X tem como base uma reta, que passa pelo vértice de origem e pelo vértice subsequente no sentido anti-horário; eixo Y é perpendicular ao eixo X no plano da face e eixo Z é o eixo ortogonal aos eixos X e Y da face. Esses três eixos têm como origem o vértice de origem da face.

A utilização dos comandos nativos do AutoCAD para cópia, movimentação e rotação podem ser utilizados normalmente. Quando um objeto é copiado o novo objeto é automaticamente numerado. Assim como foi determinado para a criação de elementos, só é permitida a cópia de barras e faces. Quando uma barra é copiada, novos nós são criados automaticamente nas extremidades da nova barra.

As barras são representadas por um seguimento de reta e por um texto que indica sua numeração. Os nós são representados por círculos ou quadrados concêntricos nos três planos principais (plano  $xy$ , plano  $xz$  e planos  $yz$ ) e por um texto que indica sua numeração. Os círculos indicam a liberação à rotação do nó nos planos em que são apresentados e os quadrados representam sua restrição à rotação também nos seus planos de desenho. Podem existir, ainda, linhas paralelas em um dos três planos citados que representam a restrição à translação do nó naquele plano. As faces são representadas por um símbolo de direção de trabalho localizado no centro geométrico do plano da face e por um texto que indica sua numeração.

Para melhor controlar a visualização do modelo, cada entidade é criada em uma camada de desenho própria. Os textos das entidades são também criados em camadas próprias.

Os carregamentos podem ser lançados após a definição da geometria dos elementos. Além dos tipos de carregamentos permitidos, descritos anteriormente, foi criada outra classe de controle que permite a criação e o gerenciamento de diferentes casos de carregamento. As cargas podem ser inseridas nos elementos, sendo sempre referenciadas por um determinado caso de carregamento.

Após o lançamento da geometria dos elementos e dos casos de carregamento pode-se gerar e exportar arquivos texto com informações sobre os elementos do modelo unifilar. São informados os nós, suas coordenadas, suas restrições ou liberações, os valores, a direção e o caso de carregamento de cargas nodais. Para as barras são informados sua numeração, sua incidência nodal, o tipo de barra, os pontos inicial e final de aplicação de cargas e o caso de carregamento de cargas aplicadas. Para as faces são informados: sua numeração,

sua direção de trabalho, o tipo de representação, seu vértice de origem e demais vértices, o valor e o caso de carregamento de cargas aplicadas.

## 4.2 Estrutura Global do Aplicativo

### 4.2.1 Classes Básicas: Node, Bar, Face

Foram criadas três classes básicas derivadas de classes do ObjectARX para representar os elementos estruturais do modelo unifilar: classe **Bar**, classe **Node** e classe **Face**. As classes **Node** e **Face** foram criadas a partir da derivação da classe *AcDbEntity*. A classe **Bar** foi criada a partir da derivação da classe *AcDbCurve*. A derivação da classe *AcDbCurve* foi escolhida para que a função *OFFSET* do AutoCAD pudesse ser utilizada sobre barras, mantendo-se a conectividade, a numeração automática de barras e de nós e a criação automática dos nós.

Essas classes são independentes, mas devido às inter-relações existentes entre elas foi utilizado o conceito de classes *friend*, ou amiga, na criação das mesmas.

Na POO, ao declarar-se uma classe *friend*, essa classe passa a ter acesso aos membros privados da classe da qual ela é *friend*. Essa declaração é um modo alternativo à implementação da estrutura Todo-Parte quando essa é efetivada por agregação de classes. Na estruturação Todo-Parte a classe "Todo" é entendida como aquela formada pela agregação de objetos de outras classes. As classes agregadas são chamadas classes "Parte". Nesse tipo de estrutura a classe "Todo" só pode acessar diretamente os dados públicos da classe "Parte". Os membros privados devem ser acessados através de funções próprias de acesso público oferecidas pela classe "Parte".

Pela declaração de classes *friend* não existe mais a estrutura Todo-Parte, pois uma classe não contém objetos de outras classes como atributos. A classe *friend* tem livre acesso aos membros privados da classe na qual ela é declarada *friend*. Assim, esse tipo de declaração diminui as características de encapsulamento da POO por alterar as restrições de acesso.

Isso pode ser perigoso, pois, a restrição de acesso é um mecanismo de segurança da POO, impedindo que classes derivadas façam alterações indesejáveis na classe base. Assim, o uso excessivo de classes *friend* pode prejudicar o funcionamento da classe base e de suas demais classes derivadas.

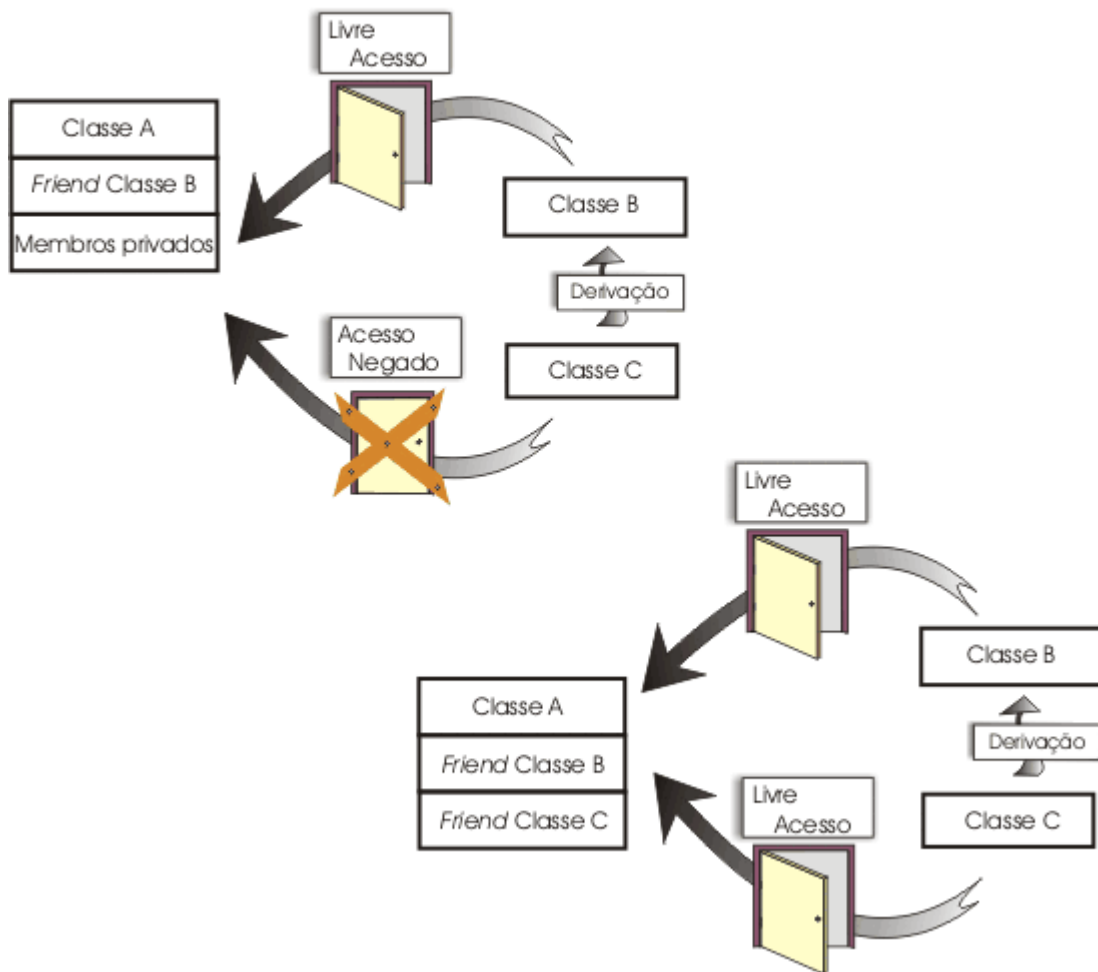
Outro fator importante é que a característica *friend* não é transmitida por herança. Uma classe derivada de uma classe *friend* não tem acesso aos dados privados da classe da qual a classe base é *friend*. Ou seja, se B é *friend* de A, B tem livre acesso aos dados privados de A. Porém sendo C derivada de B, C não é necessariamente *friend* de A. Apesar de C herdar os atributos e métodos de B, C não tem acesso aos dados privados de A. Nesse caso, para C ter acesso aos membros privados de A (o mesmo acesso dado a B), é necessário que A também considere C uma classe *friend*.(FIG. 4.1)

Na estrutura Todo-Parte, apesar da restrição de acesso, uma classe derivada da classe "Todo" herda normalmente os objetos da classe "Parte", continuando a ser constituída pela aglutinação de classes.

No PREMOLD optou-se por alterar o tipo de ligação entre as classes originais do Modelador 3D. Não foram utilizadas classes *friend*. Em seu lugar optou-se por adotar as estruturas Generalização/Especialização e Todo-Parte na geração de algumas classes como será descrito no item 4.4.

A seguir é apresentada uma breve descrição das classes básicas originais do Modelador 3D. Para um estudo mais aprofundado sugere-se a consulta às dissertações: "Modelador 3D para Estruturas via CAD - Arquitetura do Sistema / Banco de Dados / Visualização" de Fernando Poinho Malard (MALARD, 1998) e "Modelador 3D para Estruturas via CAD – Engenharia / Visualização / Banco de Dados" de Alexandra Hunter (HÜNTER, 1998).

## Relacionamento entre Classes *Friend* e suas derivadas



**FIGURA 4.1** - Relacionamento entre classes *friend* e suas classes derivadas

Para as classes **Node**, **Bar** e **Face** as funções obrigatórias para derivação da classe *AcDbEntity*, descritas no item 3.3.4, foram definidas através de Polimorfismo. Além daquelas funções foi derivada a função *getOsnapPoints*, que retorna os pontos geométricos principais da entidade, como os pontos inicial, final e ponto central de uma reta. Também foram definidas funções para definição dos pontos, que serão retornados por aquela função e a função *list*, que retorna informações sobre a entidade. A definição da função *list* é fundamental para um melhor gerenciamento do modelo criado, ao permitir que sejam informados ao usuário dados não gráficos do elemento como, por exemplo, as restrições nodais ou o tipo de barra utilizado. Para as classes **Bar** e **Face** foram definidas através de Polimorfismo funções que retornam e movem os pontos a partir dos quais é possível

estender a entidade (*getStrechPoints* e *moveStrechPointsAt*) e a função *deepClone* que determina o comportamento de uma entidade quando essa é copiada. Através dessa função é possível determinar quais os atributos serão copiados ou como eles serão copiados. Para a classe **Bar**, a função *explode*, que gerencia o comportamento da entidade quando essa é dividida nas entidades básicas que a compõem, também foi definida através de Polimorfismo. Também foram definidas as funções da classe *AcDbCurve* que permitem gerenciar o comando OFFSET do AutoCAD para que o mesmo seja corretamente implementado para a entidade **Bar**. Foram elas: *isPlanar*, *getStartParam*, *getPointAtParam*, *getFirstDeriv*, *getClosesPointTo*, *getOffsetCurves*.

Além dessas funções definidas através de polimorfismo, foram criadas funções de acesso e de definição dos seus atributos privados. As funções de acesso do Modelador 3D retornam o valor do atributo e são sempre identificadas pela palavra *get* junto ao nome do atributo. Suas funções de definição aplicam sobre o valor do atributo um valor repassado para as mesmas e são identificadas pelas palavras *set* ou *append* junto com o nome do atributo. As funções de uso mais específico de cada classe serão apresentadas a seguir juntamente com seus atributos.

As classes **Reactornode**, **Reactorbar** e **Reactorface** citadas a seguir são classes de controle criadas pelo aplicativo e serão descritas no item 4.2.2.

### a) Classe Node

Derivada de *AcDbEntity* possui como classes *friend* as classes **Bar** e **Reactornode**.

Seus atributos privados recebem os seguintes nomes, definições e funções de acesso e de atribuição:

**TABELA 4.1 - Atributos e métodos da classe Node**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
lps	AcGePoint3D	Ponto de coordenadas tridimensionais que armazena a posição do nó	setSp getSp
lpsText	AcGePoint3D	Ponto de coordenadas tridimensionais que armazena a posição do texto do nó	setText
enormal	AcGeVectort3D	Vetor tridimensional normal ao plano de representação do texto do nó	setNormal
NodeId	Adesk::Int16	Inteiro que armazena o número do nó na estrutura unifilar fornecido pela classe <b>Config1</b>	setNodeId getNodeId
BarList	AcDbObjectIdArray	Lista de <i>AcDbObjectIds</i> que armazena o identificador (ID) de cada barra ligada ao nó	AppendBarList GetBar
CenterLoad	AcGeDoubleArray	Lista de números reais que armazena os carregamentos concentrados do nó	InsertCLoad getLoadVal
CenterLoadDir	AcDbIntArray	Lista de inteiros que indicam a direção de aplicação das cargas aplicadas no nó	InsertCLoad getLoadDir
CenterLoadCas	AcDbIntArray	Lista de inteiros que indicam o caso ao qual uma carga pertence	InsertCLoad getLoadCas
KindRelease	Adesk::Boolean	Variável booleana que determina se a representação gráfica das restrições nodais será ou não apresentada na tela do AutoCAD	setKindRelease
KindGeom	Adesk::Boolean	Variável booleana que determina se a representação gráfica da entidade nó será ou não apresentada na tela do AutoCAD	setKindGeom
KindCaso	Adesk::UInt8	Inteiro positivo através do qual é determinado quais os casos de carregamento serão representados na tela do AutoCAD	setKindCaso
KindLoad	Adesk::UInt8	Inteiro positivo através do qual são determinados quais os carregamentos serão representados na tela do AutoCAD	setKindLoad
nrelRot	Adesk::UInt8	Inteiro positivo que codifica a liberação ou restrição à rotação do nó nos três eixos principais	setRot
nrelTrans	Adesk::UInt8	Inteiro positivo que codifica a liberação ou restrição à translação do nó nos três eixos principais	setTrans

As listas **CenterLoad**, **CenterLoadDir** e **CenterLoadCas** são manipuladas em paralelo, ou seja, uma determinada carga terá seu valor armazenado na posição *i* da lista **CenterLoad**, sua direção será armazenada na mesma posição *i* da lista **CenterLoadDir** e seu caso de carregamento será armazenado na posição *i* da lista **CenterLoadCas**. Quando uma carga é removida os valores a ela correspondentes são removidos das três listas.

Seus métodos específicos são:

- **drawLoad**: desenha na tela do AutoCad a representação das cargas do nó;
- **AppendBarList**: adiciona o identificador de uma barra à lista **BarList** do nó;
- **RemoveBar**: remove a identificação de uma barra da lista **BarList** do nó;
- **ContainsBar**: verifica se a lista **BarList** do nó contém o identificador de uma barra;
- **EmptyListBar**: verifica se a lista **BarList** do nó está vazia;
- **GetBar**: retorna o identificador armazenado em determinada posição da **BarList**. Essa posição é repassada pela função;
- **GetBarListLen**: retorna o tamanho da lista de barras do nó;
- **setNodeId**: busca através da classe **Config1** a numeração para um novo nó.

São ainda definidas funções para inserção de uma carga no nó (**insertCLoad**) e para remover todas as cargas referentes a um determinado caso do nó (**removeCaso**). A primeira função faz a inserção dos devidos valores relativos a uma carga nas três listas descritas anteriormente e a segunda retira todos os valores relacionados a um determinado caso das três listas.

## b) Classe Bar

Derivada de *AcDbCurve* possui como classes *friend* as classes **Node** e **Reactorbar**.

Seus atributos privados recebem os seguintes nomes, definições e funções de acesso e de atribuição:

**TABELA 4.2 - Atributos e métodos da classe Bar**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
lps	AcGePoint3D	Ponto de coordenadas tridimensionais que indica o ponto inicial da barra.	setSp getSp
StartNode	AcDbObjectId	Armazena o Identificador do nó inicial da barra.	setStartNode getStartNode
lpe	AcGePoint3D	Ponto de coordenadas tridimensionais que indica o ponto final da barra.	setEp getEp
EndNode	AcDbObjectId	Armazena o Identificador do nó final da barra.	setEndNode getEndNode
lpsText	AcGePoint3D	Ponto de coordenadas tridimensionais que armazena a posição do texto da barra.	setText getText
enormal	AcGeVector3D	Vetor tridimensional normal ao plano de representação do texto da barra.	setNormal
BarId	Adesk::Int16	Inteiro que armazena o número da barra na estrutura unifilar.	setBarId getBarId
TpBar	Adesk::UInt8	Inteiro que indica o tipo da barra: principal, secundária ou contraventamento.	setTpBar getTpBar
LoadComList	AcGeDoubleArray	Lista de números reais que armazena os valores das cargas distribuídas totais.	appendLoadCom getLoadCom
ComLoadCas	AcDbIntArray	Lista de números inteiros que armazena os casos das cargas distribuídas totais.	appendLoadCom getLoadCasCom
LoadDistList	AcGeDoubleArray	Lista de números reais que armazena os valores das cargas distribuídas parciais.	appendLoadPart getLoadPart
LengthDistList	AcGeDoubleArray	Lista de números reais que armazena o comprimento de aplicação das cargas distribuídas parciais.	appendLengthPart getLengthPart
DistanceDistList	AcGeDoubleArray	Lista de números reais que armazena as distâncias das cargas distribuídas parciais ao nó inicial da barra.	appendDistancePart getDistancePart
PartLoadCas	AcDbIntArray	Lista de números inteiros que armazena os casos das cargas distribuídas parciais.	appendLoadPart getLoadCasPart
LoadCenList	AcGeDoubleArray	Lista de números reais que armazena os valores das cargas concentradas.	appendLoadCen getLoadCen
DistanceCenList	AcGeDoubleArray	Lista de números reais que armazena as distâncias das cargas concentradas ao nó inicial da barra.	appendDistanceCen getDistanceCen

Atributo	Tipo	Descrição	Funções de acesso e atribuição
CenLoadCas	AcDbIntArray	Lista de números inteiros que armazena os casos das cargas concentradas.	appendLoadCen getLoadCasCen
KindGeom	Adesk::Boolean	Variável booleana que determina se a representação gráfica da entidade barra será ou não apresentada na tela do AutoCAD	setKindGeom
KindCaso	Adesk::UInt16	Inteiro através do qual são determinados quais os casos de carregamento serão representados na tela do AutoCAD	setKindCaso
KindLoad	Adesk::UInt16	Inteiro através do qual são determinados quais os carregamentos serão representados na tela do AutoCAD	setKindLoad

Seus métodos específicos são:

- **textFit**: calcula, a partir do ponto médio da barra, a posição do seu texto;
- **drawLoad**: desenha na tela do AutoCAD a representação das cargas da barra;
- **verifyLp**: Posiciona corretamente uma carga concentrada na barra.
- **removeLoadCen**: exclui da lista de cargas concentradas uma determinada carga cuja posição na lista é passada como parâmetro da função;
- **removeLoadPart**: exclui da lista de cargas distribuídas parciais uma determinada carga cuja posição na lista é passada como parâmetro da função;
- **removeLoadCom**: exclui da lista de cargas distribuídas totais uma determinada carga cuja posição na lista é passada como parâmetro da função;
- **StretchLoad**: Calcula os trechos de carregamento distribuído constante ao percorrer a entidade **Bar**, identificando os trechos inicial e final de cada carregamento. Permite a representação de trechos de carregamento diferenciados com alturas zenitais distintas de acordo com a intensidade da carga resultante no trecho.
- **PointLoad**: Calcula o ponto de aplicação de cada carga concentrada e sua altura zenital de acordo com a intensidade da carga, detectando cargas aplicadas num mesmo ponto e somando seus valores.
- **removeCaso**: retira todos os valores das listas de cargas relacionados a um determinado caso;
- **getLengthBar**: retorna o comprimento da barra;

- **getLoadValLenCom**: retorna o comprimento da lista de valores de cargas distribuídas totais;
- **getLoadValLenCen**: retorna o comprimento da lista de valores de cargas concentradas;
- **getLoadValLenPart**: retorna o comprimento da lista de valores de cargas distribuídas parciais;
- **setBarId**: busca através da classe **Config1** a numeração para uma nova barra.

### c) Classe Face

Derivada de *AcDbEntity* possui como classe *friend* a classe **Reactorface**.

Seus atributos privados recebem os seguintes nomes, definições e funções de acesso e de atribuição:

**TABELA 4.3** - Atributos e métodos da classe **Face**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
lps	AcGePoint3D	Ponto de coordenadas tridimensionais que indica o centro geométrico da face.	setSp
lpsText	AcGePoint3D	Ponto de coordenadas tridimensionais que armazena a posição do texto da face.	setText
enormal	AcGeVectort3D	Vetor tridimensional normal ao plano de representação do texto da face.	setNormal
FaceId	Adesk::Int16	Inteiro que armazena o número da face na estrutura unifilar.	setFaceId getFaceId
vertxList	AcGePoint3DArray	Lista de pontos de coordenadas tridimensionais que indicam os vértices da face.	getVertex appendVertex
Repre	Adesk::int16	Inteiro que indica o tipo de representação da face: vão de laje, vão vazio ou vão de escada.	Não há funções de acesso e de atribuição
ang	double	Número real que armazena o ângulo de trabalho da face em relação ao eixo X local.	SetAng GetAng
pti	AcGePoint3D	Ponto de coordenadas tridimensionais que indica o ponto de origem do eixo X local.	Getpti

Atributo	Tipo	Descrição	Funções de acesso e atribuição
ptf	AcGePoint3D	Ponto de coordenadas tridimensionais que indica o ponto final da face.	Não há funções de acesso e de atribuição
ptaux	AcGePoint3D	Ponto de coordenadas tridimensionais auxiliar.	Não há funções de acesso e de atribuição
Control	Adesk::Int16	Número inteiro que codifica o tipo de geometria da face: quadrada, retangular, poligonal de quatro faces, poligonal de mais de quatro faces. <sup>1</sup>	Não há funções de acesso e de atribuição
Per	double	Número real que armazena o percentual utilizado para desenho da direção de trabalho da face.	Não há funções de acesso e de atribuição
ComLoadCas	AcDbIntArray	Lista de números inteiros que armazena os casos das cargas distribuídas na face.	SetLoad GetLoadCas
ComLoad	AcGeDoubleArray	Lista de números reais que armazena os valores das cargas distribuídas na face.	SetLoad GetLoad
KindGeom	Adesk::Boolean	Variável booleana que determina se a representação gráfica da entidade face será ou não apresentada na tela do AutoCAD	SetKindGeom
KindCaso	Adesk::Int16	Inteiro através do qual são determinados quais os casos de carregamento serão representados na tela do AutoCAD	SetKindCaso
KindLoad	Adesk::Int16	Inteiro através do qual são determinados quais os carregamentos serão representados na tela do AutoCAD	SetKindLoad

Seus métodos específicos são:

- **textFit**: calcula, a partir do centro geométrico da face, a posição do seu texto;
- **drawRepre1**, **drawRepre2**, **drawRepre3**: desenha na tela do AutoCad a representação das faces do tipo face plana, vão vazio ou vão de escada respectivamente;
- **drawLoad**: desenha na tela do AutoCAD a representação das cargas da face;

---

<sup>1</sup> Apesar de ser possível a criação de faces triangulares, não foi definido um código específico para as mesmas, sendo codificadas apenas as faces quadradas, retangulares e poligonais (de quatro ou mais vértices).

- **getVlen**: retorna o tamanho da lista de vértices da face;
- **getLoadValLen**: retorna o tamanho da lista de valores das cargas distribuídas totais da face;
- **removeLoadCom**: exclui da lista de cargas distribuídas totais uma determinada carga cujos valor e caso são passados como parâmetros da função;
- **removeCaso**: exclui da lista de cargas distribuídas totais todas as cargas relativas a um determinado caso de carregamento que é passado como parâmetro da função;
- **setSp**: armazena o centro geométrico da face;
- **setFaceId**: busca através da classe **Config1** a numeração para uma nova face.
- **nullArray**: retira todos os dados da lista de vértices da face;
- **setCG**: calcula o centro geométrico da face a partir de seus vértices;
- **getCG**: retorna o centro geométrico da face.

#### 4.2.2 Classes De Controle: **Config1**, **Casos**, **CurrCaso**, **ReactroNode**, **ReactorBar**, **ReactorFace**, **CommandReactor**.

Além das classes **Node**, **Bar** e **Face** foram criadas classes para o gerenciamento do aplicativo. Essas classes, por serem apenas de controle, não possuem representação gráfica, sendo, portanto derivadas da classe *AcDbObject*. Apenas a classe **CommandReactor**, que monitora o acionamento ou término de comandos do AutoCAD, é derivada de *AcEditorReactor*. A classe **Config1** é a responsável pelo gerenciamento da numeração dos elementos. A definição de carregamentos é feita pela classe **Casos**. A classe **CurrCaso** armazena os dados do caso corrente. Foi implementado um reator persistente para cada entidade personalizada do Modelador 3D. Esses reatores são responsáveis por efetuar ajustes no objeto **Config1** pelo acionamento do comando ERASE sobre cada uma dessas entidades. Assim, as classes **Reactornode**, **Reactorbar**, **Reactorface** são utilizadas para gerenciar a recuperação da numeração de elementos apagados do banco de dados armazenando-os no objeto da classe **Config1**. Também a reversão dessa operação é possível, caso o comando seja desfeito. As classes **Reactornode** e **Reactorface** gerenciam o comando de exclusão de nós e faces respectivamente. A classe **Reactorbar**, além de gerenciar a exclusão de barras, verifica se os nós da mesma podem ser excluídos pela ausência de outras barras ligadas a eles. Caso não existam outras barras ligadas a um determinado nó é enviada uma notificação de que esse será apagado e o objeto da classe

**Reactornode** procede a recuperação de seu número para o objeto da classe **Config1**. Cada entidade criada e inserida no banco de dados recebe como reator persistente um objeto da respectiva classe "*reactor*".

Para armazenar os objetos do aplicativo é criado um dicionário principal dentro do NOD chamado **DATA\_OBJECTS**. Esse dicionário contém os objetos **Config1**, **Casos** e **CurCaso** além de três outros dicionários: "*faces\_dict*", "*nodes\_dict*" e "*bars\_dict*". Esses dicionários armazenam os reatores de faces, nós e barras respectivamente. Cada reator é armazenado no banco de dados, no seu dicionário próprio, no momento em que é criado.

Para as classes **Config1**, **Casos** e **CurrCaso** foram definidas através de polimorfismo as funções *dwgInFields* e *dwgOutFields*, além de serem criadas as funções de acesso e atribuição aos membros privados das mesmas. Para as classes **Reactornode**, **Reactorbar** e **Reactorface** foi definida apenas a função *erased*. Já a classe **CommandReactor**, por ser uma classe de notificação de comandos do AutoCAD, teve a implementação das funções: *commandWillStart*, *commandEnded*, *commandCancelled* e *enddeepclone*.

### a) Numeração da Estrutura: Classe **Config1**

A classe **Config1**, responsável pela numeração das entidades criadas, é derivada da classe *AcDbObject*.

Nessa classe não foram criadas as funções de acesso e de atribuição. Seus atributos privados recebem os seguintes nomes, tipos e definições:

**TABELA 4.4 - Atributos da classe Config1**

Atributo	Tipo	Descrição
CurNodeN	Adesk::Int16	Inteiro que armazena o número corrente a ser utilizado para o próximo nó a ser criado.
curBarN	Adesk::Int16	Inteiro que armazena o número corrente a ser utilizado para a próxima barra a ser criada.
curFaceN	Adesk::Int16	Inteiro que armazena o número corrente a ser utilizado para a próxima face a ser criada.
avNodeN	AcDbIntArray	Lista de inteiros que armazena os números de elementos excluídos do banco de dados, disponíveis para serem utilizados nos próximos nós a serem criados.
avBarN	AcDbIntArray	Lista de inteiros que armazena os números de elementos excluídos do banco de dados, disponíveis para serem utilizados nas próximas barras a serem criadas.
avFaceN	AcDbIntArray	Lista de inteiros que armazena os números de elementos excluídos do banco de dados, disponíveis para serem utilizados nas próximas faces a serem criadas.

### b) Casos de Carregamento: Classe Casos

A classe **Casos** representa os casos de carregamento criados pelo usuário. Ela é derivada de *AcDbObject*.

Seus atributos privados recebem os seguintes nomes, definições e funções de acesso e de atribuição:

**TABELA 4.5 - Atributos e métodos da classe Casos**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
Num	Adesk::Int16	Inteiro que representa o número dado ao caso de carregamento.	setNum getNum
Desc	char*	Ponteiro para caractere que armazena a descrição do caso de carregamento.	setDesc getDesc
Enabled	Adesk::Boolean	Variável booleana que indica se o caso está habilitado ou desabilitado.	setEnabled getEnabled

### c) Carregamento Corrente: Classe **CurrCaso**

A classe **CurrCaso** representa o caso de carregamento corrente estabelecido pelo usuário. Ela é derivada de *AcDbObject*.

Seus atributos privados recebem os seguintes nomes, definições e funções de acesso e de atribuição:

**TABELA 4.6** - Atributos e métodos da classe **CurrCaso**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
Curr	Adesk::Int16	Inteiro que representa o número dado ao caso de carregamento corrente.	setCurr getCurr
CurrName	char*	Ponteiro para caractere que armazena o nome do caso de carregamento corrente.	setCurrName getCurName

### d) Reator Da Entidade Node: Classe **ReactorNode**

A classe **ReactorNode** é derivada de *AcDbObject* e tem como classes *friend* as classes **Bar** e **Node**.

A classe **ReactorNode** não possui atributos, apenas possui a função *erased* da classe *AcDbObject* definida através de Polimorfismo. Essa função retorna ao objeto **Config1** a numeração de um nó excluído do banco de dados. Essa numeração é incluída na posição adequada da lista **avNodeN** de números válidos para serem usados na numeração de novos nós. Caso a exclusão do nó seja desfeita o seu número é retirado da lista **avNodeN**.

### e) Reator Da Entidade Bar: Classe Reactorbar

A classe **Reactorbar** é derivada de *AcDbObject* e tem como classes *friend* as classes **Bar** e **Node**.

Assim como a classe **Reactornode**, a classe **Reactorbar** não possui atributos, apenas possui a função *erased* da classe *AcDbObject* definida através de Polimorfismo. Essa função além de retornar ao objeto **Config1** a numeração de uma barra excluída do banco de dados, incluída na posição adequada da lista **avBarN** de números válidos para serem usados na numeração de novas barras, verifica se os nós da barra podem ser excluídos. Caso os nós não possuam outras barras listadas em seu atributo **BarList** será enviada uma notificação para a classe **Reactornode** para que essa processe a recuperação do número do nó excluído. Caso a exclusão da barra seja desfeita o seu número é retirado da lista **avBarN**.

### f) Reator Da Entidade Face: Classe Reactorface

A classe **Reactorface** é derivada de *AcDbObject* e tem como classe *friend* a classe **Face**.

Também essa classe possui apenas a função *erased* da classe *AcDbObject*, definida através de Polimorfismo. Essa função funciona de modo análogo á função *erased* da classe **Reactornode**, tratando da exclusão de faces e utilizando a lista **avFaceN** do objeto **Config1** para armazenar o número das faces excluídas.

### g) Monitoramento De Comandos Do AutoCAD: Classe CommandReactor

A classe **CommandReactor** é derivada da classe *AcEditorReactor* e responde a eventos específicos do AutoCAD como quando um desenho é carregado ou descarregado, quando comandos do AutoCAD são iniciados ou terminados, etc. A classe *AcEditorReactor* é

derivada da classe *AcRxObject*, uma vez que esse tipo de reator não é um objeto do banco de dados. A classe base para a derivação de classes de reatores possui uma série de funções virtuais que podem ser definidas através de Polimorfismo. Como já mencionado, para a classe **CommandReactor** foram definidas as funções *commandWillStart*, *commandEnded*, *commandCancelled* e *enddeepclone*. Além da definição dessas funções foi criada uma variável de controle que armazena os identificadores de barras copiadas, a *clBarIds*.

As funções *commandWillStart* e *commandCancelled* gerenciam o travamento ou a liberação de determinadas camadas (*layers*) durante a execução ou término de determinados comandos. Isso é feito, por exemplo, para evitar a cópia dos nós quando uma barra é copiada. A função *commandWillStart* também torna nula ou vazia a lista que armazena os identificadores das barras que serão copiadas. Já as funções *commandEnded* e *enddeepclone* gerenciam a criação de novos nós para as barras geradas pelos comandos COPY, MIRROR e ARRAY, ajustam e renumeram as mesmas. A função *commandEnded* executa verificação e ajustes para as entidades BAR e NODE. Primeiro é verificado se as novas barras foram criadas sobre barras existentes através da função **verifybar**, criando novos nós caso seja necessário através da função **verifynode**. Caso as barras tenham sido copiadas sobre barras já existentes, as barras copiadas são excluídas do banco de dados e sua numeração retorna para o objeto **Config1**. A função *enddeepclone* armazena os identificadores das barras selecionadas para cópia na variável **clBarIds**. Após a notificação de fim de comando de cópia, a notificação de *commandEnded* é chamada.

### 4.2.3 Funções Globais

Além das classes criadas, foram definidas algumas funções globais que permitem o gerenciamento de ações ou que efetuam a criação de camadas, de reatores, a ligação desses com as entidades referidas, verificações de consistência, etc. Essas funções estão distribuídas em alguns arquivos em todos os projetos criados. A seguir são listadas aquelas mais importantes e específicas do aplicativo Modelador 3D.

- **verifyNode**: Recebe como parâmetro um ponto de coordenadas tridimensionais que representa os ponto de inserção do nó. Percorre o banco de dados abrindo as entidades e verificando se essas são do tipo **Node**, caso afirmativo, verifica se existe algum nó com ponto coincidente com seu parâmetro. Caso exista esse nó, é retornado o identificador do mesmo.
- **verifyBar**: Recebe como parâmetros dois pontos de coordenadas tridimensionais que representam os pontos inicial e final da barra a ser criada. Percorre o banco de dados abrindo as entidades e verificando se essas são do tipo **Bar**, caso afirmativo, verifica se existe alguma barra com pontos coincidentes com seus parâmetros. Caso haja alguma barra com pontos coincidentes o identificador dessa barra é capturado e disponibilizado pela função.
- **verifyFace**: Recebe como parâmetro uma lista de pontos de coordenadas tridimensionais que representam os vértices da face a ser criada. Percorre o banco de dados abrindo as entidades e verificando se essas são do tipo **Face**, caso sejam verifica se existe alguma face com vértices coincidentes com seu parâmetro. Caso haja alguma face com vértices coincidentes seu identificador é capturado e tornado disponível pela função.
- **AppendNodebar**: Função que trata casos especiais de criação de barras a partir dos seguintes dados: partindo de um nó até um ponto final dado, partindo de um ponto qualquer até um nó existente, e barras ligando dois nós existentes. Essa função verifica e atualiza os dados sobre a conectividade entre nós e barras. É feita a atualização da lista de barras dos nós criados com inserção da barra nessa lista. A verificação de consistência entre barras e seus nós final e inicial é feita comparando-se o ponto inicial e final das barras com o ponto de inserção de seus nós, caso esses não coincidam os pontos da barra são atualizados. É realizada, ainda, a ordenação correta de nó inicial e final para a barra criada.
- **createBar**: Função chamada pelo comando **BAR** para criar as entidades do tipo **Bar**.
- **changetypeBAR**: Permite a alteração do tipo de barra: principal, secundária ou de contraventamento.
- **createFaces** e **createoneFace**: Chamadas pelos comando **FACE** e **FACES**. A função **createoneface** verifica a existência de uma fronteira fechada, delimitada por entidades do tipo **Bar**, a partir de um ponto dado. Caso essa fronteira exista, efetua a criação de uma entidade **Face** englobando toda a região fechada. A função **createFaces** percorre

todo o banco de dados identificando os pontos inicial das barras. A partir desses pontos, é dado um pequeno implemento em suas coordenadas e é feita a busca por uma fronteira fechada próxima a eles. Caso essa fronteira exista, é chamada a função **createoneFace** para a criação da entidade **Face** naquele ponto.

- **changeFace**: Essa função permite alterar o tipo da face criada. As faces ao serem criadas são definidas como planas. Com o uso dessa função pode-se alterar seu tipo para vazio ou vazio de escada.
- **changeWork**: Chamada pelo comando **CHWORK**, essa função permite ao usuário alterar o ângulo de trabalho de uma face. Esse ângulo é definido previamente pelo aplicativo conforme a geometria da face criada e representa o ângulo entre o eixo X global e a direção de trabalho da face.
- **verifyConsist**: Efetua a verificação de consistência de nós e barras de forma conjunta. Essa função é chamada para ajustar a posição de barras em função das transformações aplicadas aos nós.
- **Renum\_Nodes** e **Renum\_Bars** e **Renum\_Faces**: Efetuam a renumeração de nós, barras e faces respectivamente, através da ordenação dos números de cada entidade, baseado em um critério de organização que depende do tipo de entidade a ser renumerada.
- **createCasos**: Essa função é chamada quando o aplicativo é carregado e quando um novo desenho é criado. Ela acessa o dicionário NOD do AutoCAD para a criação de um novo dicionário chamado **CASOS\_OBJ**. Dentro desse dicionário a função cria o objeto **Casos** que recebe um número de identificação, uma descrição, uma sigla e uma definição para seu estado (habilitado ou desabilitado).
- **redrawEnts**: atualiza o estado de apresentação das entidades na tela gráfica do AutoCAD conforme os valores de seus atributos de definição de estado como: **kindLoad**, **KindGeom**, etc.
- **exportData**, **exportNodesAll**, **exportBarsAll**, **exportFacesAll**: funções que armazenam os números de identificação dos elementos e geram arquivos de exportação com seus dados.
- **drawload**: desenha a representação dos carregamentos tendo como parâmetros: um ponteiro para *AcGiWorldDraw*; o ponto de inserção da carga; a altura da carga e o valor da carga.

- **createCenLoad**: Cria uma carga concentrada na entidade **Bar**. Verifica se o valor da carga é nulo ou se o ponto de aplicação é maior que o comprimento da barra.
- **setLower**: Garante que o nó inicial de uma barra seja sempre o nó de menor numeração de identificação. Verifica se o nó estabelecido como nó inicial para a barra coincide com o nó de menor numeração. Caso os nós não coincidam, eles são trocados para manter sempre o como nó inicial das barras o nó de menor numeração, facilitando a apreensão da estrutura pelo usuário;
- **pointLoad**: Calcula o ponto de aplicação da carga e sua altura de representação conforme seu valor. Soma os valores das cargas aplicadas em um mesmo ponto.
- **creatDisLoad**: Insere cargas distribuídas parciais ou totais na barra, tendo o ponto inicial de aplicação da carga e seu comprimento de aplicação. Efetua as seguintes verificações: se o valor da carga é nulo; se a soma da distância mais o comprimento da carga é maior que o comprimento da barra; se o comprimento da carga é nulo.
- **CIBar**: Insere e edita cargas nas barras.
- **CIFace**: Insere e edita cargas nas faces.
- **createCurrCaso**: cria um objeto da classe **CurrCaso**.
- **setCurrCaso**: Transforma um caso dados em caso corrente.
- **getCurrCaso**: retorna a sigla do caso corrente.
- **setCurrCasoNum**: retorna o número do caso corrente.
- **getCaso**: retorna as informações de um caso de carregamento através de sua sigla.
- **removeEntityCasos**: remove de todas as entidades as cargas relacionadas a um determinado caso.
- **delCaso**: exclui um determinado caso de carregamento a partir de sua sigla.
- **modifyCaso**: modifica as informações de um caso de carregamento.
- **getAvNumber**: retorna o número válido para ser atribuído a um novo caso de carregamento.
- **searchAvNumber**: verifica se um determinado número já está sendo utilizado por algum caso de carregamento.
- **isEnabled**: verifica se um caso de carregamento está habilitado ou não.
- **getCasoName**: retorna o nome de um caso a partir de seu número.
- **getNumberOfCasos**: retorna o número de objetos do tipo **Casos** existentes.

- **createPerBar**: cria um objeto **ReactorBar** e declara o mesmo como reator persistente para uma entidade **Bar** criada.
- **createPersistent**: cria um objeto **ReactorFace** e declara o mesmo como reator persistente para uma entidade **Face** criada.
- **createPerNode**: cria um objeto **ReactorNode** e declara o mesmo como reator persistente para uma entidade **Node** criada.

#### 4.2.4 Comandos do Aplicativo

##### a) Comandos Globais

**DISPLAY**: Efetua o controle de visualização das entidades e dos carregamentos. Esse controle é feito através da manipulação das variáveis de controle de cada entidade: **kindGeom**, **kindCaso**, **kindLoad** e **kindRelease**, estando a última presente apenas na entidade **Node**. Esse comando atua de forma global, permitindo a alteração da visualização de todas as entidades de uma só vez. Atualiza as variáveis de controle de visualização das entidades através das funções globais **UpdateNode**, **UpdateBars** e **UpdateFaces**.

**Renum**: Efetua a renumeração automática das entidades conforme critérios pré-estabelecidos pelo usuário.

**Casos**: Cria e gerencia os casos de carregamento, estabelecendo qual o caso corrente, quais os casos ativados e quais os casos desativados.

**EXDATA**: Cria um arquivo de extensão CAD para exportação dos dados do desenho criado. Os dados das entidades são colhidos e formatados no arquivo escolhido. É permitida a exportação dos dados de todas as entidades do arquivo de desenho através da função *buildArray*, ou de elementos específicos selecionados na tela de desenho. Utiliza a função global *exportData*.

### **b) Comandos Relativos a Entidade Node:**

- **NREL:** Permite ao usuário alterar as restrições ou liberações dos nós.
- **CLNODE:** Insere cargas concentradas em um conjunto de nós ou em apenas um nó.

### **c) Comandos Relativos a Entidade Bar:**

- **LOADS:** aplica, edita e remove cargas das barras. Aciona a função **CLBar** que testa se a entidade é uma barra e então abre o quadro de diálogo para a edição das cargas.
- **DISLOAD:** Insere cargas distribuídas parciais ou completas nas entidades. Aciona a função **createDisLoad**.
- **CENLOAD:** Aplica cargas concentradas nas entidades selecionadas. Aciona a função **createCenLoad**. Para a inserção das cargas o usuário deve informar o valor da carga e a distância desta até o nó inicial da barra.
- **CHBAR:** altera os parâmetros das barras.
- **BAR:** comando para criação de barras de forma seqüencial. Aciona a função **createBar**. Para as barras devem ser dados: um ponto inicial e um ponto final. Caso sejam fornecidos outros pontos após o segundo ponto, é criada uma nova barra com ponto inicial coincidente com o ponto final da barra anterior e o próximo ponto dado será seu ponto final.

### **d) Comandos Relativos a Entidade Face**

- **CLFACE:** Aplica cargas distribuídas em uma face ou em um conjunto de faces.
- **CHWORK:** Permite alterar o ângulo de trabalho de uma face.
- **CHFACE:** Permite alterar o tipo de vão de laje, vão vazio ou vão de escada.
- **FACE:** Permite a criação de uma face em uma região horizontal escolhida pelo usuário.
- **FACES:** Cria faces automaticamente em todas as regiões horizontais fechadas delimitadas por três ou mais barras.

### 4.3 Modularidade Original x Nova Estruturação do Código

Como citado no item 3.2, uma das características de um sistema orientado a objetos é a Modularidade: capacidade de separação de um conjunto de módulos, cada um dos quais contendo classes com independência de funcionamento. Os módulos independentes podem ser compilados separadamente.

O desenvolvimento de aplicativos com o uso do Visual C++ é feito através de *Workspaces*, ou ambientes de trabalho. Cada aplicativo é criado dentro de um ambiente de trabalho próprio. Dentro de cada ambiente de trabalho podem ser criados vários projetos. Cada projeto é composto por um conjunto de arquivos de declaração e de definição de classes e de funções globais, arquivos de biblioteca, arquivos compilados, etc. Cada projeto dá origem a uma biblioteca de vínculo dinâmico, uma DLL. Assim, para cada arquivo ARX gerado existe um projeto correspondente.

No desenvolvimento do Modelador 3D houve a intenção de fazer-se uso da Modularidade para possibilitar o trabalho em equipe. O Modelador 3D foi dividido em vários módulos, cada um contendo mais de um projeto. Foram criados 12 projetos que reúnem funções semelhantes que incidem sobre cada tipo de elemento gerado. Um módulo central foi gerado para gerenciar os demais projetos. Porém, o conceito de Modularidade não foi alcançado em sua totalidade. A incidência das diversas funções agrupadas em um módulo sobre classes personalizadas definidas em outros módulos exigiu que cada um desses módulos tivesse acesso aos arquivos de declaração e de definição de tais classes. Assim, os módulos originais não são completamente independentes. Para possibilitar esse acesso às classes personalizadas, cada módulo passou a possuir uma cópia dos arquivos dessas classes. Esses arquivos são repetidos em praticamente todos os projetos. A repetição de arquivos tornou o código bastante extenso. A separação em vários projetos também dificulta o entendimento total do código, pois funções e classes estão distribuídas em vários arquivos. Outro fator que dificulta a apreensão do código é a distribuição de funções globais em vários arquivos, não só em arquivos próprios para essas funções, mas também em arquivos de definição de classes. Essa alocação de funções globais em arquivos de classes induz um observador menos atento a confundir funções globais com métodos das classes. Em alguns casos não é possível entender se pretendia-se criar realmente uma

função global, pois apesar de sua definição e declaração estar fora da classe, as funções aproximam-se mais de métodos de uma determinada classe do que de funções globais ou utilitárias. Esse é o caso, por exemplo, das funções **createCurrCaso**, **setCurrCaso**, **getCurrCaso** e **setCurrCasoNum** declaradas e definidas nos arquivos da classe **CurrCaso**, mas fora da definição e declaração dessa classe.

Alguns arquivos próprios para o armazenamento das funções globais são os arquivos **Utils.cpp**, **Utils.h**, **UtilsDBX.cpp**, **UtilsDBX.h**, **DictUtils.cpp** e **DictUtils.h**. Segundo MALARD (2002), hoje, após um maior conhecimento sobre o desenvolvimento de aplicativos através do uso do **ObjetARX**, o tipo de estruturação utilizado no desenvolvimento do Modelador 3D mostrou-se bastante confuso, não sendo aconselhável sua utilização para o desenvolvimento de novos aplicativos.

A seguir são listados os projetos criados no Modelador 3D e sua descrição:

- **main.arx**: projeto central que gerencia o carregamento e o descarregamento sequencial dos demais projetos na memória do computador;
- **basics.arx**: projeto que controla a numeração das entidades; implementa os reatores de comandos; controla os ajustes e modificações necessárias após a cópia e exclusão de entidades;
- **casos.arx**: projeto que implementa o comando **CASOS**; cria e gerencia os objetos que controlam os casos de carregamento; controla a exclusão dos casos de carregamento;
- **clface.arx**: projeto que implementa o comando **CLFACE** que gerencia a inclusão de cargas distribuídas na entidade **Face**;
- **clnode.arx**: projeto que implementa o comando **CLNODE** que gerencia a inclusão de cargas concentradas na entidade **Node**;
- **display.arx**: projeto que implementa o comando **DISPLAY** para o controle de visualização das três entidades do modelador;
- **export.arx**: projeto que implementa o comando **EXDATA** para exportar os dados das entidades do Modelador;
- **loadbar.arx**: projeto que implementa os comandos **CENLOAD** e **DISLOAD** que gerenciam a inserção de cargas concentradas e distribuídas nas entidades **Bar**.
- **loads.arx**: projeto que implementa o comando **LOADS** que gerenciam a visualização e edição de cargas concentradas e distribuídas nas entidades **Bar**.

- **nbface.arx**: projeto de criação e implementação das entidades **Node**, **Bar** e **Face**, implementação dos reatores de entidades; implementa os comandos **BAR**, **FACE**, **FACES**, **CHBAR**, **CHFACE**, e **CHWORK**;
- **nrelease.arx**: projeto que implementa o comando **NREL** para definição das restrições ou liberações da entidade **Node** quanto a rotações e translações;
- **renum.arx**: projeto que implementa o comando **RENUM** para renumeração das entidades; e
- **toolbar.arx**: projeto que cria e implementa as barras de ferramentas para o acionamento dos comandos do Modelador 3D.

Vê-se então que, no Modelador 3D original existem diversos projetos ARX onde arquivos de definição de classes personalizadas e arquivos de gerenciamento do aplicativo são agrupados.

Como foi mencionado no item 3.3.1, a partir da versão 2000 do ObjectARX foi criada uma nova estruturação para os aplicativos que utilizam essa biblioteca. Os arquivos passaram a serem distribuídos em dois projetos básicos. O projeto DBX engloba os arquivos de declaração e de definição das classes personalizadas. O projeto ARX engloba os arquivos de gerenciamento do aplicativo, de declaração e de definição e dos comandos próprios do aplicativo e os arquivos de declaração e de definição das classes derivadas do MFC para geração dos quadros de diálogo. O projeto DBX deve ser criado como dependente do projeto ARX. O projeto DBX pode ser compilado separadamente, porém o projeto ARX, quando compilado executa primeiro a compilação do projeto DBX a ele vinculado.

Na reestruturação do aplicativo foram suprimidos todos os módulos e projetos originais e foram criados novos projetos ARX e DBX, estruturados conforme o exposto anteriormente. A eliminação dos diversos módulos teve como objetivo reduzir o código e melhor organizar seus arquivos. Todas as classes originais do Modelador 3D foram recriadas e inseridas no projeto DBX. Na sua maioria, as classes foram reestruturadas, como será abordado no próximo item.

Os comandos do Modelador 3D não foram implementados, pois o modelamento desenvolvido no PREMOLD altera o relacionamento entre entidades e ainda utiliza uma

nova maneira de criar essas entidades. A criação de barras e faces, por exemplo, passa a ser realizada através dos comandos de criação dos elementos **Pilar**, **Viga** e **Painel** do PREMOLD, descritos no capítulo 5. Os comandos para renumeração, para a inserção de carregamentos nas entidades e para a criação de casos de carregamentos não foram implementados nesta etapa de desenvolvimento do PREMOLD.

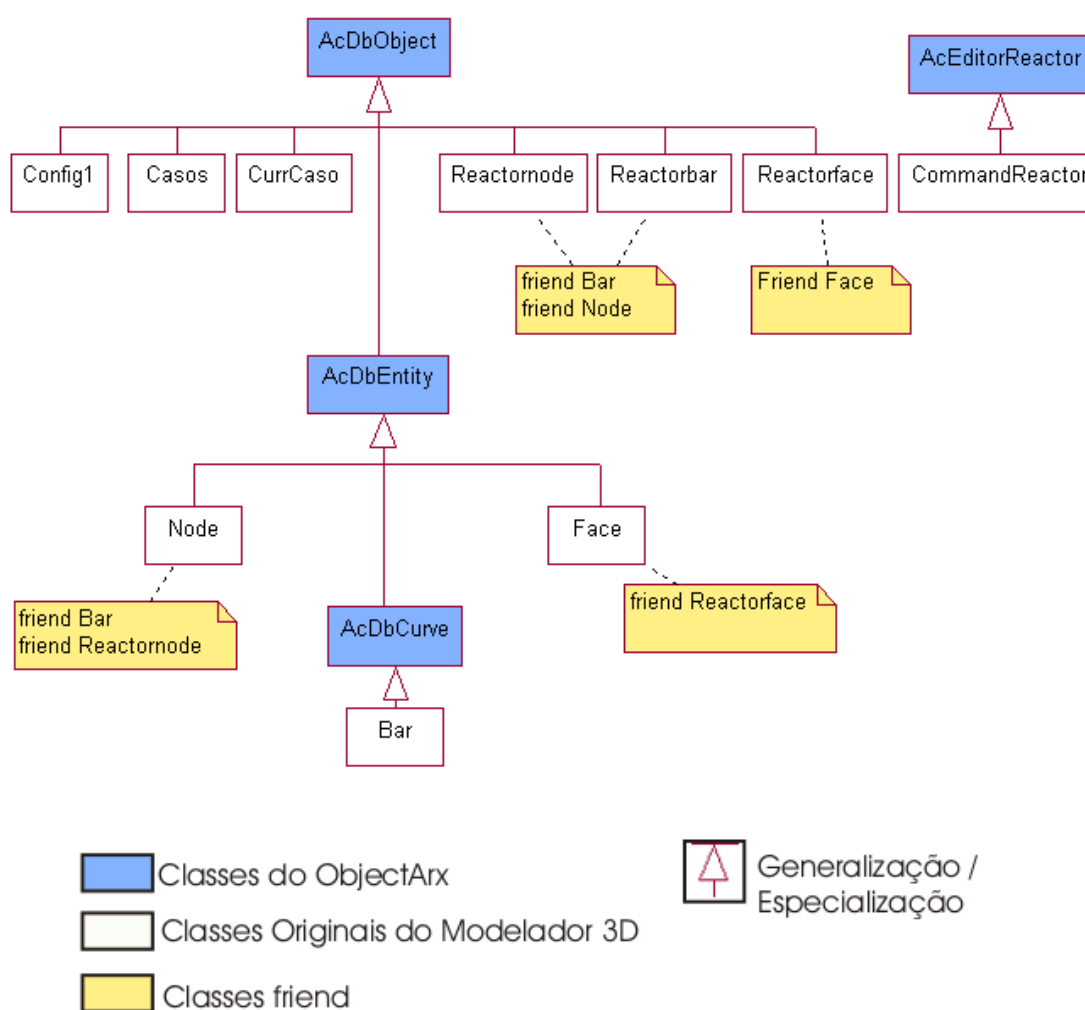
As funções globais, originariamente inseridas em arquivos de classes personalizadas, foram transferidas para arquivos utilitários que receberam os mesmos nomes dos arquivos utilitários do Modelador 3D original. Todas as funções globais e específicas do Modelador 3D foram remodeladas para adequarem-se à nova estrutura do código e à nova estruturação de herança de classes definida.

#### 4.4 Estrutura de Classes Original e Alterações Iniciais

A estrutura de classes original do Modelador 3D utiliza a estrutura Generalização/Especialização na criação das classes personalizadas. Essas são derivadas de determinadas classes base do ObjectARX. Não foi utilizada a estrutura Todo-Parte na construção das classes. Também não foi realizada a derivação de novas classes utilizando-se classes personalizadas como classes base. Como alternativa à estrutura Todo-Parte, várias classes foram declaradas *friend*, como exposto no item 4.2.1. A estrutura original de classes do Modelador 3D é mostrada na FIG. 4.2.

Para melhor utilizar os paradigmas da Programação Orientada a Objetos, optou-se por alterar a estrutura original das classes do Modelador 3D. Na reestruturação foi abandonado o uso de classes *friend* e em seu lugar adotou-se, na construção das classes, as estruturas Generalização/Especialização e Todo-Parte, quando necessárias. Posteriormente, durante o desenvolvimento do PREMOLD, esse tipo de tratamento mostrou-se mais adequado. No PREMOLD as duas estruturas foram utilizadas para gerar as classes necessárias à sua implementação.

Todos os atributos das classes passaram a ser privados, o que acarretou a necessidade de criação de funções de acesso e de atribuição adequadas para cada atributo. Alguns tipos de atributos foram alterados. Isso ocorreu principalmente para os atributos que armazenam conjuntos de caracteres. O Modelador 3D declara esses atributos geralmente como ponteiros para variáveis do tipo *char*. Na reestruturação realizada optou-se por trabalhar, sempre que possível, com a variável do tipo *CString* do C++ que já possui várias funções de acesso e de configuração predefinidas.



**FIGURA 4.2** - Estrutura original de classes do Modelador 3D

Algumas alterações foram feitas quanto à forma de retorno de dados dos métodos das classes e das funções globais. Sempre que possível foi utilizado o mesmo padrão de declaração de funções adotado pelo ObjectARX. Esse utiliza o retorno da função para informar se a mesma foi desempenhada com sucesso ou se houve alguma falha no

processo. Assim, as funções retornam sempre uma mensagem de erro que pode ser tratada pela entidade que a acionou. No Modelador 3D o retorno de dados tratados é feito através do retorno das funções. Na reestruturação realizada, o retorno dos valores passa a ser feito, sempre que possível, através dos parâmetros dessas funções.

A não utilização de classes *friend* acarretou, também, um grande esforço na alteração de todos os métodos das classes personalizadas e das funções globais, devido à restrição de acesso aos membros ou aos atributos privados das classes.

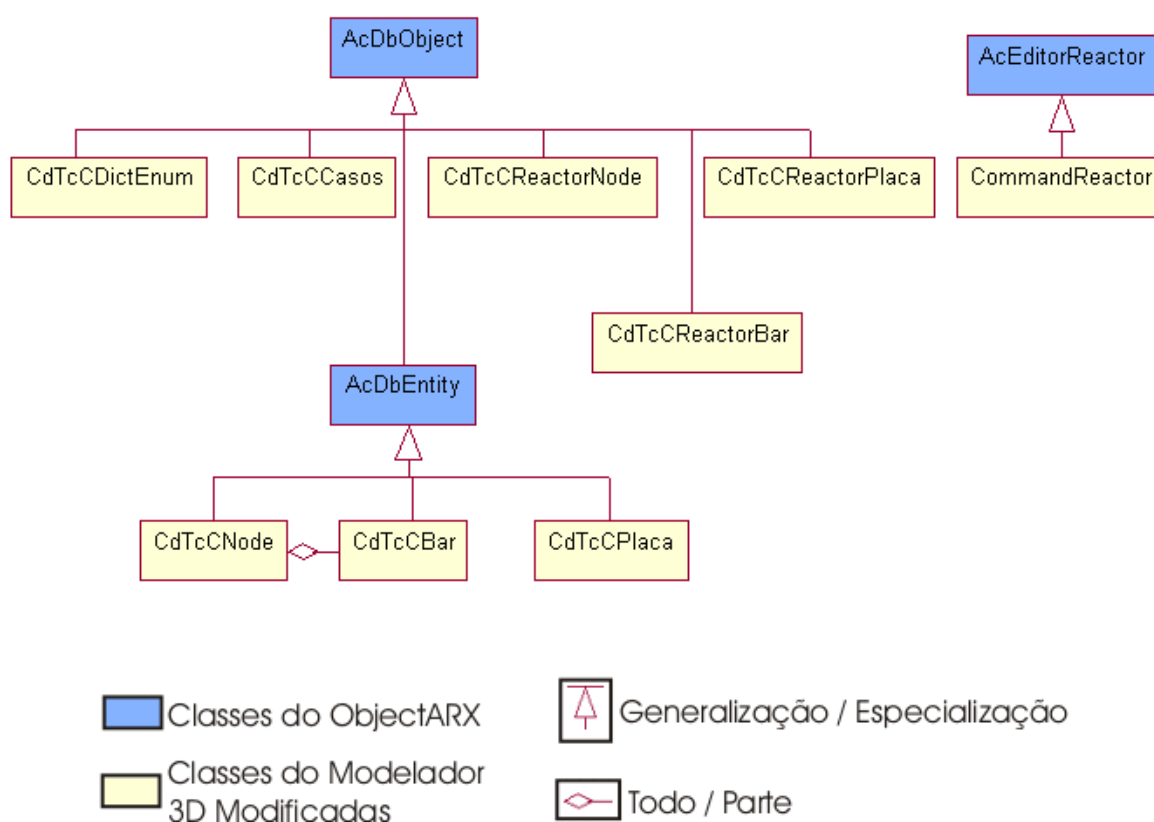
O nome dos atributos, os nomes de suas funções de atribuição e de acesso e os nomes das classes também foram alterados.

A partir da versão 2000 do ObjectARX é possível utilizar um assistente para a criação de classes (*wizard* do Object ARX). Esse assistente nomeia as classes com quatro letras iniciais que correspondem à sigla do desenvolvedor da ADN antes do nome dado à mesma. No caso do CADTEC a sigla adotada foi CdTc. Assim, todas as classes personalizadas criadas no CADTEC passaram a receber, antes de seu nome, as letras CdTc. No PREMOLD, após a sigla CdTc, adotou-se a letra "C" para indicar classe. Os objetos da classe são referidos pelo nome dado a essa sem a sigla "CdTcC". No caso da classe **Face** seu nome foi alterado para **CdTcCPlaca** pois o uso da palavra face passa a restringir-se às faces de um objeto sólido tratado pelo PREMOLD. A classe **Config1**, por tratar-se de uma classe que gerencia a numeração das entidades, e devido ao fato de seus objetos serem armazenados em um dicionário, passou a ser chamada de **CdTcCDictEnum**. As classes reestruturadas receberam então os nomes **CdTcCBar**, **CdTcCNode**, **CdTcCPlaca**, **CdTcCDictEnum**, **CdTcCCasos**, **CdTcCReactorNode**, **CdTcCReactorBar**, **CdTcCReactorPlaca**, **CdTcCCommandReactor**, etc.

Com o uso do assistente do ObjectARX, o atributo é criado automaticamente, sempre com os caracteres "m\_" antes do nome do mesmo. As funções de acesso e de atribuição também são criadas automaticamente. As funções de acesso recebem o nome do atributo sem os caracteres "m\_", enquanto as funções de atribuição recebem o nome do atributo precedido da palavra "set". Esse tipo de nomenclatura foi adotado na reestruturação do aplicativo.

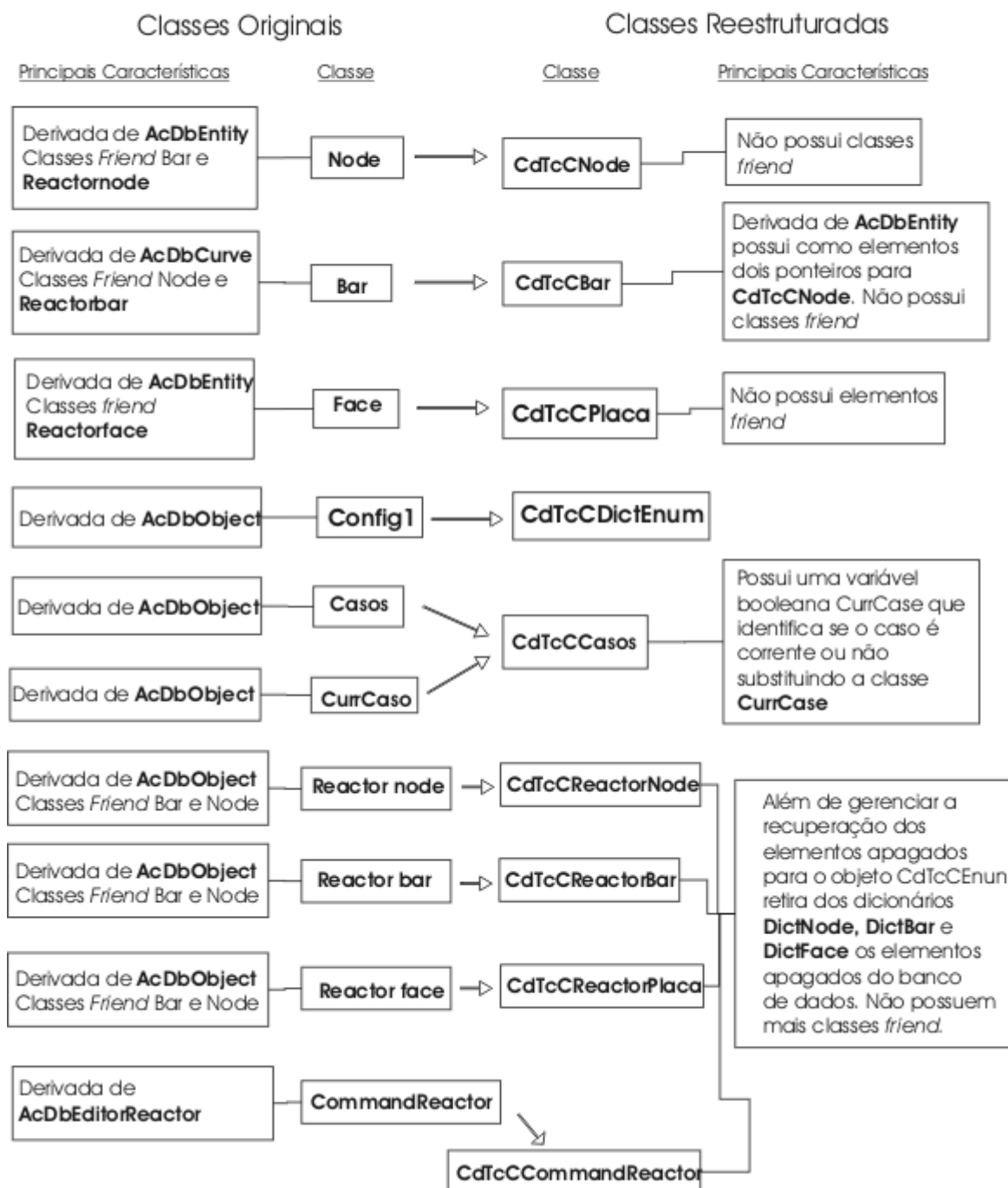
A única classe original do Modelador 3D suprimida foi a classe **CurrCaso** que armazena o caso corrente. Esse passa a ser controlado pela própria classe **CdTcCCasos** através de uma variável booleana. Essa variável recebeu o nome de **m\_CurCaso** e identifica se o objeto **Caso** é o caso corrente ou não. Se esse atributo for configurado como verdadeiro o objeto representa o caso corrente.

A seguir é apresentada na FIG. 4.3 a nova estrutura de classes adotada e na FIG. 4.4 as alterações básicas realizadas sobre as classes originais do Modelador 3D.



**FIGURA 4.3** - Nova estrutura de classes do Modelador 3D

## Diagrama de Reestruturação de Classes De Modelador 3D



**FIGURA 4.4** - Alterações realizadas sobre as classes originais do Modelador 3D

## 4.5 Nova Estrutura de Classes

Apresenta-se agora uma descrição das alterações básicas realizadas sobre cada uma das classes originais do Modelador 3D. Como mencionado no item 4.4, praticamente todos os atributos sofreram alteração no seu nome e no nome de suas funções de acesso e de atribuição. Ambas as funções passam a retornar uma mensagem de erro e a transferência de dados passa a ser feita pelos seus parâmetros. Como, além dessas alterações, não foi alterada a funcionalidade básica da maioria dos métodos das classes, serão comentados apenas aqueles métodos criados ou aqueles que sofreram alguma alteração de funcionalidade.

### 4.5.1 Classe CdTcCNode

A **CdTcCNode** (reestruturação da classe **Node**) é derivada de *AcDbEntity* possui os seguintes atributos e métodos:

**TABELA 4.7 - Atributos e métodos da classe CdTcCNode**

Atributo original	Novo atributo	Alteração sofrida	Funções de acesso e atribuição
lps	m_lps	Troca de nomes do atributo e das suas funções.	setLps lps
lpsText	m_lpsText	Foi criado um método de acesso ao conteúdo desse atributo: a função LpsText.	setText LpsText
enormal	m_enormal	Foi criado um método de acesso ao conteúdo desse atributo: a função enormal.	setEnormal enormal
NodeId	m_nodeId	Foi criada uma nova função setNodeId, além da função inicialmente existente. Essa função recebe como parâmetro uma variável booleana e é utilizada para a numeração de entidades copiadas.	setNodeId nodeId
BarList	m_BarList	Este atributo passa a ser um <i>AcDbIntArray</i> e esta lista passa a armazenar o número de identificação das barras ligadas ao nó e não uma variável do tipo <i>AcDbObjectID</i> .	AppendBarList GetBar

Atributo original	Novo atributo	Alteração sofrida	Funções de acesso e atribuição
CenterLoad	m_CenterLoad	Troca de nome do atributo.	InsertCLoad getLoadVal
CenterLoadDir	m_CenterLoadDir	Troca de nome do atributo.	InsertCLoad getLoadDir
CenterLoadCas	m_CenterLoadCas	Troca de nome do atributo.	InsertCLoad getLoadCas
KindRelease	m_KindRelease	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>kindRelease</b> .	setKindRelease kindRelease.
KindGeom	m_KindGeom	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>kindGeom</b> .	setKindGeom kindGeom
KindCaso	m_KindCaso	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>kindCaso</b> .	setKindCaso kindCaso
KindLoad	m_KindLoad	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>kindLoad</b> .	setKindLoad kindLoad
nrelRot	m_nrelRot	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>nrelRot</b> .	setNrelRot nrelRot
nrelTrans	m_nrelTrans	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>nrelTrans</b> .	setNrelTrans nrelTrans

Como foi visto no item 3.3.3, as variáveis do tipo *AcDbObjectId* não são mantidas de uma edição do AutoCAD para outra, mas são transformados em variáveis do tipo *handle* quando o desenho do AutoCAD é gravado. Essas são novamente transformados em *AcDbObjectId* quando o desenho é reaberto. Não se encontra na literatura técnica, garantias de que a mesma codificação fornecida originariamente a uma entidade, ou a um objeto, seja mantida em outras edições. Nessas novas edições a codificação original poderia até mesmo já estar sendo usada por outro objeto. Para evitar possíveis erros nessa codificação que inviabilizassem a correta identificação de uma entidade em diferentes edições, optou-se por adotar o número único para nós, barras e placas criado pelo próprio Modelador 3D através da classe **Config1**, reestruturada como classe **CdTcCDictEnum**. Esse número, ao fazer parte dos atributos de uma entidade, é guardado juntamente com a entidade quando um arquivo é gravado. Além disso os atributos da classe **CdTcCDictEnum**, responsáveis pela numeração progressiva das entidades criadas, são também gravados no arquivo do desenho quando esse é gravado. Ao ser reaberto um

arquivo de desenho do AutoCAD, a numeração das entidades continua a ser realizada a partir do último número utilizado pela classe **CdTcCDictEnum** quando o arquivo foi fechado anteriormente.

Segundo essas considerações, optou-se por alterar o tipo da variável **m\_BarList** que passou a armazenar números inteiros ao invés de armazenar variáveis do tipo *AcDbObjectId*. Esse inteiro representa a numeração das barras fornecida pela classe **CdTcCDictEnum**. Com isso os métodos específicos **AppendBarList**, **RemoveBar**, **ContainsBar**, **EmptyListBar**, **GetBar**, **GetBarListLen**, também passam a trabalhar com uma lista de variáveis do tipo *Adesk::Int16*, que equivale a um número inteiro, ao invés de trabalhar com variáveis do tipo *AcDbObjectId*. Essas funções não sofreram alteração de funcionalidade, continuando a desempenhar o mesmo papel que desempenhavam na estrutura original.

Através de sobrecarga de função foi criada outra função **setNodeId** além da função original. Essa nova função recebe como parâmetro uma variável booleana e é utilizada para controlar a numeração das entidades **Node** durante a cópia de elementos que as possuam, por exemplo, durante a cópia de barras. A função original foi mantida para efetuar a numeração de entidade **Node** criadas durante o lançamento dos elementos estruturais.

#### 4.5.2 Classe **CdTcCBar**

Essa classe (reestruturação da classe **Bar**) é derivada de *AcDbEntity*. A derivação original da classe base *AcDbCurve* visa possibilitar a implementação do comando OFFSET do AutoCAD sobre entidades da classe **Bar**. O comando OFFSET não se aplica a entidades compostas por sólidos. No PREMOLD não se tem mais uma barra desprovida de volume. A entidade barra só é criada quando um elemento sólido é criado. Assim, não é mais possível a utilização do comando OFFSET sobre essa entidade. Por isso adotou-se, na reestruturação, a classe *AcDbEntity* como classe base para a classe **CdTcCBar**.

Além da derivação da classe *AcDbEntity*, essa classe passa a ser constituída através da estrutura Todo-Parte. Ela passa a possuir dois ponteiros para objetos da classe

**CdTcCNode** como atributos: **m\_startNode** e **m\_endNode**. Esses representam os nós inicial e final da barra, respectivamente. Como os nós já armazenam as coordenadas do ponto onde estão localizados, foram suprimidos os atributos **lps** e **lpe** da classe original.

Os atributos dessa classe passam a receber os seguintes nomes e funções de acesso e de atribuição:

**TABELA 4.8 - Atributos e métodos da classe **CdTcCBar****

Atributo original	Novo atributo	Alteração sofrida	Funções de acesso e atribuição
lps	---	Atributo suprimido.	Funções suprimidas
StartNode	m_startNode	Ponteiro para um objeto da classe <b>CdTcCNode</b> que representa o nó inicial da barra.	setStartNode startNode
lpe	---	Atributo suprimido.	Funções suprimidas
EndNode	m_endNode	Ponteiro para um objeto da classe <b>CdTcCNode</b> que representa o nó final da barra.	setEndNode endNode
lpsText	m_lpsText	Não sofreu alterações além da troca do nomes do atributo e das suas funções.	setLpText lpText
enormal	m_enormal	Criada a função de acesso <b>enormal</b> .	setEnormal enormal
BarId	m_BarId	Foi criada outra função de atribuição de numeração para a barra para controlar essa numeração em barras copiadas.	setBarId(Adesk::Int16&) setBarId(Adesk::Boolean, CString&) BarId
TpBar	m_TpBar	Troca do nomes do atributo e das suas funções.	setTpBar tpBar
LoadComList	m_LoadComList	Troca do nome do atributo.	appendLoadCom getLoadCom
ComLoadCas	m_ComLoadCas	Troca do nome do atributo.	appendLoadCom getLoadCasCom
LoadDistList	m_LoadDistList	Troca do nome do atributo.	appendLoadPart getLoadPart
LengthDistList	m_LengthDistList	Troca do nome do atributo.	appendLengthPart getLengthPart
DistanceDistList	m_DistanceDistList	Troca do nome do atributo.	appendDistancePart getDistancePart

Atributo original	Novo atributo	Alteração sofrida	Funções de acesso e atribuição
PartLoadCas	m_PartLoadCas	Troca do nome do atributo.	appendLoadPart getLoadCasPart
LoadCenList	m_LoadCenList	Troca do nome do atributo.	appendLoadCen getLoadCen
DistanceCenList	m_DistanceCenList	Troca do nome do atributo.	appendDistanceCen getDistanceCen
CenLoadCas	m_CenLoadCas	Troca do nome do atributo.	appendLoadCen getLoadCasCen
KindGeom	m_kindGeom	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>kindGeom</b> .	setKindGeom kindGeom
KindCaso	m_kindCaso	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>kindCaso</b> .	setKindCaso kindCaso
KindLoad	m_kindLoad	Foi criado um método de acesso ao conteúdo desse atributo: a função <b>kindLoad</b> .	setKindLoad kindLoad

Seus métodos específicos: **textFit**, **drawLoad**, **verifyLp**, **removeLoadCen**, **removeLoadPart**, **removeLoadCom**, **StretchLoad**, **PointLoad**, **removeCaso**, **getLoadValLenCom**, **getLoadValLenCen**, **getLoadValLenPart** e **setBarId** mantiveram sua funcionalidade original. Algumas alterações foram necessárias principalmente para sua adequação à nova estrutura criada. As principais alterações ocorreram na manipulação de dados da classe **CdTcNode**. Essa manipulação passa a ser feita através dos métodos de acesso e de atribuição daquela classe. O método **getLengthBar** foi suprimido, pois os elementos sólidos que utilizarão a barra possuirão como atributo uma variável do tipo real que armazenará o comprimento da entidade. Esse atributo, com seus métodos de acesso e de atribuição passam, então, a substituir a função **getLengthBar**.

O método original **setBarId** foi alterado para retornar um parâmetro que é o próprio número da barra. Como essa função é chamada sobre elementos já criados, quando os mesmos são alterados e inseridos novamente no banco de dados, esse dado retornado passa a ser utilizado para evitar a renumeração dessas barras.

Além do método original, foi criado um novo método **setBarId** que além de retornar o número da barra, recebe o tipo de elemento ao qual a barra está associada. Esse novo método tornou-se necessário para a correta identificação do elemento que possui a barra. Esse dado é necessário para o armazenamento dos dados da barra em um dicionário de elementos utilizado pelo PREMOLD.

### 4.5.3 Classe CdTcCPlaca

A classe **CdTcCPlaca** (reestruturação da classe **Face**) é derivada da classe *AcDbEntity* e possui os seguintes atributos e métodos:

**TABELA 4.9 - Atributos e métodos da classe CdTcCPlaca**

Atributo original	Novo atributo	Alteração sofrida	Funções de acesso e atribuição
lps	m_lps	Criada a função de acesso <b>lps</b> .	setLps lps
lpsText	m_lpsText	Criada a função de acesso <b>lpsText</b> .	setLpsText lpsText
enormal	m_enormal	Criada a função de acesso <b>enormal</b> .	setEnormal enormal
FaceId	m_PlacaId	Não sofreu alterações além da troca de nomes do atributo e das suas funções.	setPlacaId placaId
vertxList	m_vertices	Criada a função de acesso <b>vertices</b> .	setVertices vertices
Repre	m_repre	Criadas as funções de atribuição e de acesso.	setRepre repre
ang	m_ang	Troca de nomes do atributo e das suas funções.	setAng ang
pti	m_pti	Ponto de coordenadas tridimensionais que indica o ponto de origem do eixo X local.	appendLoadCom getpti
ptf	m_ptf	Troca de nomes do atributo e das suas funções.	setPtf ptf

Atributo original	Novo atributo	Alteração sofrida	Funções de acesso e atribuição
ptaux	---	Atributo suprimido.	----
Control	m_control	Criadas as funções de atribuição e de acesso	setControl control
Per	m_per	Criadas as funções de atribuição e de acesso	setPer per
ComLoadCas	m_ComLoadCas	Não sofreu alterações além da troca do nome do atributo.	setLoad getLoadCas
ComLoad	m_ComLoad	Troca do nome do atributo.	setLoad getLoad
KindGeom	m_KindGeom	Criada a função de acesso <b>kindGeom</b> .	setKindGeom kindGeom
KindCaso	m_KindCaso	Criada a função de acesso <b>kindCaso</b> .	setKindCaso kindCaso
KindLoad	m_KindLoad	Criada a função de acesso kindLoad	SetKindLoad kindLoad

A funcionalidade de seus métodos específicos **textFit**, **drawRepre1**, **drawRepre2**, **drawRepre3**, **drawLoad**, **getVlen**, **getLoadVallEn**, **removeLoadCom**, **removeCaso** e **setFaceId** não foi alterada. As funções **setCG** e **getCG** foram suprimidas pois o centro de gravidade passa a ser um atributo da classe **CdTcCElem\_Estrutural** do PREMOLD.

#### 4.5.4 Classe CdTcCDictEnum

A classe **Config1** foi renomeada **CdTcCDictEnum**. Essa alteração visa uma melhor adequação do nome da classe à sua funcionalidade. Como os objetos dessa classe são armazenados em um dicionário e são responsáveis pela numeração dos elementos, foi escolhido o nome **CdTcCDictEnum**. A derivação da classe *AcDbObject* e todos os seus atributos foram mantidos. A única alteração nessa classe foi a inclusão dos caracteres "m\_" antes dos nomes originais de seus atributos, além da criação de funções de acesso e de atribuição para esses, seguindo a nomenclatura citada no item 4.4.

#### 4.5.5 Classe CdTcCCasos

A classe **CdTcCCasos** foi obtida pela reestruturação das classes originais **Casos** e **CurrCaso**. Como foi mencionado no item 4.4 a classe **CurrCaso** foi suprimida e o controle do caso de carregamento corrente passa a ser feito por uma nova variável do tipo booleana inserida na classe **CdTcCCasos**: "**m\_CurrCaso**". Essa variável indica o caso corrente através de seu estado. Se for atribuído o valor verdadeiro para a variável **m\_CurrCaso** de um determinado caso, esse será o caso corrente. Foram criadas as funções de atribuição e de acesso **setCurrCaso** e **currCaso** respectivamente.

Apenas um caso pode ser o corrente em determinado momento. Por isso a função **setCurrCaso**, além de atribuir o valor verdadeiro ao atributo **m\_CurrCaso** do caso cuja sigla lhe é repassada por parâmetro, atribui o valor falso para todos os demais objetos casos.

Também aqui foram mantidos todos os demais atributos da classe **Casos**, apenas com as alterações de nomenclatura citadas no item 4.4.

#### 4.5.6 Classes CdTcCReactorNode, CdTcCReactorBar, CdTcCReactorPlaca

As classes **CdTcCReactorNode**, **CdTcCReactorBar** e **CdTcCReactorPlaca** foram obtidas pela reestruturação das classes **Reactornode**, **Reactorbar** e **Reactorface**, respectivamente. Elas são derivadas da classe *AcDbObject*. Assim como as classes originais, as novas classes não possuem atributos, mas apenas a função *erased* foi definida para cada uma delas através de Polimorfismo, tendo sido mantida sua funcionalidade original na etapa de reestruturação. Assim, o retorno da numeração dos elementos excluídos do banco de dados para o objeto da classe **CdTcCDictEnum** continua a ser gerenciado por essas classes.

Durante o desenvolvimento do PREMOLD novas funcionalidade foram acrescentadas a essas classes e serão discutidas no capítulo 5 após a apresentação das classes do PREMOLD.

#### 4.5.7 Classe CdTcCCommandReactor

Essa classe é a reestruturação da classe **CommandReactor**.

Com a alteração da estrutura de ligação entre as classes **CdTcCNode** e **CdTcCBar**, uma barra passa a possuir duas entidades do tipo **Node**. Os comandos COPY, ARRAY e MIRROR, ao copiarem uma barra, copiam todos os seus elementos e, conseqüentemente, os nós são copiados. Assim, não é mais necessária a verificação de consistência feita pela função *commandEnded* da classe original. Pelo mesmo motivo não se justifica mais o travamento ou a liberação de determinadas camadas (*layers*) realizados originalmente no Modelador 3D pelas funções *commandEnded*, *commandWillStart* e *commandCancelled*, durante a execução ou término de determinados comandos. O gerenciamento da numeração das novas entidades copiadas passa a ser feito pela função *deepclone* definida nas classes **CdTcCNode** e **CdTcCBar**. A variável de controle *clBarIds* não é mais necessária pois ela era utilizada para listar os identificadores das barras copiadas e possibilitar a criação de nós nas extremidades dessas barras.

Nesta etapa de reestruturação das classes originais do Modelador 3D apenas criou-se uma nova classe de gerenciamento dos comandos do AutoCAD que será útil no desenvolvimento do PREMOLD. A implementação de suas funções foi então realizada durante o desenvolvimento do PREMOLD, para gerenciar a interferência de um elemento no seu banco de dados quando o elemento é copiado ou excluído do banco de dados do AutoCAD. A descrição completa dessa classe será realizada no capítulo 5 após a apresentação das classes do PREMOLD.

## 4.6 Reestruturação das Funções Globais

Todas as funções globais foram modificadas para adequarem-se à nova estrutura de classes do Modelador 3D. Sua funcionalidade básica não foi alterada, apenas o modo de acessar os atributos das classes foi alterado, ou mesmo o tipo de atributo acessado foi alterado.

Todas as funções globais foram transferidas para os arquivos DictUtils ou Utils, não sendo mantida nenhuma função global junto aos arquivos de classes do Modelador 3D. Os arquivos DictUtils (.cpp e .h) englobam funções para criação de dicionários do Moderador 3D, assim como funções que criam ou manipulam objetos armazenados nesses dicionários.

Nos arquivos Utils (cpp e .h) foram inseridas funções para criação ou manipulação de entidades do Moderador 3D, além de funções que realizam algumas atividades básicas como: criação e gerenciamento de camadas de desenho (*layers*); conversão de tipos de variáveis, criação de entidades nativas do AutoCAD; inserção de uma entidade no banco de dados do AutoCAD; atualização do estado de entidades.

# 5

## O MODELADOR PREMOLD

No item 1.6 foi mencionada a necessidade da realização inicial de algumas atividades que serviriam de base para o desenvolvimento do PREMOLD. Foram elas:

- Pesquisa sobre os fundamentos e pré-requisitos de projeto e execução de estruturas de concreto pré-moldado;
- Capacitação e treinamento no manuseio das ferramentas básicas utilizadas no desenvolvimento do aplicativo (AutoCAD 2002, ObjectARX 2002 e Visual C++ 6.0);  
e
- Estudo do funcionamento e do código fonte do Modelador 3D com posterior implementação de alterações na sua estrutura.

Após a conclusão das etapas iniciais, discutidas nos capítulos 2, 3 e 4, iniciou-se o desenvolvimento do PREMOLD através do desenvolvimento conceitual do aplicativo desejado. Nessa etapa foram definidos:

- O sistema estrutural a ser modelado;
- A funcionalidade pretendida para o aplicativo inicial e suas prováveis limitações de aplicação;
- Os tipos de elementos a serem modelados com levantamento dos dados essenciais para sua representação;
- Os critérios para o lançamento da estrutura através do aplicativo a ser desenvolvido;
- Os tipos de ligações modeladas e o tratamento dado às mesmas através da automação pretendida.

Após a definição dos conceitos e diretrizes iniciais, passou-se ao desenvolvimento computacional do aplicativo com:

- Declaração e definição de classes;
- Implementação de funções globais;
- Definição e implementação dos comandos próprios do PREMOLD.

## **5.1 Desenvolvimento Conceitual do PREMOLD: Concepção e Definições Gerais do Aplicativo**

Nos itens seguintes encontra-se uma explanação mais detalhada de cada uma das etapas do desenvolvimento conceitual do aplicativo.

### **5.1.1 Escolha do Sistema Estrutural a Ser Modelado**

Foi visto no item 2.4 que existem dois sistemas estruturais básicos para edifícios de múltiplos pavimentos em concreto pré-moldado: o sistema reticulado e o sistema de parede portante. Na concepção do PREMOLD, optou-se por desenvolver, numa primeira etapa de

trabalho, um modelador sólido para o sistema descrito no item 2.4.1: sistema estrutural reticulado com elemento de eixo reto. A escolha do sistema estrutural a ser inicialmente modelado foi determinada pelo desenvolvimento do PREMOLD a partir do Modelador 3D, uma vez que aquele fora desenvolvido para estruturas metálicas, nas quais o sistema estrutural muito se assemelha ao sistema reticulado com elemento de eixo reto.

### **5.1.2 Delimitação do Produto Pretendido**

O PREMOLD foi concebido como um modelador sólido específico para estruturas de concreto pré-moldado. Além do lançamento geométrico da estrutura através de elementos sólidos, ele realiza o lançamento automático de barras, nós e placas do modelo estrutural. Para o modelamento de barras, nós e placas são utilizadas as classes do Modelador 3D reestruturadas conforme exposto no capítulo 4.

A alteração da estrutura de classes do Modelador 3D gerou uma nova estrutura de classes que foi utilizada pelo PREMOLD. Os elementos sólidos são modelados por novas classes criadas a partir das classes do ObjectARX por Herança, através da estrutura de Generalização/Especialização, no desenvolvimento do novo aplicativo. Muitas dessas classes foram criadas utilizando-se também a estrutura Todo-Parte com agregação das classes reestruturadas do Modelador 3D, ou de outras classes do ObjectARX.

Como primeiro passo na criação dessas classes foi desenvolvida uma classe base para representar elementos estruturais genéricos. A partir dessa, foram derivadas duas classes: uma para representar os elementos estruturais de eixo reto e outra para representar os elementos estruturais planos. Essas duas classes representam tipos diferentes de elementos estruturais. Os elementos da primeira classe podem ser representados, no modelo estrutural, como barras delimitadas por dois nós. Os elementos planos podem ser representados por superfícies planas finitas, delimitadas por vértices. Todas as demais classes, que têm a função de representar os elementos sólidos tratados nesta etapa de desenvolvimento do PREMOLD, foram desenvolvidas tendo uma dessas três classes de elementos estruturais como classe base.

Foram criadas, ainda, classes de objetos (elementos sem representação gráfica), para melhor organizar os dados da estrutura gerada e facilitar sua exportação para arquivos de texto.

A numeração dos elementos gerados continua a ser feita automaticamente, do mesmo modo como era realizada no Modelador 3D.

O modelamento da estrutura de concreto pré-moldado, a partir do lançamento dos elementos sólidos, exigiu o desenvolvimento de novos comandos em substituição aos comandos originais do Modelador 3D. Os elementos sólidos, ao serem gerados, criam consigo os elementos do sistema estrutural reticulado adotado. Os dados sobre coordenadas nodais, incidência nodal e dados sobre as propriedades geométricas dos elementos são armazenados em objetos dentro de dicionários próprios, criados pelo PREMOLD. Esses dicionários, somados aos atributos das classes personalizadas que representam os elementos estruturais tratados, formam o banco de dados do PREMOLD. A partir desse banco de dados é possível exportar arquivos contendo todos os dados da estrutura gerada e de seus elementos. Esses dados poderão ser adequadamente tratados e enviados automaticamente a programas de análise e de dimensionamento em trabalhos futuros, efetuando a interligação desse sistema CAD com os sistemas CAE e ainda com sistemas CAM.

### **5.1.3 Tipos de Elementos a Serem Modelados com Levantamento dos Dados Essenciais para sua Representação**

O sistema estrutural reticulado com elemento de eixo reto é composto, basicamente, por pilares e vigas. Um dos sistemas estruturais resultantes da aplicação desse tipo de elemento apresenta a forma básica de pilares engastados na fundação e vigas articuladas nos pilares. Assim, nesta primeira etapa de desenvolvimento do aplicativo PREMOLD, optou-se por restringir-se o lançamento de estruturas àquelas concebidas segundo essa forma básica.

Além de pilares e vigas foi implementado o lançamento de painéis de piso e de blocos de fundação.

Os elementos de concreto pré-moldado, apesar de seguirem uma certa padronização de formas básicas, como discutido no item 2.2, apresentam uma grande variação de tipos de seções com dimensões variando conforme o fabricante.

Para o desenvolvimento desse aplicativo optou-se por gerar um banco de dados baseado em seções padronizadas por uma das empresas de pré-moldados do Estado de Minas Gerais. Essa decisão foi tomada em função de uma parceria anterior do CADTEC com essa empresa de pré-moldados que resultou na geração de uma biblioteca, para o AutoCAD, dos elementos estruturais catalogados pela empresa (PREMO, 1999). Através dessa biblioteca é possível gerar seções e vistas, lateral e em planta, dos elementos catalogados (pilares, vigas, painéis, telhas, etc.). A representação gráfica desses elementos é feita através de desenhos bidimensionais.

A nomenclatura e os parâmetros utilizados nessa biblioteca foram aproveitados para o desenvolvimento do PREMOLD. Todas as seções catalogadas na referida biblioteca foram parametrizadas e inseridas no banco de dados do PREMOLD.

A parametrização das seções possibilita a futura criação de novas seções semelhantes às existentes, através da alteração dos parâmetros utilizados. Nesta primeira etapa do trabalho não foram criados dispositivos para a inserção de novas seções ou para a alteração dos parâmetros pelo usuário. Foi prevista a criação de seções personalizadas apenas para as formas retangulares, com altura e largura de pilares e de vigas retangulares, assim como de lajes planas, definidas pelo usuário.

Além desses parâmetros para a definição da seção transversal dos elementos, são necessários outros dados para modelar um elemento sólido e efetuar as operações necessárias para tratar sua interação com outros elementos na estrutura lançada. São eles: altura e largura da seção transversal e o seu centro geométrico, comprimento do elemento, seu ponto de inserção, seus eixos locais, o tipo do elemento e tipo de sua seção transversal. Outros dados necessários para a manipulação dos elementos também devem fazer parte de seus atributos. Algumas propriedades geométricas podem ser obtidas do próprio AutoCAD, diretamente dos elementos sólidos ou de regiões planas tratadas por essa

plataforma. Foram, então, somados aos atributos do elemento as seguintes propriedades geométricas de sua seção transversal: seus raios de giração, seus momentos de inércia, seus momentos principais, seus eixos principais, seu produto de inércia e seu volume.

#### **5.1.4 Critérios para o Lançamento da Estrutura**

O lançamento dos elementos, durante o modelamento sólido, foi concebido baseado na montagem real de uma estrutura. A ordem de lançamento dos elementos foi estabelecida para que os tratamentos de conectividade e de interferências entre esses elementos, nas ligações, pudessem ser realizados automaticamente pelo aplicativo. Assim, um elemento só pode ser lançado sobre elementos que lhe sirvam de base numa estrutura real.

O pilar é o único elemento do PREMOLD que não necessita de uma outra entidade que lhe sirva de base para ser lançado. Assim, ele deve ser o primeiro elemento a ser inserido em um projeto gerado através do PREMOLD.

Vigas só podem ser lançadas sobre pilares ou sobre outras vigas, gerando-se vigas principais e vigas secundárias definidas no item 2.7.3.

Em um sistema reticulado com elementos de eixo reto, os elementos estruturais principais são os pilares e as vigas. Assim, os painéis de piso ou de vedação devem ser apoiados sobre vigas. Nesta primeira etapa de desenvolvimento do PREMOLD foi definida apenas a inserção dos painéis de piso. Para o lançamento desses painéis é necessária a existência prévia de duas vigas de apoio.

Para os blocos de fundação, a ordem de lançamento foi alterada: os blocos de fundação devem ser lançados após a inserção dos pilares. Essa alteração teve por finalidade possibilitar o dimensionamento do bloco de fundação de acordo com a seção do pilar ao qual ele servirá de base. Com esse procedimento é, ainda, possível alterar o pilar inserido inicialmente para que esse tenha seu comprimento acrescido da altura mínima de engaste estabelecida pela NBR 9062/85 (ABNT, 1985).

Com esses pré-requisitos para o lançamento dos elementos estruturais, o tratamento das ligações entre elementos é feito automaticamente, levando-se em consideração os tipos de elementos que compõem a ligação e suas seções transversais. Evita-se, ainda, que elementos sejam lançados em posições incompatíveis com sua funcionalidade. Por outro lado, esse tipo de lançamento requer do programador um grande esforço no sentido de prever todas as possibilidades de lançamento de um tipo de elemento sobre outro.

No desenvolvimento do PREMOLD foi tratado, principalmente, o lançamento de elementos segundo uma malha tridimensional modular formada por retas. Os elementos são lançados em planos horizontais ou verticais, com comprimento em uma direção paralela a um dos eixos da malha modular. O tratamento de elementos inseridos em uma angulação diferente da descrita foi iniciado para vigas. Tal procedimento não está concluído, faltando ainda implementar o tratamento da interferência dessas vigas, chamadas vigas esconsas, com seus pilares de apoio e demais vigas a eles ligadas.

### **5.1.5 Tipos de Ligações Modeladas e seu Tratamento Inicial**

Para o tratamento do sistema reticulado com elementos de eixo reto, com pilares engastados na fundação e ligação articulada entre pilares e vigas, nas ligações rígidas entre pilares e blocos de fundação, foi desenvolvido o lançamento automatizado de ligações por meio de cálice de fundação. Nas ligações articuladas entre pilares e vigas pressupõe-se o uso de apoios de elastômero, de chumbadores ou de chapa metálica soldada no topo das vigas. Esses acessórios para as ligações articuladas (chapas metálicas, chumbadores e elastômero) não foram tratados nesta etapa de trabalho, porém as folgas necessárias entre os elementos foram aqui previstas e são automaticamente lançadas nas ligações.

## **5.2 Desenvolvimento Computacional do PREMOLD**

Após a definição do aplicativo a ser desenvolvido e de suas diretrizes gerais, iniciou-se o desenvolvimento computacional do PREMOLD. Como primeiro passo desse

desenvolvimento, foi realizado um estudo para definição da estrutura hierárquica adequada às classes que seriam construídas para o aplicativo. Após a definição da estrutura de classes, essas foram criadas, assim como as funções globais e os comandos do aplicativo. A criação de classes, de funções e de comandos não foi realizada numa seqüência fixa e bem determinada. As atividades foram intercaladas conforme a identificação de novas necessidades a partir do andamento dos trabalhos. A interface do aplicativo com o usuário, também foi desenvolvida em etapas, as quais seguiram o desenvolvimento dos comandos do PREMOLD.

### 5.2.1 Estrutura de Classes do PREMOLD

No levantamento de dados para o desenvolvimento do PREMOLD e identificação dos elementos estruturais a serem tratados, foi observado que os elementos de uma estrutura segundo o sistema reticulado com elementos de eixo reto são, em sua maioria, elementos lineares. Esses elementos são representados por uma barra delimitada por dois nós, correspondendo à entidade **Bar** do Modelador 3D. Porém, existem alguns elementos estruturais que compõem, ou fazem parte desse tipo de estrutura como fechamento ou como fundação, que não se encaixam nessa definição. Nessa situação encontram-se os blocos de fundação, os consolos, as telhas, os acessórios e os painéis de piso ou de parede.

Os painéis de parede não possuem uma função estrutural no sistema tratado, sendo apenas responsáveis pelo fechamento da edificação. Porém, nos sistemas estruturais de parede portante, esses elementos desempenham papel estrutural e, por isso, foram aqui listados, prevendo-se seu uso em implementações futuras, onde outros sistemas estruturais poderão ser modelados. Identificou-se, então, que os elementos do tipo painel (tanto de piso quanto de vedação) podem ser representados por uma superfície plana delimitada por vértices, correspondendo à entidade **Placa** (originariamente nomeada **Face** no Modelador 3D).

Para representar, no modelo computacional, todos os elementos de concreto pré-moldado com função estrutural, foi criada uma classe derivada da classe *AcDbEntity* do ObjctcARX. Essa nova classe torna-se a classe base para todas as demais classes de entidades criadas no PREMOLD, recebendo o nome de **CdTcCElem\_Estrutural**. Todos os demais elementos, que possam ser entendidos como "um tipo de elemento estrutural", serão modelados,

através de classes derivadas dessa classe base, pela estrutura Generalização/Especialização. Os demais objetos necessários ao controle e ao funcionamento do PREMOLD serão modelados por classes derivadas, principalmente, da classe *AcDbObject* do ObjectARX.

A classe **CdTcCElem\_Estrutural** não possui nenhum objeto a ela associado, mas possui atributos e métodos capazes de representar qualquer elemento estrutural, ou seja, ela possui a abstração dos dados comuns a todos os elementos estruturais. Dessa classe base foram derivadas duas novas classes que representam elementos estruturais mais especializados: a classe **CdTcCElem\_Linear** e a classe **CdTcCElem\_Plano**.

A classe **CdTcCElem\_Linear** é formada, também, pela estrutura Todo-Parte através da agregação de um objeto (na verdade um ponteiro para objeto) da classe **CdTcCBar**. Esse objeto é o responsável por representar os elementos lineares no modelo estrutural tratado.

Na criação da classe **CdTcCElem\_Plano**, também foi utilizada a estrutura Todo-Parte. Essa classe é formada pela agregação de um ponteiro para um objeto da classe **CdTcCPlaca**. Esse objeto é o responsável por representar os elementos formados por uma superfície plana no modelo estrutural tratado.

Os consolos não possuem uma representação gráfica no sistema estrutural reticulado com elementos de eixo reto (item 2.4.1). Porém, para o sistema com elementos formados por trechos de eixo reto, os consolos possuem um comprimento maior e são representados como uma barra engastada no pilar, conforme definido no item 2.4.2. Assim, prevendo um futuro desenvolvimento do PREMOLD, contemplando esses sistemas, a classe **CdTcCConsolo** foi criada a partir da derivação da classe **CdTcCElem\_Linear**. Essa derivação também é útil para a representação de vigas curtas em balanço engastadas no pilar. Nesse caso, essas vigas engastadas podem ser representadas através do lançamento desses consolos de comprimento prolongado, sem o lançamento de vigas sobre os mesmos.

Os blocos de fundação, por não se enquadrarem em nenhuma dessas duas representações (por meio de barras ou por meio de superfícies planas), foram representados por uma

classe derivada diretamente da classe **CdTcCElem\_Estrutural**. A classe que representa os blocos de fundação foi chamada **CdTcCBloco**.

Utilizando-se de herança com a estrutura Generalização/Especialização, foram criadas as classes para representar os demais elementos estruturais tratados pelo PREMOLD.

Os pilares e as vigas são elementos lineares, representados por barras delimitadas por dois nós no sistema estrutural. Assim, esses elementos são modelados por classes derivadas da classe **CdTcCElem\_Linear**. A classe **CdTcCPilar** representa os pilares, enquanto a classe **CdTcCViga** representa as vigas.

Os painéis de piso, por serem do tipo elemento estrutural plano, foram representados pela classe **CdTcCPainel** derivada da classe **CdTcCElem\_Plano**.

Após terem sido criadas as classes para representar os elementos estruturais tratados pelo PREMOLD, foram criadas classes de controle que armazenam, em dicionários próprios, determinados dados dos elementos criados. Esses dados, armazenados nos objetos dessas classes, facilitam a busca, ou a pesquisa por tipos específicos, de elementos no banco de dados do AutoCAD, o estudo da conectividade entre os elementos, e a geração dos arquivos de exportação de dados.

Além do uso das classes de controle do Modelador 3D (reestruturadas) foram criadas as classes: **CdTcCDictNode**, **CdTcCDictBar** e **CdTcCDictElement**. O objeto da classe **CdTcCDictNode** armazena no dicionário **NodeDict** a numeração dos nós inseridos no banco de dados e suas coordenadas globais. O objeto da classe **CdTcCDictBar** armazena no dicionário **BarDict** a numeração das barras inseridas no banco de dados, a numeração de seu nós inicial e de seu nó final e o tipo de elemento que é representado pela barra (viga ou pilar). O objeto da classe **CdTcCDictElement** armazena no dicionário **ElementDict** a numeração dos elementos inseridos no banco de dados, o seu tipo (vigas, pilares e painéis) e o tipo de seção dos mesmos. Todos os demais dados dos elementos fazem parte dos atributos de cada classe que os representa. O armazenamento de dados na própria entidade evita ambigüidades e possíveis erros de identificação. Para acessar esses dados, o elemento é mapeado, dentro do banco de dados do AutoCAD, pelas funções reestruturadas do

Modelador 3D (funções **verifyNode**, **verifyBar**, **verifyPlaca**) e por novas funções criadas no desenvolvimento do PREMOLD para busca de entidades desse aplicativo (**verifyElement** e **verifyPainel**). Essas funções percorrem todo o banco de dados do AutoCAD e selecionam o elemento cujos atributos correspondam aos dados que tais funções recebem como parâmetro. Após a identificação do elemento ele pode ser aberto para leitura e seus atributos podem ser acessados através dos métodos de acesso definidos em sua respectiva classe.

Para o desenvolvimento dos desenhos de detalhamento de elementos estruturais foram criadas classes que representam: os formatos de desenho normatizados pela ABNT através da norma NBR 10068, "Folha de desenho - Leiatura e dimensões" (ABNT, 1987), suas margens, um conjunto de notas que pode ser inserido pelo usuário e um carimbo padrão desenvolvido para os trabalhos do CADTEC. Essas classes foram desenvolvidas pelos alunos de iniciação científica do CADTEC sob a coordenação das alunas de mestrado Regina Célia Guedes Leite, autora do PREMOLD, e Rosália Gusmão de Lima, como parte do treinamento e capacitação desses alunos nas ferramentas AutoCAD, C++ e ObjectARX. Após esse trabalho de capacitação dos alunos, os trabalhos foram reunidos, aproveitando-se as melhores propostas desenvolvidas por cada aluno, e foram criadas as classes **CdTcCFormato**, **CdTcCLegenda** e **CdTcCNotas** que foram incorporadas ao aplicativo PREMOLD.

A estrutura de classes do PREMOLD é apresentada na FIG. 5.1 a seguir. Nessa estrutura foram incluídas as classes reestruturadas do Modelador 3D, pois essas passam a fazer parte do PREMOLD.



foi ainda definida a função *deepClone*, que determina o comportamento de uma entidade quando essa é copiada. Através dessa função é possível determinar quais os atributos serão copiados ou como eles serão copiados.

### a) Classe **CdTcCElem\_Estrutural**

Essa classe foi criada pela derivação da classe *AcDbEntity* do ObjectARX e pela agregação de um ponteiro para um objeto da classe *AcDb3dSolid*, também do ObjectARX.

O objeto da classe *AcDb3dSolid* é o responsável pela representação gráfica do elemento estrutural.

A classe **CdTcCElem\_Estrutural** é capaz de representar todos os tipos de elementos estruturais de uma estrutura de concreto pré-moldado, utilizando para isso os seguintes atributos e métodos:

**TABELA 5.1 - Atributos e métodos da classe **CdTcCElem\_Estrutural**.**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_secao	CString	Armazena o tipo de seção do elemento estrutural	setSecao getSecao
m_CG	AcGePoint3D	Ponto de coordenadas tridimensionais que armazena o centro de gravidade do elemento estrutural	setCG cG
m_Insert_point	AcGePoint3D	Ponto de coordenadas tridimensionais que armazena o ponto de inserção do elemento estrutural no banco de dados do AutoCAD	setInsert_point insert_point
m_Zlocal	AcGeVector3D	Vetor tridimensional normal ao plano da seção transversal do elemento estrutural	setZLocal zLocal
m_XLocal	AcGeVector3D	Vetor tridimensional contido no plano da seção transversal do elemento estrutural, paralelo á base dessa seção, passando pelo ponto inicial de desenho da mesma. Esse ponto é definido para cada seção gerada no banco de dados.	setXLocal xLocal

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_YLocal	AcGeVectort3D	Vetor tridimensional contido no plano da seção transversal do elemento estrutural, paralelo à altura dessa seção, passando pelo ponto inicial de desenho.	setYLocal yLocal
m_radiiGyration	double*	Lista de números reais que armazena os raios de giração de uma seção transversal do elemento estrutural	setRadiiGyration radiiGyration
m_prinMoments	double*	Lista de números reais que armazena os momentos de inércia principais de uma seção transversal do elemento estrutural	setPrinMoments prinMoments
m_momInertia	double*	Lista de números reais que armazena os momentos de inércia de uma seção transversal de um elemento estrutural, relativos aos seus eixos locais	setMomInertia momInertia
m_prodInertia	double*	Lista de números reais que armazena os produtos de inércia de uma seção transversal de um elemento estrutural	setProdInertia prodInertia
M_prinAxes	AcGeVectro3d*	Lista de vetores tridimensionais que armazena os eixos principais de uma seção transversal de um elemento estrutural	setPrinAxes prinAxes
m_tipo	CString	Variável que armazena o tipo de um elemento estrutural	setTipo getTipo
m_psolid	AcDb3dSolid	Ponteiro para uma entidade do tipo sólido do AutoCAD (classe <i>AcDb3dSolid</i> ), responsável pela representação gráfica do elemento estrutural.	setSolid
m_comprimento	double	Variável do tipo real que armazena o comprimento de um elemento estrutural, sendo essa a dimensão do elemento na direção do eixo Z local	setComprimento comprimento
m_altura	double	Variável do tipo real que armazena a altura de um elemento estrutural, sendo essa a dimensão do elemento na direção do eixo Y local	setAltura altura
m_largura	double	Variável do tipo real que armazena a largura de um elemento estrutural, sendo essa a dimensão do elemento na direção do eixo X local	setLargura largura
m_volume	double	Variável do tipo real que armazena o volume de um elemento estrutural.	setVolume Volume

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_arrh	AcGeDoubleArray	Lista de números reais que representam os parâmetros na direção do eixo X local da seção transversal do elemento estrutural.	setArrh arrh
m_arrb	AcGeDoubleArray	Lista de números reais que representam os parâmetros na direção do eixo Y local da seção transversal do elemento estrutural.	setArrb arrb
m_pontos	AcGePoint3dArray	Lista de pontos de coordenadas tridimensionais que representam os vértices da seção transversal do elemento estrutural.	SetPontos Pontos

A função **worldDraw** foi definida por Polimorfismo. Nessa função após a execução da função virtual da classe base, foram acrescentados comandos para gerar o elemento sólido através da manipulação do ponteiro **m\_psolid** referente a um sólido da classe *AcDb3dSolid* definido na classe **CdTcCElem\_Estrutural**.

Também foram criados os métodos virtuais **setSolid** e **DrawElement**. Esses métodos deverão ser definidos por Polimorfismo em todas as classes derivadas dessa classe. O primeiro método possibilita a configuração e modelagem do objeto sólido a ser criado através da definição do ponteiro **m\_psolid**. O segundo método gera a representação gráfica do elemento estrutural modelado ao chamar sobre o ponteiro **m\_psolid** a função *draw* da classe *AcGiWorldDraw*.

Os métodos de acesso e de atribuição para os atributos dessa classe seguem a padronização de passagem de parâmetros e de retorno de mensagens de erro do ObjectARX, descrito no item 4.4. Apenas as funções de acesso às variáveis do tipo *CString* retornam o valor do atributo ao invés de retornarem mensagens de erro. As funções de acesso para esse tipo de variável não funcionam adequadamente quanto os valores são retornados através de seus parâmetros.

### b) Classe **CdTcCElem\_Linear**

Essa classe foi criada pela derivação da classe **CdTcCElem\_Estrutural** do PREMOLD e pela agregação de um ponteiro para um objeto da classe **CdTcCBar**.

Ela é capaz de representar todos os elementos estruturais do tipo linear, ou seja, todos aqueles que são representados por uma barra delimitada por dois nós no modelo estrutural.

Ela herda todos os atributos e métodos da classe base, sendo criados para a classe **CdTcCElem\_Linear** os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.2 - Atributos e métodos da classe **CdTcCElem\_Linear****

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_bar	CdTcCBar*	Ponteiro para objeto da classe <b>CdTcCBar</b> . Esse objeto será o responsável pela representação do elemento no modelo estrutural. A numeração do elemento estrutural será a mesma numeração dada à sua entidade <b>Bar</b> .	SetBar Bar

### c) Classe **CdTcCElem\_Plano**

Essa classe foi criada pela derivação da classe **CdTcCElem\_Estrutural** do PREMOLD e pela agregação de um ponteiro para um objeto da classe **CdTcCPlaca**.

Ela é capaz de representar todos os elementos estruturais do tipo plano, ou seja, todos aqueles que são representados por uma superfície plana delimitada por vértices no modelo estrutural.

Ela herda todos os atributos e métodos da classe base, sendo criados para a classe **CdTcCElem\_Plano** os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.3 - Atributos e métodos da classe CdTCElem\_Plano**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_placa	CdTcCPlaca*	Ponteiro para objeto da classe <b>CdTcCPlaca</b> . Esse objeto será o responsável pela representação do elemento no modelo estrutural. A numeração do elemento estrutural será a mesma dada á sua entidade <b>Placa</b> .	SetPlaca Placa

#### d) Classe CdTcCBloco

Essa classe foi derivada diretamente da classe **CdTcCElem\_Estrutural**. Ela representa os objetos do tipo bloco de fundação, os quais não são representados por meio de barras nem de superfícies planas no modelo estrutural.

Os blocos de fundação podem ser representados apenas por um nó ou um ponto de apoio para os pilares. Apesar disso, nesta etapa de desenvolvimento do PREMOLD, os blocos de fundação não possuem um objeto do tipo **Node**. Os nós de apoio da estrutura são representados pelos nós iniciais dos pilares do primeiro pavimento.

Foram criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.4 - Atributos e métodos da classe CdTcCBloco**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_BaseY	double	Número real que armazena o valor da dimensão da base do bloco de fundação na direção do eixo Y local	setBaseY baseY
m_BaseX	double	Número real que armazena o valor da dimensão da base do bloco de fundação na direção do eixo X local	setBaseYX baseX
m_profund_furo	double	Número real que armazena o valor da altura do colarinho do bloco de fundação na direção do eixo Z local	setProfund_furo Profund_furo
m_alt_furo	double	Número real que armazena o valor da dimensão do colarinho do bloco de fundação na direção do eixo Y local	setAlt_furo alt_furo
m_larg_furo	double	Número real que armazena o valor da dimensão do colarinho do bloco de fundação na direção do eixo X local	setLarg_furo larg_furo
m_h_soco	double	Número real que armazena o valor da altura da base do bloco de fundação na direção do eixo Z local	setH_soco h_soco
m_h_nicho	double	Número real que armazena o valor da profundidade do vão do colarinho do bloco de fundação na direção do eixo Z local	setH_nicho h_nicho
m_psolid1	AcDb3dSolid*	Ponteiro para objeto da classe <i>AcDb3dSolid</i> que modela o sólido que representa o colarinho do bloco de fundação.	
m_psolid2	AcDb3dSolid *	Ponteiro para objeto da classe <i>AcDb3dSolid</i> que modela o sólido que representa a base do bloco de fundação.	

As funções **worldDraw** e **DrawElement** foram definidas por Polimorfismo. Em ambas as funções após a execução das funções virtuais da classe base, foram acrescentados comandos para gerar o elemento sólido através da manipulação dos ponteiros para objetos da classe *AcDb3dSolid*, definidos na classe **CdTcCBloco**: **m\_psolid1** e **m\_psolid2**.

### e) Classe CdTcCConsolo

Essa classe foi derivada da classe **CdTcCElem\_Linear**. Ela representa os objetos do tipo consolo e herda todos os atributos e métodos da classe base, sendo criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.5 - Atributos e métodos da classe CdTcCConsolo**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_alt_chanfro	double	Número real que armazena o valor da altura da parte inclinada do consolo trapezoidal na direção do eixo Y local	setAlt_chanfro alt_chanfro
m_profund	double	Número real que armazena o valor do comprimento do consolo na direção do eixo X local	setProfund profund
m_tipocons	int	Número inteiro que armazena a codificação para o tipo de consolo: 1 para consolo retangular e 0 para consolo trapezoidal	setTipocons tipocons

### f) Classe CdTcCPilar

Essa classe foi criada pela derivação da classe **CdTcCElem\_Linear** e pela agregação de uma lista de objetos da classe **CdTcCConsolo**. Ela representa os objetos do tipo pilar, herdando todos os atributos e métodos da classe base.

A agregação de objetos da classe **CdTcCConsolo** deve-se ao fato de que os consolos são objetos que não têm existência independente. Os consolos vão sempre existir ligados a um pilar ou até mesmo a uma viga. Assim, a inserção de vigas junto aos pilares no sistema estrutural reticulado pressupõe o lançamento de consolos nos pilares, nos pontos de ligação entre esses elementos.

Existem de outros tipos de ligação entre pilares e vigas como por exemplo vigas apoiadas no topo dos pilares e consolos metálicos. Nesta etapa de desenvolvimento foram tratadas apenas as ligações entre vigas e pilares por meio de consolos de concreto pré-moldado.

Para essa classe foram criados os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.6 - Atributos e métodos da classe CdTcCPilar**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_altura_furo	Double	Número real que armazena a altura, em relação à base do pilar, em que o tubo de descida de água pluvial é desviado da direção do eixo Z local do pilar, saindo em uma das faces do pilar	SetAltutra_furo altura_furo
m_diam_furo	Double	Número real que armazena o valor do diâmetro do furo de descida de água pluvial	setDiam_furo diam_furo
m_rotacao_z	double	Número real que armazena o valor da rotação dos eixos locais X e Y do pilar em relação aos eixos globais X e Y do AutoCAD	setRotacao_z rotacao_z
m_psolid1	AcDb3dSolid*	Ponteiro para objeto da classe <i>AcDb3dSolid</i> que modela o sólido que representa o consolo inserido no pilar em determinado instante..	
m_psolid2	AcDb3dSolid *	Ponteiro para objeto da classe <i>AcDb3dSolid</i> que modela o sólido que representa o tubo de descida de água do pilar	
m_consolos	Acrray<CdTcCConsolo*, CdTcCConsolo*>	Lista de que armazena ponteiros para objetos da classe <b>CdTcCConsolo</b> que representam os consolos inseridos no pilar.	setConsolo getConsolos

As funções **setConsolo** e **getConsolos** recebem um parâmetro (um número inteiro) que determina uma posição da lista de ponteiros para objetos do tipo consolo. A função **setConsolo** adiciona um ponteiro para objetos do tipo consolo na lista **m\_consolos** na posição determinada por aquele parâmetro, enquanto a função **getConsolos** retorna o ponteiro para objetos do tipo consolo armazenado na lista **m\_consolos** na posição determinada pelo parâmetro.

Foram criadas, ainda, as funções específicas: **resetConsolo** e **getConsoloslength**. A primeira função retira todos os dados armazenados na lista **m\_consolos**, enquanto a segunda retorna o número de consolos armazenados nessa lista.

As funções **worldDraw** e **DrawElement** foram definidas por Polimorfismo da mesma maneira realizada para a classe **CdTcCBloco**. Em ambas as funções, após a execução das funções virtuais da classe base, foram acrescentados comandos para gerar o elemento sólido através da manipulação dos ponteiros para *AcDb3dSolid* definidos na classe **CdTcCPilar**: **m\_psolid1** e **m\_psolid2**.

### g) Classe **CdTcCViga**

A classe **CdTcCViga** foi criada pela derivação da classe **CdTcCElem\_Linear** e pela agregação de uma lista de objetos da classe **CdTcCConsolo**. Ela representa os objetos do tipo viga, herdando todos os atributos e métodos da classe base.

A agregação de objetos da classe **CdTcCConsolo**, na formação dessa classe, deve-se a três motivos básicos:

- Na ligação com pilares, geralmente as vigas são apoiadas sobre os consolos inseridos nos pilares. O encaixe das vigas nos consolos pode ser feito através de consolos externos, de consolos embutidos ou através de consolos semi-embutidos. Os consolos externos são colocados abaixo do nível inferior das vigas, não exigindo cortes nas mesmas junto ao apoio. Os consolos embutidos (ou semi-embutidos no caso de consolos trapezoidais) são posicionados em uma altura superior à base da viga, exigindo a redução da seção transversal dessa junto aos apoios, formando dentes de concreto na viga, ou dentes de Gerber (EL DEBS, 2000). (FIG. 2.29)
- As vigas também podem servir de apoio para outras vigas. Algumas vigas já possuem uma seção transversal com formato que possibilita o apoio de painéis ou de vigas sem a necessidade de outros elementos de apoio. Em outros casos, a seção transversal da viga não possibilita esse apoio direto. Esse é o caso de vigas apoiadas em vigas retangulares

em altura inferior ao topo das últimas. Nesse caso, as vigas de apoio podem receber consolos para suportar as vigas secundárias. (FIG. 2.30)

- Pode ocorrer, ainda, que, na ligação entre viga principal e viga secundária, a primeira não sofra alterações, mas que a segunda sofra reduções em sua seção transversal para ser encaixada na principal. Ou ainda, podem ocorrer reduções de seção na ligação entre duas vigas fora do pilar (ligação comum em sistema reticulado com elementos formados por trechos de eixo reto). Nesse caso, ambas as vigas podem sofrer redução de seção transversal quando são moldados dentes de apoio em ambas, um servindo de apoio, ou consolo, para a outra viga. (FIG. 2.29 e FIG. 2.30)

Assim, no lançamento da representação gráfica da viga, é necessário repassar à sua função de modelamento, **setSolid**, qual o tipo de apoio usado, sua posição e suas dimensões. No caso de vigas apoiadas em consolos embutidos, ou semi-embutidos, a viga é devidamente reduzida no apoio. Os consolos são dimensionados no tamanho da redução necessária e são subtraídos do sólido original da viga durante sua modelagem. No caso de vigas principais com consolos, esses são anexados à viga.

Para a classe **CdTcCViga** foram criados os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.7 - Atributos e métodos da classe CdTcCViga**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_espelha	Adesk::Boolean	Variável booleana que indica se a seção transversal de uma viga será inserida espelhada ou não em relação ao modelo de inserção da viga. Caso a variável assumo valor verdadeiro a seção será espelhada antes da inserção da viga.	setEspelha espelha
m_psolid1	AcDb3dSolid*	Ponteiro para objeto da classes <i>AcDb3dSolid</i> que modela o sólido que representa o consolo inserido junto ao nó inicial da viga.	

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_psolid2	AcDb3dSolid *	Ponteiro para objeto da classes <i>AcDb3dSolid</i> que modela o sólido que representa o consolo inserido junto ao nó final da viga.	
m_psolid3	AcDb3dSolid*	Ponteiro para objeto da classes <i>AcDb3dSolid</i> que modela o sólido que representa o consolo que inserido junto à viga para apoio de uma viga secundária.	
m_consolos	Acrray<CdTcCConsolo*, CdTcCConsolo*>	Lista de que armazena ponteiros para objetos da classe <b>CdTcCConsolo</b> que representam os consolos inseridos junto aos nós inicial e final da viga ou ao longo da mesma.	setConsolo getConsolos

Foram criadas as funções específicas **resetConsolo** e **getConsoloslength**. Essas duas funções, assim como as funções **setConsolo** e **getConsolos**, possuem a mesma funcionalidade que as funções homônimas definidas na classe **CdTcCPilar**.

As funções **worldDraw** e **DrawElement** foram definidas por Polimorfismo da mesma maneira realizada para a classe **CdTcCBloco**. Em ambas as funções, após a execução das funções virtuais da classe base, foram acrescentados comandos para gerar o elemento sólido através da manipulação dos ponteiros para *AcDb3dSolid* definidos na classe **CdTcCViga**: **m\_psolid1** e **m\_psolid2**.

## h) Classe **CdTcCPainel**

Essa classe obtida pela derivação da classe **CdTcCElem\_Plano**. Ela representa os objetos do tipo painel (de piso ou de vedação) e herda todos os atributos e métodos da classe base, sendo criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.8 - Atributos e métodos da classe CdTcCPainel**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_ptslice	AcGePoint3d	Coordenada do ponto de inserção de um painel retangular para efetuar um corte no painel modelado.	setPtslice ptslice
m_slice	Adesk::Boolean	Variável booleana que determina se o painel inserido será cortado na direção de seu comprimento.	setSlice slice
m_psolid1	AcDb3dSolid*	Ponteiro para objeto da classe <i>AcDb3dSolid</i> que modela os furos de um painel alveolar.	
m_furos	CArray<AcDb3dSolid*, AcDb3dSolid*>	Vetor que armazena os sólidos para construção dos furos do painel alveolar.	
m_diam_furo	Double	Número real que armazena o diâmetro dos furos do painel alveolar.	setDiam_furo diam_furo

Para realizar cortes longitudinais em um painel, foi criada a função global **GeraSlice**, que modela um painel retangular com dimensões externas iguais às dimensões de um outro painel que está sendo modelado. Esse novo painel é posicionado no mesmo alinhamento do painel original, com ponto de inserção coincidente com um ponto predefinido ao longo do painel original. Através da subtração entre dois sólidos do AutoCAD, o painel auxiliar é subtraído do painel original, sendo realizado, então, o corte longitudinal do painel original em uma posição definida. Essa função, inserida no arquivo **GeraSecao** do DBX, é acionada pelo método **setSolid** dessa classe.

As funções **worldDraw** e **DrawElement** foram definidas por Polimorfismo da mesma maneira realizada para a classe **CdTcCBloco**. Em ambas as funções, após a execução das funções virtuais da classe base, foram acrescentados comandos para gerar o elemento sólido através da manipulação do ponteiro para *AcDb3dSolid*, definido na classe **CdTcCPainel**: **m\_psolid1**.

### i) Classe CdTcCDictElemt

Essa classe foi derivada da classe *AcDbObject* do ObjectARX. Ela representa elementos sem representação gráfica. Seus objetos armazenam dados de controle dos elementos estruturais presentes no banco de dados do AutoCAD e são inseridos no dicionário **ElementDict**, próprio do PREMOLD. Os dados armazenados por seus objetos são: a numeração do elemento, o tipo de elemento (viga, pilar, painel) e o nome da seção transversal do elemento. Para armazenar tais dados foram criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.9** - Atributos e métodos da classe **CdTcCDictElement**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_elementId	Adesk::Int16	Número identificador do elemento, fornecido pela classe <b>CdTcCDictEnum</b>	setElementId elementId
m_type	CString	Variável que armazena o tipo do elemento referenciado (pilar, viga, painel)	setType getType
m_element	CString	Variável que armazena o nome da seção transversal do elemento referenciado	setElement element

### j) Classe CdTcCDictNode

Essa classe foi derivada da classe *AcDbObject* do ObjectARX. Seus objetos armazenam dados de controle das entidades do tipo **Node** presentes no banco de dados do AutoCAD através da criação de elementos lineares. Esses objetos são inseridos no dicionário **NodeDict**, próprio do PREMOLD. Os dados armazenados por seus objetos são: a numeração da entidade **Node** e a coordenada tridimensional de inserção dessa entidade. Para armazenar tais dados foram criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.10 - Atributos e métodos da classe CdTcCDictNode**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_NodeId	Adesk::Int16	Número identificador da entidade <b>Node</b> , fornecido pela classe <b>CdTcCDictEnum</b>	setNodeId nodeId
m_pto	AcGePoint3d	Ponto de coordenadas tridimensionais que armazena as coordenadas de inserção da entidade <b>Node</b> referenciada	setPto pto

### k) Classe CdTcCDictBar

Essa classe foi derivada da classe *AcDbObject* do ObjectARX. Seus objetos armazenam dados de controle das entidades do tipo **Bar**, presentes no banco de dados do AutoCAD através da criação de elementos lineares. Esses objetos são inseridos no dicionário **BarDict**, próprio do PREMOLD. Os dados armazenados por seus objetos são: a numeração da entidade **Bar**, a numeração das entidades **Node**, que representam os nós inicial e final da entidade **Bar**, e o tipo de elemento que gerou a entidade **Bar** (pilar ou viga). Para armazenar tais dados foram criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.11 - Atributos e métodos da classe CdTcCDictBar**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_BarId	Adesk::Int16	Número identificador da entidade <b>Bar</b> fornecido pela classe <b>CdTcCDictEnum</b>	setBarId barId
m_endNodeId	Adesk::Int16	Número identificador da entidade <b>Node</b> que representa o nó final da barra	setEndNodeId endNodeId
m_startNodeId	Adesk::Int16	Número identificador da entidade <b>Node</b> que representa o nó inicial da barra	setStartNodeId startNodeId
m_tipo	CString	Variável que armazena o tipo do elemento referenciado (pilar ou viga)	setTipo gettipo

## 1) Classe CdTcCFormato

Essa classe foi derivada da classe *AcDbEntity* do ObjectARX e representa os formatos de desenho e suas margens.

Foram criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.12** - Atributos e métodos da classe **CdTcCFormato**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_num_formato	double	Número real que representa o tipo de formato a ser desenhado	setNum_formato num_formato
m_ins_pt	AcGePoint3d	Ponto de coordenadas tridimensionais que armazena o ponto de inserção de um formato	setIns_pt ins_pt
m_H	double	Número real que armazena o valor da altura do formato	setH h
m_B	double	Número real que armazena o valor da largura do formato	setB b
m_margeme	double	Número real que armazena o valor da margem esquerda do formato	setMargeme margeme
m_margemd	double	Número real que armazena o valor da margem direita do formato	setMargemd margemd
m_margems	double	Número real que armazena o valor da margem superior do formato	setMargems margems
m_margemi	double	Número real que armazena o valor da margem inferior do formato	setMargemi margemi

Além das funções obrigatórias descritas no item 3.3.4, as funções **worldDraw** e **setLayer** foram definidas por Polimorfismo. Em ambas as funções, após a execução das funções virtuais da classe base, foram acrescentados comandos para gerar a representação gráfica da entidade criada e definir sua camada de inserção.

### m) Classe CdTcCLegenda

Essa classe foi derivada da classe *AcDbEntity* do ObjectARX e representa as legendas, ou carimbos, a serem inseridos nos formatos de desenho.

Foram criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.13 - Atributos e métodos da classe CdTcCLegenda**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_FolhaNumero	double	Número real que representa o número da folha de desenho quando existe mais de um formato para um mesmo projeto	setfolhaNumero folhaNumero
m_ProjetoNumero	double	Número real que representa o número de identificação de um projeto	setprojetoNumero projetoNumero
m_InsPoint	AcGePoint3d	Ponto de coordenadas tridimensionais que armazena o ponto de inserção da legenda	setinsPoint insPoint
m_Altura	double	Número real que armazena o valor da altura total da legenda	setAltura altura
m_Largura	double	Número real que armazena o valor da largura total da legenda	setLargura largura
m_DesenhoNumero	double	Número real que representa o número de identificação de um desenho	setdesenhoNumero desenhoNumero
m_Escala	CString	Conjunto de caracteres que armazena o tipo de escala utilizada no desenho	setEscala escala
m_Verifica	CTime	Variável do C++ que armazena a data de verificação do desenho	setVerifica verifica
m_Desenho	CTime	Variável do C++ que armazena a data de execução do desenho	setDesenho desenho
m_Aprovacao	CTime	Variável do C++ que armazena a data de aprovação do desenho	setAprovacao aprovacao
m_Projeto	CTime	Variável do C++ que armazena a data do projeto	setProjeto projeto
m_Calculo	CTime	Variável do C++ que armazena a data de cálculo da estrutura	setCalculo calcula
m_Titulo	CString	Conjunto de caracteres que armazena o título do projeto ao qual pertence o desenho	setTitulo titulo

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_Obra	CString	Conjunto de caracteres que armazena o nome da obra do desenho	setObra obra
m_EngResp	CString	Conjunto de caracteres que armazena o nome do engenheiro responsável pelo projeto estrutural ao qual pertence o desenho	setEngResp engResp

Também nessa classe foram definidas por Polimorfismo as funções **worldDraw** e **setLayer** além das obrigatórias descritas no item 3.3.4.

#### n) Classe CdTcNota

Essa classe foi derivada da classe *AcDbEntity* do ObjectARX e representa as notas, ou observações, inseridas no desenho pelo usuário.

Foram criados para essa classe os seguintes atributos e métodos de acesso e de atribuição:

**TABELA 5.14** - Atributos e métodos da classe **CdTcNota**

Atributo	Tipo	Descrição	Funções de acesso e atribuição
m_InsPoint	AcGePoint3d	Ponto de coordenadas tridimensionais que armazena o ponto de inserção da nota	setInsPoint insPoint
m_Nota	CString	Conjunto de caracteres que armazena a informação que deve ser apresentada na nota de número N na tela gráfica. Foram criadas 12 variáveis m_Nota (m_Nota1, m_Nota2, etc..)	setNotaN notaN

Assim como para as classes **CdTcCFormato** e **CdTcCLegenda**, também para essa classe foram definidas por Polimorfismo as funções **worldDraw** e **setLayer** além das obrigatórias descritas no item 3.3.4.

### 5.2.3 Adição de funcionalidade às classes reestruturadas do Modelador 3D

As classes **CdTcCReactorNode**, **CdTcCReactorBar** e **CdTcCReactorPlaca** continuam a possuir os mesmos atributos e funções definidos no Modelador 3D, porém a função *erased* de cada classe passa a desempenhar novas funções, além das funções até então definidas.

A classe **CdTcCCommandReactor** não recebeu, no processo de reestruturação, a mesma funcionalidade original definida no Modelador 3D (ver item 4.5.7). A implementação de nova funcionalidade para essa classe foi deixada, propositadamente, para uma etapa posterior à criação e implementação das classes do PREMOLD.

As novas funcionalidades inseridas na função *erased* das classes **CdTcCReactorNode**, **CdTcCReactorBar** e **CdTcCReactorPlaca** e as novas funcionalidades da classe **CdTcCCommandReactor** são assim descritas.

#### a) Classe **CdTcCReactorNode**

Além de retornar a numeração das entidades do tipo **Node** excluídas do banco de dados para a variável **m\_AvNodeId** da classe **CdTcCDictEnum**, a função *erased* retira do dicionário **NodeDict** o objeto que armazena os dados do nó excluído.

#### b) Classe **CdTcCReactorBar**

Quando um elemento estrutural do tipo linear é excluído do banco de dados do AutoCAD, o objeto do tipo barra por ele possuído também é excluído. Assim, além de retornar a numeração das entidades do tipo **Bar** excluídas do banco de dados para a variável **m\_AvBarId** da classe **CdTcCDictEnum**, a função *erased* retira do dicionário **ElementDict** o objeto que armazena os dados do elemento estrutural excluído. Do

dicionário **BarDict** é retirado o objeto que armazena os dados da barra possuída por aquele elemento.

### c) Classe **CdTcCReactorPlaca**

Assim como ocorre para o objeto do tipo barra, quando um elemento estrutural do tipo plano é excluído do banco de dados do AutoCAD, o objeto do tipo placa por ele possuído também é excluído. Portanto, a função *erased* da classe **CdTcCReactorPlaca**, além de retornar a numeração das entidades do tipo **Placa** excluídas do banco de dados para a variável **m\_AvPlacaId** da classe **CdTcCDictEnum**, retira do dicionário **ElementDict** os objetos que armazenam os dados das placas excluídas.

### d) Classe **CdTcCCommandReactor**

A implementação de suas funções *commandEnded* e *commandCancelled* foi realizada para gerenciar a interferência de um elemento no banco de dados do PREMOLD quando o primeiro é copiado ou excluído do banco de dados do AutoCAD.

No PREMOLD, o gerenciamento da numeração das novas entidades copiadas passa a ser feito pela função *deepclone* definida nas classes **CdTcCNode** e **CdTcCBar**. Porém o preenchimento das coordenadas de uma entidade do tipo **Node** no dicionário **NodeDict** é realizado antes que a entidade copiada tenha seus dados atualizados. Portanto, após a realização do comando de cópia, essa classe de reatores efetua a atualização das coordenadas das entidades do tipo **Node** no referido dicionário. Caso o comando seja cancelado o objeto do dicionário criado para armazenar os dados da entidade copiada é excluído.

O gerenciamento das interferências entre entidades durante os comandos de cópia, de movimentação ou de exclusão do banco de dados do AutoCAD não foi implementado nesta etapa do trabalho.

## 5.2.4 Funções Globais do PREMOLD

No presente trabalho são consideradas funções globais do aplicativo aquelas funções que não são métodos de nenhuma classe em especial, mas que manipulam ou criam objetos de determinada classe.

As funções globais do PREMOLD são declaradas e definidas em arquivos próprios chamados de Utils, DictUtils ou GeraSecao. Os arquivos Utils e DictUtils são reestruturações dos arquivos de mesmo nome do Modelador 3D. Neles foram inseridas, além das funções globais daquele aplicativo, as funções globais do PREMOLD. O arquivo Utils contém funções que criam ou manipulam entidades do PREMOLD, enquanto o arquivo DictUtils contém funções que criam ou manipulam os objetos armazenados nos dicionários do PREMOLD, além de funções que criam esses dicionários. O arquivo GeraSecao contém as funções responsáveis por gerar, a partir dos parâmetros das seções transversais catalogadas pelo aplicativo, os pontos de coordenadas tridimensionais que representam os vértices dessas seções. Esse arquivo é inserido apenas no projeto DBX, uma vez que suas funções são utilizadas apenas pelo método **setSolid** de cada classe de entidades do PREMOLD. Os arquivos Utils e DictUtils são inseridos tanto no projeto ARX quanto no projeto DBX, possuindo, em cada projeto, apenas as funções ali utilizadas. Cada nome de arquivo aqui citado representa o arquivo de declaração, ou arquivo cabeçalho (.h) e o arquivo fonte, ou arquivo de definição (.cpp), conforme citado no item 3.2.

### a) Funções declaradas e definidas nos arquivos DictUtils do projeto DBX.

Nesses arquivos são declaradas e definidas as funções **CreateNode**, **CreateBar** e **CreateElement**, as quais criam os objetos a serem armazenados nos dicionários **NodeDict**, **BarDict** e **ElementDict**, respectivamente. Para as funções **CreateNode** e **CreateBar**, foi feita a sobrecarga dessas funções para que os objetos dos dicionários **NodeDict** e **BarDict** pudessem ser criados de duas formas diferentes. A primeira forma das funções abre o dicionário correspondente para escrita antes da criação do objeto, enquanto a segunda forma cria um objeto em um dicionário que já se encontra aberto para escrita. A segunda

forma recebe, além dos parâmetros repassados para a primeira forma, um outro parâmetro que é um ponteiro para dicionário que aponta para aquele onde o elemento será criado. Os parâmetros dessas funções representam os dados de uma determinada entidade que devem ser armazenados no objeto de dicionário a ser criado.

Para a função **CreateElement** há apenas um modo de implementação: o dicionário para inserção do objeto é aberto pela própria função, não sendo necessário repassá-lo como parâmetro.

Além dessas funções são definidas funções que buscam, dentro dos dicionários citados, os elementos ali armazenados. São elas: **getNodeData**, **getBarData** e **getElementData**. Essas funções retornam os objetos dos dicionários **NodeDict**, **BarDict** e **ElementDict** respectivamente. Elas são utilizadas no comando responsável por gerar os arquivos contendo os relatórios quantitativos e qualitativos sobre a estrutura e seus elementos constituintes, assim como pelo comando que gera os desenhos de detalhamento da estrutura.

## **b) Funções declaradas e definidas nos arquivos DictUtils do projeto ARX.**

Além das funções listadas para o arquivo DictUtils do DBX, são declaradas no arquivo DictUtils do ARX as funções **createEnumDict**, **createElementDict**, **createNodeDict** e **createBarDict** responsáveis por criar os dicionários do PREMOLD sempre que o aplicativo é carregado ou quando um novo desenho é criado com o aplicativo já carregado.

## **c) Funções declaradas e definidas nos arquivos Utils dos projetos DBX e ARX.**

As funções inseridas nesses arquivos são basicamente as funções do Modelador 3D reestruturadas. Foram acrescentadas apenas as funções **verifyElement** e **verifyPainel**

criadas no desenvolvimento do PREMOLD. Essas funções possuem a mesma funcionalidade básica das funções **verifyNode**, **verifyBar** e **verifyPlaca** (função **verifyFace** renomeada), responsáveis pela busca de entidades do Modelador 3D no banco de dados do AutoCAD.

As novas funções de busca de entidades do PREMOLD foram desenvolvidas com base nas funções do Modelador 3D e tornaram-se essenciais à manipulação das entidades para: exportação de dados; geração de desenhos de detalhamento; futuras implementações como, por exemplo, a edição de entidades.

No Modelador 3D, as entidades são mapeadas no banco de dados do AutoCAD através de coordenadas que representam: o ponto de inserção para entidades do tipo **Node** (função **verifyNode**); os pontos de inserção dos nós inicial e final para a entidade **Bar** (função **verifyBar**); uma lista de pontos de coordenadas tridimensionais que representam os vértices de uma região para a entidade **Placa** (função **verifyPlaca**). Essas funções, após a identificação de uma entidade com coordenadas correspondentes às coordenadas utilizadas na busca, retornam o identificador da entidade encontrada.

A função **verifyElement** percorre o banco de dados do AutoCAD à procura de entidades do tipo **Pilar** ou **Viga** que possuam como identificação de sua entidade **Bar** um número igual aquele repassado como parâmetro para a função. Caso seja encontrada uma entidade, **Pilar** ou **Viga**, que preencha esse requisito, seu identificador é retornado por essa função.

De modo semelhante à função anterior, a função **verifyPainel** procura, por todo o banco de dados do AutoCAD, entidades do tipo **Painel** cuja identificação de placa seja um número igual ao seu parâmetro. Uma vez tendo encontrado a entidade **Painel** desejada, a função retorna seu identificador.

#### **d) Funções declaradas e definidas nos arquivos GeraSecao do projeto DBX.**

Esse arquivo possui a declaração e a definição das funções **GeraSecao** e **GeraSlice**.

A função **GeraSecao** recebe como parâmetros o tipo de seção a ser gerada, sua altura, sua largura e as listas de dimensões parametrizadas horizontais e verticais, específicas da seção. Sendo informados esses parâmetros, a função **GeraSecao** cria os pontos de coordenadas tridimensionais que representam os vértices da seção escolhida. Esses pontos serão utilizados pelo método **setSolid** de cada classe de entidades do PREMOLD para gerar uma região plana a partir da qual o sólido da entidade é obtido. A criação do sólido é então feita aplicando-se sobre a região criada o comando EXTRUDE do AutoCAD, ao longo do eixo Z local da entidade, no comprimento definido para a mesma.

A função **GeraSlice** tem a mesma funcionalidade básica da função **GeraSecao**, porém, os pontos criados por ela correspondem aos vértices de uma seção retangular que será utilizada para gerar um sólido que efetuará um corte longitudinal no último painel inserido em uma viga, para que o vão da viga seja totalmente preenchido por um número fracionário de painéis. Esse procedimento será descrito no item 5.2.5.d.

### 5.2.5 Os Comandos Do PREMOLD

Os comandos do PREMOLD podem ser divididos em três grupos segundo suas funcionalidades básicas:

- Comandos que promovem o modelamento geométrico da estrutura e o preenchimento dos bancos de dados do AutoCAD e do PREMOLD:
  - Comando "**GeraMalha**";
  - Comando "**GeraPilar**";
  - Comando "**GeraViga**";
  - Comando "**GeraPainel**"; e
  - Comando "**GeraBloco**".
- Comando que gera, em um arquivo próprio, os relatórios qualitativos e quantitativos sobre a estrutura modelada pelo PREMOLD:
  - Comando "**GeraRelatorio**".

- Comandos que geram os desenhos de detalhamento da estrutura.
  - Comando "**GeraFormato**";
  - Comandos "**GeraLegenda**" e "**EditaLegenda**";
  - Comando "**GeraNota**" e "**EditaNota**";
  - Comando "**GeraDesenho**";
  - Comando "**GeraCarimbo**".

Os comandos **GeraFormato**, **GeraLegenda**, **EditaLegenda**, **GeraNota** e **EditaNota** foram desenvolvidos pelos alunos de iniciação científica do CADTEC sob coordenação das alunas de mestrado Regina Célia Guedes Leite, autora do PREMOLD, e Rosália Gusmão de Lima. Esse desenvolvimento fez parte do treinamento e capacitação desses alunos nas ferramentas AutoCAD, C++ e ObjectARX. Esses comandos são utilizados, não só pelo PREMOLD, mas também pelo aplicativo TowerCAD (MAGALHÃES, 2002). No TowerCAD esses comandos ainda não estavam completamente desenvolvidos e receberam outros nomes como: **CriarDesenho** e **Notas**.

Os comandos **GeraRelatorio** e **GeraDesenho** foram idealizados pela aluna de mestrado Regina Célia Guedes Leite, especificamente para o PREMOLD. Eles foram desenvolvidos, sob a coordenação dessa aluna, pelos alunos de iniciação científica Renata Spyer Las Casas e Marcelo Paixão Pinto Rodrigues, respectivamente.

Na enumeração seguinte encontra-se uma descrição da implementação de todos os comandos utilizados pelo PREMOLD. A descrição da utilização dos comandos é feita no APÊNDICE A onde é apresentado o Manual de Utilização do PREMOLD.

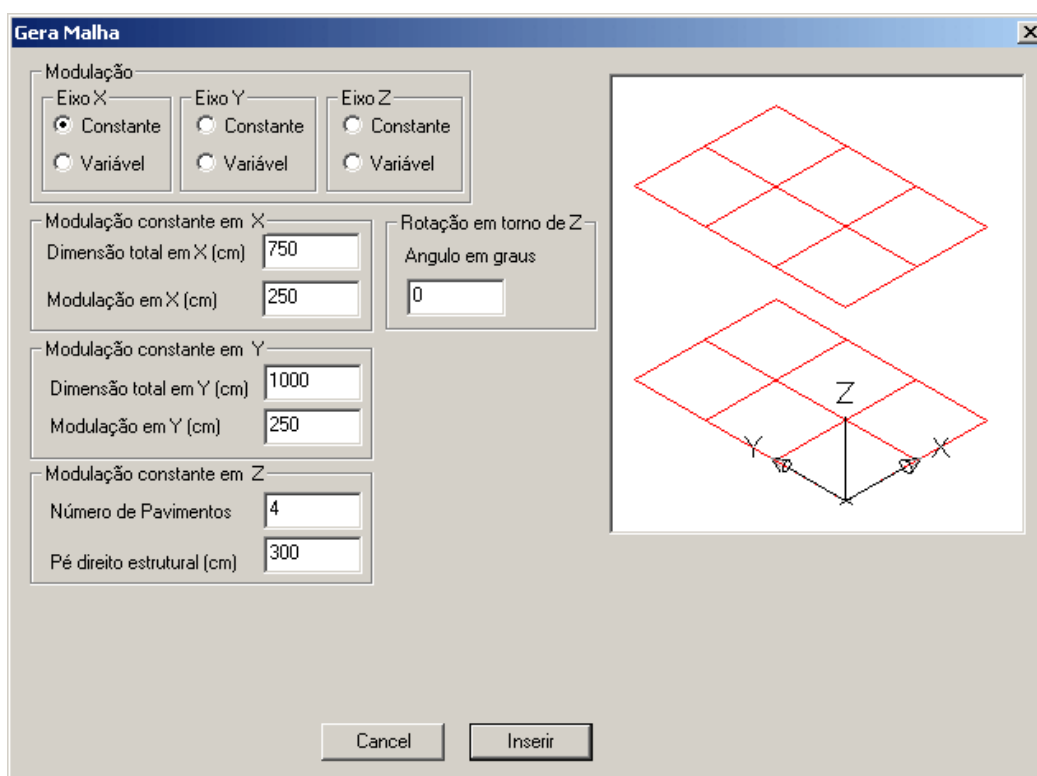
#### **a) Comando "GeraMalha":**

Esse comando foi desenvolvido com a finalidade de facilitar o lançamento de uma estrutura modulada através de uma malha tridimensional, com módulos definidos em três direções ortogonais entre si. O acionamento desse comando permite ao usuário definir uma malha composta por linhas do próprio AutoCAD, ou seja, por entidades nativas do AutoCAD, com espaçamentos regulares nas direções dos eixos X, Y, e Z globais, ou em

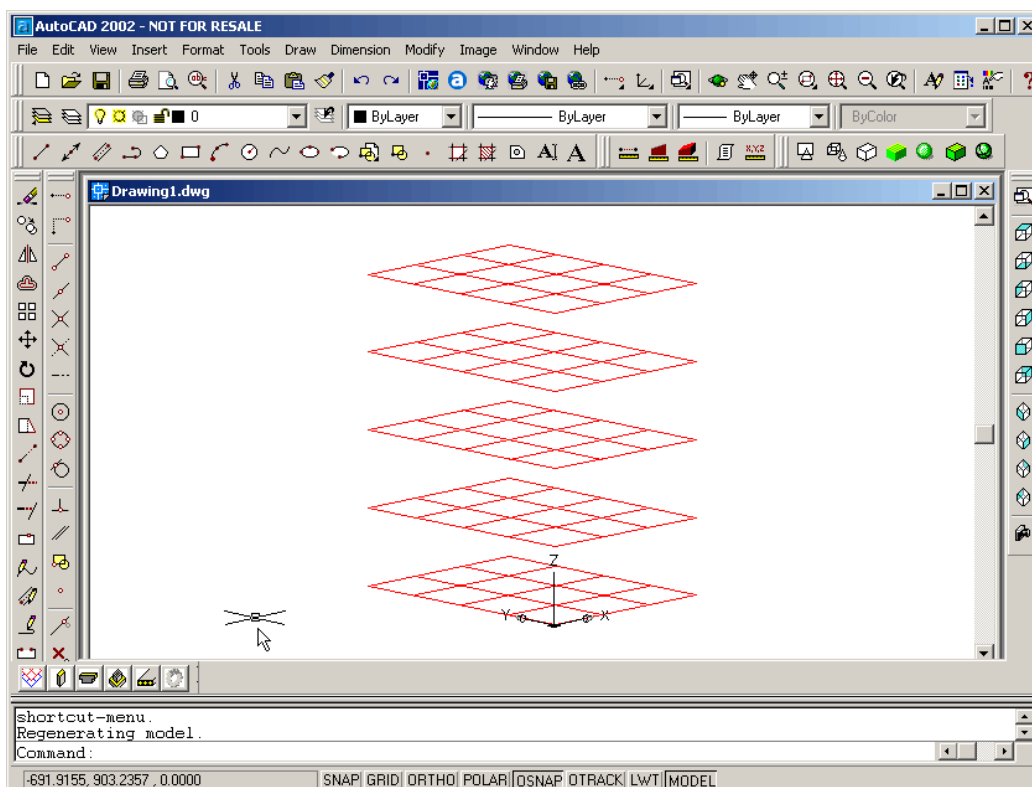
um conjunto de eixos ortogonais entre si e rotacionados de um ângulo qualquer em torno do eixo Z global.

Uma vez que as linhas foram geradas como entidades nativas do AutoCAD, esse comando não trabalha com as classes do PREMOLD.

A partir das dimensões básicas definidas pelo usuário, foi desenvolvido um código recursivo que cria o número de linhas necessárias para formar a malha. As linhas possuem comprimento definido pelo número de divisões em uma determinada direção e pelo módulo naquela direção. É criada automaticamente uma camada de desenho própria para a malha (*layer* MALHA), cujo tipo de linha é definido como contínuo e cuja cor de desenho é definida como vermelho. O ângulo de rotação da malha é tomado a partir da direção positiva do eixo X global em sentido anti-horário.



**FIGURA 5.2** – Quadro de diálogo do comando **GeraMalha**.



**FIGURA 5.3** – Malha gerada com dimensões especificadas segundo a figura 5.2.

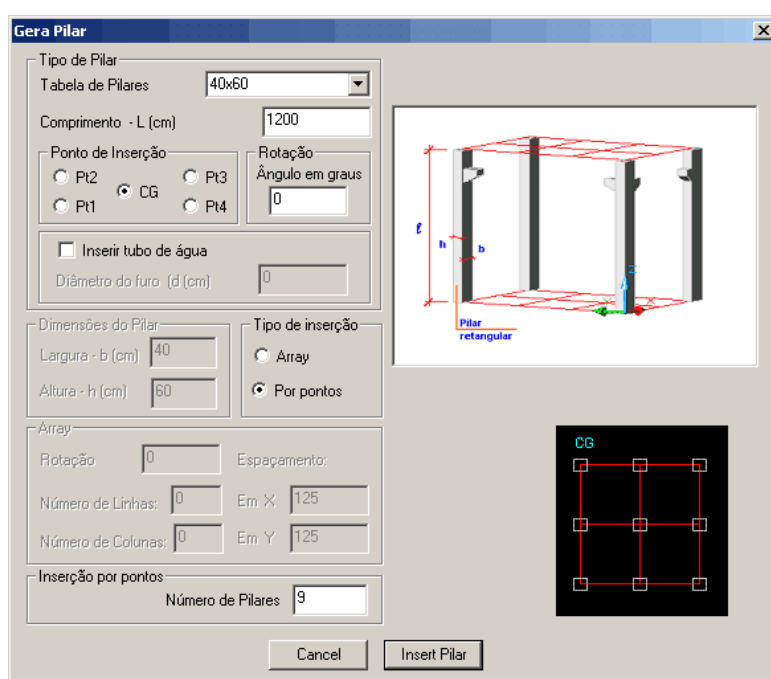
### b) Comando "GeraPilar":

Através de um quadro de diálogo desenvolvido para esse comando, o usuário escolhe as configurações do pilar a ser criado, assim como o número de pilares. Entre os dados a serem informados pelo usuário estão: a seção do pilar, o comprimento do pilar, o ângulo de rotação com o eixo X global (o sentido anti-horário foi definido como positivo para esse ângulo), o número de pilares, a existência e o diâmetro do tubo de descida de água, o modo de inserção do pilar e o ponto de inserção do pilar.

O pilar pode ser inserido através de cinco pontos de sua base: pelo centro de gravidade de sua seção transversal ou por qualquer um de seus quatro vértices. Se forem inseridos mais de um pilar o usuário pode escolher entre inserir os pilares um a um, escolhendo as coordenadas de inserção na tela do AutoCAD ou digitando as mesmas através da linha de comando, ou inseri-los de uma só vez através de uma matriz formada por "m" linhas e "n" colunas. Nesse caso, os pilares serão inseridos com espaçamentos constantes entre linhas e

colunas. O número de linhas e colunas, assim como o número de pilares é definido pelo usuário.

Após o preenchimento do quadro de diálogo pelo usuário (FIG. 5.4), o aplicativo, através do comando **GeraPilar**, procede à leitura dos dados informados e dos dados da seção escolhida armazenados no banco de dados do aplicativo.



**FIGURA 5.4** – Quadro de diálogo do comando **GeraPilar** para inserção de nove pilares pelo seu centro de gravidade, com seção padrão de 40cmx60cm e altura de 12m.

Após a leitura dos dados, é criada uma entidade da classe **CdTcCPilar**. Os atributos dessa entidade (como parâmetros da seção transversal, altura, ponto de inserção, ângulo de rotação, tubo de descida de água, etc.) são então preenchidos à partir dos dados colhidos, ou da manipulação desses pelo próprio aplicativo. Para o preenchimento dos atributos da entidade são utilizados os métodos de atribuição definidos no item 5.2.2.f.

Após a configuração da entidade é acionado o método **setSolid** específico da classe **CdTcCPilar**. Através desse método o sólido de representação do pilar é modelado. As entidades **Bar** e **Node**, que definem a entidade **Pilar** no modelo estrutural, são configuradas durante a modelagem do sólido gerado. Os dados para essas entidades são

obtidos a partir dos dados geométricos estabelecidos para o pilar e pela numeração automática realizada pela classe **CdTcCDictEnum**. As coordenadas do nó inicial são definidas pelas coordenadas do centro geométrico da seção transversal do pilar sobre o plano horizontal que passa por seu ponto de inserção. As coordenadas do nó final são definidas pela soma do comprimento do pilar, multiplicado por seu eixo Z, às coordenadas do nó inicial. A barra é definida como o seguimento de reta que liga os nós inicial e final do pilar. As restrições dos nós da barra são lançadas automaticamente conforme o sistema estrutural predefinido e o tipo de vinculação adotada entre os elementos. Quando um pilar é lançado, o PREMOLD define o mesmo engastado na base e livre no topo. Assim, o nó inicial recebe a codificação de restrição ao deslocamento, indicado pelo símbolo “-“, e à rotação nas três direções correspondentes aos eixos locais. O nó final recebe a codificação referente à liberdade de deslocamento e de rotação, indicada pelo símbolo “+“, também nas direções dos eixos locais. A FIG. 5.5 indica as restrições de pilares inseridos num desenho. Para esta etapa de trabalho, apenas este tipo de vinculação foi definido para um pilar sem elementos nele apoiados. As vinculações do nó final serão alteradas automaticamente ao serem inseridas vigas em cota igual à cota desse nó. Essa alteração é realizada durante a execução do comando **GeraViga**.

São, então, criados objetos das classes **CdTcCDictNode**, **CdTcCDictBar** e **CdTcCDictElement** para armazenar os dados básicos das entidades criadas (**Node**, **Bar** e **Pilar** respectivamente) nos dicionários próprios do PREMOLD.

Concluída a modelagem da entidade **Pilar**, essa é inserida no banco de dados do AutoCAD e sua representação é mostrada na tela gráfica (FIG.5.6).

Relatorio - Notepad

File Edit Format Help

\*\*\*\*\*  
 RELATÓRIO GEOMÉTRICO  
 \*\*\*\*\*

NÓS

ID	Coordenadas			Restrições					
	X	Y	Z	Rx	Ry	Rz	Tx	Ty	Tz
1	5.33	376.04	0.00	-	-	-	-	-	-
2	5.33	376.04	300.00	+	+	+	+	+	+
3	255.33	376.04	0.00	-	-	-	-	-	-
4	255.33	376.04	300.00	+	+	+	+	+	+
5	255.33	1.04	0.00	-	-	-	-	-	-
6	255.33	1.04	300.00	+	+	+	+	+	+
7	5.33	1.04	0.00	-	-	-	-	-	-
8	5.33	1.04	300.00	+	+	+	+	+	+

BARRAS

ID	Tipo	Seção	Nó inicial	Nó final	Comprimento
1	PILAR	20 X 20	1	2	300.00
2	PILAR	20 X 20	3	4	300.00
3	PILAR	20 X 20	5	6	300.00
4	PILAR	20 X 20	7	8	300.00

ID	Área	Volume	Momentos de Inércia		
			Ix	Iy	Iz
1	400.00	120000	+5.33e+004	+1.7e+010	+0
2	400.00	120000	+5.33e+004	+2.48e+010	+0
3	400.00	120000	+5.33e+004	+7.83e+009	+0
4	400.00	120000	+5.33e+004	+1.15e+007	+0

\*\*\*\*\*

FIGURA 5.5 – Relatório com indicação das restrições nodais de quatro pilares criados pelo comando **GeraPilar**.

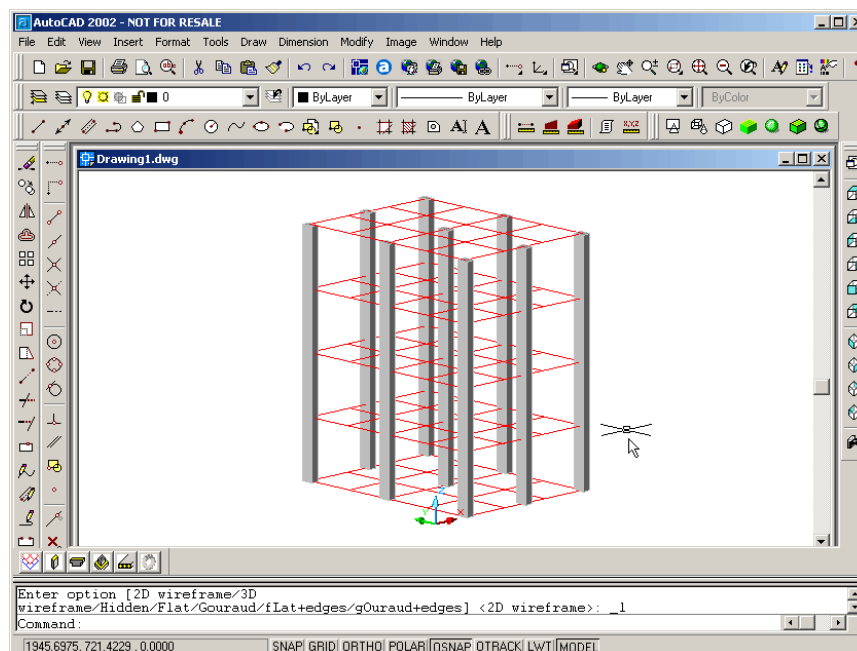


FIGURA 5.6 – Pilares definidos como mostrado na FIG. 5.4, inseridos no banco de dados do AutoCAD e do PREMOLD.

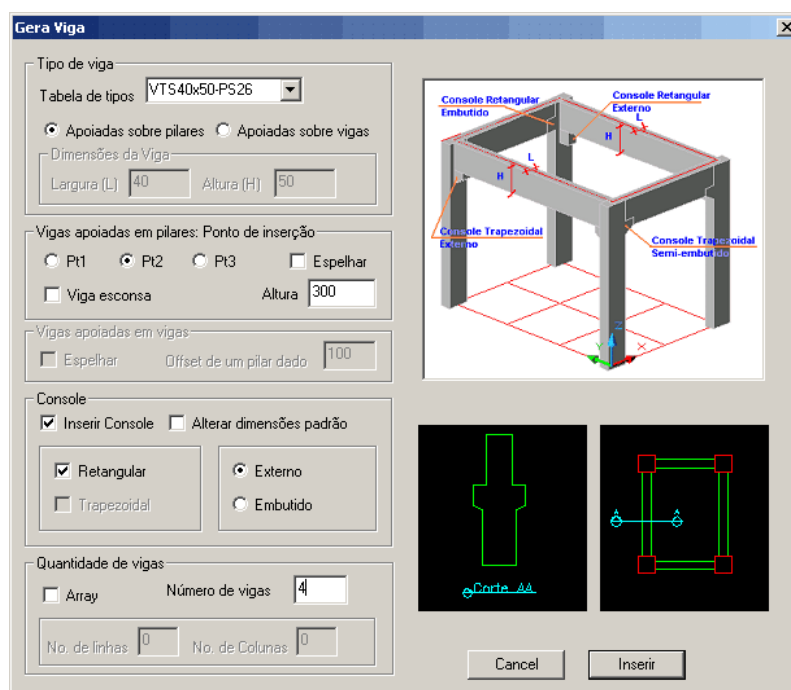
### c) Comando "**GeraViga**":

Os passos iniciais para a definição dos dados da viga a ser gerada são semelhantes aos descritos para a criação dos pilares. Através do quadro de diálogo criado para o comando **GeraViga** o usuário escolhe a seção da viga a ser criada, sua altura de inserção, o tipo de viga (se principal ou secundária), o tipo de consolo a ser inserido nos pilares de base no caso de viga principal, o número de vigas e seu modo de inserção.

As vigas podem ser inseridas faceando o pilar em duas de suas laterais ou centradas em relação ao pilar. Caso existam consolos, esses podem ser retangulares ou trapezoidais, externos, embutidos ou semi-embutidos (esse último apenas no caso de consolos trapezoidais). As dimensões dos consolos, assim como as folgas entre a viga e seus elementos de apoio, são definidas através de uma configuração geral do sistema, mas podem ser alteradas pelo usuário.

Para a inserção das vigas não foi desenvolvida, nesta etapa de trabalho, a inserção através de matriz como foi realizado para os pilares. A inserção de vigas deve ser feita individualmente quando são escolhidos pelo usuário os elementos de apoio para a viga a ser criada.

Após o preenchimento do quadro de diálogo pelo usuário (fig. 5.7), o aplicativo, através do comando **GeraViga**, procede à leitura dos dados informados e dos dados da seção escolhida armazenados no seu banco de dados.



**FIGURA 5.7** – Quadro de diálogo do comando **GeraViga** para inserção de vigas do tipo VTS sobre os pilares internos da estrutura da FIG. 5.5.

Após a leitura dos dados é criada uma entidade da classe **CdTcCViga**. Os elementos que servirão de base para a viga são abertos para a leitura e seus dados geométricos são obtidos: coordenadas de seu centro geométrico, altura, largura, parâmetros da seção transversal, ponto de inserção, eixos locais, etc. Esses dados são obtidos através de seus métodos de acesso, descritos nos itens 5.2.2.f e 5.2.2.g, sendo utilizados para realizar-se o correto posicionamento da viga em relação aos elementos de apoio e também para configurar seus eixos locais e seu comprimento.

A posição da viga é, então, calculada automaticamente pelo aplicativo a partir dos elementos de apoio indicados pelo usuário. Entre os elementos escolhidos, o elemento de apoio inicial e o elemento de apoio final são definidos segundo parâmetros estabelecidos no código fonte do PREMOLD. Essa definição não depende da ordem de escolha desses elementos pelo usuário. O elemento de apoio inicial será sempre aquele que possui as menores coordenadas globais para seu ponto de inserção. Os demais atributos da entidade **Viga** (como parâmetros da seção transversal, altura etc.) são então preenchidos a partir dos dados colhidos do quadro de diálogo ou da manipulação desses por funções utilitárias. Para

o preenchimento dos atributos da entidade **Viga** são utilizados os métodos de atribuição definidos no item 5.2.2.g.

São criadas e configuradas duas entidades da classe **CdTcCConsolo**. Essas entidades são repassadas tanto para os elementos de apoio da viga, quanto para a própria viga a ser criada. Elas formarão os consolos de apoio da viga e definirão sua seção na região do apoio (a existência ou não dos dentes de Gerber). Os atributos dessas entidades são preenchidos com os dados obtidos do quadro de diálogo do comando **GeraViga** e pelos dados obtidos dos elementos de apoio selecionados. Para o preenchimento dos atributos da entidade **Consolo** são utilizados os métodos de atribuição definidos no item 5.2.2.e.

Após a configuração da entidade, é acionado o método **setSolid** da classe **CdTcCViga** e, através desse, a representação da viga é modelada. A configuração das entidades **Bar** e **Node**, que definem a entidade **Viga** no modelo estrutural, são configuradas do mesmo modo realizado para a entidade **Pilar**. As coordenadas do nó inicial da entidade **Viga** são definidas pelas coordenadas de um ponto obtido pela projeção do centro geométrico da seção transversal da viga sobre a entidade **Bar** do elemento de apoio definido como inicial. As coordenadas do nó final são definidas pelas coordenadas de um ponto obtido pela projeção do centro geométrico da seção transversal da entidade **Viga** sobre a entidade **Bar** do elemento de apoio definido como final.

A criação dos objetos das classes **CdTcCDictNode**, **CdTcCDictBar** e **CdTcCDictElement** é feita de modo análogo ao realizado no comando **GeraPilar**.

As entidades de apoio são remodeladas com as alterações necessárias para a inserção da viga. Os pilares de apoio recebem os consolos para suporte da viga inserida e as vigas principais, caso necessário, também recebem consolos. Os consolos dos elementos de apoio são modelados pelos dados repassados pelas entidades criadas do tipo **Consolo**. Os métodos **setSolid** das entidades de apoio são, então, acionados. Caso hajam consolos a serem inseridos nas entidades de apoio, os sólidos formados por aqueles são adicionados a essas entidades, passando a fazer parte das mesmas. Assim, não existem entidades **Consolo** isoladas no modelo criado pelo PREMOLD, essas sempre estarão inseridas em outras entidades como pilares ou vigas. Uma vez que as entidades de apoio já haviam sido criadas

anteriormente, com barras e nós predefinidos, o seu método **setSolid** aciona, então, apenas a modelagem geométrica do sólido representativo da entidade, não alterando os dados de suas entidades constituintes (**Bar** e **Node**). Apenas os dados relativos às vinculações do nó final do pilar podem ser alterados caso a cota de inserção da viga seja igual à cota do nó final do pilar. Nesse caso, o PREMOLD considera a ligação da viga com o pilar como uma ligação articulada onde as rotações da viga em torno do eixo longitudinal dessa são impedidas, mas as rotações da viga em torno do eixo perpendicular ao anterior são permitidas. As rotações do nó final do pilar não são impedidas, porém o deslocamento do pilar, na direção do comprimento da viga, fica impedido (FIG. 5.8). Após a remodelagem, as entidades de apoio são inseridas novamente no banco de dados do AutoCAD com sua representação gráfica atualizada.

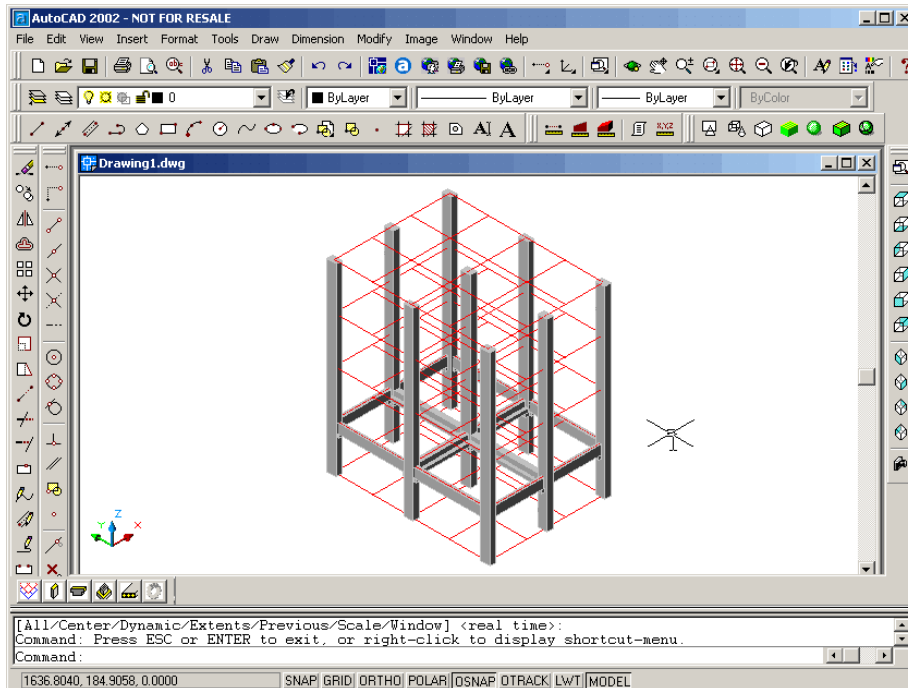
```

*****
                          RELATÓRIO GEOMÉTRICO
*****
NÓS
*****
ID      Coordenadas      Restrições
-----
      X      Y      Z      Rx  Ry  Rz  Tx  Ty  Tz
-----
1       5.90   376.52  0.00  -   -   -   -   -   -
10      5.90   365.52  275.00 -   -   +   -   +   -
11      16.90   1.52   275.00 -   -   +   +   -   -
12      244.90  1.52   275.00 -   -   +   +   -   -
13      255.90  12.52  275.00 -   -   +   -   +   -
14      255.90  365.52  275.00 -   -   +   -   +   -
15      16.90   376.52  275.00 -   -   +   +   -   -
16      244.90  376.52  275.00 -   -   +   +   -   -
2       5.90   376.52  300.00 +   +   +   -   -   +
3      255.90  376.52  0.00   -   -   -   -   -   -
4      255.90  376.52  300.00 +   +   +   -   -   +
5      255.90  1.52   0.00   -   -   -   -   -   -
6      255.90  1.52   300.00 +   +   +   -   -   +
7       5.90   1.52   0.00   -   -   -   -   -   -
8       5.90   1.52   300.00 +   +   +   -   -   +
9       5.90   12.52  275.00 -   -   +   -   +   -
*****
BARRAS
*****
ID  Tipo      seção      Nó inicial  Nó final  Comprimento
-----
1   PILAR     20 x 20    1           2         300.00
2   PILAR     20 x 20    3           4         300.00
3   PILAR     20 x 20    5           6         300.00
4   PILAR     20 x 20    7           8         300.00
5   VIGA     VR10x50    9           10        353.00
6   VIGA     VR10x50    11          12        228.00
7   VIGA     VR10x50    13          14        353.00

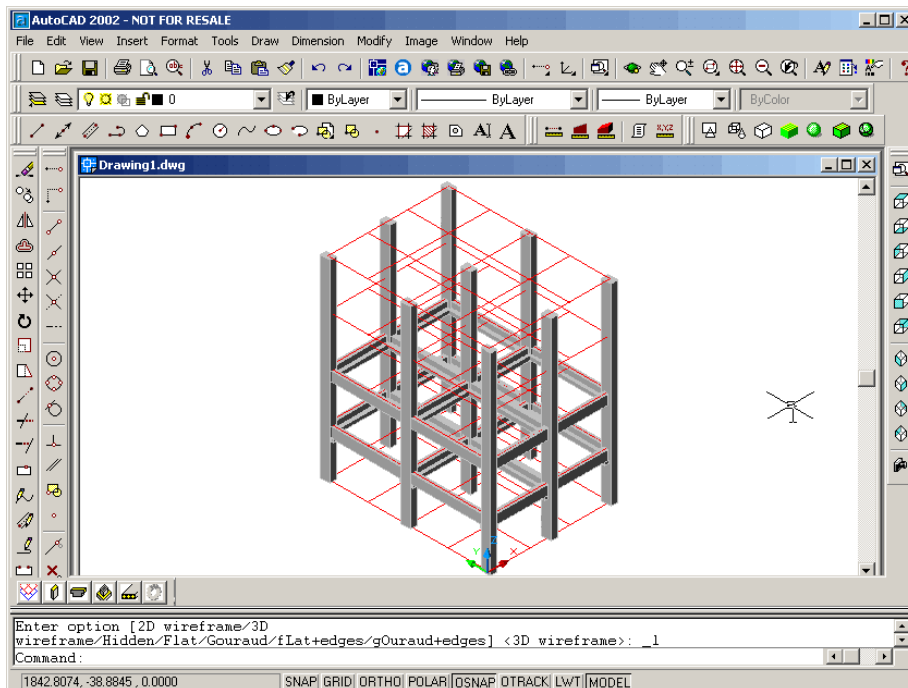
```

**FIGURA 5.8** – Relatório com dados sobre as restrições nodais de nós inicial e final de pilares e vigas criados pelo PREMOLD.

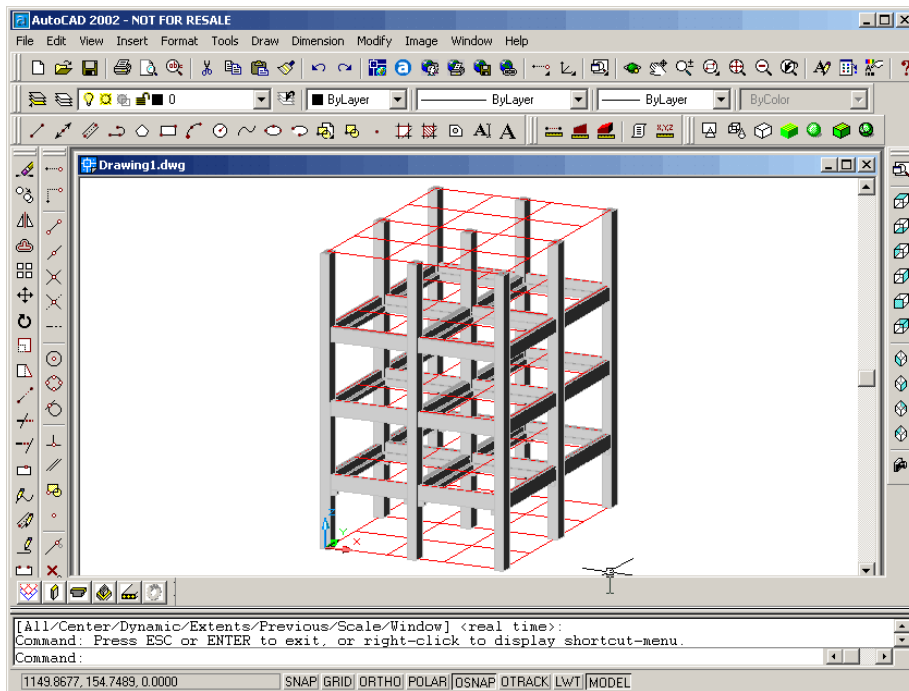
Concluída a modelagem da entidade **Viga**, essa é também inserida no banco de dados do AutoCAD e sua representação é mostrada na tela gráfica (FIG. 5.9 a FIG. 5.12).



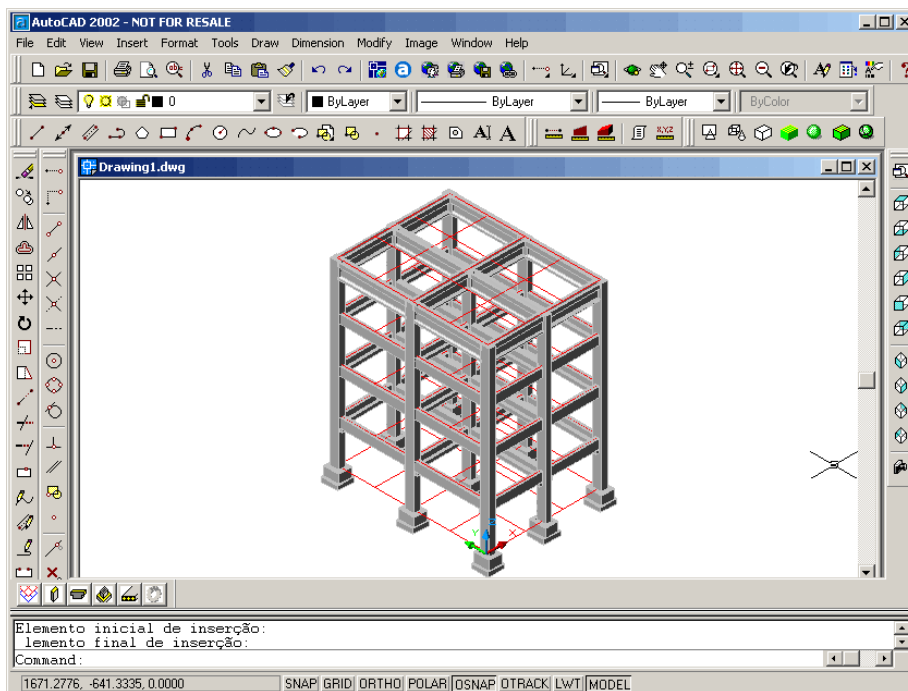
**FIGURA 5.9** – Vigas inseridas no primeiro pavimento da estrutura da FIG. 5.6.



**FIGURA 5.10** - Vigas inseridas no segundo pavimento da estrutura mostrada na FIG. 5.9.



**FIGURA 5.11 - Vigas inseridas no terceiro pavimento da estrutura mostrada na FIG. 5.10.**



**FIGURA 5.12 – Inserção de vigas de fechamento da estrutura da Fig. 5.11.**

#### d) Comando "GeraPainel":

O comando **GeraPainel** muito se assemelha ao comando **GeraViga** na manipulação de elementos de apoio para a configuração de alguns atributos da entidade **Painel** criada. Porém nesse comando não há a criação de entidades **Consolo** e os elementos de apoio são sempre entidades da classe **CdTcCViga**.

A entidade **Painel** será apoiada diretamente sobre as entidades **Viga**, e seu ponto de inserção dependerá apenas da seção das entidades de apoio, sendo predefinido para cada uma das seções do banco de dados do PREMOLD. Algumas vigas possuem uma seção transversal com formato próprio para receber determinado tipo de painel de piso. Caso a seção do painel escolhido não seja aquela para a qual a viga foi desenvolvida, o aplicativo emite um alerta ao usuário de que o painel escolhido não é usualmente utilizado com o tipo da viga escolhida como elemento de apoio. O painel escolhido é inserido e posicionado automaticamente sobre a viga na posição em que essa deve receber seu painel de piso específico. Para vigas retangulares a posição de inserção dos painéis foi definida como superior à viga, faceando seu lado externo. Entende-se por lado externo da viga aquele oposto ao lado próximo à viga de apoio final do painel.

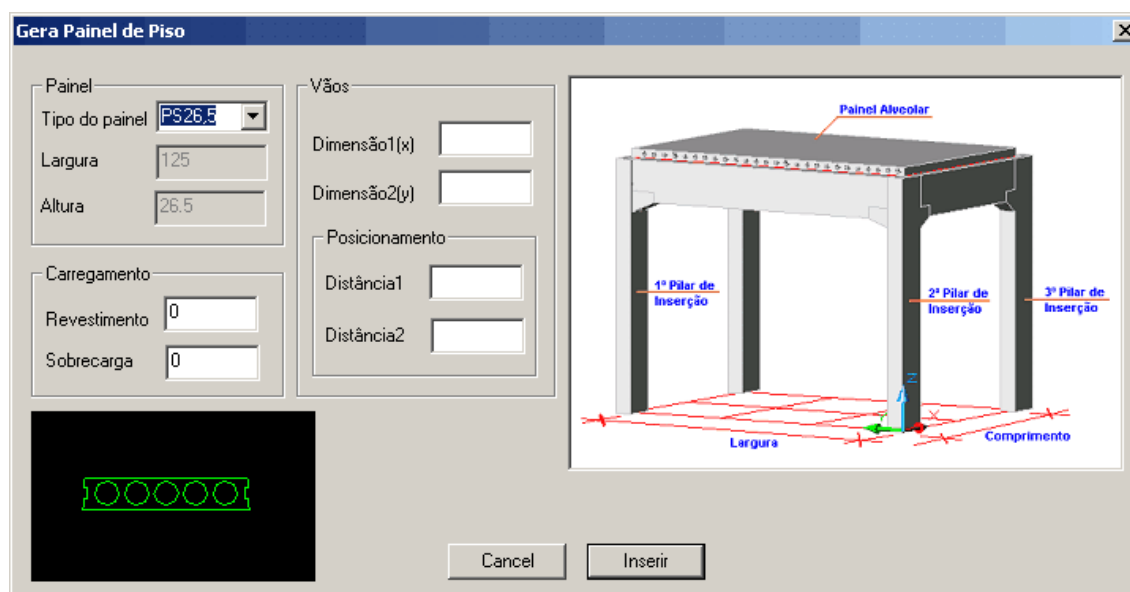
A inserção dos painéis pode ser feita dos seguintes modos: inserção de um único painel, inserção de um número inteiro de painéis contidos no comprimento da viga de apoio mais curta, ou ainda, em número fracionário de painéis. No último caso, o comprimento da menor viga é preenchido pelo número de painéis inteiros por ele contidos e o último painel é cortado no sentido longitudinal, para completar o vão, no comprimento determinado pela menor viga. Esse modo de inserção é escolhido pelo usuário. O aplicativo realiza automaticamente todos os cálculos e procedimentos necessários para preencher o vão conforme a escolha do usuário.

Conforme foi mostrado no item 4.1 a entidade **Face** original do Modelador 3D modela uma superfície plana horizontal delimitada por barras. O parágrafo anterior mostra que um vão entre duas vigas pode ser preenchido parcialmente ou completamente por painéis de piso, e a superfície formada por tais painéis necessita apenas de duas barras de apoio para ser formada. Assim, os painéis de piso de concreto pré-moldado podem ser representados

por uma superfície plana horizontal, mas não necessitam de barras em todo seu contorno para serem definidos. Apenas os vértices da projeção horizontal do painel são necessários para definir uma superfície horizontal que o represente no modelo estrutural. Assim, para melhor representar o painel de concreto pré-moldado, foi realizada a reestruturação da classe original **Face** para a classe **CdTcCPlaca**, apresentada no capítulo 4. A classe **CdTcCPlaca** passa, então, a fazer parte do painel de piso modelado pelo PREMOLD.

Para o comando **GeraPainel** também foi desenvolvido um quadro de diálogo através do qual o usuário escolhe a seção do painel a ser inserido (FIG. 5.13). O tipo de inserção é realizado através das linhas de comando do AutoCAD.

Após a escolha do tipo de seção do painel, o usuário deve escolher as vigas que lhe servirão de apoio. Foi desenvolvida a inserção de painéis de piso para preenchimento de apenas um vão a cada acionamento do comando **GeraPainel**.



**FIGURA 5.13** – Quadro de diálogo do comando **GeraPainel**.

Após as escolhas feitas pelo usuário, é criada, automaticamente, uma entidade da classe **CdTcCPainel** e os parâmetros da seção transversal escolhida são obtidos do banco de dados do PREMOLD. As vigas que servirão de base para o painel são abertas para a leitura de seus dados geométricos como: coordenadas de seu centro geométrico, altura, largura e parâmetros da seção transversal, ponto de inserção, eixos locais, etc., são obtidos através

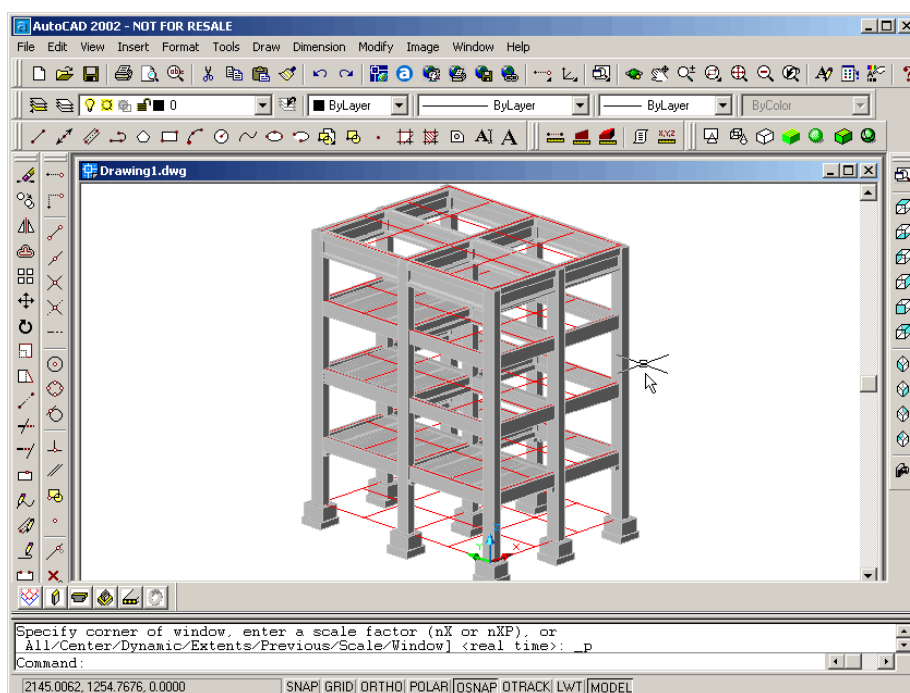
de seus métodos de acesso, descritos nos itens 5.2.2 g. Esses dados são utilizados para obter-se o correto posicionamento do painel em relação às vigas de apoio e também para configurar seus eixos locais e seu comprimento.

A posição do painel é calculada automaticamente pelo aplicativo a partir dos elementos de apoio indicados pelo usuário. Também aqui o elemento de apoio inicial e o elemento de apoio final são definidos segundo parâmetros estabelecidos no código fonte do PREMOLD, não dependendo da ordem de escolha desses elementos pelo usuário. O elemento de apoio inicial será aquele que possui as menores coordenadas globais para seu ponto de inserção. Os atributos da entidade **Painel** são preenchidos a partir dos dados diretamente colhidos do quadro de diálogo ou através da manipulação desses dados.

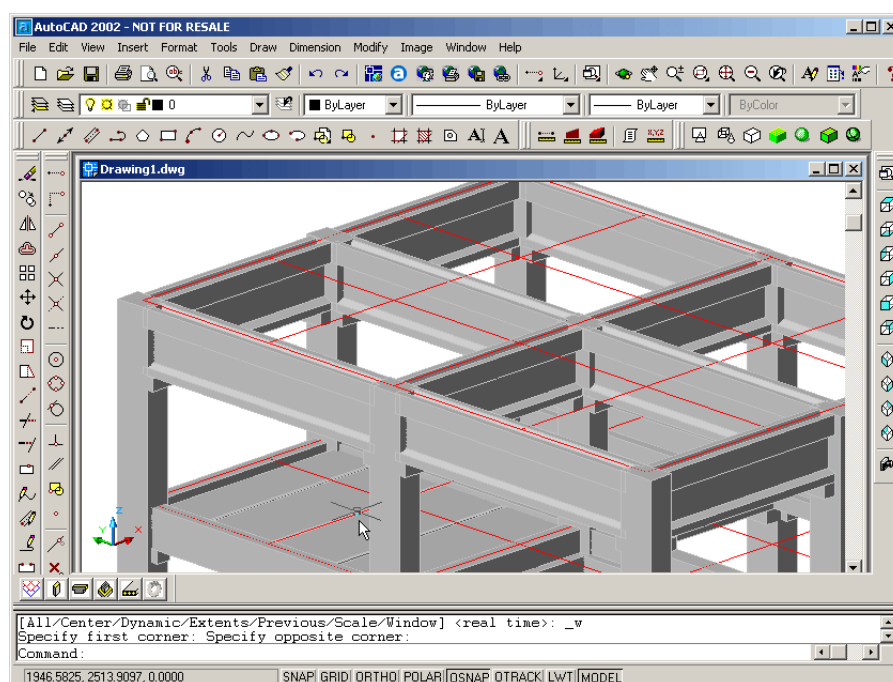
A entidade **Painel** possui uma entidade **Placa** para sua representação no modelo estrutural. A configuração da entidade **Placa** é realizada durante a modelagem do sólido gerado, através da função **setSolid** da classe **CdTcCPainel**. Os dados para essa entidade são obtidos a partir dos dados geométricos estabelecidos para o painel e pela numeração automática realizada pela classe **CdTcCDictEnum**. O vértice inicial é definido pelo vértice do painel de menores coordenadas globais, esse coincide com o ponto de inserção da viga tomada como elemento de apoio inicial. As coordenadas do segundo vértice são definidas como as do vértice inicial somas à largura do painel multiplicada pelo eixo Z local da viga de apoio inicial. O vértice final coincide com o ponto de inserção da viga tomada como elemento de apoio final. As coordenadas do terceiro vértice são definidas como as do vértice final somadas à largura do painel multiplicada pelo eixo Z local da viga de apoio final.

É, então, criado um objeto da classe **CdTcCDictElement** para armazenar os dados básicos da entidade **Painel** criada no dicionário **ElementDict** do PREMOLD.

Concluída a modelagem da entidade **Painel**, essa é inserida no banco de dados do AutoCAD e sua representação é mostrada na tela gráfica (FIG. 5.14 a 5.17).



**FIGURA 5.14** – Inserção de painéis do tipo alveolar no primeiro, segundo e terceiro pavimentos da estrutura mostrada na FIG. 5.13.



**FIGURA 5.15** – Detalhe do encaixe de vigas com pilares e do apoio de painéis sobre vigas da estrutura da FIG. 5.14.

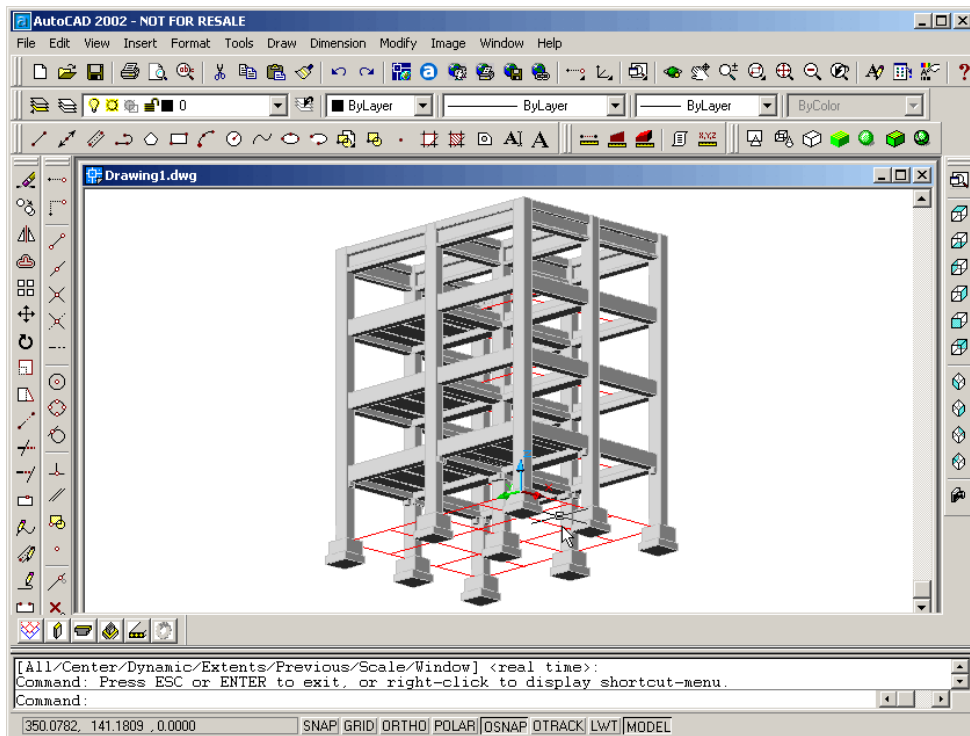


FIGURA 5.16 – Outro ângulo de visão da estrutura da FIG. 5.14.

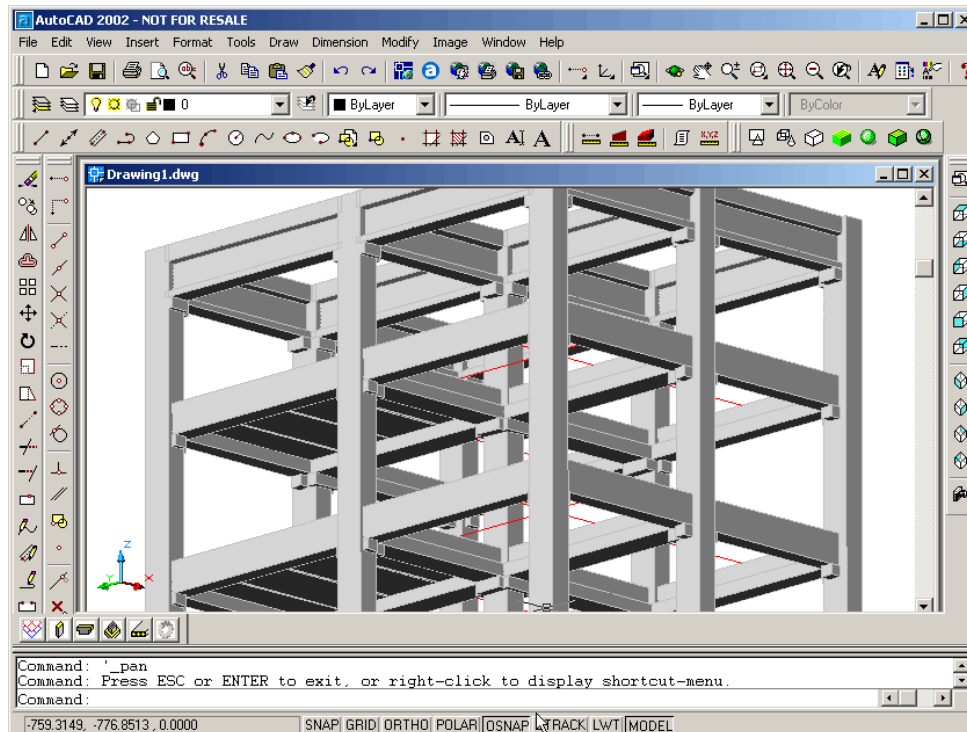


FIGURA 5.17 – Vista mais aproximada da estrutura mostrada na FIG.5.16.

### e) Comando "GeraBloco":

Através de um quadro de diálogo desenvolvido para esse comando (FIG. 5.18), o usuário escolhe as configurações mínimas do bloco de fundação a ser criado, assim como o número de blocos. Entre os dados a serem informados pelo usuário estão: espessura do colarinho, as dimensões da base do bloco de fundação, altura do colarinho, folga entre o pilar e o colarinho e engaste através de superfície lisa ou rugosa.

O bloco de fundação é inserido sob pilares preexistentes no banco de dados. Seu ponto de inserção coincide com o centro geométrico da base do pilar. Os blocos são inseridos individualmente, após terem sido escolhidos, na tela gráfica, os pilares aos quais deseja-se anexar os blocos de fundação. O comando operará de modo iterativo permitindo que o usuário escolha um número de pilares igual ao número de blocos a serem inseridos.

Após o preenchimento do quadro de diálogo pelo usuário, o aplicativo, através do comando **GeraBloco**, procede à leitura dos dados informados. A entidade **Pilar**, para a qual será criado um bloco de fundação, é aberta para leitura e seus dados geométricos (como parâmetros da seção transversal, ponto de inserção e ângulo de rotação, etc.) são então obtidos através dos métodos de acesso descritos no item 5.2.2.f. Para o dimensionamento do bloco de fundação são consideradas as prescrições geométricas mínimas estabelecidas pela NBR 9062/85 (ABNT, 1985) para blocos de fundação com colarinho. Caso o usuário tenha definido uma dimensão menor que aquela estabelecida pela citada norma, o usuário é informado da desconformidade do dado fornecido, devendo alterar o mesmo para um valor igual ao maior ao valor mínimo estabelecido por norma.

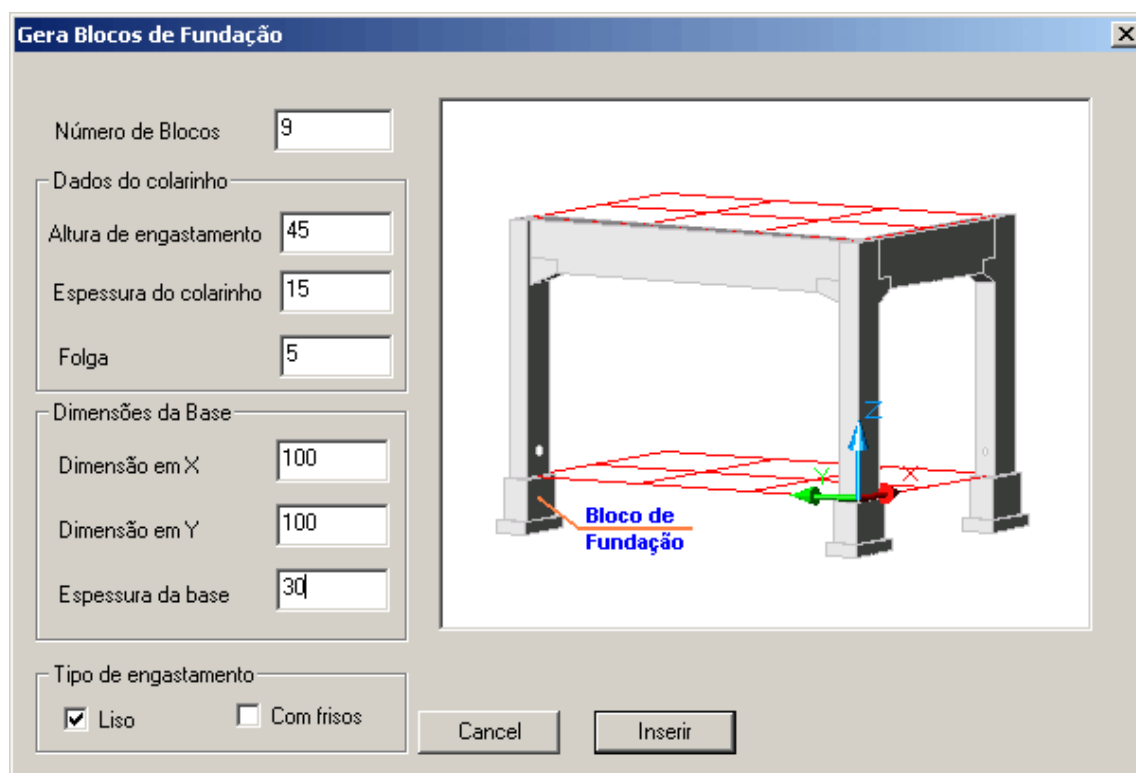
Para o preenchimento dos atributos da entidade são utilizados os métodos de atribuição definidos no item 5.2.2.d.

Após a configuração da entidade é acionado o método **setSolid** específico da classe **CdTcCBloco**. Através desse método o sólido de representação do bloco é modelado. Como foi mencionado no item 5.2.1, nesta etapa de desenvolvimento do PREMOLD, não foram criados, para o bloco de fundação, elementos que o representem no modelo

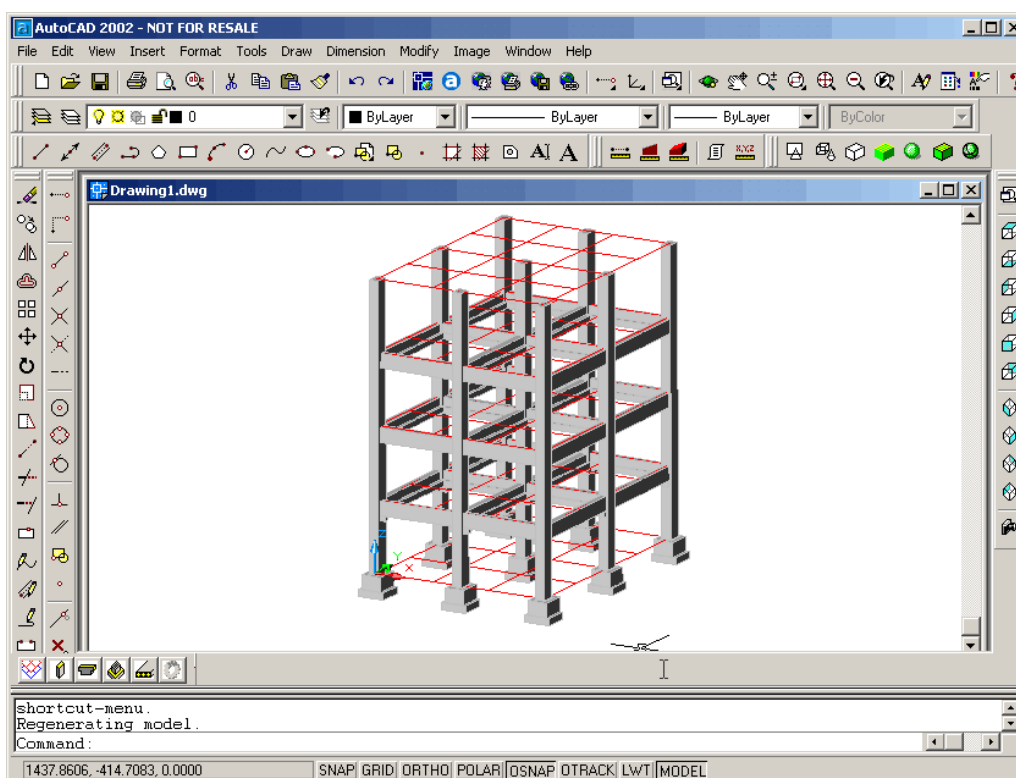
estrutural e, portanto, o bloco não recebe numeração e não é inserido nos dicionários do PREMOLD.

Durante a criação do bloco de fundação, o pilar no qual o bloco é inserido tem seu comprimento aumentado do comprimento de engaste definido para o bloco de fundação. Para redefinir a representação gráfica do pilar, esse é remodelado através de sua função **setSolid**, sendo inserido novamente no banco de dados do AutoCAD.

Concluída a modelagem da entidade **Bloco**, essa é inserida no banco de dados do AutoCAD e sua representação é mostrada na tela gráfica (FIG. 5.19).



**FIGURA 5.18** – Quadro de diálogo do comando **GeraBloco** para inserção de nove blocos sob os pilares da estrutura mostrada na FIG. 5.11.



**FIGURA 5.19** – Blocos de fundação inseridos sob os pilares da Estrutura da FIG. 5.11.

#### f) Comando "GeraRelatorio":

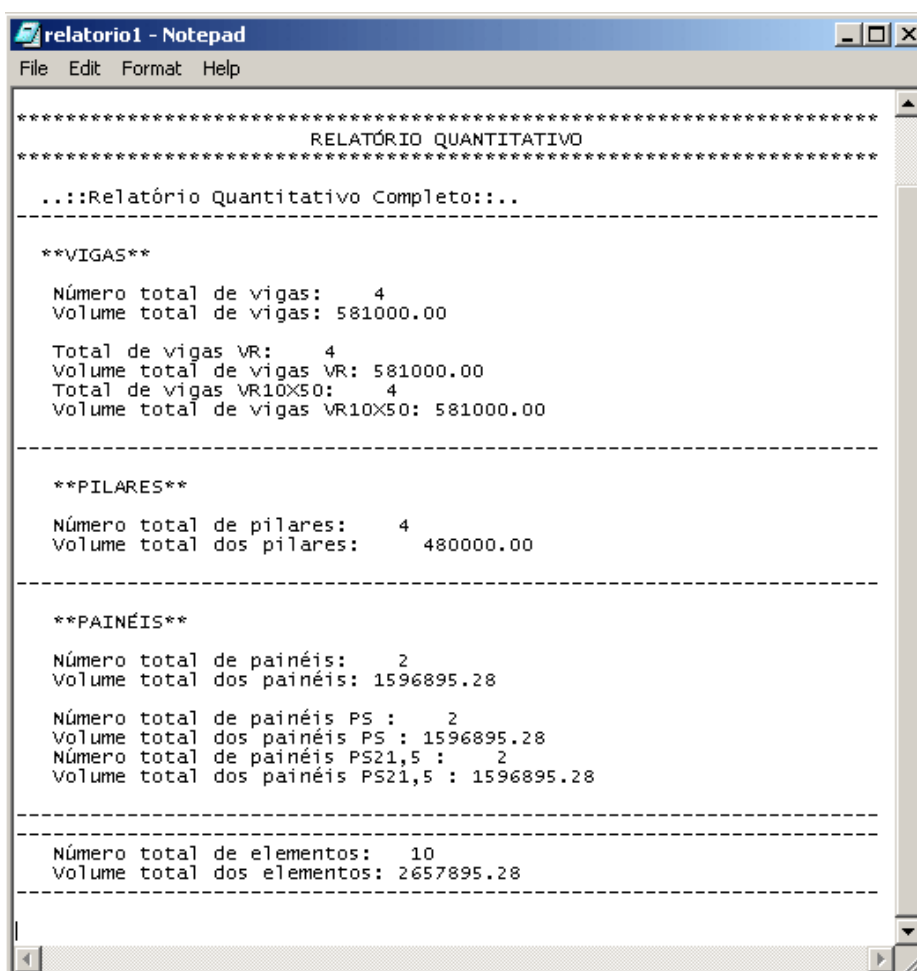
O comando **GeraRelatorio** foi desenvolvido para percorrer todo o banco de dados do PREMOLD e do AutoCAD, mapeando as entidades do PREMOLD neles inseridas, listando, em um arquivo de texto, informações qualitativas e quantitativas sobre essas entidades.

Após a escolha pelo usuário do nome do arquivo no qual as informações serão gravadas e o tipo de relatório a ser gerado, o comando **GeraRelatorio** percorre os dicionários de entidades do PREMOLD para colher os números de identificação das entidades ali referenciadas e os demais dados ali contidos. De posse do número de identificação da entidade ou das coordenadas de pontos pertencentes a essas entidades, o comando aciona as funções **verifyNode**, **verifyBar**, **verifyPlaca**, **verifyPainel** e **verifyElement** para encontrar uma entidade das classes **CdTcCNode**, **CdTcCBar**, **CdTcCPlaca**,

**CdTcCPainel**, **CdTcCPilar** ou **CdTcCViga** que correspondam às entidades cujos dados foram armazenados nos dicionários do PREMOLD. Caso a entidade seja encontrada, a busca é interrompida e o identificador da entidade (seu *Object ID*) é informado pela função acionada. De posse do identificador de uma entidade, essa pode ser aberta para leitura e seus atributos podem ser obtidos através de seus métodos de acesso.

Podem ser gerados alguns tipos de relatórios através do comando **GeraRelatorio**:

- Relatórios: G/Q, Geométrico (FIG. 5.5 e FIG. 5.8) ou Quantitativo (FIG. 5.20);
- Relatórios: Parcial ou Total.



```
relatorio1 - Notepad
File Edit Format Help
*****
                        RELATÓRIO QUANTITATIVO
*****
...:Relatório Quantitativo Completo:...
-----
**VIGAS**

Número total de vigas:      4
Volume total de vigas: 581000.00

Total de vigas VR:      4
Volume total de vigas VR: 581000.00
Total de vigas VR10X50:  4
Volume total de vigas VR10X50: 581000.00
-----
**PILARES**

Número total de pilares:    4
Volume total dos pilares:  480000.00
-----
**PAINÉIS**

Número total de painéis:    2
Volume total dos painéis: 1596895.28

Número total de painéis PS :  2
Volume total dos painéis PS : 1596895.28
Número total de painéis PS21,5 :  2
Volume total dos painéis PS21,5 : 1596895.28
-----
Número total de elementos:  10
Volume total dos elementos: 2657895.28
-----
```

**FIGURA 5.20** – Relatório Quantitativo Total gerado pelo PREMOLD.

O Relatório Geométrico informa apenas o tipo dos elementos gerados na estrutura e os dados geométricos desses elementos. O Relatório Quantitativo informa o tipo dos elementos, as quantidades desses elementos, o volume por tipo de elemento e volume total

da estrutura. O Relatório G/Q informa os dados geométricos e os dados quantitativos dos elementos.

Os relatórios: G/Q, Geométrico e relatório Quantitativo, podem ser gerados de forma parcial (Relatório Parcial) quando apenas um determinado tipo de elemento é pesquisado no banco de dados, ou de forma total (Relatório Total) quando são gerados dados sobre todas as entidades do PREMOLD inseridas na estrutura.

### **g) Comando "GeraFormato":**

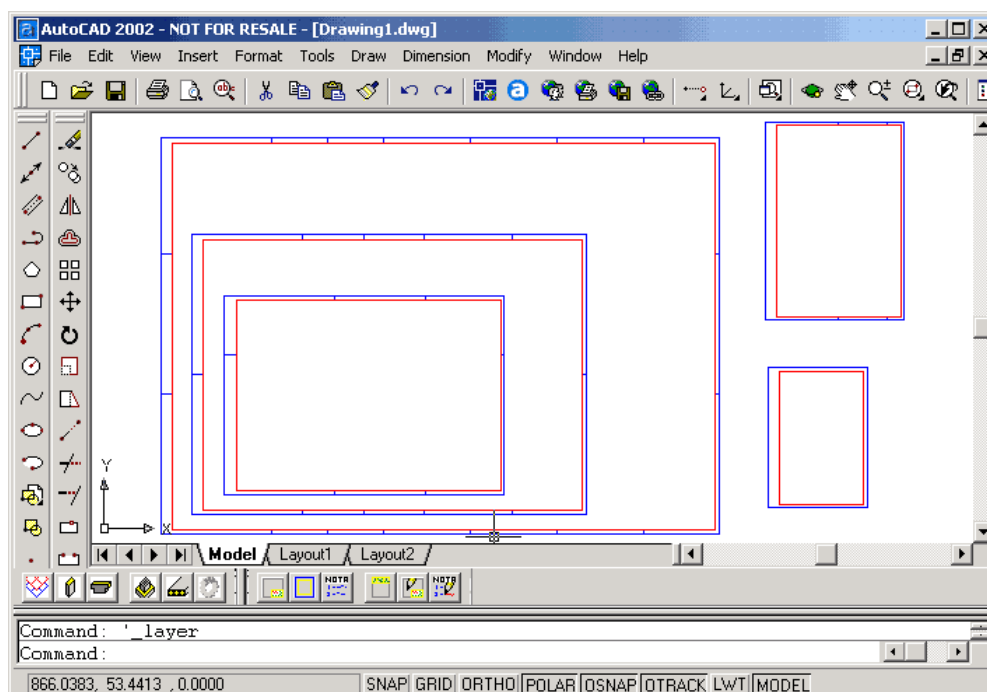
O comando **GeraFormato** cria um formato padronizado pela ABNT através da norma NBR 10068, "Folha de desenho - Leiaute e dimensões" (ABNT, 1987), inserindo suas margens e marcas usuais para facilitar a dobra do formato.

Podem ser gerados os formatos A0, A1, A2, A3, A4 e um formato personalizado cujas dimensões e margens são fornecidas pelo usuário através da barra de comandos do AutoCAD, permitindo uma flexibilização na criação dos formatos. As dimensões do formato e das margens são padronizadas e informadas em milímetros (FIG. 5.21).

Ao acionar o comando, o usuário deve escolher o tipo de formato a ser gerado e o ponto da tela gráfica onde o mesmo será inserido. As dimensões dos formatos padrão e de margens para os mesmos é armazenada em um banco de dados de formatos. Caso o usuário prefira inserir um formato personalizado o aplicativo solicitará todas as dimensões desse e das margens a serem incluídas. Após a leitura dos dados é criada uma entidade da classe **CdTcCFormato**. Os dados do formato a ser gerado são então repassados para essa entidade através de seus métodos de atribuição.

Após a configuração da entidade **CdTcCFormato**, o desenho do formato escolhido e suas margens são apresentados na tela gráfica através de entidades nativas do AutoCAD. Para a representação dos limites do formato de desenho e de suas margens são utilizadas entidades da classe *AcDbPolyline* do ObjectARX.

Os formatos são inseridos na escala 1:1 com os desenhos dimensionados em milímetros uma dimensão de desenho equivalente a um milímetro.



**FIGURA 5.21** – Formatos gerados pelo comando **GeraFormato**.

#### **h) Comandos "GeraLegenda" e "EditaLegenda":**

O comando **GeraLegenda** cria uma legenda padronizada pelo PREMOLD. Essa legenda contém espaços para os seguintes dados:

- Obra, Título, Engenheiro Responsável;
- Datas de: Cálculo, Projeto, Desenho, Verificação e Aprovação;
- Números de: Projeto, Desenho e Folha;
- Escala.

Para a configuração da legenda são necessárias duas dimensões básicas: largura e altura. As demais dimensões da legenda são proporcionais a sua largura e a sua altura. O tamanho da letra usada para preenchimento dos campos da legenda é proporcional à altura da

legenda adequando-se, assim, a diferentes alturas sem causar distorções no preenchimento dos espaços.

Os dados informados na legenda podem receber uma configuração básica padrão. Essa configuração básica deve ser inserida no código fonte do PREMOLD. Caso seja necessário alterar qualquer dado dessa configuração básica pode-se fazê-lo, através do quadro de diálogo do desse comando (FIG. 5.22), durante a inserção da legenda no formato já desenhado, ou alterá-la posteriormente através de um comando chamado **EditaLegenda**.

O diálogo 'Premold - Editor de legendas' contém os seguintes campos e controles:

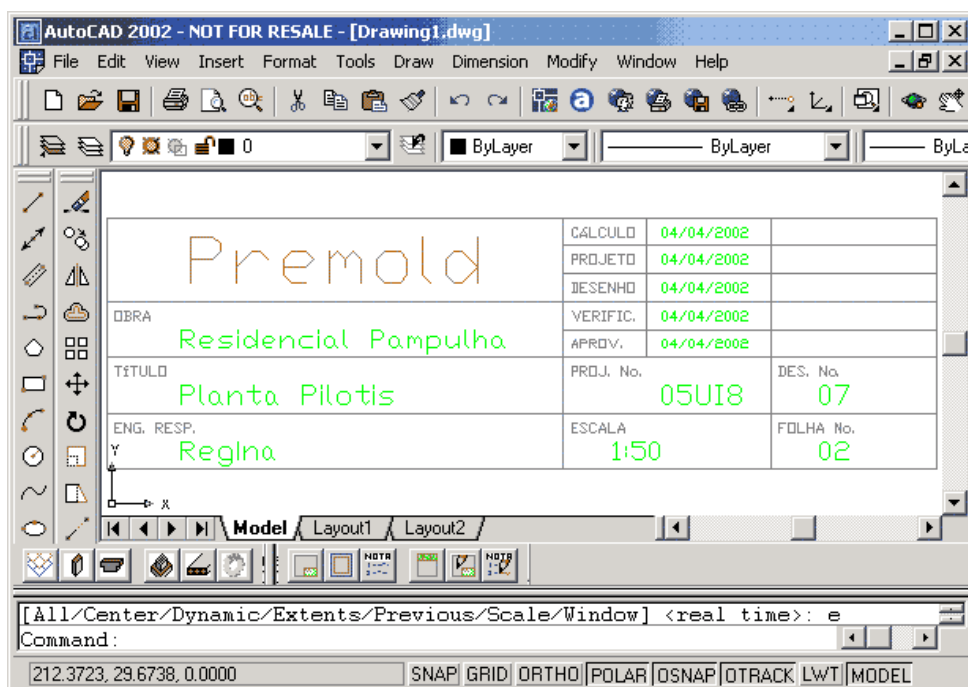
- Dados da legenda:** Um campo de texto contendo o texto 'Premold'.
- Obra:** Um campo de texto para o nome da obra.
- Título:** Um campo de texto para o título da legenda.
- Engenheiro Responsável:** Um campo de texto para o nome do responsável.
- Largura:** Um campo de texto para a largura da legenda.
- Altura:** Um campo de texto para a altura da legenda.
- Calculo:** Um menu suspenso com o valor '01-04-2002'.
- Projeto:** Um menu suspenso com o valor '01-04-2002'.
- Desenho:** Um menu suspenso com o valor '01-04-2002'.
- Verificação:** Um menu suspenso com o valor '01-04-2002'.
- Aprovação:** Um menu suspenso com o valor '01-04-2002'.
- Proj. No.:** Um campo de texto para o número do projeto.
- Des. No.:** Um campo de texto para o número do desenho.
- Escala:** Um campo de texto para a escala.
- Folha No.:** Um campo de texto para o número da folha.
- Botões:** 'Desenhar' e 'Cancelar'.

**FIGURA 5.22** – Quadro de diálogo do comando **GeraLegenda**.

Após a leitura dos dados a serem inseridos na legenda e de suas dimensões, é criada uma entidade da classe **CdTcCLegenda**. Seus atributos são, então, configurados através dos seus métodos de atribuição, recebendo os dados informados pelo usuário no quadro de diálogo. Após essa configuração dos atributos da entidade **Legenda**, essa é inserida no banco de dados do AutoCAD, sendo apresentada na tela gráfica (FIG. 5.23).

O comando **EditaLegenda** abre o mesmo quadro de diálogo criado para o comando **GeraLegenda** e recupera, nos campos apropriados, os valores ou conteúdos dos atributos de uma entidade Legendada já criada através do comando **GeraLegenda**. É possível,

então, alterar qualquer desses campos e o comando procederá a alteração necessária no conteúdo do atributo correspondente. Com isso a entidade **Legenda** será modificada no banco de dados e sua representação gráfica será atualizada



**FIGURA 5.23** – Legenda criada através do comando **GeraLegenda**.

### i) Comandos "GeraNota" e "EditaNota":

O comando **GeraNota** cria uma lista de notas a ser inserida em um desenho. O número de notas é definido pelo usuário, estando limitado a 12 notas individuais.

Através de um quadro de diálogo desenvolvido para esse comando o usuário define o número de notas que deseja inserir e o conteúdo das mesmas (FIG. 5.24). O aplicativo faz, então, a leitura do número de notas e habilita para o usuário tantos campos quantos forem as notas a serem criadas. Após o preenchimento dos campos pelo usuário, é feita a leitura dos campos específicos do quadro de diálogo contendo os dados a serem incluídos em cada nota. Após essa leitura é criada uma entidade da classe **CdTcNota** e os dados informados pelo usuário são repassados para essa entidade através de seus métodos de atribuição. Após a configuração dos conteúdos dos atributos da entidade **Nota** com os dados

informados pelo usuário através do quadro de diálogo do comando **GeraNota**, ela é inserida no banco de dados do AutoCAD e sua representação é mostrada na tela gráfica (FIG. 2.25). Ao serem inseridas no desenho as notas são numeradas automaticamente e dispostas na tela gráfica uma abaixo da outra.

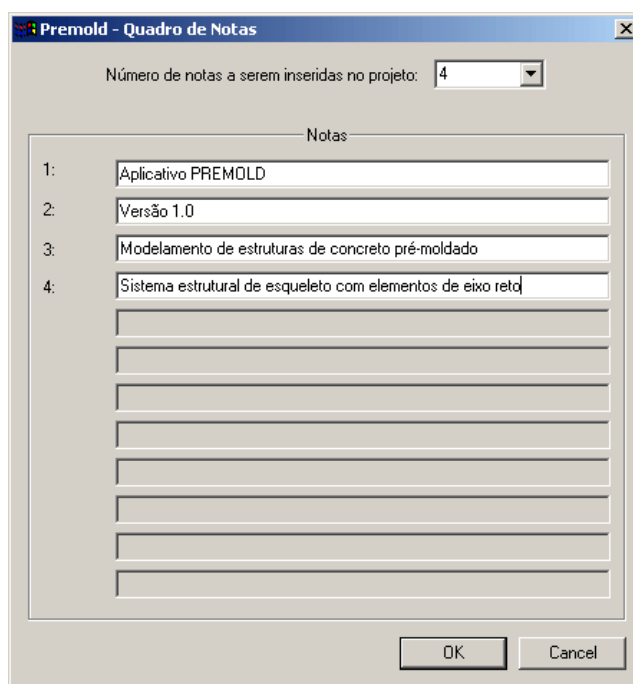


FIGURA 5.24 – Quadro de diálogo do comando **GeraNota**.

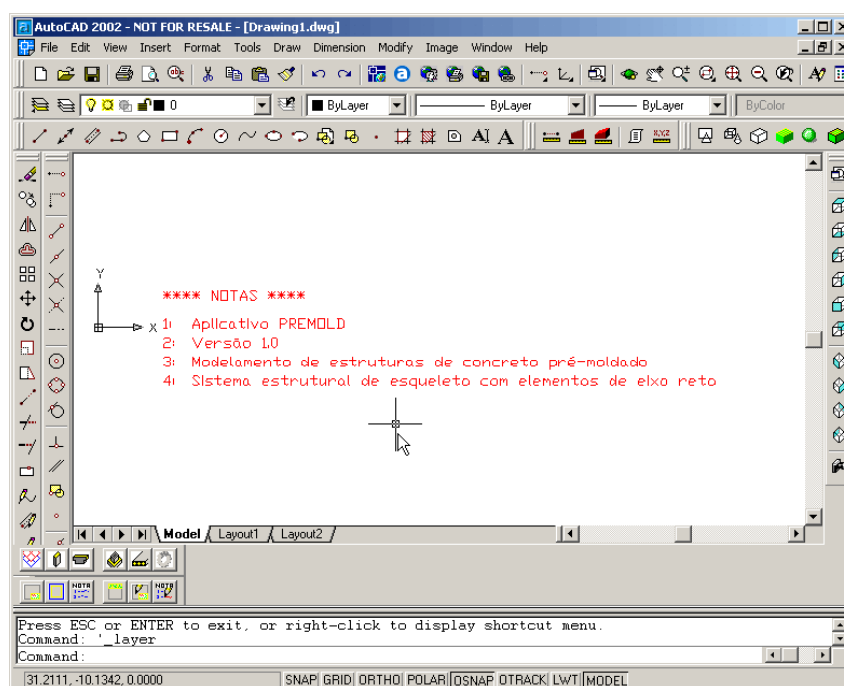


FIGURA 5.25 – Nota gerada pelo comando **GeraNota**

Do mesmo modo que o comando **EditaLegenda**, o comando **EditaNota** abre o mesmo quadro de diálogo criado para o comando **GeraNota** e recupera, nos campos apropriados, os conteúdos dos atributos de uma entidade **Nota** já criada pelo comando **GeraNota**. É possível então se alterar qualquer desses campos e a entidade **Nota** será modificada.

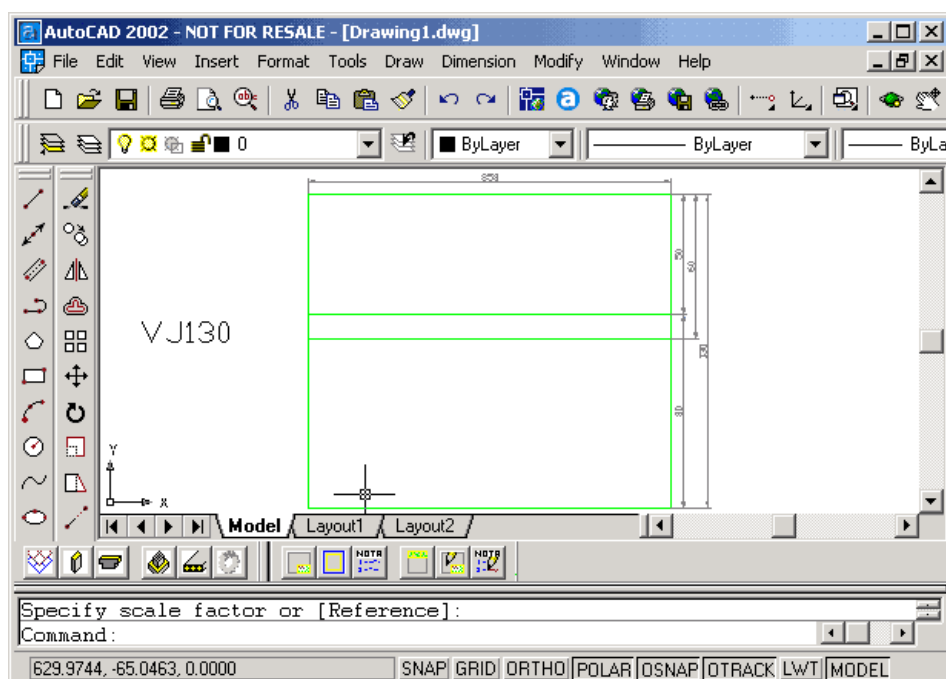
#### **j) Comando "GeraDesenho":**

O comando **GeraDesenho** automatiza a criação de desenhos de fôrma para os elementos lançados na estrutura pelo PREMOLD segundo os critérios da NRB 7191 (ABNT, 1982). O usuário deve escolher o elemento a ser detalhado através de seu número de identificação. A partir da escolha feita pelo usuário, o aplicativo abre automaticamente a entidade selecionada para leitura e obtém os seus dados geométricos necessários à criação dos desenhos da seção transversal e das vistas lateral e superior da peça.

Após a leitura dos dados da peça a ser desenhada, esse comando aciona os comandos **GeraFormato**, **GeraNota** e **GeraLegenda** para a criação do desenho completo da peça.

Para a geração do desenho não foi criada uma classe específica e a representação gráfica dos elementos é feita através de entidades nativas do AutoCAD como linhas, dimensionamentos, textos, etc. As dimensões da peça, seu tipo, o nome de sua seção transversal e seu número identificador são automaticamente inseridos no desenho. O trabalho com entidades nativas possibilita ao usuário alterar facilmente qualquer desenho gerado, inserindo as modificações ou adequações necessárias e não contempladas pelo aplicativo.

Esse comando ainda está em desenvolvimento e muitas implementações ou melhorias ainda devem ser realizadas para que os desenhos gerados estejam realmente completos (FIG. 5.26).



**FIGURA 5.26** – Desenho de forma de uma viga VJ, inserida em uma estrutura gerada pelo PREMOLD, criado pelo comando **GeraDesenho**.

### k) Comando GeraCarimbo:

O comando **GeraCarimbo** é um complemento do comando **GeraLegenda**, acrescentando ao formato uma nova legenda contendo os dados da peça como: acabamento (ACAB), o peso de concreto armado da peça em quilos (PCA), o volume da peça em metros cúbicos, o peso de somente do concreto simples da peça em quilos (PCS), o tipo de aço utilizado, o cobrimento de concreto adotado, a resistência característica à compressão do concreto em MPa, o comprimento da peça em centímetros e a quantidade de peças iguais inseridas na estrutura (FIG. 5.27).

A legenda criada para a informação desses dados é usualmente inserida na parte superior do formato, embora um ponto inferior também possa ser utilizado para sua inserção. Essa legenda é formada por entidades nativas (linhas). A inserção dos dados, é ser realizada pela linha de comandos do AutoCAD.

Essa legenda possui uma largura predefinida, pois é normalmente inserida em formatos A4. A alteração dessa largura, por parte do usuário, poderá ser implementada, possibilitando a adequação da legenda a diferentes formatos.

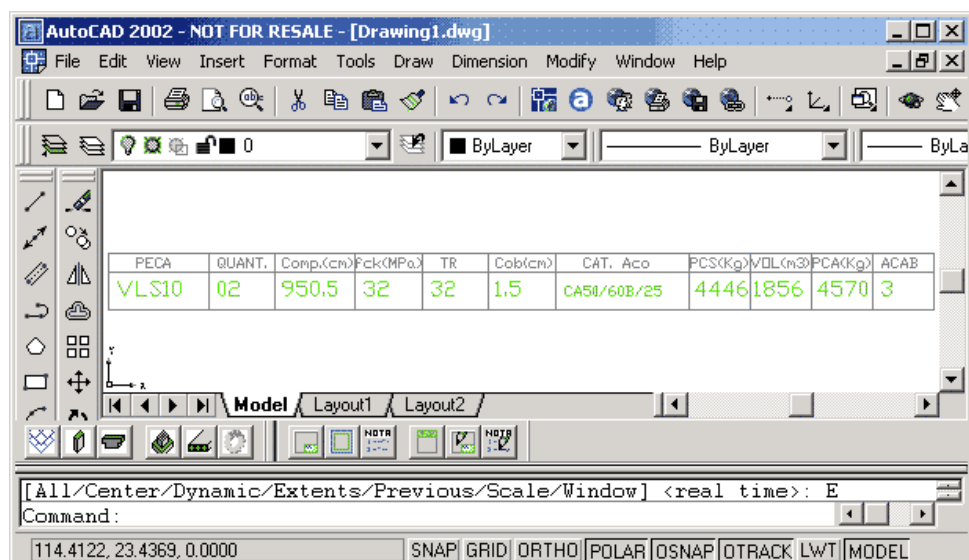


FIGURA 5.27 – Carimbo gerado pelo comando **GeraCarimbo**.

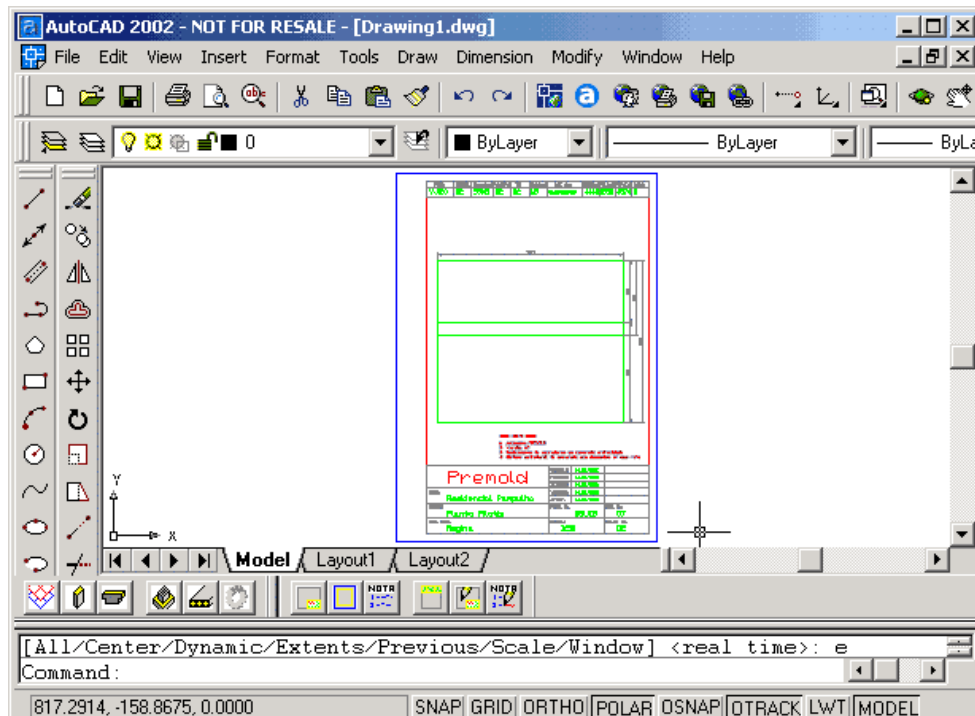


FIGURA 5.28 – Desenho completo gerado pelo comando **GeraDesenho**, acionar dentro desse os comando **GeraFormato**, **GeraLegenda**, **GeraCarimbo** e **GeraNota**

# 6

## CONSIDERAÇÕES FINAIS

### 6.1 O Trabalho Desenvolvido e as Contribuições Geradas

Neste trabalho são apresentadas as atividades realizadas no desenvolvimento de um aplicativo CAD para o modelamento sólido de estruturas de concreto pré-moldado. A etapa de estudo e de reformulação do Modelador 3D, aliada à concepção do escopo conceitual do sistema computacional a ser desenvolvido, também foram muito importantes para a realização deste trabalho. A partir desse estudo um novo aplicativo CAD, denominado PREMOLD, pôde ser construído baseado em uma estrutura de classes que permitiu o modelamento sólido de estruturas de concreto pré-moldado. O escopo conceitual do projeto do aplicativo PREMOLD prevê a continuidade do seu desenvolvimento com o aprimoramento do aplicativo e de suas funções pela inserção de novos sistemas estruturais, de novos elementos e de novas tipologias de ligações, em trabalhos futuros.

### **6.1.1 Contribuições Gerais para o grupo CADTEC**

O desenvolvimento do aplicativo PREMOLD transcendeu a geração de uma ferramenta para a automatização do processo de concepção de estruturas pré-moldadas para também produzir resultados de caráter mais coletivo, cujos beneficiários foram todos os membros do grupo de pesquisa do CADTEC.

Além disso o presente trabalho iniciou, pioneiramente, uma cultura de reaproveitamento de trabalhos anteriores, com as devidas correções dos mesmos e atualizações. As etapas de estudo e de reestruturação do Modelador 3D serviram para ganhar tempo e para descobrir novos caminhos para a criação de uma hierarquia de classes mais adequada aos paradigmas da Programação Orientada a Objetos (POO).

Este trabalho mostrou, também, a importância do desenvolvimento de uma padronização para as classes mais genéricas geradas no CADTEC, a fim de se ter um maior reaproveitamento de códigos.

O estudo realizado sobre o Modelador 3D foi também importante para aglutinar todas as informações sobre aquele aplicativo, que se encontravam distribuídas em duas dissertações de mestrado. Essa divisão dos assuntos nas dissertações dificultava uma compreensão global do trabalho desenvolvido.

O trabalho de capacitação de dois alunos de iniciação científica realizado com a finalidade de viabilizar sua participação no projeto teve como resultado não só a execução de tarefas menores do projeto, mas principalmente a qualificação dos mesmos nesta tecnologia CAD. Além disso o presente projeto criou uma relação de interatividade entre os grupos de trabalho do CADTEC, o que muito contribuiu para o aprofundamento do conhecimento de todos.

Nesse sentido, as contribuições do presente trabalho extrapolam o desenvolvimento do projeto, para influenciar positivamente outros trabalhos desenvolvidos no CADTEC.

## **6.1.2 Contribuições Específicas para a Automação de Processo Produtivos**

O aplicativo desenvolvido nesta etapa de trabalho construiu as bases para um trabalho mais amplo de modelamento e detalhamento de estruturas de concreto pré-moldado. As classes geradas foram concebidas prevendo implementações futuras que permitam a inclusão de novos sistemas estruturais, de novos tipos de elementos, de novas funcionalidades e, principalmente, de novas classes. Além disso o desenvolvimento foi feito levando-se em conta o reaproveitamento da estrutura de classes para a implementação de outros aplicativos.

O uso da herança de classes com a estrutura Generalização/Especialização e da agregação de classes com a estrutura Todo/Parte permitiu a geração de todas as classes utilizadas neste trabalho, possibilitando também, a inclusão no aplicativo de novas classes derivadas daquelas. O aprimoramento das classes base também pode ser realizado sem prejuízo ao aplicativo desenvolvido ou para outros aplicativos, que utilizem essas classes. Esses fatores contribuem para o reaproveitamento dos códigos já gerados, uma das características da POO que até então não estava sendo adequadamente aplicada aos os trabalhos desenvolvidos no CADTEC.

Este trabalho serviu, também, para mostrar as vantagens e desvantagens do uso de entidades e de objetos personalizados no desenvolvimento de aplicativos para o AutoCAD. Essas entidades e objetos permitem a geração de um banco de dados próprio, capaz de armazenar os dados necessários para integrar diversas etapas do processo de desenvolvimento de projetos de estruturas, através do uso de sistemas automatizados. O completo gerenciamento dessas entidades exige do programador um elevado trabalho de definição de funções básicas para a sua manipulação. O uso de entidades nativas do AutoCAD dispensa tais implementações mais avançadas, porém dificulta o gerenciamento do banco de dados gerado e sua vinculação às entidades gráficas representadas na tela.

Embora o esforço computacional requerido para se utilizar entidades personalizadas seja considerável, conclui-se que esse procedimento é a maneira mais eficiente para automatizar o processo produtivo.

### **6.1.3 Contribuições Específicas para a Indústria de Concreto Pré-Moldado**

O trabalho desenvolvido para esta dissertação ainda não está suficientemente operacional para se colocar em uma linha de produção de uma indústria de pré-moldados. Entretanto, o estágio atual do aplicativo mostra claramente o potencial desta tecnologia na automação deste tipo de processo. O aumento da produtividade, a precisão dos resultados e a melhoria da qualidade dos produtos são os resultados esperados.

Uma primeira versão do aplicativo poderá em breve ser aplicada na indústria, assim que novas implementações para generalizar o lançamento da estrutura sejam realizadas. Recursos do aplicativo para combinar um maior número de tipo de elementos em posições e direções variadas, aliados a implementações para a edição e alteração da estrutura, tornarão o aplicativo suficientemente operacional para causar impacto na produção de pré-moldados.

A utilização do AutoCAD como plataforma gráfica para o desenvolvimento do PREMOLD facilita a interação entre os projetos arquitetônico e estrutural. O uso de uma mesma plataforma gráfica elimina a etapa de tradução da geometria da estrutura obtida a partir do projeto arquitetônico para modeladores ou para pré-processadores desenvolvidos em outra plataforma. Além disso, na concepção de edificações em concreto pré-moldado, geralmente a arquitetura utiliza a própria estrutura na definição da estética final da edificação, usando a estrutura aparente e explorando suas particularidades como ligações e encaixes. O PREMOLD servirá ao arquiteto como uma importante ferramenta para explorar a forma final da estrutura que será lançada segundo critérios de norma estabelecidos pela NBR 9062/85 (ABNT, 1985), introduzidos no aplicativo.

A geração de um modelo sólido tridimensional, já na etapa de pré-dimensionamento da estrutura, ou ainda em seu lançamento inicial, tem grande apelo comercial. Assim, o PREMOLD permitirá ao cliente uma visualização clara da estrutura a ser construída, facilitando a sua comercialização. Os relatórios quantitativos e qualitativos gerados pelo PREMOLD, a partir de seu banco de dados, permitem o lançamento de um orçamento básico da estrutura logo após seu modelamento inicial.

## 6.2 Limitações do Aplicativo e Propostas para Trabalhos Futuros

Nesta primeira etapa de desenvolvimento do PREMOLD foram desenvolvidos recursos para lançamento tridimensional sólido de um número considerável de elementos, o tratamento de um tipo de ligação entre esses elementos e o controle de lançamento de nós, barras e placas do modelo unifilar para o sistema estrutural reticulado com elementos de eixo reto. Apesar da automação alcançada no modelamento da estrutura, vários aspectos ainda necessitam de aprimoramento e outros ainda devem ser implementados. As limitações do PREMOLD, considerando-se o desenvolvimento até aqui alcançado, e as atividades as serem desenvolvidas em trabalhos futuros são descritas a seguir:

### a) Malha:

A malha lançada nesta etapa de trabalho possui módulos constantes em cada direção, não sendo possível, ainda, criar espaçamentos variáveis. Em desenvolvimentos futuros será criada uma classe para gerenciar a malha gerada e um controle sobre a malha que possibilite referenciar o objeto e a malha no nível em que eles são inseridos. Evitar-se-á, assim, que o lançamento de uma malha muito grande, com diversos níveis, gere um desenho confuso, dificultando a visualização do espaço de trabalho pelo usuário. Uma outra importante implementação será a habilitação ou não das linhas inseridas em determinadas coordenadas para se trabalhar cada nível da estrutura por vez. Além disso será criada uma numeração ou codificação para as linhas da malha numa projeção em planta da estrutura (numeração dos eixos da estrutura). Essa codificação torna-se muito útil nos projetos de locação de pilares ou de fundação.

#### b) Edição de entidades:

Após a inserção de um elemento, seja ele pilar, viga, painel ou bloco é importante poder editar esse elemento, ou seja, criar dispositivos para editar elementos já inseridos no banco de dados do AutoCAD e do PREMOLD. As ferramentas de edição de entidades serão também implementadas futuramente.

Outra implementação importante será o tratamento das interferências entre elementos, quando um elemento que suporta outros elementos é movido, copiado ou excluído do banco de dados. Nesses casos apenas o objeto que sofre a ação e seus dados, inseridos no banco de dados do PREMOLD, são tratados. Assim, caso um pilar, que sirva de base para duas vigas, seja copiado, apenas o pilar com seus consoles será copiado. Se esse mesmo pilar for excluído do banco de dados, as vigas apoiadas sobre ele ficarão sem apoio, mas não serão excluídas.

As interferências entre elementos, que não tenham uma relação direta de apoio e elemento apoiado, também ainda não foram tratadas. Por exemplo, a interferência entre um painel horizontal e o pilar que ultrapassa o nível de inserção do painel precisa ser implementada.

#### c) Entidade **Pilar**:

Para os pilares foram inseridas apenas as seções retangulares ou quadradas. As demais seções listadas no item 2.2.1 não foram ainda trabalhadas.

A inserção de outros tipos de pilares na biblioteca do aplicativo deve ser desenvolvida, possibilitando a personalização dessa biblioteca. Tal procedimento deve ser implementado, para pilares e para os demais elementos.

No caso de pilares com tubo de descida de água, a saída lateral do tubo e a terminação superior do tubo, geralmente em forma de funil, serão implementações futuras.

Outra implementação necessária é a ligação de topo entre pilares ou o lançamento de pilares nascendo sobre vigas ou sobre cortinas.

Para o comprimento de engastamento dos pilares em blocos de fundação será inserida a criação de ranhuras no pilar, cujo prolongamento ainda é liso.

d) Entidade **Viga**:

O lançamento das vigas está ainda limitado à face do pilar perpendicular ao plano vertical que define a direção de lançamento, não sendo possível ainda lançar vigas lateralmente ao pilar. Ou seja, os consoles estão sempre na mesma direção das vigas, sendo necessária a criação da opção de lançar vigas perpendiculares à direção dos consoles.

As vigas são lançadas individualmente devendo ser desenvolvido o lançamento através de matrizes com "m" linhas e "n" colunas como é feito para os pilares.

O lançamento de vigas "I" com seção variável junto aos apoios, o lançamento de vigas com altura variável ao longo do comprimento e o lançamento de vigas inclinadas em relação ao plano horizontal da estrutura são importantes implementações futuras.

O acabamento de vigas calha sobre os pilares também deve ser desenvolvido.

Para viga composta por partes concretadas na obra durante a montagem da estrutura deve ser prevista a altura de concretagem "in loco" para que a mesma possa ser detalhada no desenho de forma.

Muitas vezes é necessário prever furos, ou vãos, nas vigas para a passagem de tubos ou dutos. Esses furos podem ser gerados através de uma ferramenta própria a ser desenvolvida e que efetuará tais cortes nos elementos em pontos definidos e na forma e dimensões escolhidas pelo usuário.

O tratamento de interferências de vigas esconsas com os pilares de apoio e com as demais vigas apoiadas sobre os pilares também está para ser desenvolvido.

No lançamento das vigas deve ser previsto, ainda, o lançamento de consoles tipo capitel nos pilares. Esse tipo de console é lançado em torno de todo o pilar sendo indicado para receber vigas esconsas e para os sistemas estruturais em pavimentos sem vigas.

e) Entidade **Painel**:

Na atual versão os painéis são lançados para cobrir o comprimento de uma viga de apoio. Caso sejam lançados painéis para cobrir duas vigas consecutivas, separadas apenas pelo pilar de apoio comum a elas, são necessárias, outras opções de inserção de painéis visando cobrir o vão que corresponde a este intervalo entre as vigas, preenchido pelo pilar de apoio. Também o tratamento da interferência entre o painel de piso e o pilar, caso esse ultrapasse o nível de inserção do painel, é uma importante implementação futura.

Até o momento o aplicativo lida com o lançamento de painéis horizontais, que devem ser lançados sobre vãos retangulares ou quadrados. O lançamento de painéis, principalmente de lajes maciças ou de painéis alveolares, em vãos com outras configurações é uma das atividades a serem desenvolvidas na continuação dos trabalhos do PREMOLD.

O lançamento de painéis verticais, tanto de fechamento quanto estruturais, também deverá ser desenvolvido em etapas futuras.

f) Entidade **Bloco**:

Em relação à textura interna do colarinho dos blocos de fundação, é ainda necessário implementar o lançamento de ranhuras no colarinho. Atualmente esse possui paredes lisas.

Vigas baldrame para ligação entre os blocos de fundação e a criação de blocos de divisa também precisam ser implementados.

g) Lançamento da estrutura e dimensionamento preliminar:

Foi mencionado no item 6.1.3 que o lançamento da estrutura necessita de uma maior flexibilização para englobar diversos tipos de combinações entre os elementos, diversas

posições e direções para esses elementos. Novas ferramentas ou comandos devem ser criados. Opções de escolha entre diversos tipos de sistemas, de tipos de ligações entre os elementos, de tipos de acessórios utilizados, entre outros, serão proporcionadas ao usuário em futuros desenvolvimentos. Além disso para um dimensionamento preliminar da estrutura será necessário automatizar o aplicativo para inserir os carregamentos com as cargas permanentes e acidentais definidas pelo usuário e o peso próprio dos elementos. Com esse carregamento inicial, novos critérios de dimensionamento mínimo dos elementos também podem ser inseridos no aplicativo.

#### h) Detalhamento da estrutura:

Como foi mencionado no item 5.2.5.j, o comando para geração de desenhos de fôrma ainda encontra-se em fase de desenvolvimento, necessitando de correções e novas implementações. O detalhamento até agora desenvolvido se restringe a desenhos de fôrma dos elementos, não tendo sido lançado nenhum dado sobre as ferragens dos mesmos.

No comando GeraDesenho deve ser inserida a geração de listas de materiais para cada desenho gerado e listas gerais de material para toda a estrutura.

#### i) Ligações:

O tratamento das ligações, realizado nesta etapa de trabalho, restringiu-se ao estudo de interferências entre os elementos lançados, que possuem relação de dependência do tipo apoio e elemento apoiado. As folgas mínimas devem ser informadas pelo usuário e são consideradas no lançamento dos elementos. Não foram tratados, ainda, as tolerâncias previstas pela norma NBR 9062/85 (ABNT, 1985) e o comprimento mínimo para consolos e máximo para vigas.

#### j) Sistema estrutural:

Nesta etapa de trabalho os elementos são sempre lançados segundo o sistema reticulado com elementos de eixo reto. O tratamento de outros tipos de sistemas deverá ainda ser feito, porém, as bases para o modelamento de qualquer dos demais sistemas listados nos

itens 2.3 e 2.4 já foram desenvolvidas, restando implementar algumas opções de alterações: a posição de nós, o tipo de restrições lançadas nos nós, o tipo de elementos e o tipo de apoio dos elementos, etc.

k) Modelo unifilar:

Os comandos para a representação gráfica do modelo unifilar não foram implementados nesta etapa de trabalho. Para que a representação dos elementos do modelo unifilar seja possível, deverão ser implementadas algumas alterações nas dimensões definidas para textos e símbolos criados pelo Modelador 3D, pois aquele modelador trabalha com dimensões em milímetros e o PREMOLD trabalha com dimensões em centímetros.

l) Carregamentos:

Toda as classes, variáveis e funções implementadas no Modelador 3D para carregamentos foram aproveitadas no PREMOLD. O lançamento de cargas sobre os elementos não foi implementado, pois se deseja alterar o modo de lançamento proposto no Modelador 3D. Nas próximas etapas de desenvolvimento do PREMOLD, pretende-se que cada elemento inserido na estrutura possa repassar automaticamente as cargas permanentes decorrentes de sua inserção para os elementos que lhe servirão de apoio. A informação de sobrecargas e cargas variáveis pelo usuário, também deverá ser implementada.

Todas as pendências listadas nos itens acima deverão ser abordadas na continuação dos trabalhos do PREMOLD, no desenvolvimento de um sistema de manufatura integrada por computador que será tema de doutorado desta aluna de mestrado.

Além das considerações listadas, algumas modificações nas classes geradas e a criação de novas classes serão necessárias para englobar outros elementos, os acessórios de ligações e outros dados relevantes para as demais etapas produtivas da estrutura.

Até aqui vários desafios foram vencidos e novos foram vislumbrados, sempre em busca de alcançar e superar um alvo no universo do conhecimento humano, como uma pequena contribuição para o seu crescimento.

## REFERÊNCIAS BIBLIOGRÁFICAS

- (ABNT, 1978) ABNT – Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *NB-6118 -Projeto e execução de obras de Concreto Armado*, 1978.
- (ABNT, 1982) ABNT – Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *NBR 7191 -Execução de desenhos para obras de concreto simples ou armado (orig. NB16)*, 1982.
- (ABNT, 1985) ABNT – Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *NBR 9062 -Projeto e execução de estruturas de concreto pré-moldado – Procedimento (orig. NB949)*, 1985.
- (ABNT, 1987) ABNT – Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *NBR 10068 - "Folha de desenho - Leiaut e dimensões"*, 1987.
- (ABNT, 2000) ABNT - Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *Projeto de Revisão NBR 6118 – Projeto de Estruturas de Concreto*, 2000.
- (AUTODESK, 1999a) AUTODESK, Inc. *AutoCAD 2000 - ObjectARX Developer's Guide*, 1999.
- (AUTODESK, 1999b) *AutoCAD User's Guide for AutoCAD 2000, Developer Guide*, 1999.
- (AUTODESK, 1999c) AUTODESK, Inc. *AutoCAD Customization Guide for AutoCAD2000, Developer Guide*, 1999.
- (AUTODESK, 1999d) AUTODESK, Inc *AutoCAD Command Reference for AutoCAD2000, Developer Guide*, 1999.
- (AUTODESK, 1999e) AUTODESK Inc., *ObjectARX for AutoCAD 2000 On-Line Help*, 1999.

- (AUTODESK, 1999f) AUTODESK Inc., AutoCAD 2000 On-Line Help, 1999.
- (CADTEC, 2000) CADTEC, DEES, UFMG, AutoCAD 2000 Curso Básico Apostila 1a. Edição, 2000.
- (CULLENS et. al, 1997) CULLENS, C., DADIDSON, M., ROBICHAUX, P., CORRY, C., POTTS, S., GREGORY, K. Usando Visual C++: O guia de referência mais completo. Rio de Janeiro: Editora Campus Ltda, 1997. 763 p.
- (DEITEIL e DEITEIL, 1997) DEITEL, Harvey M., DEITEIL P.J, C++ How to Program. 2a. edição, Prentice Hall, New Jersey. 1130p, 1997.
- (EL DEBS, 2000) EL DEBS, Mounir Khalil. Concreto pré-moldado: fundamentos e aplicações, São Carlos: EESC-USP, 2000.
- (ELLIS e STROUSTRUP, 1993) ELLIS, M. A., STROUSTRUP, B. C++ Manual de Referência Comentado. Rio de Janeiro: Campus, 1993.
- (HORTON, 1997) HORTON, Ivor; Beginning Visual C++, Wrox Press Ltd., UK, 1997.
- (HÜNTER, 1998) HÜNTER, Alexandra, Sistema de Modelagem CAD, Modelador 3D para Estruturas de Barras via CAD Engenharia/ Visualização/Banco de Dados, Dissertação de Mestrado – UFMG, Belo Horizonte, 1998.
- (KRUGLINSKI, 1997) KRUGLINSKI, D. J., Inside Visual C++, 4 ed. United States of America: Canada Publishing Corporation, 1997. 986 p
- (MAGALHÃES, 1998) MAGALHÃES, Ana Liddy Cenni de Castro. Estudo Projeto e implementação de um modelador de sólidos voltado para aplicações em eletromagnetismo, Escola de Engenharia, Universidade Federal de Minas Gerais, 1998.

- (MAGALHÃES, 2002) MAGALHÃES, Paulo Henrique Vieira de. Automação do Processo de Modelamento e Detalhamento de Torres Via Tecnologia CAD. Escola de Engenharia, Universidade Federal de Minas Gerais, 2002.
- (MALARD, 2002) MALARD, Fernando Poinho. Comunicação pessoal.
- (MALARD, 1998) MALARD, Fernando Poinho. Sistema de Modelagem CAD, Modelador 3D para Estruturas de Barras Via CAD. Arquitetura do Sistema/Banco de Dados/Visualização. Escola de Engenharia, Universidade Federal de Minas Gerais, 1998.
- (MÄNTYLÄ, 1988) MÄNTYLÄ, M. An introduction to solid modeling. Maryland: Computer Science, 1988.401p
- (MERSQUITA, 2001) MESQUITA, Renato. Material de aula da Disciplina Projeto Assistido por Computador, 1º Semestre de 2001, UFMG
- (MICROSOFT, 1998) MICROSOFT, Visual C++ 6.0 On-Line Help. 1998
- (MONTENEGRO e PACHECO, 1994) Orientação a objetos em C++. Rio de Janeiro: Editora Ciência Moderna Ltda., 1994. 394 p.
- (OLIVEIRA, 1991) OLIVEIRA, M. C. F. Modelagem geométrica: uma introdução. Instituto de ciências matemáticas de São Carlos, Universidade de São Paulo. Relatório interno, 1991.
- (PREMO, 1999) PREMO, Engenharia Indústria Comércio Ltda. Manual Técnico: Construções Prefabricadas. Premo, 1999.
- (QUATRANI, 1998) QUATRANI, Terry. Visual Modeling with Rational Rose and UML. Addison Wesley, USA, 222p, 1998.
- (SHAH e MÄNTILÄ, 1995) SHAH, J.J., MÄNTILÄ M. Parametric and feature-based CAD/CAM: concepts, techniques and applications. New York: John Wiley, 1995. 619p.

## **APÊNDICE A**

### **MANUAL DE UTILIZAÇÃO DO PREMOLD**

## APRESENTAÇÃO

O PREMOLD, versão 1.0 foi desenvolvido pelo CADTEC (Centro de Apoio, Desenvolvimento Tecnológico e Ensino da Computação gráfica), através da dissertação de mestrado desenvolvida pela aluna Regina Célia Guedes Leite, do curso de Mestrado em Engenharia de Estruturas da UFMG.

Esse modelador permite a integração visual entre o usuário e a estrutura virtual gerada, de forma amigável e segura, permitindo a substituição da entrada puramente gráfica ou numérica por uma entrada de componentes que representam os componentes de uma estrutura de concreto pré-moldado. Esses componentes contêm, não só dados geométricos, mas os demais dados necessários a outras etapas do processo produtivo dessas estruturas. Esse modelador foi desenvolvido como um aplicativo que utiliza o AutoCAD 2002 como plataforma gráfica.

Esse modelador foi desenvolvido para a geração de uma estrutura baseada no sistema reticulado com elementos de eixo reto para edifícios de múltiplos pavimentos, com pilares engastados na fundação e vigas articuladas nos pilares. Para tanto, foram desenvolvidas entidades que representam os pilares, as vigas, os painéis de piso e os blocos de fundação dessas estruturas.

Ao ser lançada uma estrutura, todos os dados gerados são armazenados no banco de dados do AutoCAD e no banco de dados do PREMOLD, podendo ser exportados para arquivos texto, contendo relatórios geométricos e quantitativos da estrutura.

Ao serem lançados, pelo usuário, os elementos estruturais vão recebendo os elementos necessários para sua representação no modelo estrutural escolhido. Em termos do sistema estrutural adotado, os elementos estruturais são compostos por três elementos básicos: barra, nós e placas. Essas entidades são associadas a entidades estruturais da seguinte forma:

- Pilares e vigas são representados por uma barra com dois nós, um em cada extremidade;

- Painéis de piso: são representados por placas delimitadas por vértices;
- Blocos de fundação: são inseridos no ponto de inserção dos pilares que recebe o nó inicial das barras de representação dos pilares.

Estas entidades (nós, barras e placas) são numeradas automaticamente ao serem inseridas nos elementos estruturais gerados.

A ordem de lançamento dos elementos estruturais deve obedecer à ordem lógica de lançamento desses em uma estrutura real.

Alguns comandos do AutoCAD podem ser utilizados normalmente pelo usuário como comandos de cópia e de movimentação dos elementos gerados. Durante esses comandos, os dados dos elementos estruturais como sua numeração, seu ponto de inserção seus nós final e inicial são alterados e renumerados automaticamente pelo aplicativo quando necessário.

Todo o lançamento da estrutura e os dimensionamentos básicos da mesma, quando relacionados a dados geométricos, não se considerando esforços ou carregamento, são realizados automaticamente pelo aplicativo, tomando-se como base os critérios de dimensionamento apresentados pela NBR 9062/85 “*Projeto e execução de estruturas de concreto pré-moldado – Procedimento*” (ABNT, 1985).

## SUMÁRIO

APRESENTAÇÃO.....	214
SUMÁRIO .....	216
DOCUMENTAÇÃO .....	220
INSTALAÇÃO .....	220
BARRAS DE FERRAMENTAS.....	221
Barra de ferramentas de modelamento.....	222
Barra de ferramentas de detalhamento .....	223
COMANDOS DO PREMOLD.....	224
Comando GeraMalha .....	224
Comandos de Modelamento .....	226
Comando GeraPilar .....	226
Comando GeraViga .....	228
Comando GeraPainel.....	232
Comando GeraBloco .....	236
Comando GeraRelatorio.....	240
Comandos Auxiliares para o Detalhamento .....	243
Comando GeraFormato.....	243
Comando GeraLegenda.....	244
Comando EditaLegenda.....	246
Comando GeraNota.....	246
Comando EditaNota .....	248
Comando GeraCarimbo .....	248
Comando de Detalhamento.....	249
Comando GeraDesenho .....	249
CRIAÇÃO AUTOMÁTICA DO MODELO UNIFILAR .....	251
EXEMPLOS DE MODELAMENTO GERADO PELO PREMOLD .....	254
Exemplo 01 .....	254
Exemplo 02 .....	264
REFERÊNCIAS BIBLIOGRÁFICAS .....	269

## ÍNDICE DE FIGURAS

FIGURA A. 1 – BARRA DE FERRAMENTAS DE MODELAMENTO DO PREMOLD.....	222
FIGURA A. 2 - BARRA DE FERRAMENTAS DE DETALHAMENTO DO PREMOLD .....	223
FIGURA A. 3 – BOTÃO DE ACIONAMENTO DO COMANDO GERAMALHA. ....	224
FIGURA A. 4 – QUADRO DE DIÁLOGO DO COMANDO GERAMALHA.....	225
FIGURA A. 5 – MALHA TRIDIMENSIONAL GERADA PELO COMANDO GERAMALHA .....	226
FIGURA A. 6 - BOTÃO DE ACIONAMENTO DO COMANDO GERAPILAR .....	226
FIGURA A. 7 - QUADRO DE DIÁLOGO DO COMANDO GERAPILAR .....	227
FIGURA A. 8 – PILARES DE DIFERENTES SEÇÕES COM OU SEM TUBOS DE DESCIDA DE ÁGUA GERADOS PELO COMANDO GERAPILAR .....	228
FIGURA A. 9 - BOTÃO DE ACIONAMENTO DO COMANDO GERAVIGA.....	228
FIGURA A. 10 - QUADRO DE DIÁLOGO DO COMANDO GERAVIGA .....	229
FIGURA A. 11 – SEÇÕES TRANSVERSAIS DE VIDAS VLS COM POSIÇÃO ESPELHADA OU NÃO. .....	230
FIGURA A. 12 – VIGAS INSERIDAS SOBRE PILARES PELO COMANDO GERAVIGA .....	231
FIGURA A. 13 – VISTA SUPERIOR DE VIGAS INSERIDAS SOBRE PILARES FACEANDO UMA DAS FACES DOS PILARES OU CENTRADAS EM RELAÇÃO AOS PILARES. ....	232
FIGURA A. 14 - BOTÃO DE ACIONAMENTO DO COMANDO GERAPAINEL .....	232
FIGURA A. 15 - QUADRO DE DIÁLOGO DO COMANDO GERAPAINEL.....	233
FIGURA A. 16 – TIPO DE PREENCHIMENTO DO VÃO DA VIGA POR PAINÉIS DE PISO. ....	233
FIGURA A. 17 – ALERTA DADO PELO APLICATIVO QUANTO A NÃO ADEQUAÇÃO ENTRE O TIPO DE PAINEL A SER INSERIDO E O TIPO DE VIGA DE APOIO .....	234
FIGURA A. 18 – PAINÉIS INSERIDOS SOBRE VIGAS: PAINEL ALVEOLAR SOBRE VIGAS PRINCIPAIS E LAJES SOBRE VIGAS DE DIFERENTES TAMANHOS. ....	235
FIGURA A. 19 – FIG. A.13 COM O USO DO RECURSO <i>FLAT SHADED</i> DE TRATAMENTO DE SÓLIDOS DO AUTOCAD. ....	235
FIGURA A. 20 – VISTA SUPERIOR DA ESTRUTURA MOSTRADA NA FIG. A.14.....	236
FIGURA A. 21 - BOTÃO DE ACIONAMENTO DO COMANDO GERABLOCO.....	236
FIGURA A. 22 - QUADRO DE DIÁLOGO DO COMANDO GERABLOCO .....	237
FIGURA A. 23 – BLOCOS GERADOS PELO PREMOLD DIMENSIONADOS DE ACORDO COM AS DIMENSÕES DOS PILARES SOB OS QUAIS SÃO INSERIDOS, RESPEITANDO AS DIMENSÕES MÍNIMAS PRESCRITAS PELA NBR 9062/85.....	238

FIGURA A. 24 – ESTRUTURA GERADA PELO PREMOLD COM PILARES COM BLOCOS DE FUNDAÇÃO, VIGAS VL E PAINEL ALVEOLAR PREENCHENDO TODO O VÃO DAS VIGAS, APOIADO SOBRE AS ABAS DAS MESMAS. ....	238
FIGURA A. 25 – INSERÇÃO DE APENAS UM PAINEL ALVEOLAR SOBRE AS VIGAS VL, APOIADO SOBRE AS ABAS DAS MESMAS. ....	239
FIGURA A. 26 – VISTA SUPERIOR DA ESTRUTURA REPRESENTADA NA FIG. A.19. ....	239
FIGURA A. 27 - BOTÃO DE ACIONAMENTO DO COMANDO GERARELATORIO. ....	240
FIGURA A. 28 – QUADRO DO WINDOWS PARA GRAVAR O ARQUIVO QUE ARMAZENARÁ OS DADOS DO RELATÓRIO GERADO PELO PREMOLD. ....	240
FIGURA A. 29 – LINHA DE COMANDO DO AUTOCAD PARA ESCOLHA DO TIPO DE RELATÓRIO A SER GERADO: G/Q, GEOMÉTRICO OU QUANTITATIVO. ....	241
FIGURA A. 30 - – LINHA DE COMANDO DO AUTOCAD PARA ESCOLHA DO TIPO DE RELATÓRIO A SER GERADO: PARCIAL OU TOTAL ....	241
FIGURA A. 31 – QUADRO DE DIÁLOGO DO COMANDO GERARELATORIO PARA ESCOLHA DOS ELEMENTOS A SEREM LISTADOS NO RELATÓRIO PARCIAL. ....	241
FIGURA A. 32 – EXEMPLO DE RELATÓRIO GEOMÉTRICO GERADO PELO PREMOLD .....	242
FIGURA A. 33 - BOTÃO DE ACIONAMENTO DO COMANDO GERAFORMATO .....	243
FIGURA A. 34 – FORMATOS GERADOS PELO COMANDO GERAFORMATO. ....	244
FIGURA A. 35 - BOTÃO DE ACIONAMENTO DO COMANDO GERALEGENDA. ....	244
FIGURA A. 36 – QUADRO DE DIÁLOGO DO COMANDO GERALEGENDA. ....	245
FIGURA A. 37 – LEGENDA GERADA PELO COMANDO GERALEGENDA DO PREMOLD. ....	245
FIGURA A. 38 - BOTÃO DE ACIONAMENTO DO COMANDO EDITALEGENDA. ....	246
FIGURA A. 39 - BOTÃO DE ACIONAMENTO DO COMANDO GERANOTA.....	246
FIGURA A. 40 - QUADRO DE DIÁLOGO DO COMANDO GERANOTA. ....	247
FIGURA A. 41 – NOTA CRIADA PELO COMANDO GERANOTA DO PREMOLD. ....	247
FIGURA A. 42 - BOTÃO DE ACIONAMENTO DO COMANDO EDITANOTA.....	248
FIGURA A. 43- BOTÃO DE ACIONAMENTO DO COMANDO GERACARIMBO. ....	248
FIGURA A. 44 – CARIMBO GERADO PELO COMANDO GERACARIMBO DO PREMOLD. ....	249
FIGURA A. 45 - BOTÃO DE ACIONAMENTO DO COMANDO GERADESENHO .....	249
FIGURA A. 46 – DESENHO DE FORMA DE FIGA VJ CRIADO PELO COMANDO GERADESENHO DO PREMOLD. ....	250
FIGURA A. 47 – FORMATO, LEGENDA, NOTA, CARIMBO E DESENHO DEFORMA CRIADOS PELO PREMOLD. ....	251

FIGURA A. 48 – PILARES GERADOS PELO PREMOLD .....	254
FIGURA A. 49 – RELATÓRIO SOBRE A ESTRUTURA FORMADA POR QUATRO PILARES GERADOS PELO PREMOLD .....	255
FIGURA A. 50 – VIGAS RETANGULARES GERADAS PELO PREMOLD. ....	256
FIGURA A. 51 - RELATÓRIO SOBRE A ESTRUTURA FORMADA POR QUATRO PILARES E QUATRO VIGAS GERADAS PELO PREMOLD. ....	257
FIGURA A. 52 – PAINEL ALVEOLAR GERADO PELO PREMOLD. ....	258
FIGURA A. 53 - RELATÓRIO SOBRE A ESTRUTURA FORMADA POR QUATRO PILARES, QUATRO VIGAS E DOIS PAINÉIS ALVEOLARES, GERADOS PELO PREMOLD. ....	259
FIGURA A. 54 - RELATÓRIO QUANTITATIVO TOTAL SOBRE A ESTRUTURA FORMADA POR QUATRO PILARES, QUATRO VIGAS E DOIS PAINÉIS ALVEOLARES, GERADOS PELO PREMOLD. ....	260
FIGURA A. 55 – BLOCOS DE FUNDAÇÃO GERADOS PELO PREMOLD. ....	261
FIGURA A. 56 - VISTA SUPERIOR DA ESTRUTURA GERADA PELO PREMOLD. ....	262
FIGURA A. 57 - FIG. A.13 COM O USO DO RECURSO <i>FLAT SHADED</i> DE TRATAMENTO DE SÓLIDOS DO AUTOCAD. ....	263
FIGURA A. 58 – PILARES DE ALTURAS DIFERENTES. ....	264
FIGURA A. 59 – PILARES COM VIGAS INSERIDAS NO PRIMEIRO PAVIMENTO. ....	265
FIGURA A. 60 – VISTA SUPERIOR DA ESTRUTURA DA FIG. A.57. ....	265
FIGURA A. 61 – ESTRUTURA COM PILARES, VIGAS NOS PAVIMENTOS 1 E 2 E BLOCOS DE FUNDAÇÃO.....	266
FIGURA A. 62 – VISTA SUPERIOR DA ESTRUTURA MOSTRADA NA FIG. A.59.....	266
FIGURA A. 63 – COMANDO <i>FLAT SHADED</i> DO AUTOCAD SOBRE ESTRUTURA GERADA PELO PREMOLD. ....	267
FIGURA A. 64 – ESTRUTURA DA FIG. A.59 COM PAINÉIS DE PISO DO TIPO LAJE SOBRE VIGAS DO PRIMEIRO PAVIMENTO. ....	267
FIGURA A. 65 – ESTRUTURA DA FIG. A.63 SEM AS VIGAS DE FECHAMENTO DO SEGUNDO PAVIMENTO, COM INSERÇÃO DE VIGAS SECUNDÁRIAS E OUTRA DISTRIBUIÇÃO DE PAINÉIS DE PISO. ....	268

## DOCUMENTAÇÃO

Este aplicativo é utilizado em conjunto com o AutoCAD 2002. Para sua correta utilização aconselha-se a leitura detalhada deste manual.

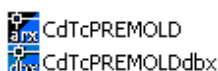
Este manual lista todos os comandos desenvolvidos nesta primeira versão do PREMOLD.

Recomenda-se a utilização dos manuais do AutoCAD 2002 para dúvidas relativas àquele programa.

## INSTALAÇÃO

Os arquivos fornecidos juntamente com este manual, no disquete de instalação, devem ser copiados para dentro do diretório de trabalho do AutoCAD.

O PREMOLD é composto por dois arquivos compilados que trabalham em conjunto, são eles:



- CdTcPREMOLD
- CdTcPREMOLDdbx

Os dois arquivos devem ser carregados para que o programa esteja ativo. Para carregar os arquivos basta digitar ARX na linha de comando do AutoCAD, optar por *Load* e então selecionar primeiramente o arquivo CdTcPREMOLDdbx e posteriormente o arquivo CdTcPREMOLD. Ou entrar na opção *Tools* do menu de tela do AutoCAD e escolher a opção *Load Application*, escolher o arquivo CdTcPREMOLDdbx, apertar o botão *Load* e repetir a operação para o arquivo CdTcPREMOLD.

Quando o aplicativo está carregado aparecem na tela gráfica duas barras de ferramentas próprias do aplicativo.

Para descarregar o aplicativo basta digitar ARX na linha de comandos do AutoCAD e escolher a opção *Unload*. Tendo feito essa escolha, basta digitar na linha de comando, primeiramente, o nome do arquivo CdTcPREMOLD e posteriormente, repetir o comando para o arquivo CdTcPREMOLDdbx. Pelo menu de tela basta entrar na opção *Tools, Load Application*, selecionar os arquivos na ordem descrita acima e apertar o botão *Unload*.

Ao descarregar o aplicativo com um desenho aberto que tenha sido gerado por aquele, todas as entidades personalizadas criadas pelo aplicativo serão transformadas em entidades *proxies*. Os *proxies* são entidades que possuem a mesma representação gráfica que as entidades personalizadas, mas não possuem a mesma funcionalidade, impossibilitando que qualquer alteração seja aplicada sobre essas. Esse fator impede qualquer alteração nos arquivos criados pelo aplicativo por outros usuários que não o possuam, aumentando a segurança dos arquivos criados pelo aplicativo.

Caso o aplicativo seja recarregado, as entidades *proxies* são automaticamente transformadas em entidades personalizadas voltando a possuir sua funcionalidade própria.

## **BARRAS DE FERRAMENTAS**

Foram criadas duas barras de ferramentas para facilitar o acionamento dos comandos pelo usuário. Através dos botões da barra de ferramenta pode-se acessar os comandos do aplicativo sem a necessidade de digitar na linha de comandos o nome dos mesmos. Os comandos do PREMOLD foram agrupados segundo sua funcionalidade em comandos de modelamento e comandos de detalhamento.

## Barra de ferramentas de modelamento



**FIGURA A. 1** – Barra de ferramentas de modelamento do PREMOLD

Comando **GeraMalha** cria uma malha tridimensional auxiliar para o lançamento da estrutural. Essa malha é formada por linhas do AutoCAD

Comando **GeraPilar** cria a representação de elementos estruturais do tipo **Pilar** através de entidades personalizadas.

Comando **GeraViga** cria a representação de elementos estruturais do tipo **Viga** através de entidades personalizadas. As vigas devem ser criadas tendo pilares ou outras vigas como elementos de apoio.

Comando **GeraPainel** cria a representação de elementos estruturais do tipo **Painel de piso** através de entidades personalizadas. Os painéis devem ser criados tendo vigas como elementos de apoio.

Comando **GeraBloco** cria a representação de elementos estruturais do tipo **Bloco de fundação** através de entidades personalizadas. Os blocos devem ser criados na base de pilares já inseridos na estrutura.

Comando **GeraRelatorio** exporta dados da estrutura modelada para um arquivo texto, em forma de relatórios que podem conter dados geométricos e quantitativos sobre toda a estrutura ou sobre um tipo particular de elemento.

## Barra de ferramentas de detalhamento



**FIGURA A. 2** - Barra de ferramentas de detalhamento do PREMOLD

Comando **GeraFormato** cria um formato de desenho padronizado pela NBR 10068/87 "*Folha de desenho - Leitura e dimensões*" (ABNT, 1987), ou personalizado pelo usuário, com margens apropriadas.

Comando **GeraLegenda** cria uma legenda, ou carimbo, com campos preenchidos com dados referentes ao desenho criado, para ser inserido em um formato de desenho.

Comando **GeraNota** cria uma enumeração de notas ou observações que podem ser preenchidas pelo usuário e inseridas em um desenho.

Comando **GeraDesenho** cria automaticamente um desenho de forma de um elemento da estrutura gerada seguindo critérios previstos pela NBR 7191/82 "*Execução de desenhos para obras de concreto simples ou armado*" (ABNT, 1982). Esse comando aciona automaticamente os comandos: **GeraFormato** e **GeraLegenda**.

Comando **EditaLegenda** permite a alteração dos campos de preenchimento de uma legenda criada pelo comando **GeraLegenda**.

Comando **EditaNota** permite a alteração das notas inseridas em uma enumeração criada pelo comando **GeraNotas**.

Comando **GeraCarimbo** cria um carimbo com campos serem preenchidos com os dados técnicos do elemento estrutural detalhado, para ser inserido em um formato de desenho.

## COMANDOS DO PREMOLD

O PREMOLD possui treze comandos, sendo quatro comandos de modelamento da estrutura com lançamento de elementos estruturais, um comando de detalhamento com geração automática de desenhos de forma de elementos da estrutura, sete comandos para criação ou edição de entidades auxiliares usadas tanto no modelamento quanto no detalhamento da estrutura, e um comando de exportação de dados da estrutura.

### Comando GeraMalha

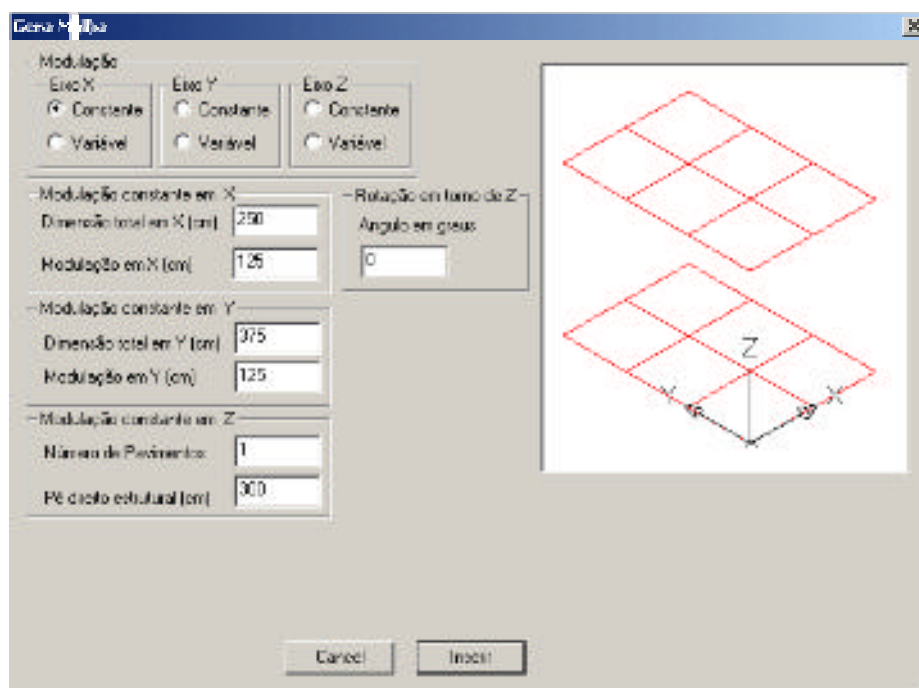


Acionar o botão **GeraMalha** da barra de ferramentas de modelamento ou digitar "GERAMALHA" na linha de comandos do AutoCAD.

**FIGURA A. 3** – Botão de acionamento do comando **GeraMalha**.

Com o quadro de diálogo **GeraMalha** aberta, escolher os espaçamentos segundo os eixos X e Y globais do AutoCAD e as dimensões totais da malha nessas direções, além do número de pavimentos e altura de um pavimento padrão. O aplicativo dividirá as dimensões totais indicadas pelo usuário em módulos com dimensões iguais aos espaçamentos escolhidos, caso

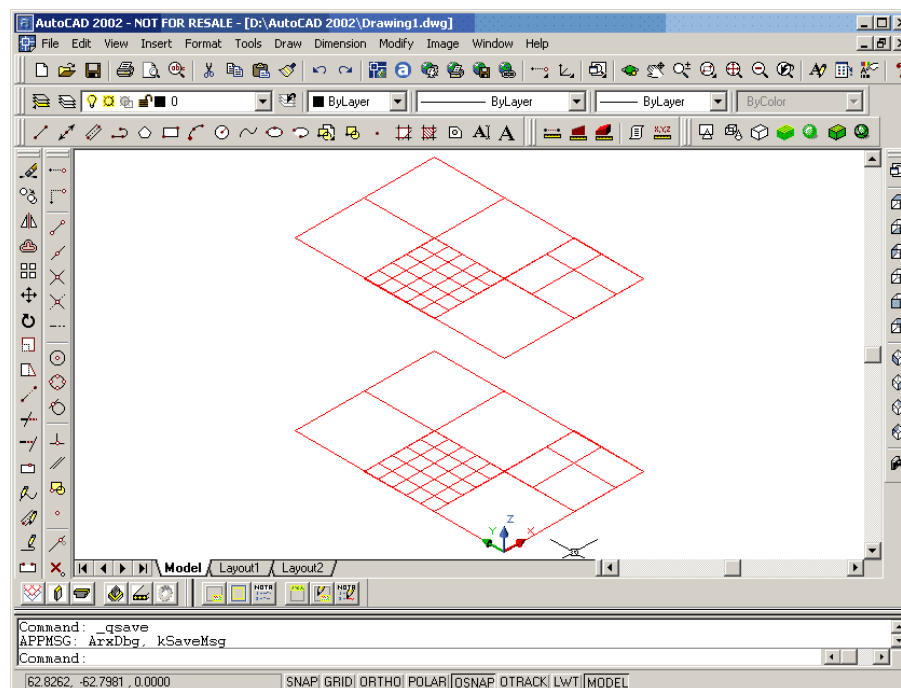
qualquer das dimensões totais não comportem um número inteiro de divisões o último módulo será desenhado para fechar a malha respeitando a dimensão total definida.



**FIGURA A. 4** – Quadro de diálogo do comando **GeraMalha**

Não é possível inserir módulos variáveis nesta versão do aplicativo, porém pode-se inserir malhas menores dentro de malhas maiores, proporcionando um maior detalhamento da malha em pontos onde esse seja necessário.

Após terem sido definidas as dimensões parciais e totais nos campos próprios do quadro de diálogo, o usuário deve acionar o botão "inserir" para escolher na tela gráfica o ponto de inserção da malha ou digitar, na linha de comandos do AutoCAD, as coordenadas de um ponto para sua inserção. Caso o usuário não deseje inserir a malha pode cancelar o comando ao acionar o botão "cancelar" que o quadro de diálogo será fechada e o comando será cancelado.



**FIGURA A. 5** – Malha tridimensional gerada pelo comando **GeraMalha**

## Comandos de Modelamento

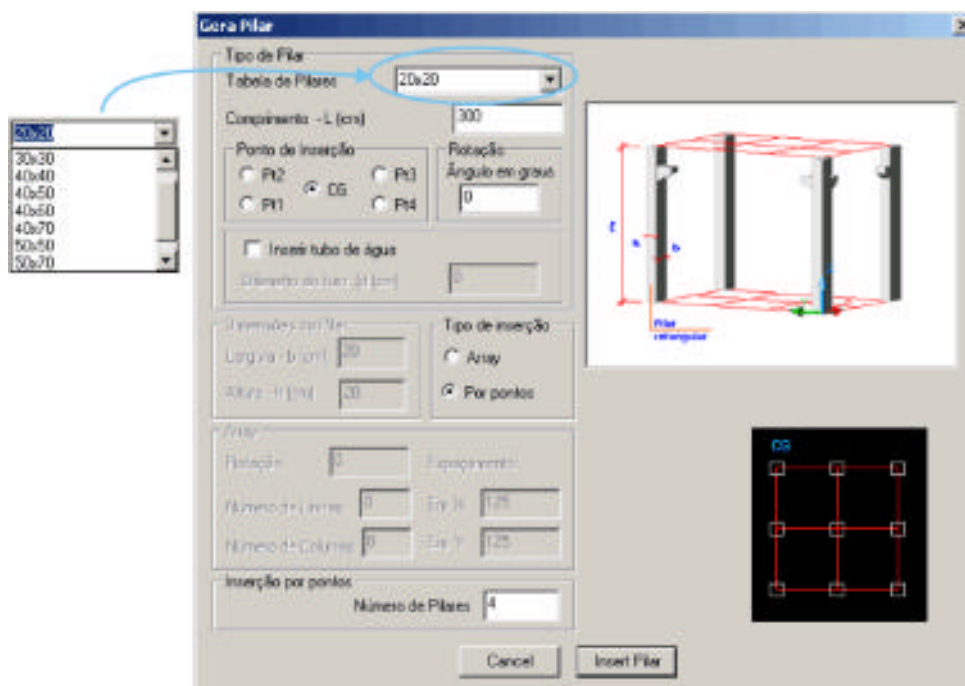
### Comando GeraPilar



Acionar o botão **GeraPilar** da barra de ferramentas de modelamento ou digitar "GERAPILAR" na linha de comandos do AutoCAD.

**FIGURA A. 6** - Botão de acionamento do comando **GeraPilar**

Com o quadro de diálogo **GeraPilar** aberta, escolher a seção transversal do pilar através da barra de rolagem do campo "Seção Padrão". Para gerar outra seção diferente das seções padrões disponíveis escolher na barra de rolagem "Seção Personalizada" e inserir nos campos largura e altura as dimensões do pilar a ser criado.



**FIGURA A. 7 -** Quadro de diálogo do comando **GeraPilar**

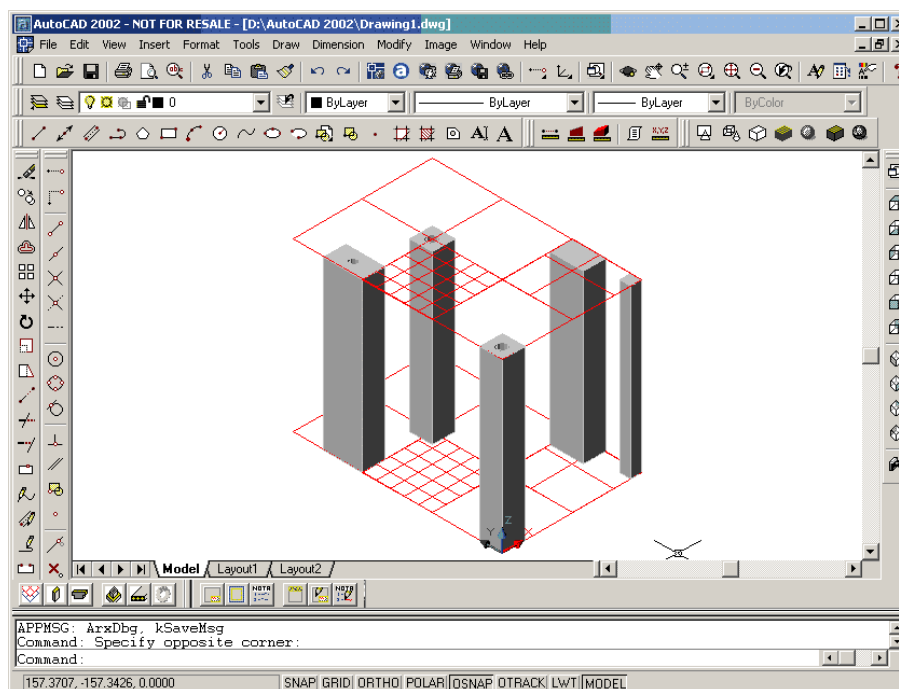
O pilar pode ser inserido pelo seu centro geométrico, ou por qualquer um dos vértices de sua seção transversal localizada em sua base. Ao ser escolhido um tipo de inserção é são mostrados exemplos do tipo escolhido no canto inferior esquerdo do quadro de diálogo.

Os pilares podem receber tubos para descida de água pluvial. Caso seja escolhida a opção de inserir esses tubos deve-se indicar o diâmetro do mesmo. Caso o diâmetro indicado seja maior que aquele que permita uma parede mínima de 10cm para a seção restante do pilar, o usuário é alertado de que o diâmetro escolhido não está de acordo com a norma NBR 90962/85 (ABNT, 1985), devendo fazer uma escolha permitida pela mesma.

Pode-se inserir mais de um pilar durante uma chamada do comando **GeraPilar**. O número de pilares deve ser indicado pelo usuário. Como padrão foi definido um número básico de 4 pilares. Os pilares podem ser inseridos individualmente, quando o usuário deve indicar os pontos de inserção dos mesmos na tela gráfica ou através de suas coordenadas digitadas no linha de comando do AutoCAD, ou através de uma matriz de "m" linhas e "n" colunas de pilares, quando o usuário deve indicar o número de linhas e colunas e o espaçamento entre as mesmas.

É necessário, ainda que o usuário indique o comprimento dos pilares a serem inseridos através do campo "Comprimento do Pilar".

Após ter preenchido os campos do quadro de diálogo, o usuário pode acionar o botão "inserir" para escolher os pontos de inserção dos pilares ou o botão "cancelar" para interromper o comando e voltar à tela gráfica do AutoCAD.



**FIGURA A. 8** – Pilares de diferentes seções com ou sem tubos de descida de água gerados pelo comando **GeraPilar**

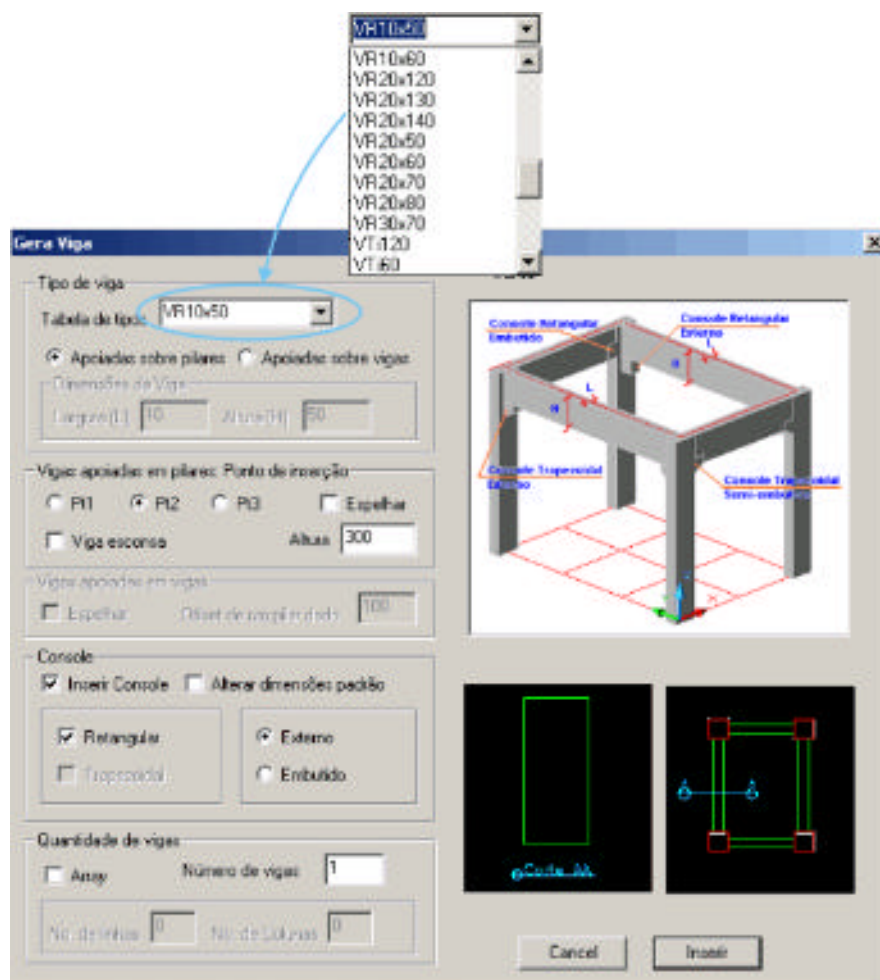
## Comando GeraViga



Acionar o botão **GeraViga** da barra de ferramentas de modelamento ou digitar "GERAVIGA" na linha de comandos do AutoCAD.

**FIGURA A. 9** - Botão de acionamento do comando **GeraViga**

Com o quadro de diálogo **GeraViga** aberta, escolher a seção transversal da viga através da barra de rolagem do campo "Tabela de tipos". É permitido ao usuário gerar uma seção retangular diferente das seções padrão disponíveis, bastando para isso, escolher na barra de rolagem "Seção Personalizada" e inserir nos campos largura e altura as dimensões da viga retangular a ser criada.



**FIGURA A. 10** - Quadro de diálogo do comando **GeraViga**

A viga deve sempre ser criada tendo dois pilares ou duas vigas como apoios. Quando possui pilares como apoio, ela pode ser inserida centrada em relação aos pilares ou faceando os pilares em uma de suas duas faces paralelas à direção do comprimento da viga. O tipo de inserção da viga é mostrado através de esquemas de inserção em um quadro no canto inferior esquerdo do quadro de diálogo. A altura de inserção da viga em relação à cota zero do projeto também deve ser informada pelo usuário.

As vigas podem ser inseridas com seção transversal espelhada em relação a um eixo vertical de simetria quando são chamadas espelhadas. A posição normal de inserção e a posição espelhada também são mostradas através de esquemas da seção transversal em um quadro no canto inferior esquerdo do quadro de diálogo, juntamente com o tipo de inserção da viga para vigas segundo o eixo X global e o eixo Y global.



**FIGURA A. 11** – Seções transversais de vigas VLS com posição espelhada ou não.

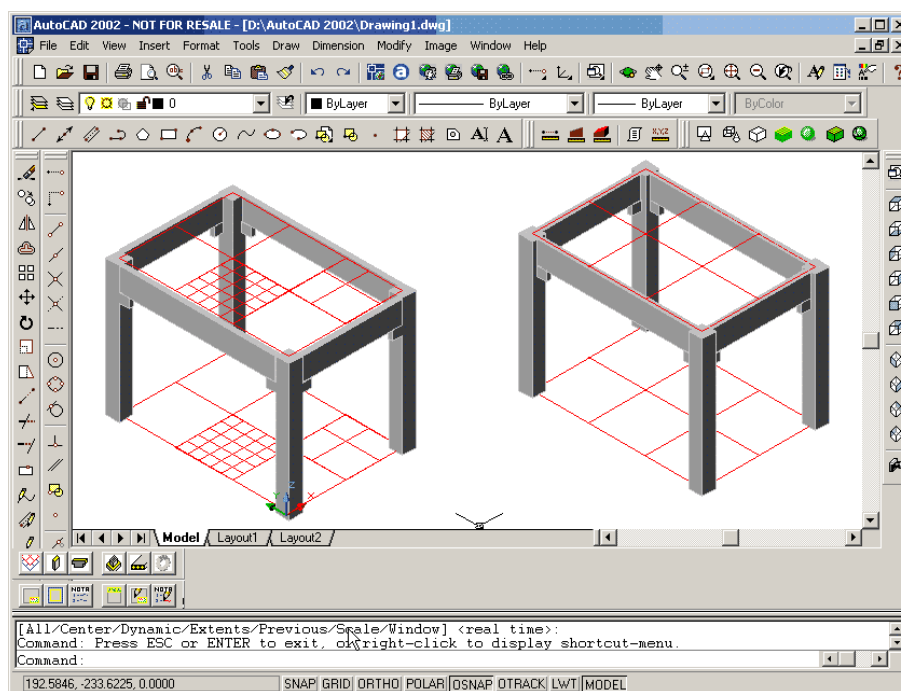
Ao serem inseridas as vigas seus pilares de apoio recebem consolos que podem ser retangulares ou trapezoidais, externos, embutidos ou semi-embutidos (apenas para consolos trapezoidais). Os tipos de consolos devem ser escolhidos pelo usuário, devendo ser iguais nos pilares de base para uma mesma viga. O padrão de inserção do consolo define o mesmo como retangular e externo. As dimensões dos consolos são definidas através das configurações globais do sistema, podendo também ser alteradas pelo usuário. Essas dimensões seguem o padrão mínimo da norma NBR 9062/85(ABNT, 1985) para a altura na extremidade do consolo em relação à sua altura junto ao pilar. Caso o usuário escolha uma altura não recomendada pela norma ele é avisado de que a dimensão escolhida não se enquadra nas determinações da norma, devendo alterá-la. Os consoles podem possuir a largura da viga ou a largura do pilar.

Pode-se inserir mais de uma viga durante uma chamada do comando **GeraViga**, devendo o número de vigas ser indicado pelo usuário. Como padrão foi definida a inserção de apenas uma viga. As vigas podem ser inseridas apenas de modo individual, quando o usuário deve indicar os apoios das mesmas, ao selecionar, na tela gráfica, os elementos que lhe servirão de apoio.

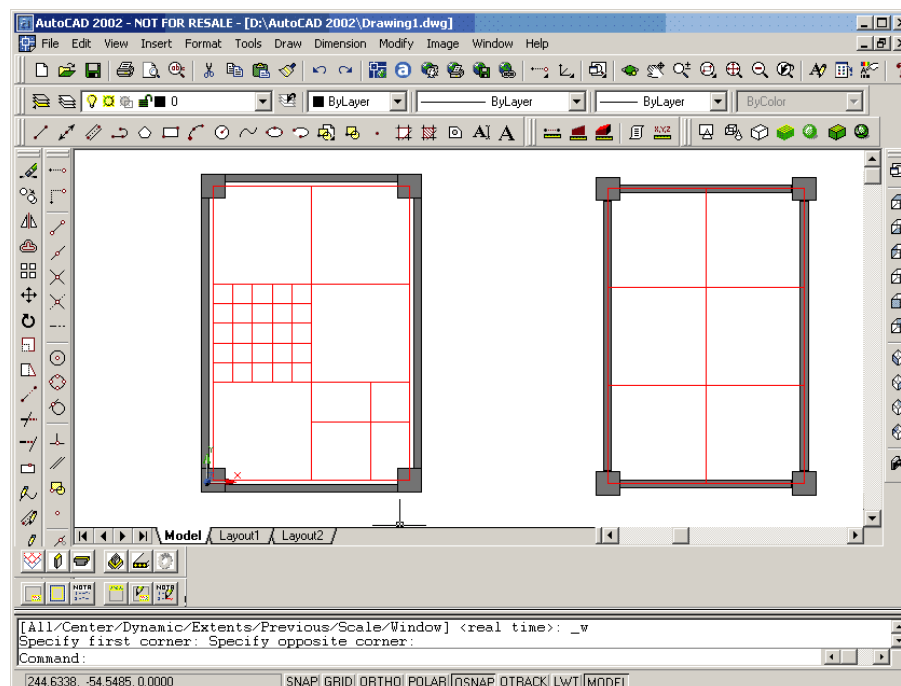
Não é necessário que o usuário indique o comprimento das vigas. Ao serem escolhidos os apoios, o aplicativo calcula automaticamente a distância entre esses e insere a viga já

descontando, no comprimento das vigas, as seções dos mesmos e as folgas definidas na configuração geral do sistema. Essas são, ainda, cortadas para a inserção dos consolos embutidos (ou semi-embutidos) conforme o tipo e as dimensões desses.

Após ter preenchido os campos do quadro de diálogo, o usuário pode acionar o botão "inserir" para escolher os apoios das vigas ou o botão "cancelar" para interromper o comando e voltar à tela gráfica do AutoCAD.

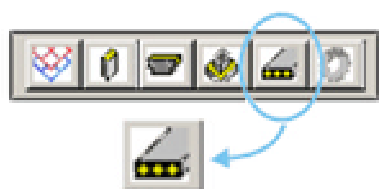


**FIGURA A. 12 – Vigas inseridas sobre pilares pelo comando GeraViga**



**FIGURA A. 13** – Vista superior de vigas inseridas sobre pilares faceando uma das faces dos pilares ou centradas em relação aos pilares.

## Comando GeraPainel



Acionar o botão **GeraPainel** da barra de ferramentas de modelamento ou digitar "GERAPAINEL" na linha de comandos do AutoCAD.

**FIGURA A. 14** - Botão de acionamento do comando **GeraPainel**

Com o quadro de diálogo **GeraPainel** aberta, escolher a seção transversal do painel através da barra de rolagem do campo "Seção Padrão". É permitido ao usuário gerar um painel do tipo "laje", diferente das seções padrão disponíveis, com seção retangular, bastando para isso, escolher na barra de rolagem "Laje" e inserir nos campos largura e altura as dimensões da laje a ser criada.

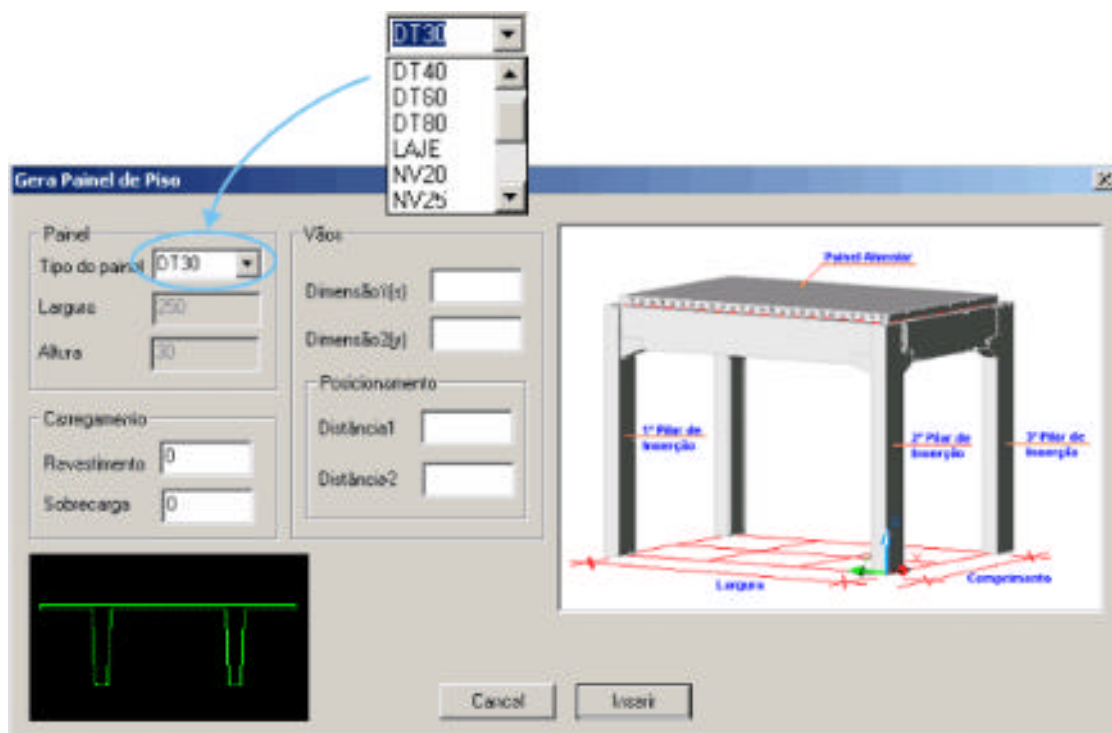


FIGURA A. 15 - Quadro de diálogo do comando **GeraPainel**

O painel de piso deve ser sempre criado tendo duas vigas como apoios. Podem ser inseridos mais de um painel em um vão durante uma única chamada do comando **GeraPainel**. Para a inserção dos painéis é sempre considerada a viga de menor comprimento entre aquelas escolhidas para determinar o vão a ser preenchido pelos painéis. Pode ser inserido apenas um painel no vão considerado, um número inteiro de painéis que caibam dentro do vão, ou um número fracionário de painéis para preenchimento completo do vão. Para a escolha do tipo de preenchimento do vão o usuário deve responder através da linha de comandos do AutoCAD à pergunta mostrada na FIG. A.16.

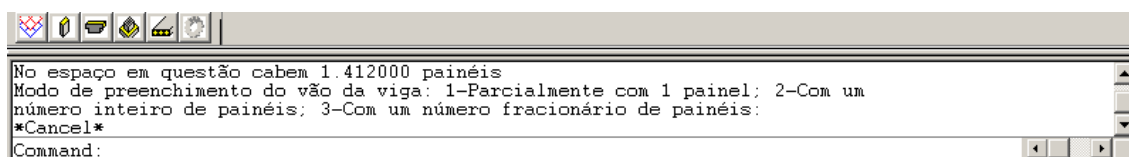
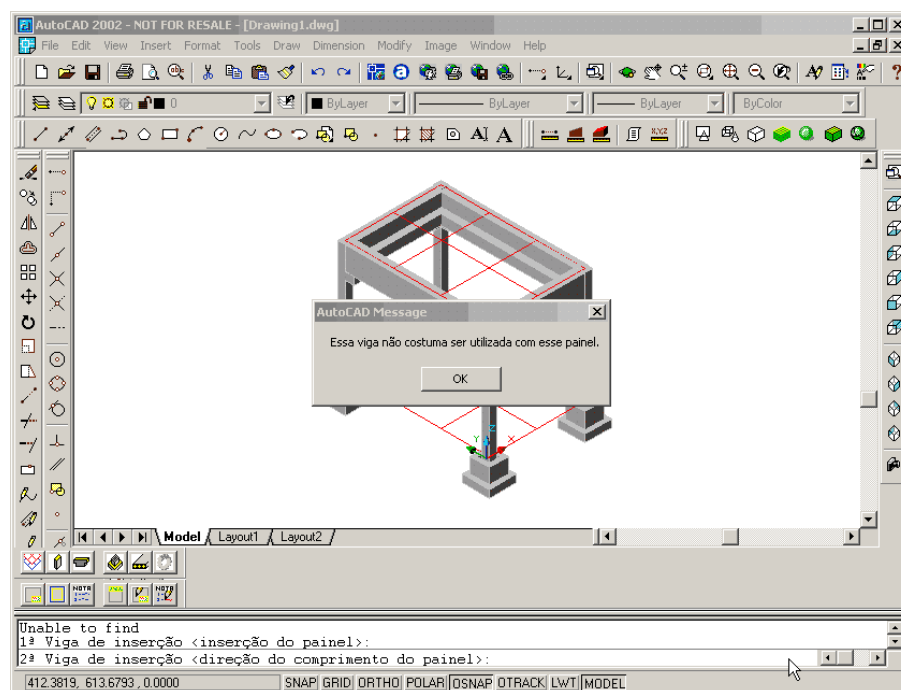


FIGURA A. 16 – Tipo de preenchimento do vão da viga por painéis de piso.

Os painéis de piso que fazem parte da biblioteca do PREMOLD geralmente são inseridos sobre um tipo específico de viga. Caso o usuário escolha um painel que geralmente não é

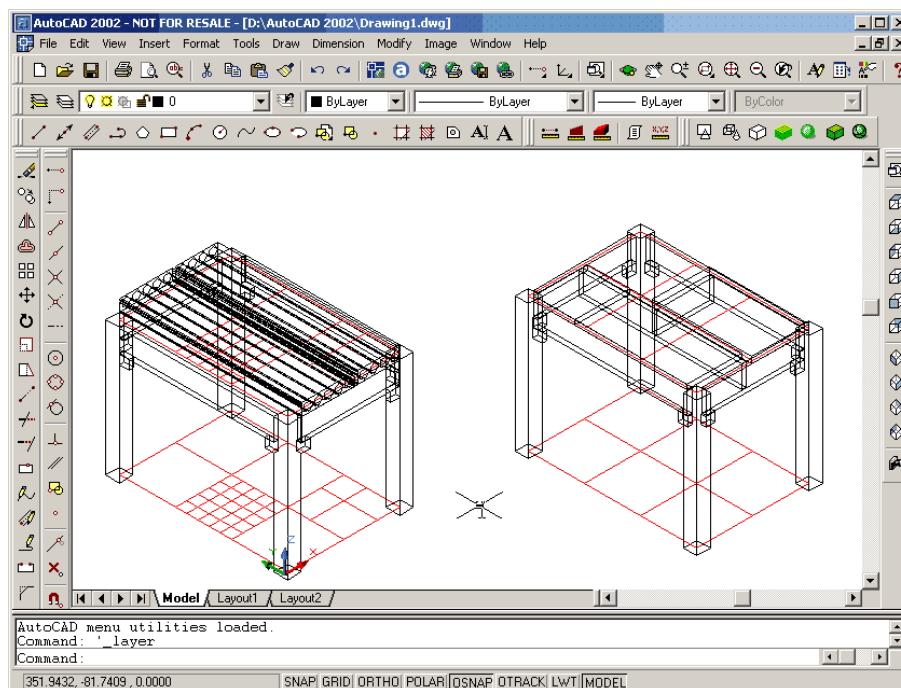
usado com o tipo de viga escolhida como apoio, o aplicativo informa ao usuário que sua escolha não é um procedimento usual e esse pode alterar ou não sua escolha.



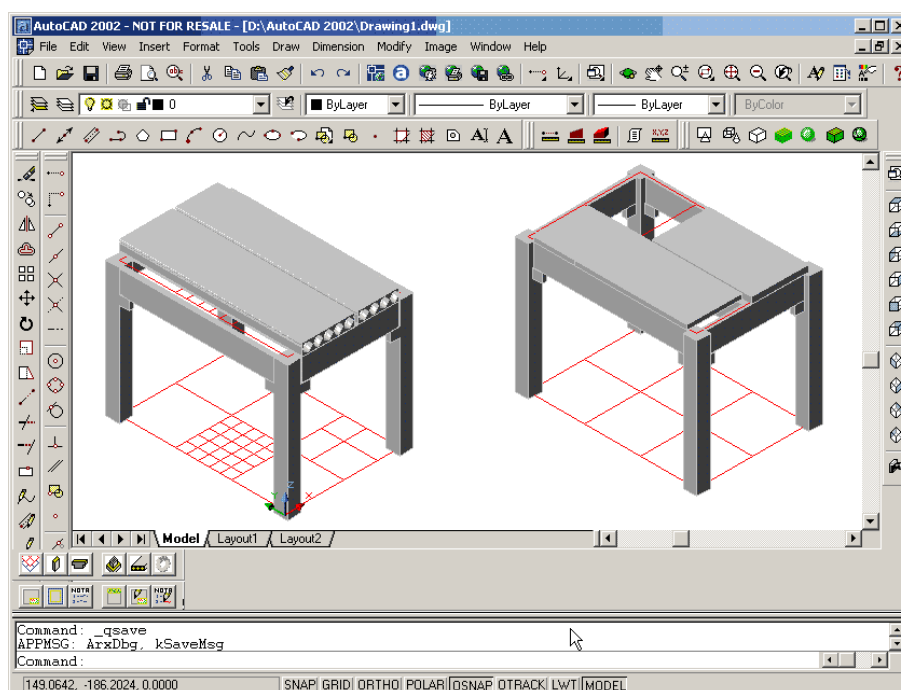
**FIGURA A. 17** – Alerta dado pelo aplicativo quanto a não adequação entre o tipo de painel a ser inserido e o tipo de viga de apoio

Não é necessário que o usuário indique o comprimento dos painéis de piso. Ao serem inseridos os painéis, o aplicativo calcula automaticamente a distância entre as vigas de apoio e insere aqueles nos pontos apropriado das vigas. Algumas vigas possuem abas para apoio de painéis. Nesse caso o aplicativo identifica o tipo de viga selecionada para servir de apoio e posiciona o painel sobre sua aba caso haja alguma. Caso contrário, o painel é apoiado sobre a alma da viga. Quando o painel é apoiado entre as abas de duas vigas, o comprimento do painel é calculado descontando as dimensões das almas das vigas de apoio e as folgas definidas na configuração geral do sistema.

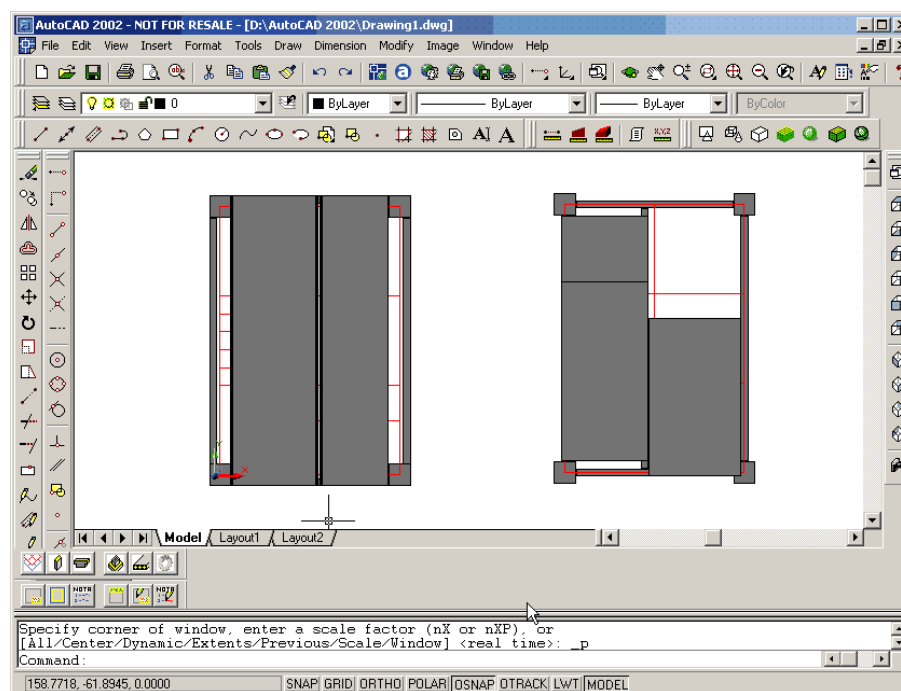
Após ter preenchido os campos do quadro de diálogo, o usuário pode acionar o botão "inserir" para escolher os apoios dos painéis ou o botão "cancelar" para interromper o comando e voltar à tela gráfica do AutoCAD.



**FIGURA A. 18** – Painéis inseridos sobre vigas: painel alveolar sobre vigas principais e lajes sobre vigas de diferentes tamanhos.



**FIGURA A. 19** – FIG. A.13 com o uso do recurso *Flat Shaded* de tratamento de sólidos do AutoCAD.



**FIGURA A. 20** – Vista superior da estrutura mostrada na FIG. A.14

## Comando GeraBloco

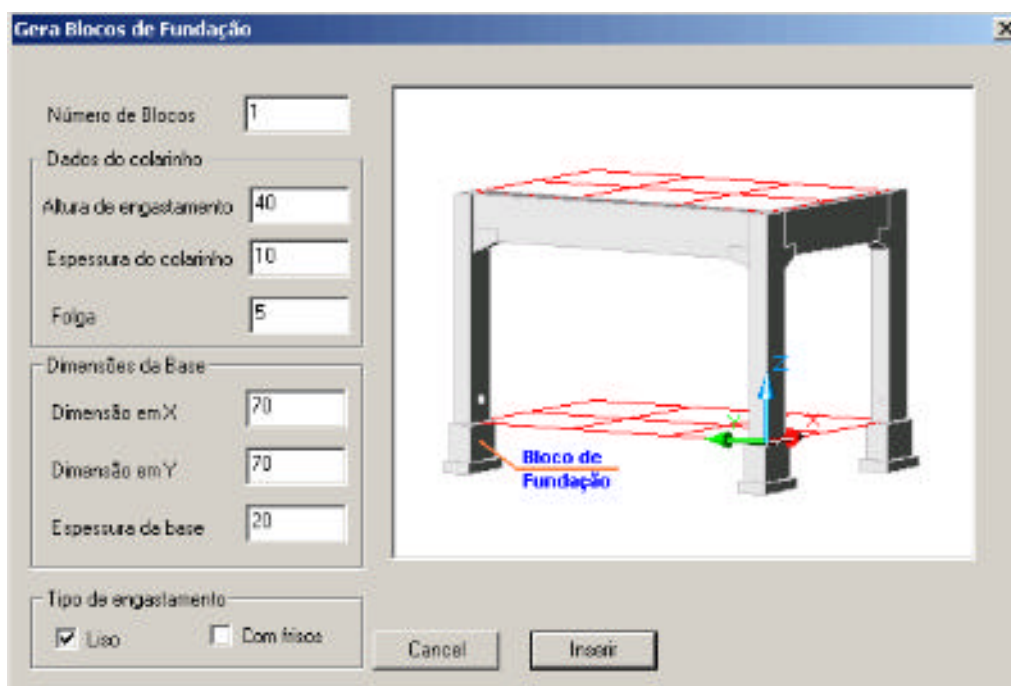


Acionar o botão **GeraBloco** da barra de ferramentas de modelamento ou digitar "GERABLOCO" na linha de comandos do AutoCAD.

**FIGURA A. 21** - Botão de acionamento do comando **GeraBloco**.

Com o quadro de diálogo **GeraBloco** aberta, escolher as dimensões mínimas dos blocos de fundação a serem criados. É permitido ao usuário gerar mais de um bloco durante o acionamento desse comando. As dimensões mínimas escolhidas pelo usuário serão usadas para todos os blocos criados durante uma seção do comando, sendo testadas para cada bloco individualmente. As seções mínimas definidas pela NBR 9062/85 são, então, calculadas conforme as dimensões dos pilares sob os quais os blocos serão criados. Caso as dimensões definidas pelo usuário sejam inferiores às definidas pela norma, o usuário é informado de que

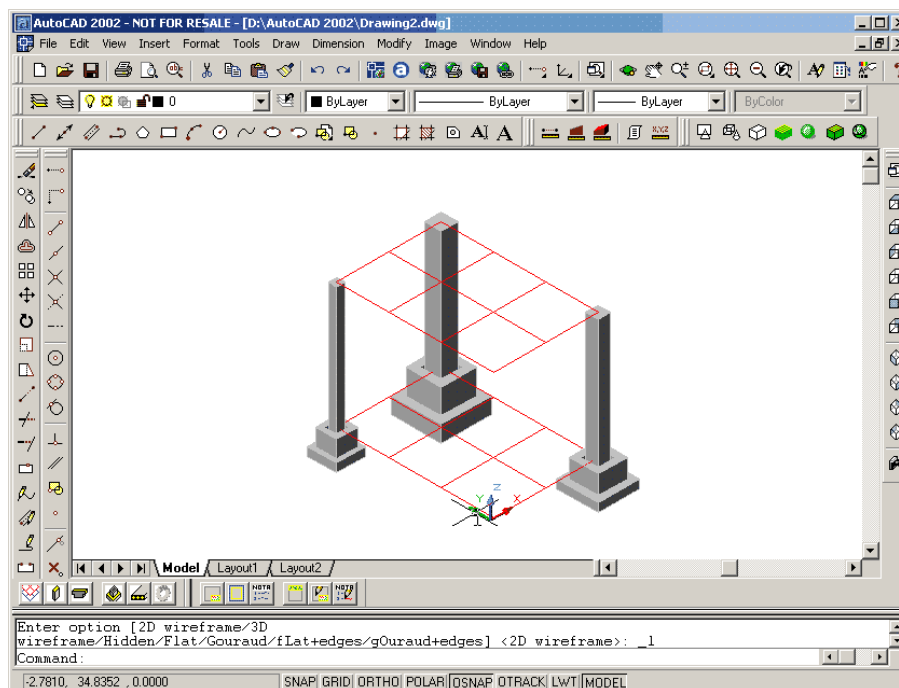
as dimensões são inadequadas, indicando também as dimensões mínimas a serem adotadas em cada caso.



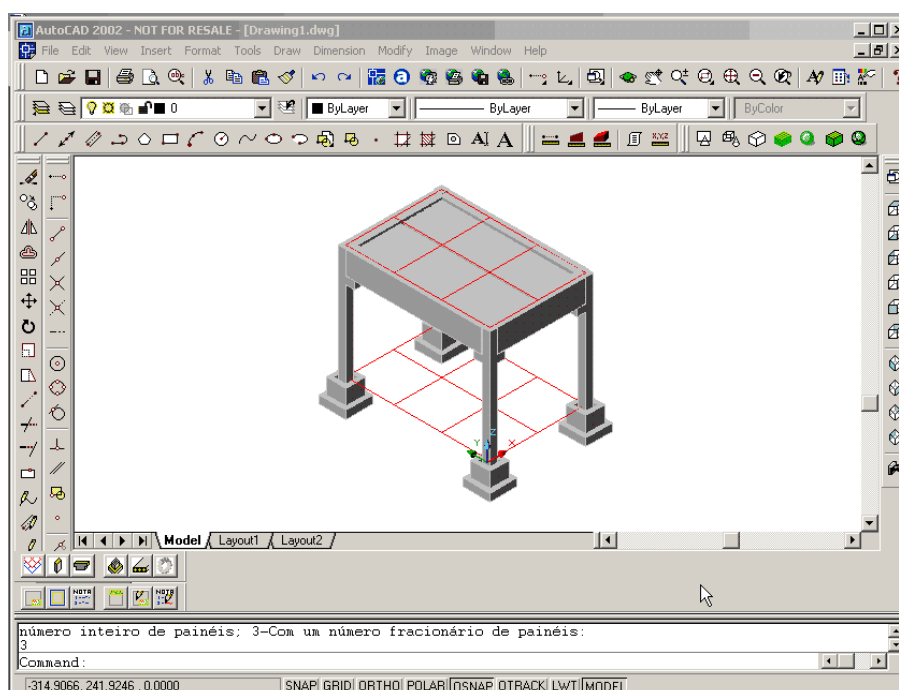
**FIGURA A. 22** - Quadro de diálogo do comando **GeraBloco**

Os blocos devem ser sempre criados como base para um pilar. Ao serem inseridos mais de um bloco de fundação durante uma única chamada do comando **GeraBloco**, o usuário deve selecionar os pilares que receberão os blocos de fundação individualmente. Para a inserção dos blocos são consideradas as dimensões do pilar sob o qual serão inseridos os blocos. O bloco é então dimensionado conforme as dimensões da seção transversal do pilar. Também o comprimento do pilar que recebe o bloco de fundação é alterado, sendo aquele aumentado do comprimento de engastamento adotado para o bloco de fundação.

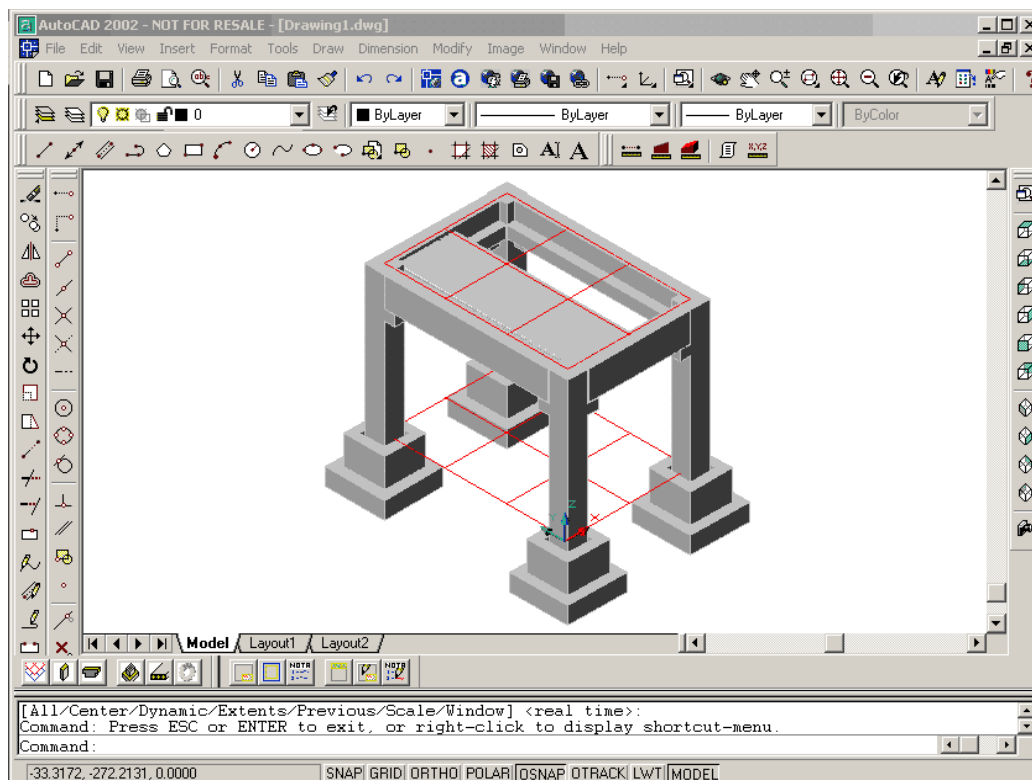
Após ter preenchido os campos do quadro de diálogo, o usuário pode acionar o botão "inserir" para escolher os pilares sob os quais serão inseridos os blocos de fundação, ou o botão "cancelar" para interromper o comando e voltar à tela gráfica do AutoCAD.



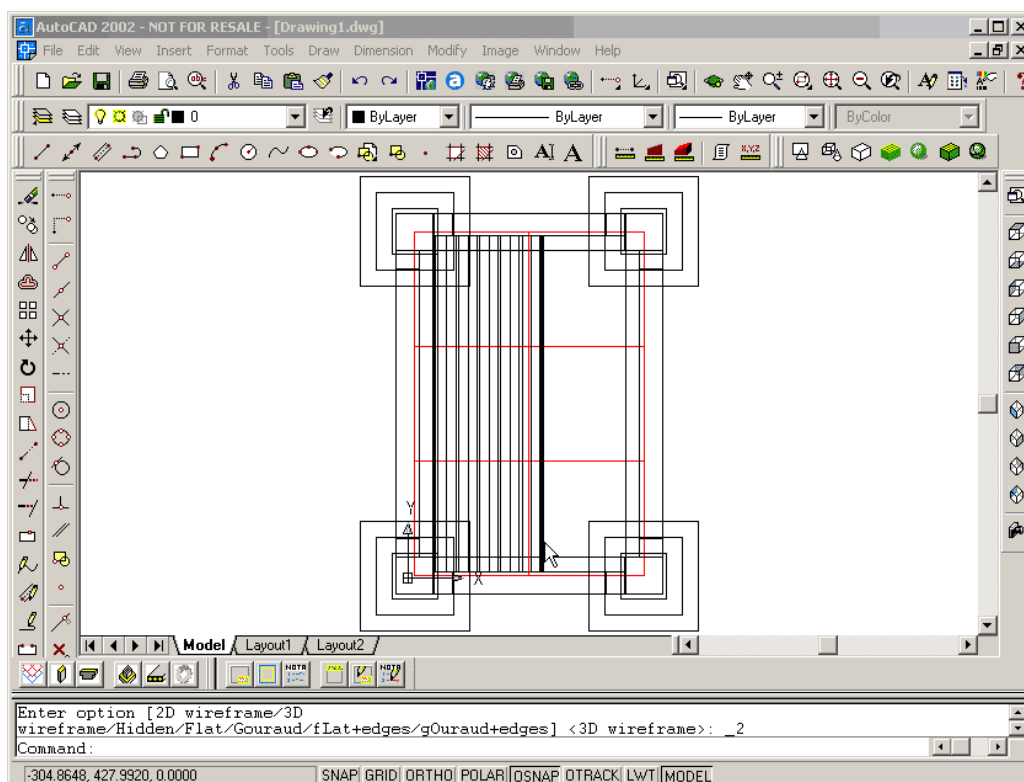
**FIGURA A. 23** – Blocos gerados pelo PREMOLD dimensionados de acordo com as dimensões dos pilares sob os quais são inseridos, respeitando as dimensões mínimas prescritas pela NBR 9062/85.



**FIGURA A. 24** – Estrutura gerada pelo PREMOLD com pilares com blocos de fundação, vigas VL e painel alveolar preenchendo todo o vão das vigas, apoiado sobre as abas das mesmas.



**FIGURA A. 25** – Inserção de apenas um painel alveolar sobre as vigas VL, apoiado sobre as abas das mesmas.



**FIGURA A. 26** – Vista superior da estrutura representada na FIG. A.19.

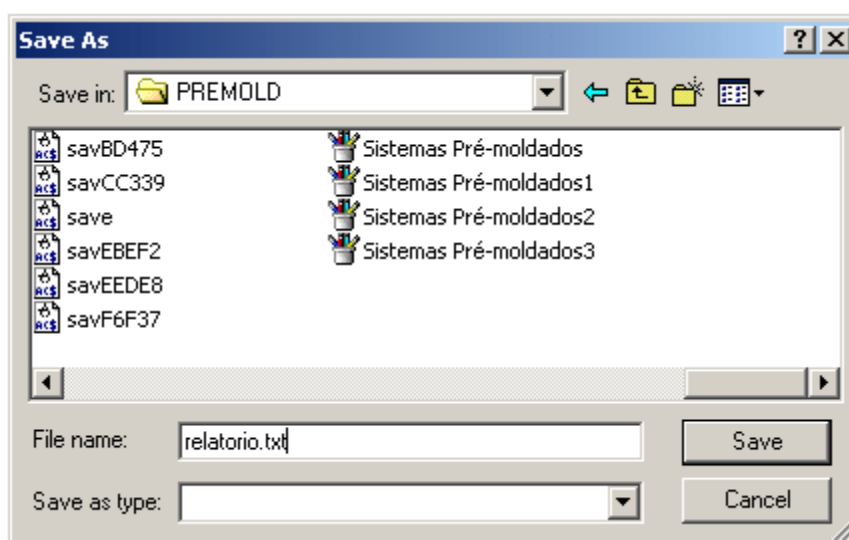
## Comando GeraRelatorio



Acionar o botão **GeraRelatorio** da barra de ferramentas de modelamento ou digitar "GERARELATORIO" na linha de comandos do AutoCAD.

**FIGURA A. 27** - Botão de acionamento do comando **GeraRelatorio**.

Ao acionar esse comando é aberto o quadro de diálogo do Windows para salvar um arquivo. O usuário deve então escolher o diretório para salvar a arquivo, o nome do arquivo a ser gerado e indicar ainda sua terminação como ".txt".



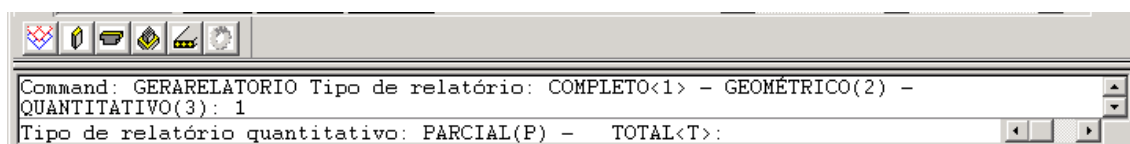
**FIGURA A. 28** – Quadro do Windows para gravar o arquivo que armazenará os dados do relatório gerado pelo PREMOLD.

Após a escolha do diretório e do nome do arquivo a ser gerado, o usuário é questionado sobre o tipo do relatório a ser gerado: G/Q, Geométrico ou Quantitativo. Ele deve então fazer a escolha através da linha de comando do AutoCAD conforme mostrado na FIG. A.22.

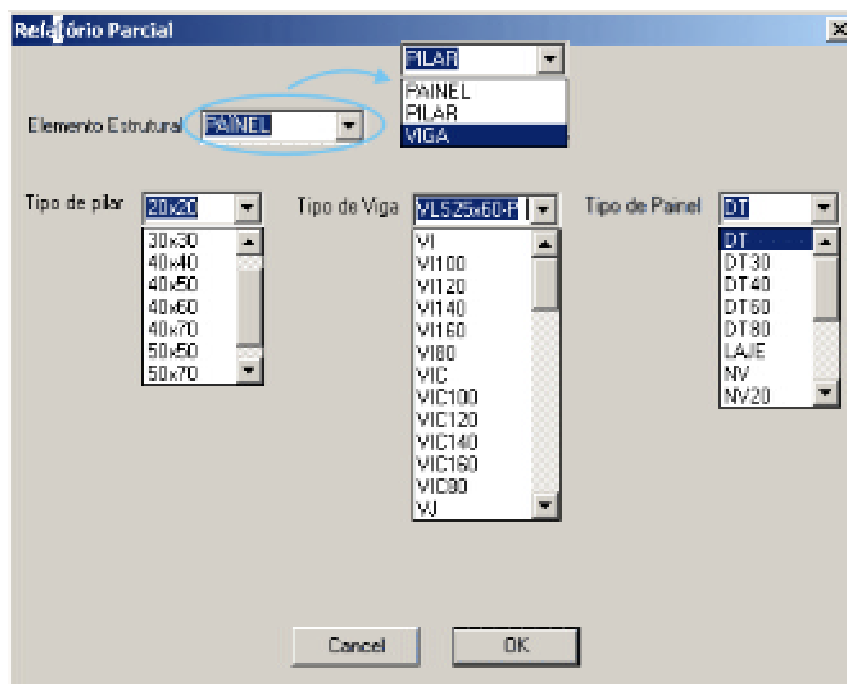


**FIGURA A. 29** – Linha de comando do AutoCad para escolha do tipo de relatório a ser gerado: G/Q, Geométrico ou Quantitativo.

O usuário pode ainda escolher entre o relatório Total aonde são gerados os dados sobre todos os elementos da estrutura criada ou o relatório Parcial quando o usuário pode escolher o tipo de elemento sobre o qual deseja obter os dados (FIG. A.29). Nesse caso é aberta a quadro de diálogo Relatório Parcial onde o usuário deve escolher o elemento através da barras de rolagem disponíveis (FIG. A.30).

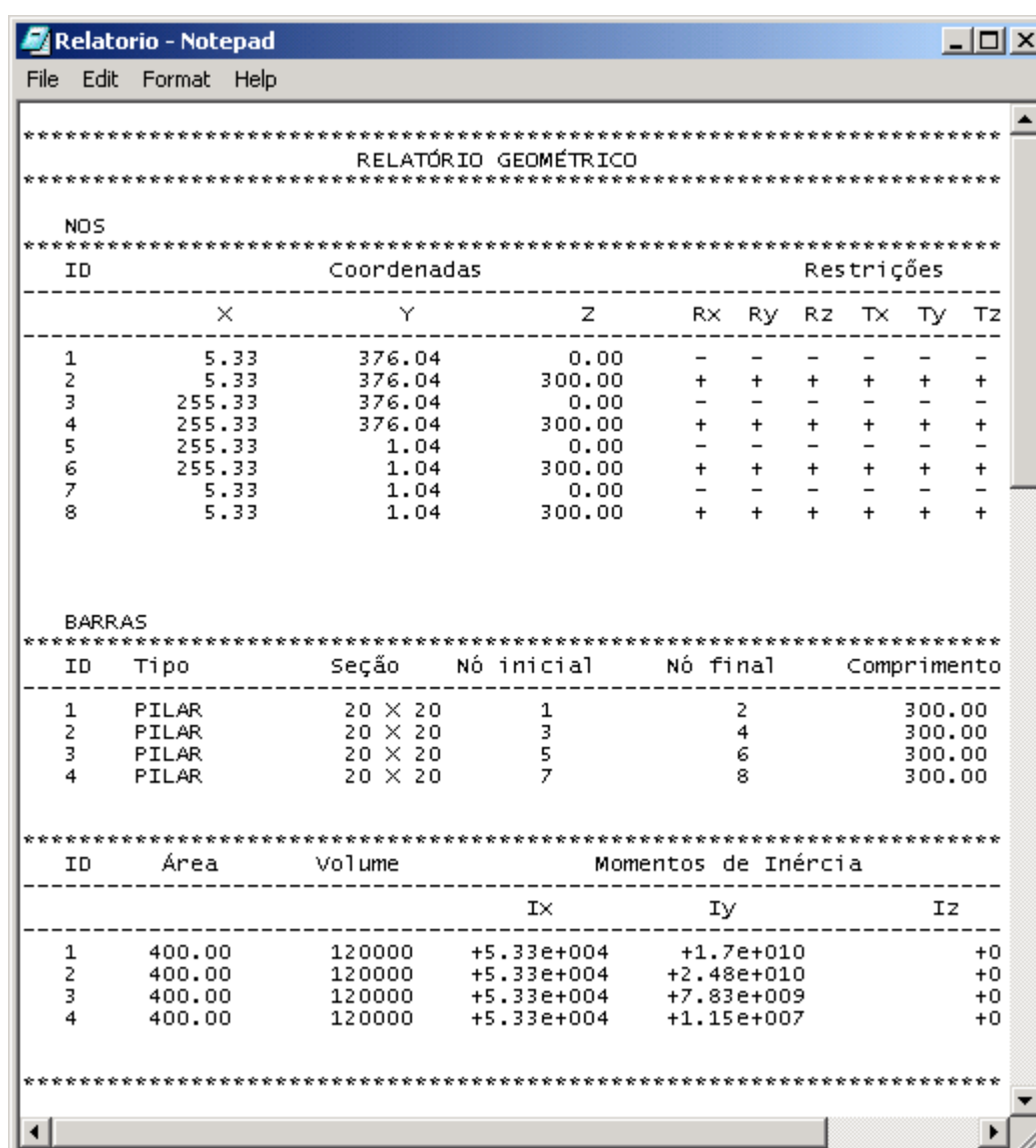


**FIGURA A. 30** - – Linha de comando do AutoCad para escolha do tipo de relatório a ser gerado: Parcial ou Total



**FIGURA A. 31** – Quadro de diálogo do comando **GeraRelatorio** para escolha dos elementos a serem listados no relatório Parcial.

O tipo de relatório escolhido é então gerado automaticamente pelo aplicativo, sendo gravado no arquivo indicado pelo usuário. Caso Já exista no diretório escolhido um arquivo de nome igual ao fornecido pelo usuário, ele é informado dessa existência e pode gravar os dados do relatório sobre esse arquivo ou fornecer outro nome para o novo arquivo a ser criado. Caso o usuário escolha gravar o relatório sobre um arquivo existente, os dados do arquivo original são perdidos, e esse é substituído pelo relatório gerado.



```

*****
                          RELATÓRIO GEOMÉTRICO
*****

NÓS
*****
ID          Coordenadas          Restrições
-----
          X          Y          Z          Rx  Ry  Rz  Tx  Ty  Tz
-----
1           5.33      376.04      0.00      -   -   -   -   -   -
2           5.33      376.04     300.00      +   +   +   +   +   +
3          255.33      376.04      0.00      -   -   -   -   -   -
4          255.33      376.04     300.00      +   +   +   +   +   +
5          255.33         1.04      0.00      -   -   -   -   -   -
6          255.33         1.04     300.00      +   +   +   +   +   +
7           5.33         1.04      0.00      -   -   -   -   -   -
8           5.33         1.04     300.00      +   +   +   +   +   +

BARRAS
*****
ID  Tipo      Seção  Nó inicial  Nó final  Comprimento
-----
1  PILAR      20 X 20  1          2          300.00
2  PILAR      20 X 20  3          4          300.00
3  PILAR      20 X 20  5          6          300.00
4  PILAR      20 X 20  7          8          300.00

*****
ID  Área      Volume      Momentos de Inércia
-----
          Ix          Iy          Iz
-----
1  400.00     120000     +5.33e+004     +1.7e+010     +0
2  400.00     120000     +5.33e+004     +2.48e+010     +0
3  400.00     120000     +5.33e+004     +7.83e+009     +0
4  400.00     120000     +5.33e+004     +1.15e+007     +0

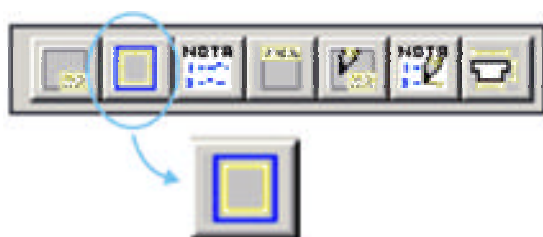
*****

```

FIGURA A. 32 – Exemplo de relatório Geométrico gerado pelo PREMOLD

## Comandos Auxiliares para o Detalhamento

### Comando GeraFormato



Acionar o botão **GeraFormato** da barra de ferramentas de detalhamento ou digitar "GERAFORMATO" na linha de comandos do AutoCAD.

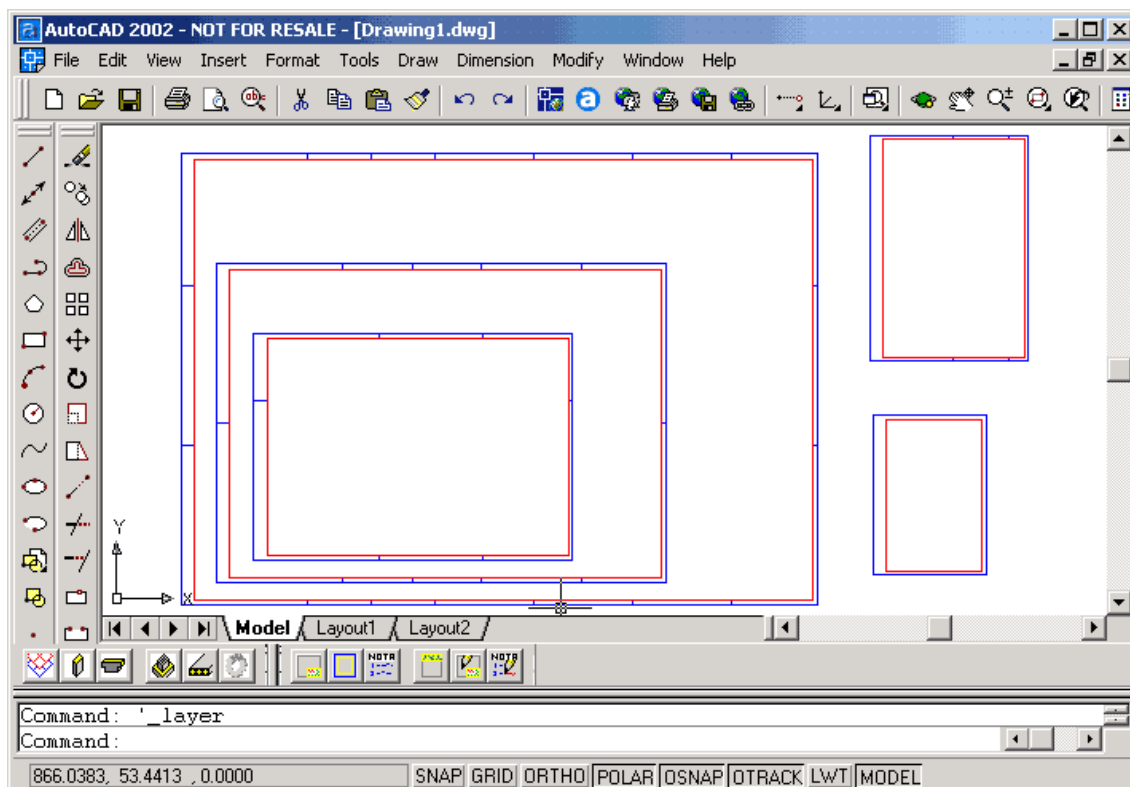
**FIGURA A. 33** - Botão de acionamento do comando **GeraFormato**

Ao acionar o comando aparece na linha de comando do AutoCAD as opções de formatos (A0, A1, A2, A3, A4) ou A5 para um formato de dimensões a serem inseridas pelo usuário. Somente os números são fornecidos nesse comando. Ao selecionar a opção 5 (personalizado), informar através da linha de comando as dimensões da base, altura e das quatro margens do formato.

Após preencher os dados do formato o usuário deve indicar o ponto de inserção do mesmo. O formato é inserido sempre pelo seu vértice inferior esquerdo. Esse ponto pode ser fornecido digitando-se suas coordenadas na linha de comando do AutoCAD ou escolhendo um ponto na tela gráfica.

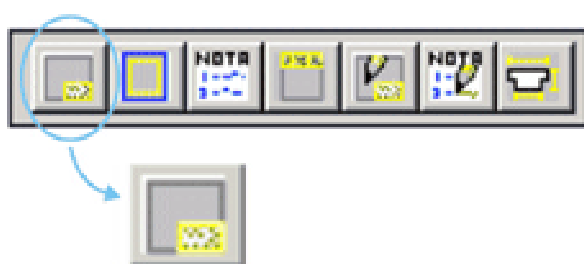
Além das margens interna e externa, esse comando desenha as marcas de dobra do desenho segundo a NBR 10068, "Folha de desenho - Leiaute e dimensões".

Esse comando permite o acionamento automático dos comandos **GeraLegenda** e **GeraCarimbo** também descritos neste manual. O usuário é questionado se deseja esse acionamento automático durante a execução do comando **GeraFormato**.



**FIGURA A. 34** – Formatos gerados pelo comando **GeraFormato**.

## Comando GeraLegenda



Acionar o botão **GeraLegenda** da barra de ferramentas de detalhamento ou digitar "GERALEGENDA" na linha de comandos do AutoCAD.

**FIGURA A. 35** - Botão de acionamento do comando **GeraLegenda**.

Com o quadro de diálogo **GeraLegenda** aberta, preencher os campos da legenda a serem preenchidos.

**Premold - Editor de legendas**

Dados da legenda

Premold

Obra

Título

Engenheiro Responsável

La largura

Altura

Calculo: 01-04-2002

Projeto: 01-04-2002

Desenho: 01-04-2002

Verificação: 01-04-2002

Aprovação: 01-04-2002

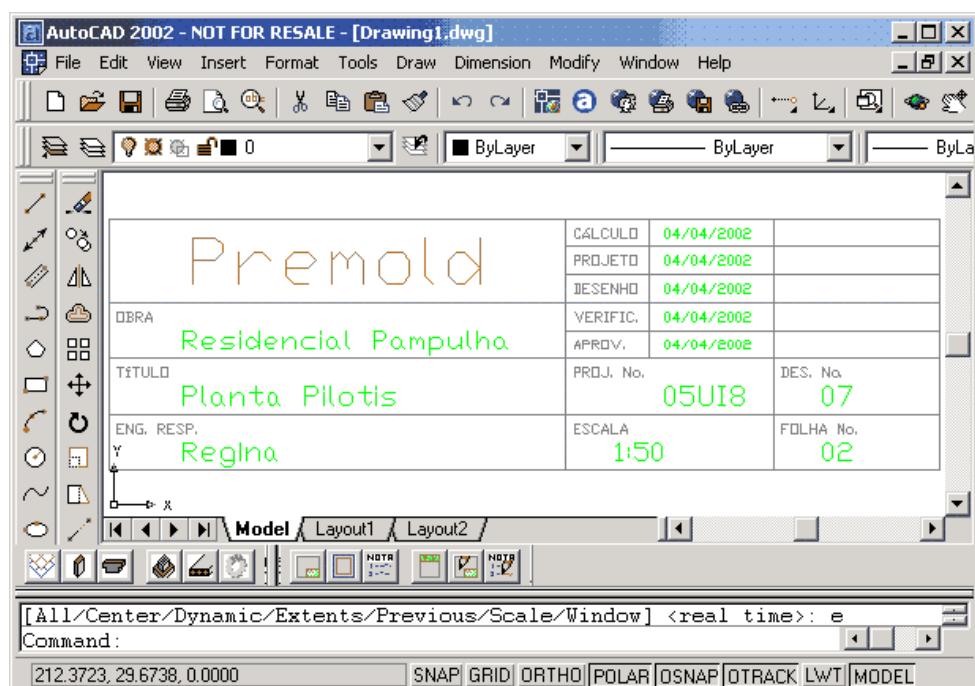
Proj. No. Des. No.

Escala Folha No.

Desenhar Cancelar

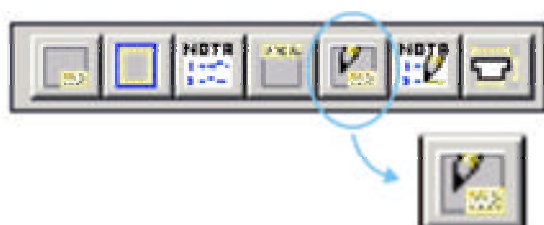
**FIGURA A. 36** – Quadro de diálogo do comando **GeraLegenda**.

Após ter preenchido os campos do quadro de diálogo, o usuário pode acionar o botão "inserir" para inserir a legenda cujos campos foram previamente preenchidos, ou o botão "cancelar" para interromper o comando e voltar à tela gráfica do AutoCAD.



**FIGURA A. 37** – Legenda gerada pelo comando **GeraLegenda** do PREMOLD.

## Comando EditaLegenda



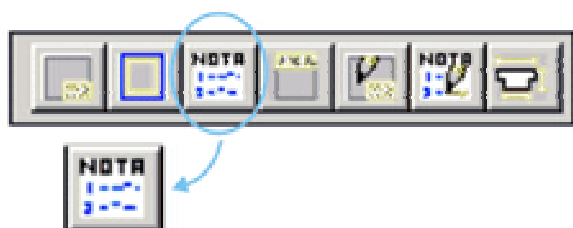
Acionar o botão **EditaLegenda** da barra de ferramentas de detalhamento ou digitar "EDITALEGENDA" na linha de comandos do AutoCAD.

**FIGURA A. 38** - Botão de acionamento do comando **EditaLegenda**.

Esse comando aciona o mesmo quadro de diálogo utilizado pelo comando **GeraLegenda**. Com esse quadro de diálogo aberto, alterar os campos da legenda previamente preenchidos.

Após ter alterado os campos do quadro de diálogo, o usuário pode acionar o botão "inserir" para inserir a legenda alterada no lugar da legenda original, ou o botão "cancelar" para interromper o comando e voltar à tela gráfica do AutoCAD.

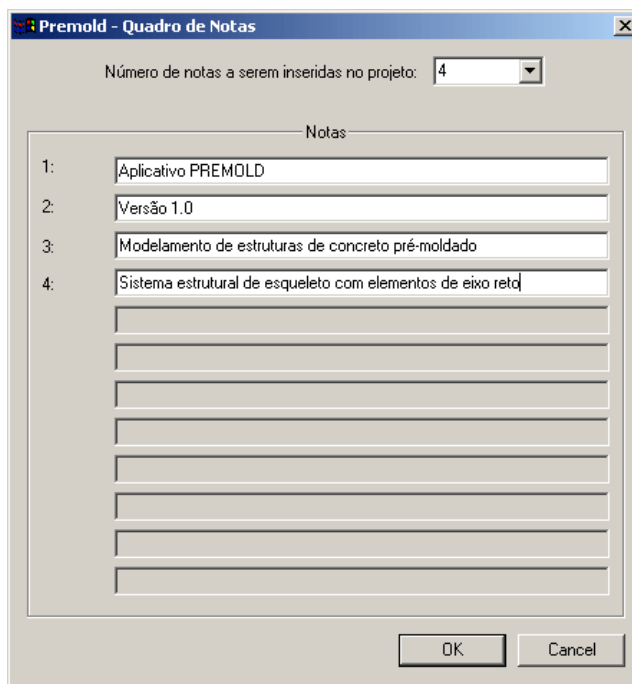
## Comando GeraNota



Acionar o botão **GeraNota** da barra de ferramentas de detalhamento ou digitar "GERANOTA" na linha de comandos do AutoCAD.

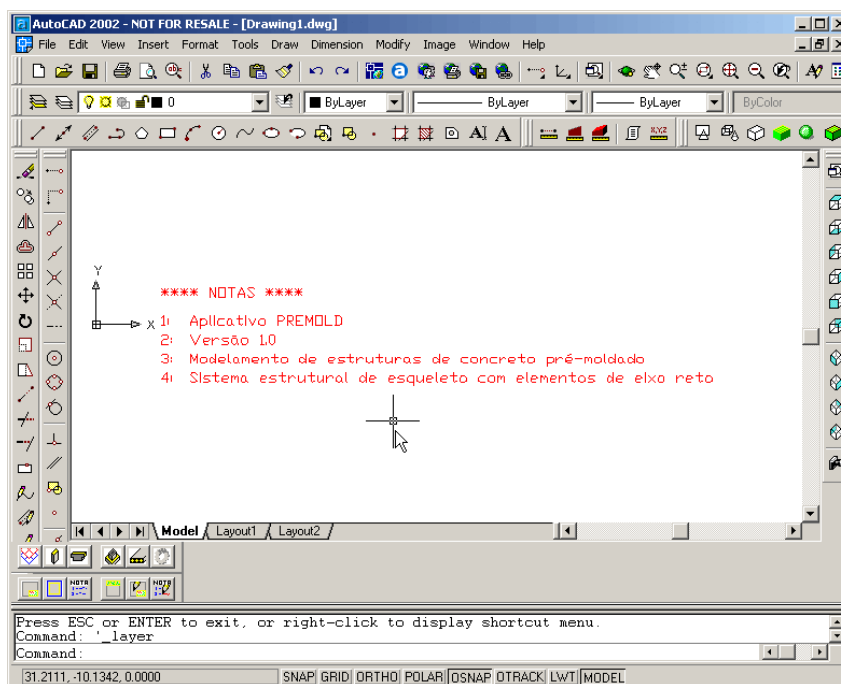
**FIGURA A. 39** - Botão de acionamento do comando **GeraNota**.

Com o quadro de diálogo **GeraNota** aberta, preencher nos devidos campos as notas a serem inseridas.



**FIGURA A. 40 -** Quadro de diálogo do comando **GeraNota**.

Após ter preenchido os campos do quadro de diálogo, o usuário pode acionar o botão "inserir" para inserir as notas cujos campos foram previamente preenchidos, ou o botão "cancelar" para interromper o comando e voltar à tela gráfica do AutoCAD.



**FIGURA A. 41 –** Nota criada pelo comando **GeraNota** do PREMOLD.

## Comando EditaNota



Acionar o botão **EditaNota** da barra de ferramentas de detalhamento ou digitar "EDITANOTA" na linha de comandos do AutoCAD.

**FIGURA A. 42** - Botão de acionamento do comando **EditaNota**.

Através da linha de comando do AutoCAD, informar o número da nota a ser alterada e preencher a linha de comando com os dizeres a serem mostrados na enumeração.

## Comando GeraCarimbo



Acionar o botão **GeraCarimbo** da barra de ferramentas de detalhamento ou digitar "GERACARIMBO" na linha de comandos do AutoCAD.

**FIGURA A. 43**- Botão de acionamento do comando **GeraCarimbo**.

O usuário deve selecionar, através da linha de comando do AutoCAD, o vértice para a inserção do carimbo gerado. O carimbo pode ser inserido pelos seus vértices superior ou inferior, ambos do lado direito. Após a escolha do vértice de inserção do carimbo, o usuário deve fornecer, através da linha de comando do AutoCAD ou através da tela gráfica, o ponto onde aquele será inserido. Após a escolha do ponto de inserção, o usuário tem a opção de preencher os campos do carimbo através da linha de comando.

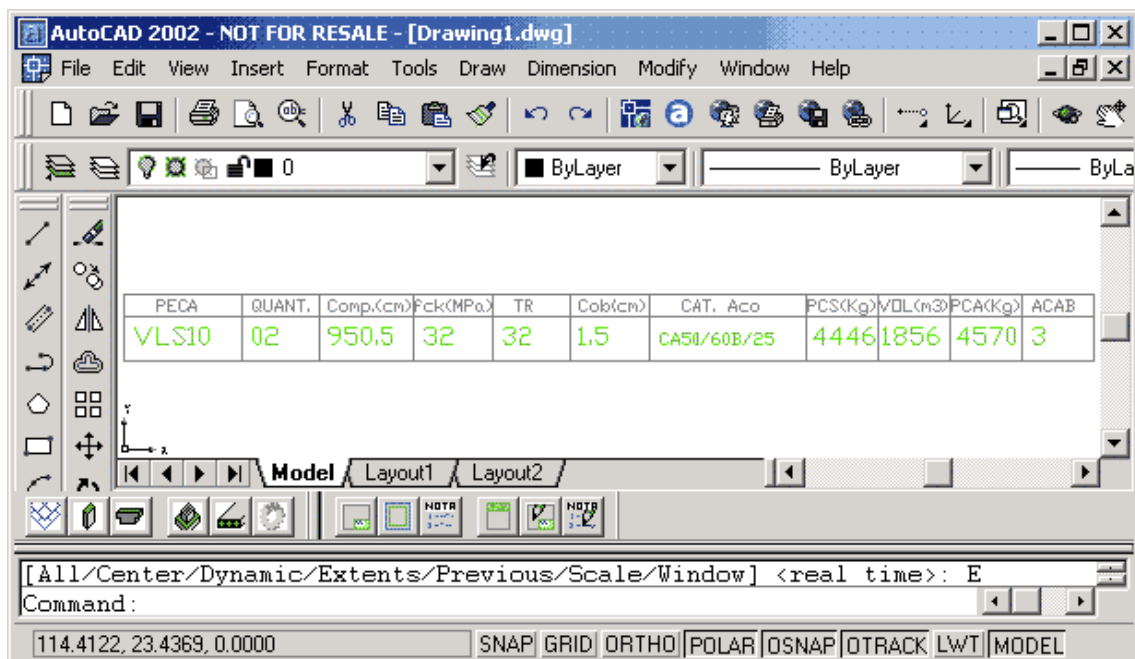


FIGURA A. 44 – Carimbo gerado pelo comando **GeraCarimbo** do PREMOLD.

## Comando de Detalhamento

### Comando GeraDesenho



Acionar o botão **GeraDesenho** da barra de ferramentas de detalhamento ou digitar "GERADESENHO" na linha de comandos do AutoCAD.

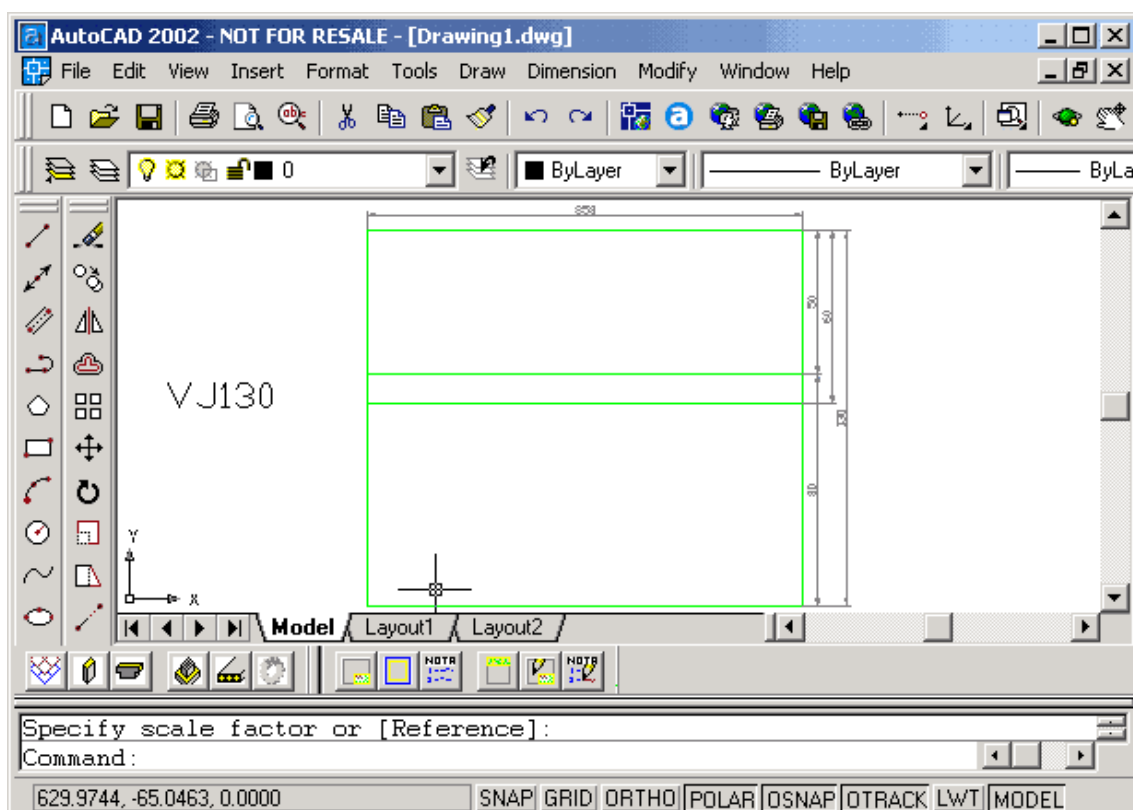
FIGURA A. 45 - Botão de acionamento do comando **GeraDesenho**

Esse comando gera o desenho de forma dos elementos lineares da estrutura. Sua implementação para elementos planos ainda está em desenvolvimento.

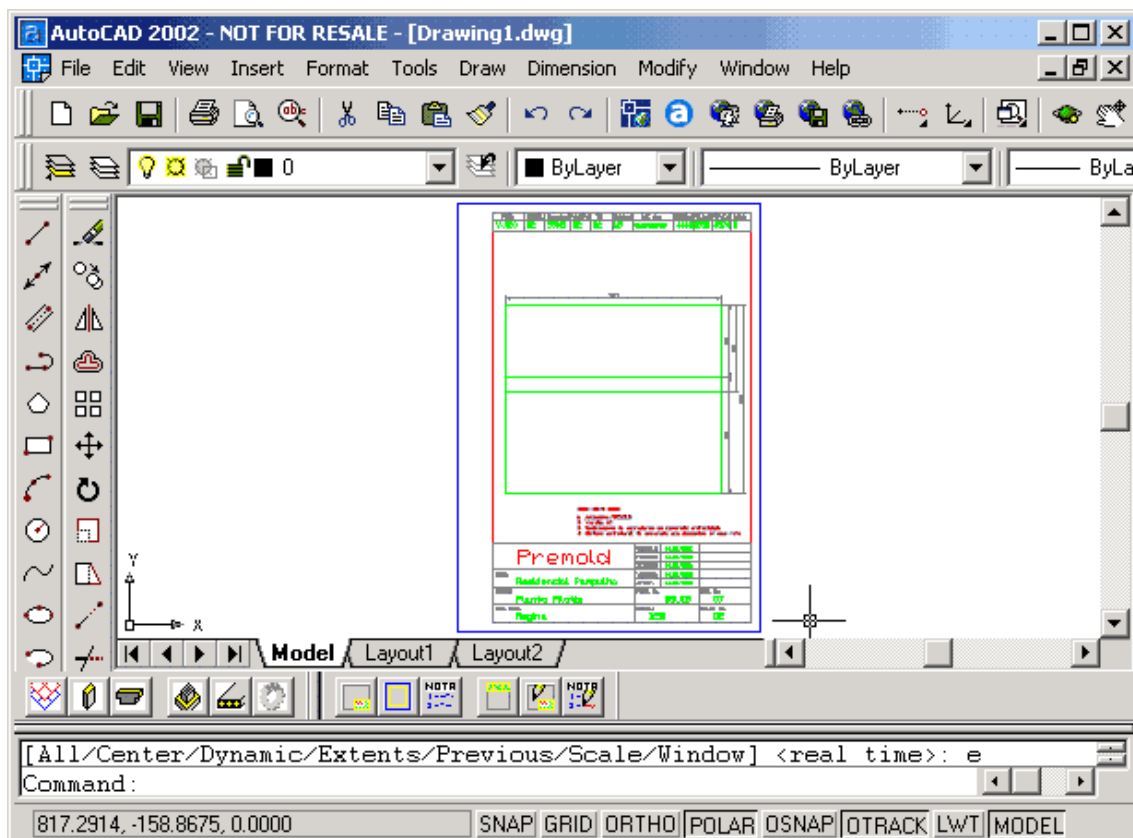
Esse comando permite o acionamento automático do comando **GeraFormato** e descrito neste manual. O usuário é questionado se deseja esse acionamento automático durante a execução do comando **GeraDesenho**.

Após ter escolhido o tipo de formato e preenchido os campos da legenda e do carimbo, caso tenha escolhido sua geração automática, o usuário pode escolher o elemento da estrutura a ser detalhado informando na linha de comando do AutoCAD o número da barra contida pelo elemento.

Os desenhos de forma desse elemento com inserção de dimensionamentos serão gerados automaticamente pelo aplicativo, devendo o usuário escolher na tela gráfica o ponto de inserção dos mesmos.



**FIGURA A. 46** – Desenho de forma de figa VJ criado pelo comando **GeraDesenho** do PREMOLD.



**FIGURA A. 47** – Formato, legenda, nota, carimbo e desenho deforma criados pelo PREMOLD.

## CRIAÇÃO AUTOMÁTICA DO MODELO UNIFILAR

O PREMOLD promove a criação automática do modelo unifilar segundo o sistema estrutural reticulado com elementos de eixo reto para edifícios de múltiplos pavimentos, com pilares engastados na fundação e vigas articuladas nos pilares.

Os pilares a serem inseridos no modelo recebem um elemento do tipo barra que possui um nó inicial e um nó final. A barra é posicionada em um eixo paralelo ao comprimento do pilar, passando pelo seu centro geométrico. O nó inicial é definido como aquele situado sobre o ponto de inserção do pilar, enquanto o nó final é aquele situado no outro extremo do comprimento do pilar.

O nó inicial do pilar é configurado com rotações e translações impedidas na direção dos três eixos globais (X, Y, Z), indicando um pilar engastado na base. Seu nó final recebe inicialmente liberação à rotação e à translação na direção dos três eixos globais. À medida que são inseridas vigas no topo do pilar as translações do mesmo vão sendo automaticamente restringidas na direção do comprimento das vigas inseridas, porém a rotação continua liberada devido às ligações articuladas com as vigas previstas no modelo estrutural adotado.

As vigas também recebem um elemento do tipo barra com dois nós ao serem inseridas no modelo estrutural. O elemento de apoio inicial da viga é aquele entre os elementos de apoio cujo ponto de inserção possui as menores coordenadas globais. A barra da viga é posicionada em um eixo paralelo ao comprimento da viga, passando pelo seu centro geométrico. O nó inicial é definido como aquele situado sobre o ponto de interseção do prolongamento da barra da viga com a barra do pilar de apoio inicial, enquanto o nó final é aquele situado sobre o ponto de interseção do prolongamento da barra da viga com a barra do pilar de apoio final.

Os painéis de piso recebem um elemento do tipo placa ao serem inseridos no modelo estrutural. Uma placa, nesse caso, é uma superfície finita delimitada por vértices. A viga de apoio inicial do painel é aquela entre as vigas de apoio cujo ponto de inserção possui as menores coordenadas globais. A placa do painel é definida de modo que o seu vértice inicial corresponde ao ponto de inserção da viga de apoio inicial. O vértice seguinte é aquele obtido pela soma do vértice inicial à largura do painel multiplicada pelo vetor que indica o eixo Z local da viga de apoio inicial. O terceiro vértice do painel corresponde ao ponto de inserção da viga de apoio final do painel e o último vértice é obtido pela soma do terceiro vértice à largura do painel multiplicada pelo vetor que indica o eixo Z local da viga de apoio final.

Os elementos do modelo unifilar são numerados automaticamente pelo aplicativo com numerações independentes para cada tipo de elemento (nós, barras e placas). A numeração das barras e placas passa a identificar o elemento estrutural que as possui (pilar, viga ou painel de piso).

As barras podem ser definidas como barras de pórtico (*Frame*), secundárias (*Secondary*) ou de contraventamento (*Bracing*). As placas podem ser definidas como vão preenchido por laje (*Slab*), vão vazio (*Empty*) ou vão de escada (*Stair*). Os pilares e as vigas apoiadas em pilares

são criados como barras principais. As vigas apoiadas sobre outras vigas são criadas como barras secundárias. Os painéis são criados como lajes.

Nos relatórios é possível obter-se o número dos nós, das barras e das placas inseridas no desenho, sua posição, a incidência nodal, o tipo de elemento que recebe tais elementos, etc. Também através do comando *list* do AutoCAD é possível visualizar vários dados das entidades personalizadas referentes ao modelo unifilar como: numeração de barras e nós ou de placas, coordenadas nodais ou dos vértices das placas, tipo do elemento, tipo de entidade personalizada que o representa, etc.

## EXEMPLOS DE MODELAMENTO GERADO PELO PREMOLD

### Exemplo 01

Geração de quatro pilares de seção 20x20 inseridos pelo seu centro geométrico.

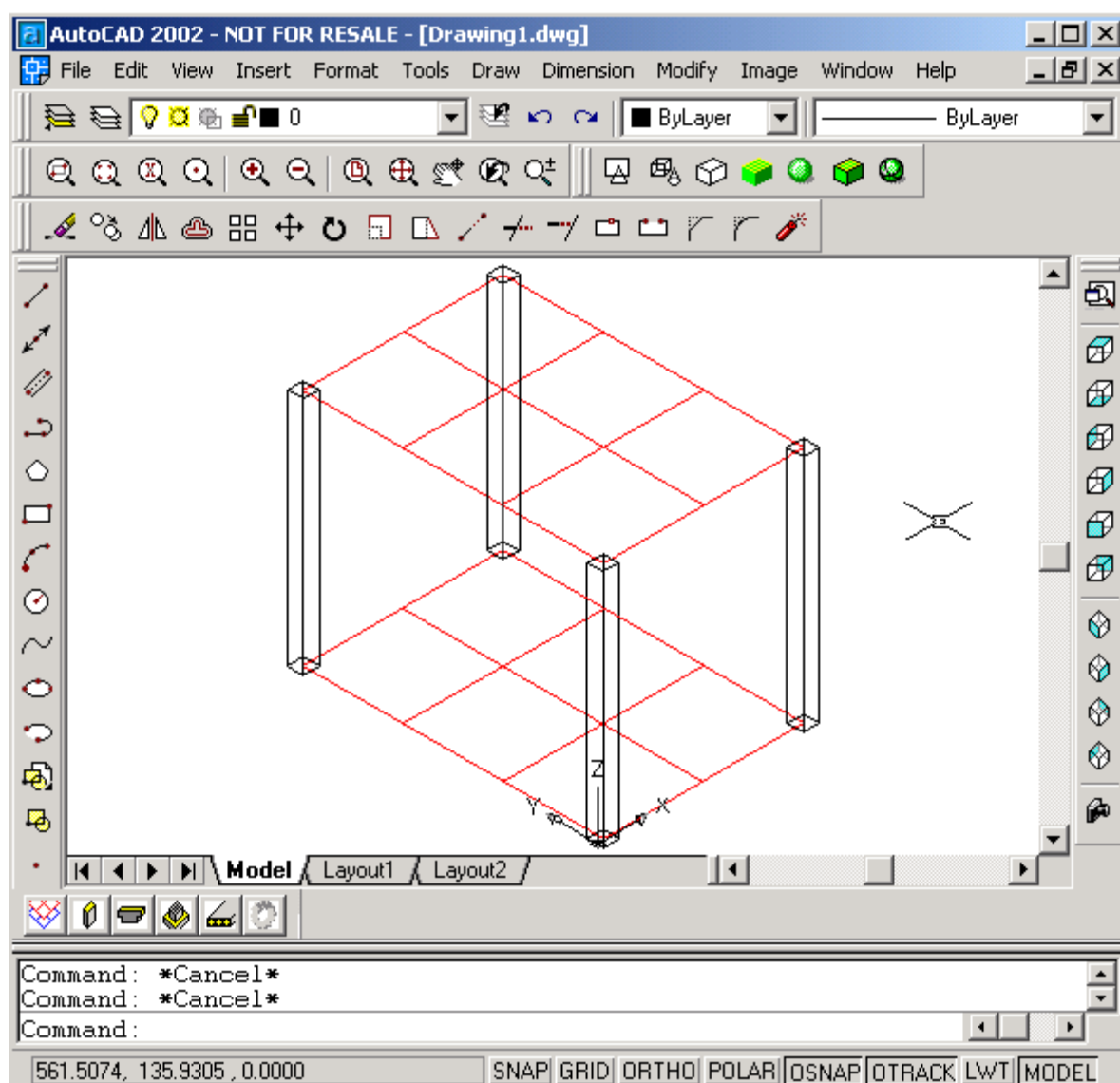


FIGURA A. 48 – Pilares gerados pelo PREMOLD

Relatório G/Q total gerado sobre os quatro pilares inseridos na estrutura. Observar as restrições nodais com os nós iniciais dos pilares restringidos quanto à translação e à rotação nas direções dos eixos globais X, Y, Z. Os nós finais são inseridos com liberações quanto à translação e à rotação nas direções dos eixos globais X, Y, Z.

```

*****
***** RELATÓRIO GEOMÉTRICO *****
*****
NOS
*****
ID          Coordenadas          Restrições
-----
          X          Y          Z          Rx  Ry  Rz  Tx  Ty  Tz
-----
1          5.33          376.04          0.00          -   -   -   -   -   -
2          5.33          376.04          300.00          +   +   +   +   +   +
3         255.33          376.04          0.00          -   -   -   -   -   -
4         255.33          376.04          300.00          +   +   +   +   +   +
5         255.33           1.04          0.00          -   -   -   -   -   -
6         255.33           1.04          300.00          +   +   +   +   +   +
7          5.33           1.04          0.00          -   -   -   -   -   -
8          5.33           1.04          300.00          +   +   +   +   +   +

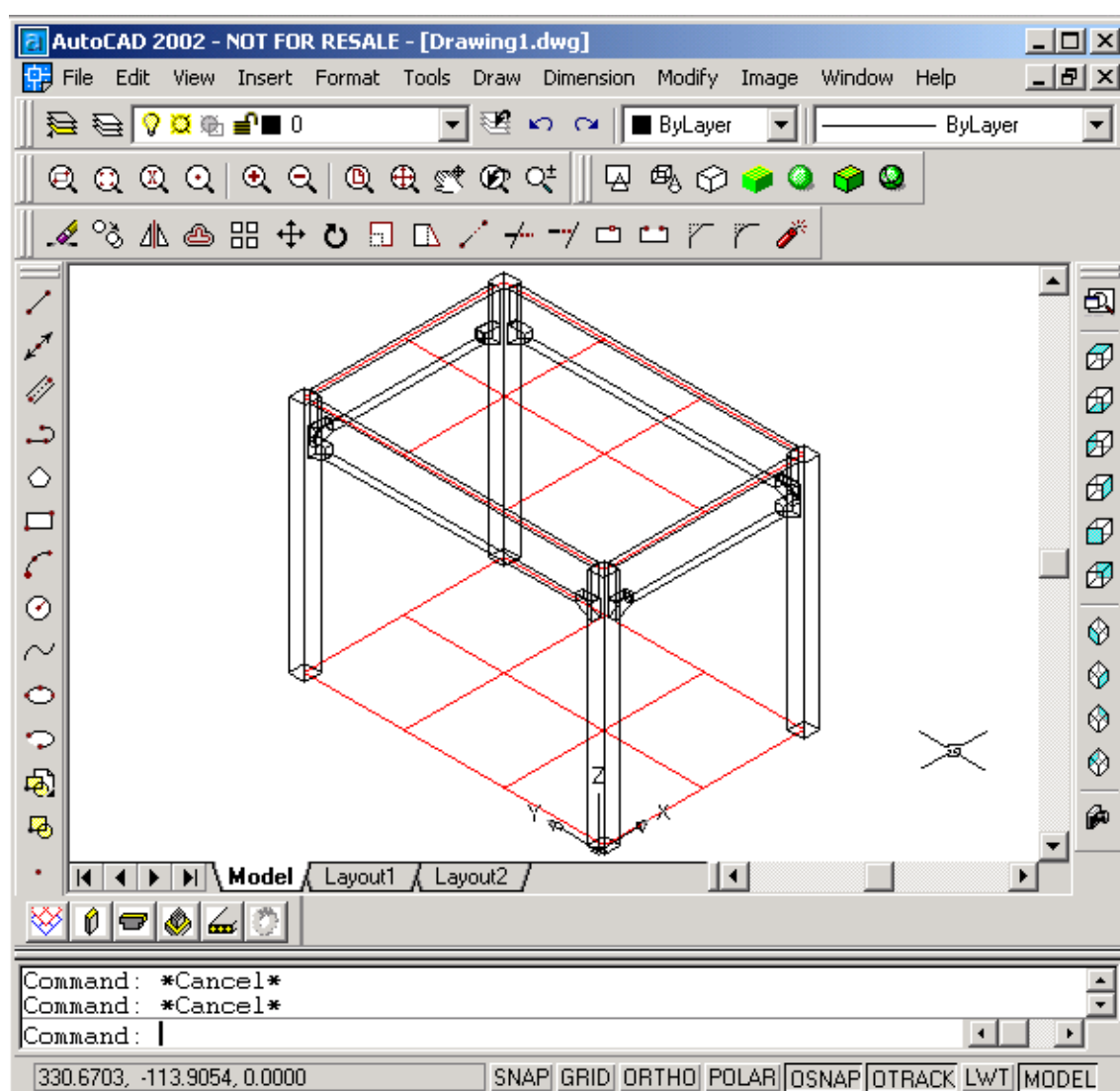
BARRAS
*****
ID  Tipo      Seção      Nó inicial      Nó final      Comprimento
-----
1  PILAR      20 X 20      1              2              300.00
2  PILAR      20 X 20      3              4              300.00
3  PILAR      20 X 20      5              6              300.00
4  PILAR      20 X 20      7              8              300.00

*****
ID  Área      volume      Momentos de Inércia
-----
          Ix          Iy          Iz
-----
1  400.00      120000      +5.33e+004      +1.7e+010      +0
2  400.00      120000      +5.33e+004      +2.48e+010      +0
3  400.00      120000      +5.33e+004      +7.83e+009      +0
4  400.00      120000      +5.33e+004      +1.15e+007      +0
*****

```

**FIGURA A. 49** – Relatório sobre a estrutura formada por quatro pilares gerados pelo PREMOLD

Vigas retangulares de seção 10x50 geradas pelo PREMOLD sobre pilares com consolos trapezoidais semi-embutidos, inseridas centradas em relação ao eixo vertical dos pilares.



**FIGURA A. 50** – Vigas retangulares geradas pelo PREMOLD.

Relatório G/Q total gerado sobre os quatro pilares e quatro vigas inseridas na estrutura. Observar as restrições nodais com os nós iniciais dos pilares restringidos quanto à translação e à rotação nas direções dos eixos globais X, Y, Z. Os nós finais dos pilares tiveram suas restrições alteradas conforme a inserção de vigas com altura igual a ao comprimento do pilar restringindo sua translação nas direções dos eixos longitudinal das vigas.

```

*****
                          RELATÓRIO GEOMÉTRICO
*****

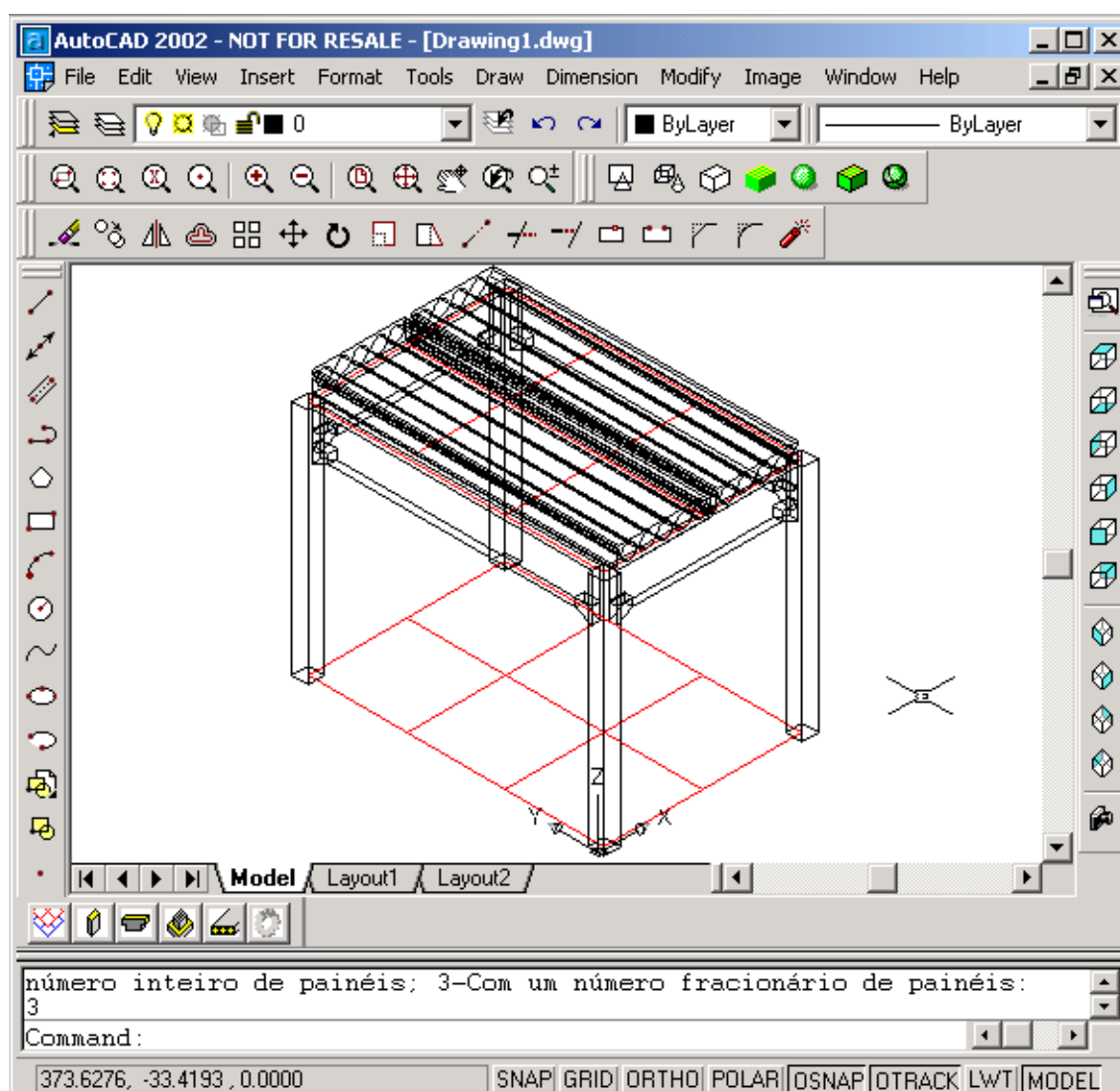
NÓS
*****
ID          Coordenadas          Restrições
-----
          X          Y          Z          RX  Ry  RZ  Tx  Ty  Tz
-----
1           5.90       376.52       0.00       -  -  -  -  -  -
10          5.90       365.52      275.00       -  -  +  -  +  -
11         16.90         1.52      275.00       -  -  +  +  -  -
12        244.90         1.52      275.00       -  -  +  +  -  -
13        255.90         12.52     275.00       -  -  +  -  +  -
14        255.90        365.52     275.00       -  -  +  -  +  -
15         16.90        376.52     275.00       -  -  +  +  -  -
16        244.90        376.52     275.00       -  -  +  +  -  -
2           5.90       376.52      300.00       +  +  +  -  -  +
3        255.90        376.52         0.00       -  -  -  -  -  -
4        255.90        376.52      300.00       +  +  +  -  -  +
5        255.90         1.52         0.00       -  -  -  -  -  -
6        255.90         1.52      300.00       +  +  +  -  -  +
7           5.90         1.52         0.00       -  -  -  -  -  -
8           5.90         1.52      300.00       +  +  +  -  -  +
9           5.90        12.52     275.00       -  -  +  -  +  -

BARRAS
*****
ID  Tipo      seção      Nó inicial  Nó final  Comprimento
-----
1  PILAR     20 X 20      1           2         300.00
2  PILAR     20 X 20      3           4         300.00
3  PILAR     20 X 20      5           6         300.00
4  PILAR     20 X 20      7           8         300.00
5  VIGA      VR10x50      9          10        353.00
6  VIGA      VR10x50     11         12        228.00
7  VIGA      VR10x50     13         14        353.00

```

**FIGURA A. 51** - Relatório sobre a estrutura formada por quatro pilares e quatro vigas geradas pelo PREMOLD.

Painel alveolar gerado pelo PREMOLD, inserido sobre vigas retangulares preenchendo todo o vão da viga de apoio. O painel alveolar tem largura padrão de 1,25m e as vigas têm comprimento de 2,28m, assim foram inseridos dois painéis sendo um inteiro e outro cortado no sentido longitudinal para preencher o vão da viga.



**FIGURA A. 52** – Painel alveolar gerado pelo PREMOLD.

Relatório sobre a estrutura formada por quatro pilares, quatro vigas e dois painéis alveolares, gerados pelo PREMOLD. O trecho do relatório sobre vigas e pilares não sofreu alterações em relação à FIG. A.4, por isso apresenta-se a seguir apenas o trecho referente aos painéis de piso inseridos no desenho.

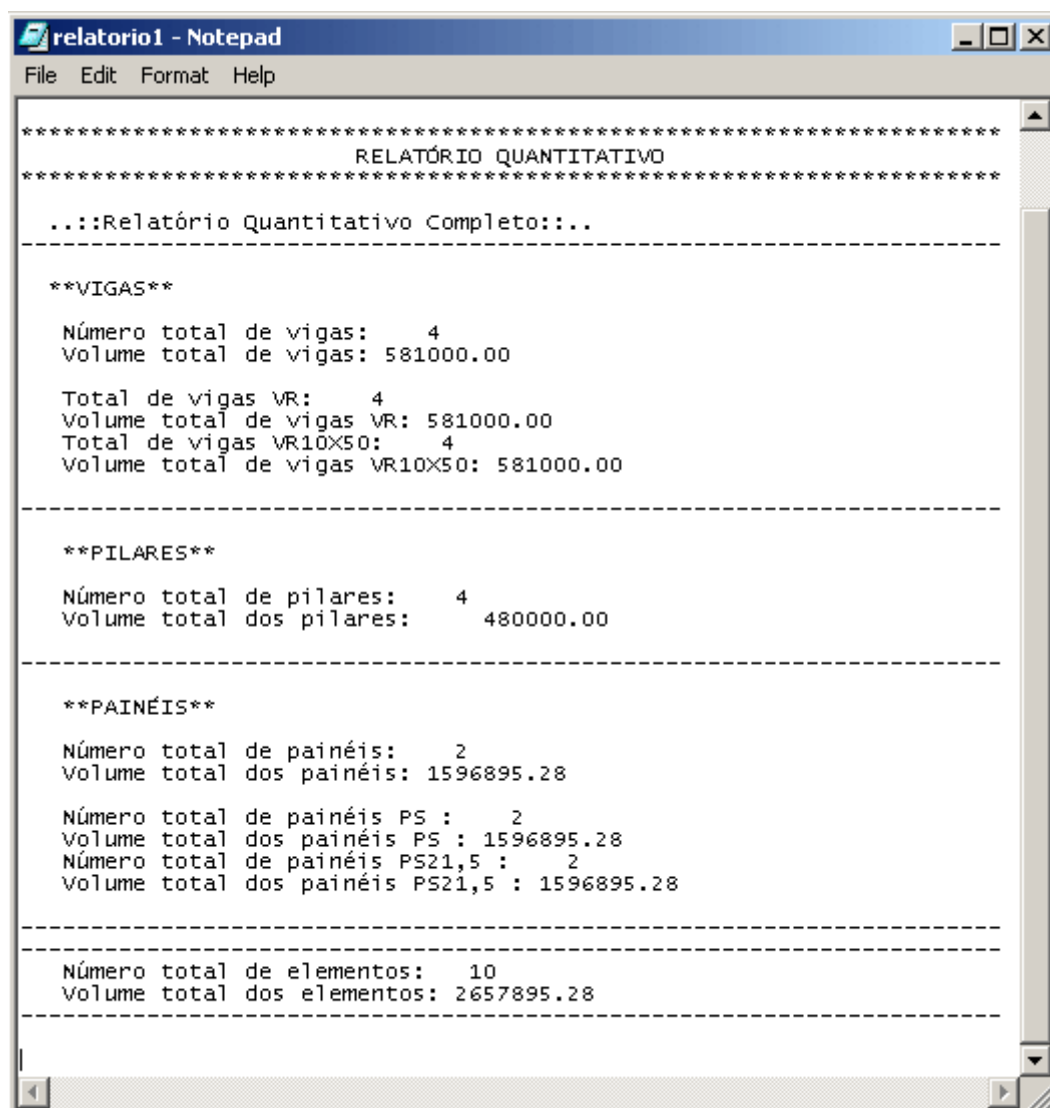
```

relatorio1 - Notepad
File Edit Format Help

FACES
*****
ID      Tipo      Direção   origem
-----
2      PS21,5      0°        (141.90,1.52,275.00)
                Demais vértices
                (244.90,1.52,275.00)(244.90,376.52,275.00)(141.90,376.52,275.00)
1      PS21,5      0°        (16.90,1.52,275.00)
                Demais vértices
                (141.90,1.52,275.00)(141.90,376.52,275.00)(16.90,376.52,275.00)
  
```

**FIGURA A. 53** - Relatório sobre a estrutura formada por quatro pilares, quatro vigas e dois painéis alveolares, gerados pelo PREMOLD.

Relatório Quantitativo Total sobre a estrutura formada por quatro pilares, quatro vigas e dois painéis alveolares, gerados pelo PREMOLD. Nesse relatório são informados os tipos dos elementos presentes na estrutura modelada, suas quantidades o volume total para cada tipo de elemento e o volume total da estrutura, somando-se todos os elementos. Nos relatórios não foram inseridos ainda, na versão 1.0 os dados sobre os blocos de fundação da estrutura.



```
relatorio1 - Notepad
File Edit Format Help
*****
                          RELATÓRIO QUANTITATIVO
*****
...:Relatório Quantitativo Completo:...
-----

**VIGAS**

Número total de vigas:      4
Volume total de vigas: 581000.00

Total de vigas VR:         4
Volume total de vigas VR: 581000.00
Total de vigas VR10X50:    4
Volume total de vigas VR10X50: 581000.00
-----

**PILARES**

Número total de pilares:    4
Volume total dos pilares:   480000.00
-----

**PAINÉIS**

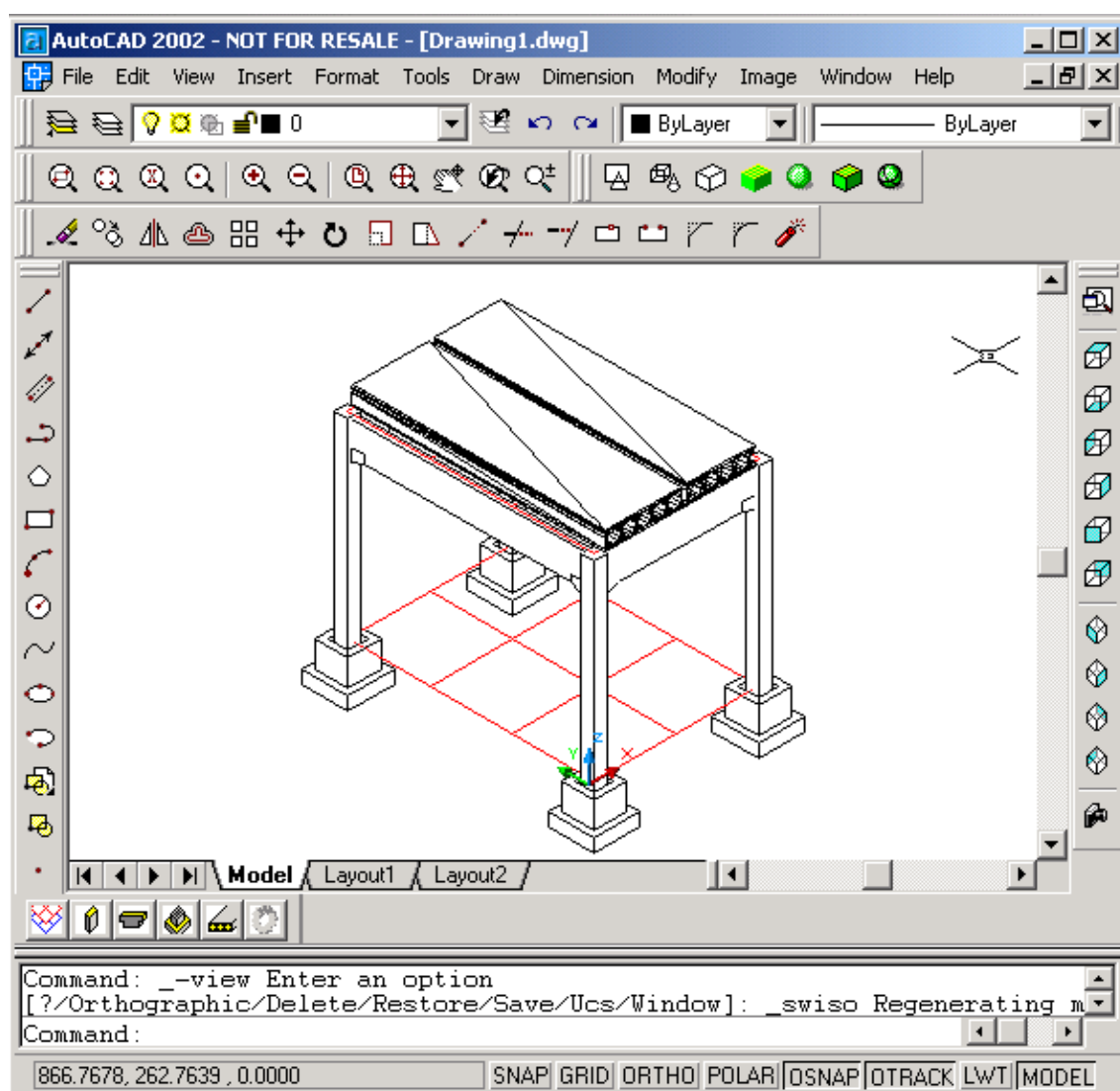
Número total de painéis:    2
Volume total dos painéis: 1596895.28

Número total de painéis PS : 2
Volume total dos painéis PS : 1596895.28
Número total de painéis PS21,5 : 2
Volume total dos painéis PS21,5 : 1596895.28
-----

Número total de elementos: 10
Volume total dos elementos: 2657895.28
-----
```

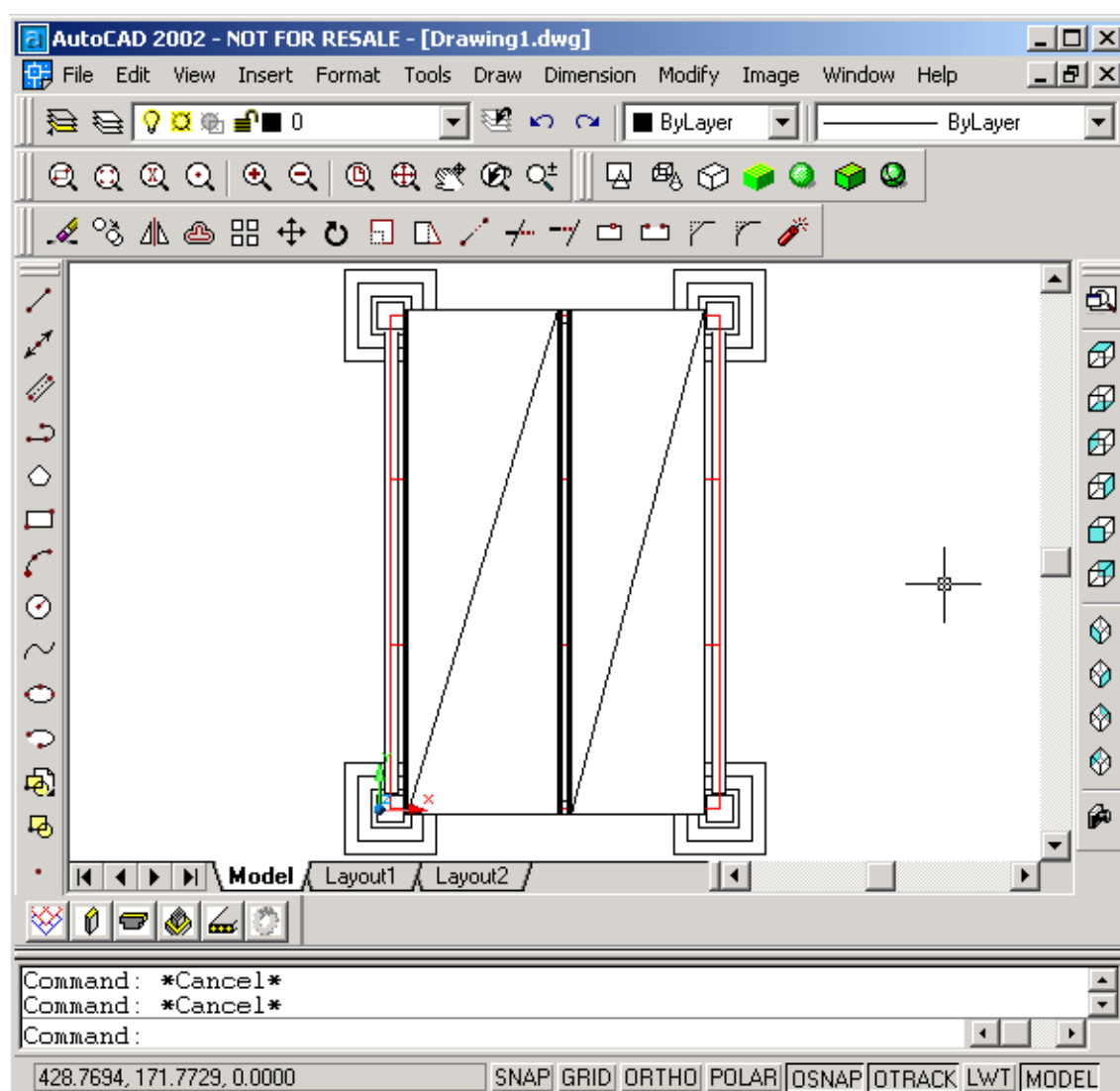
**FIGURA A. 54** - Relatório Quantitativo Total sobre a estrutura formada por quatro pilares, quatro vigas e dois painéis alveolares, gerados pelo PREMOLD.

Blocos de fundação gerados pelo PREMOLD, inseridos sob os pilares pré-existentes. Os blocos são dimensionados para responder às dimensões mínimas prescritas pela NBR 9062/85 (ABNT, 1985).



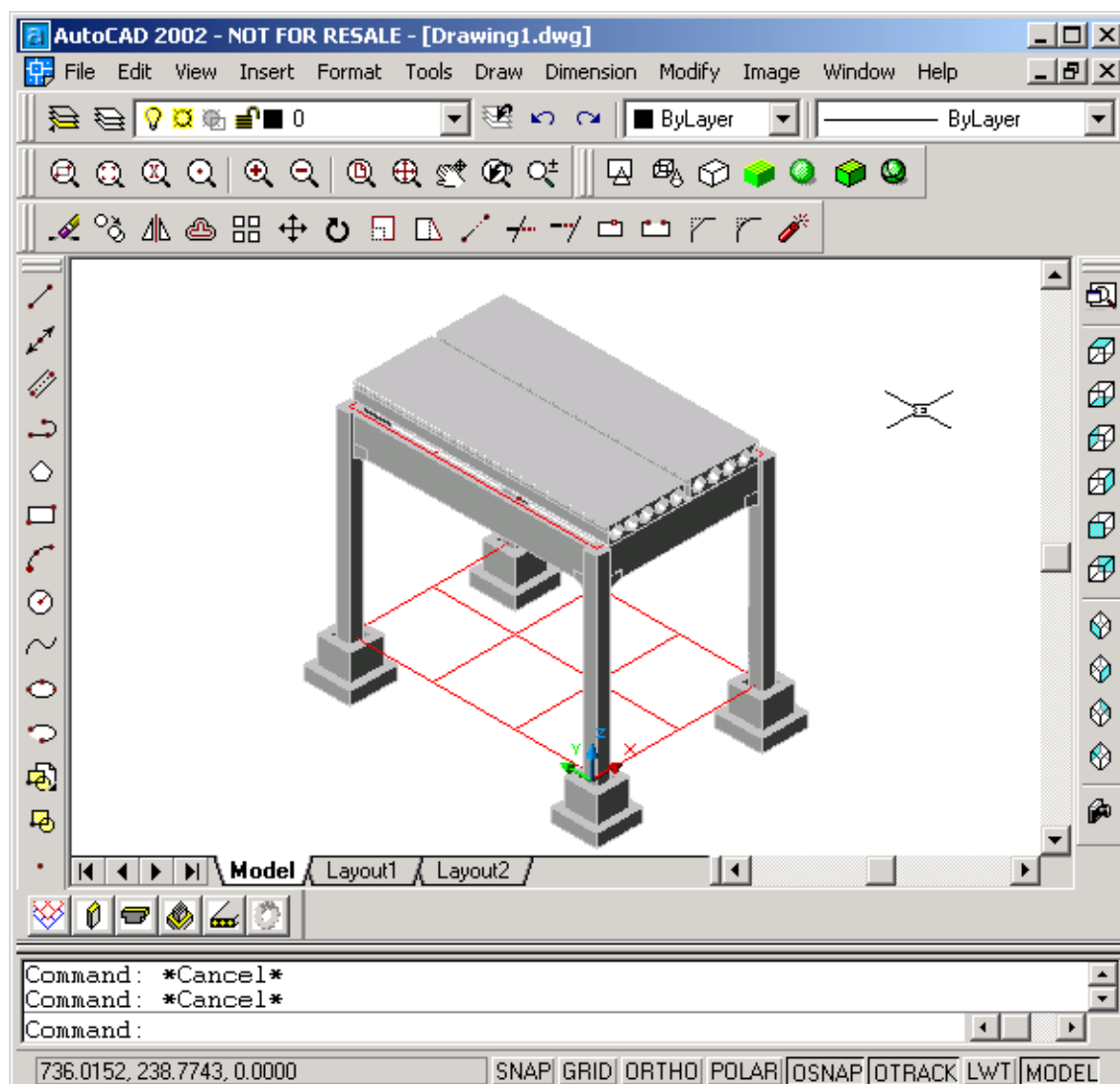
**FIGURA A. 55** – Blocos de fundação gerados pelo PREMOLD.

Vista superior da estrutura gerada pelo PREMOLD com pilares de seção 20x20, vigas retangulares de seção 10x50 inseridas centradas em relação ao eixo vertical dos pilares, com consolos trapezoidais semi-embutidos. Dois painéis de piso alveolar preenchendo todo o vão da viga de apoio, sendo um inteiro e outro cortado longitudinalmente. Blocos de fundação inseridos sob os pilares com dimensões mínimas prescritas pela NBR 9062/85 (ABNT, 1985).



**FIGURA A. 56** - Vista superior da estrutura gerada pelo PREMOLD.

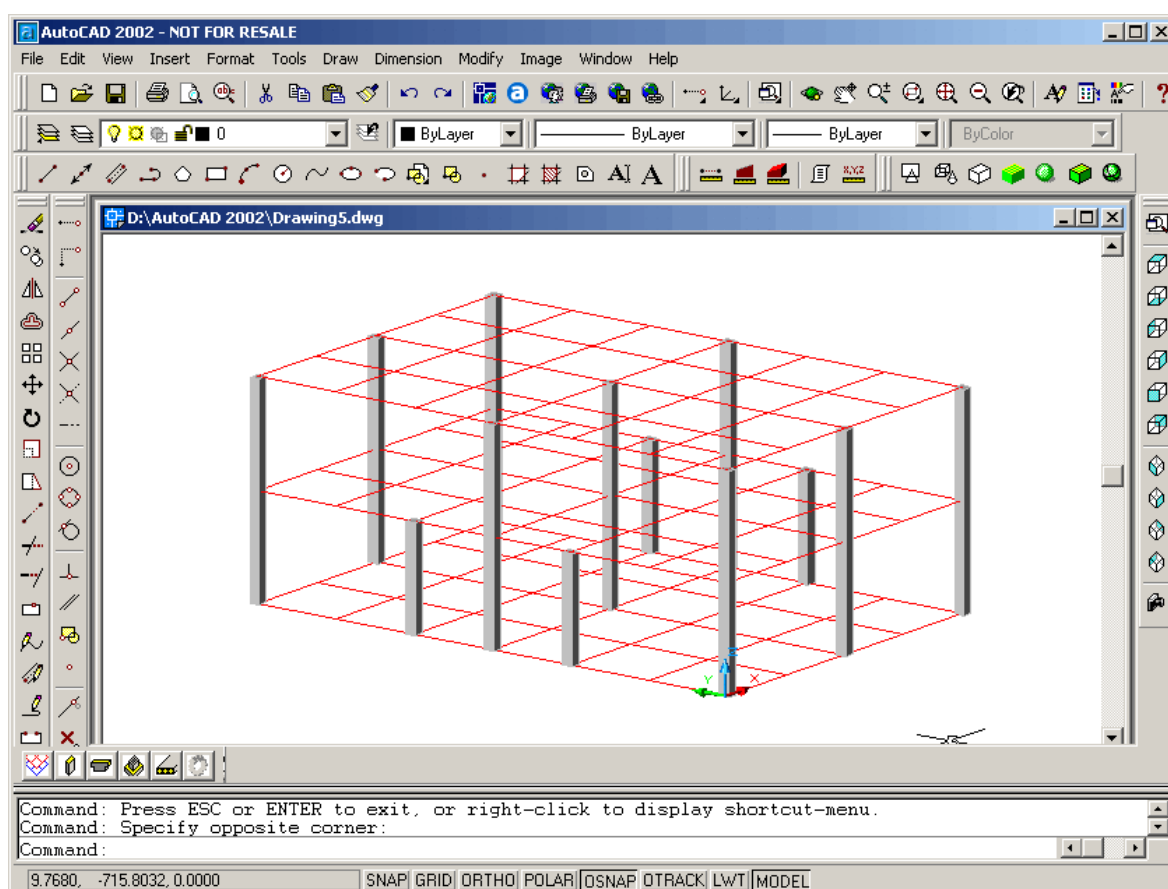
O resultado final desse modelamento é mostrado na FIG. A.55.



**FIGURA A. 57** - FIG. A.13 com o uso do recurso *Flat Shaded* de tratamento de sólidos do AutoCAD.

## Exemplo 02

O exemplo 02 mostra uma estrutura de dois pavimentos com pilares na altura do pavimento 1 e pilares com altura da edificação. Foram inseridas vigas nos pavimentos 1 e 2 apoiadas sobre pilares e vigas aplicadas sobre vigas. Foram inseridos alguns painéis de piso do tipo laje sobre vigas do primeiro pavimento preenchendo completamente o vão da viga de apoio inicial.



**FIGURA A. 58** – Pilares de alturas diferentes.

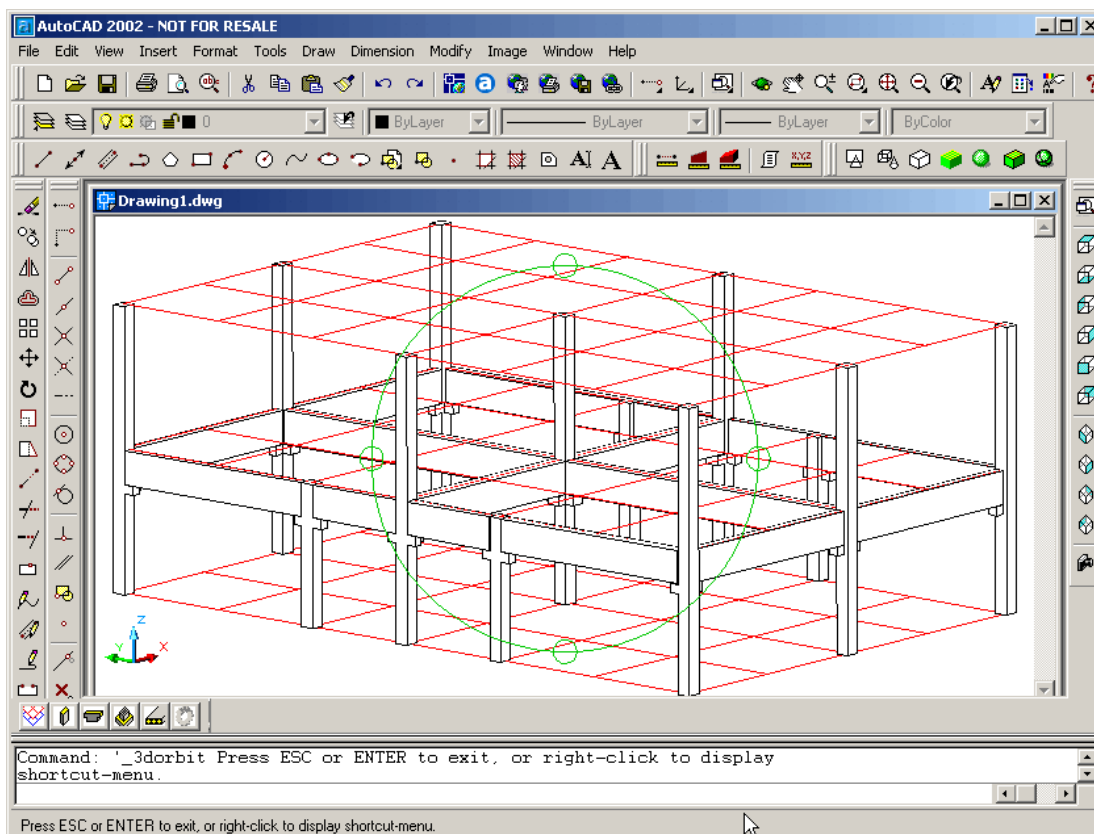


FIGURA A. 59 – Pilares com vigas inseridas no primeiro pavimento.

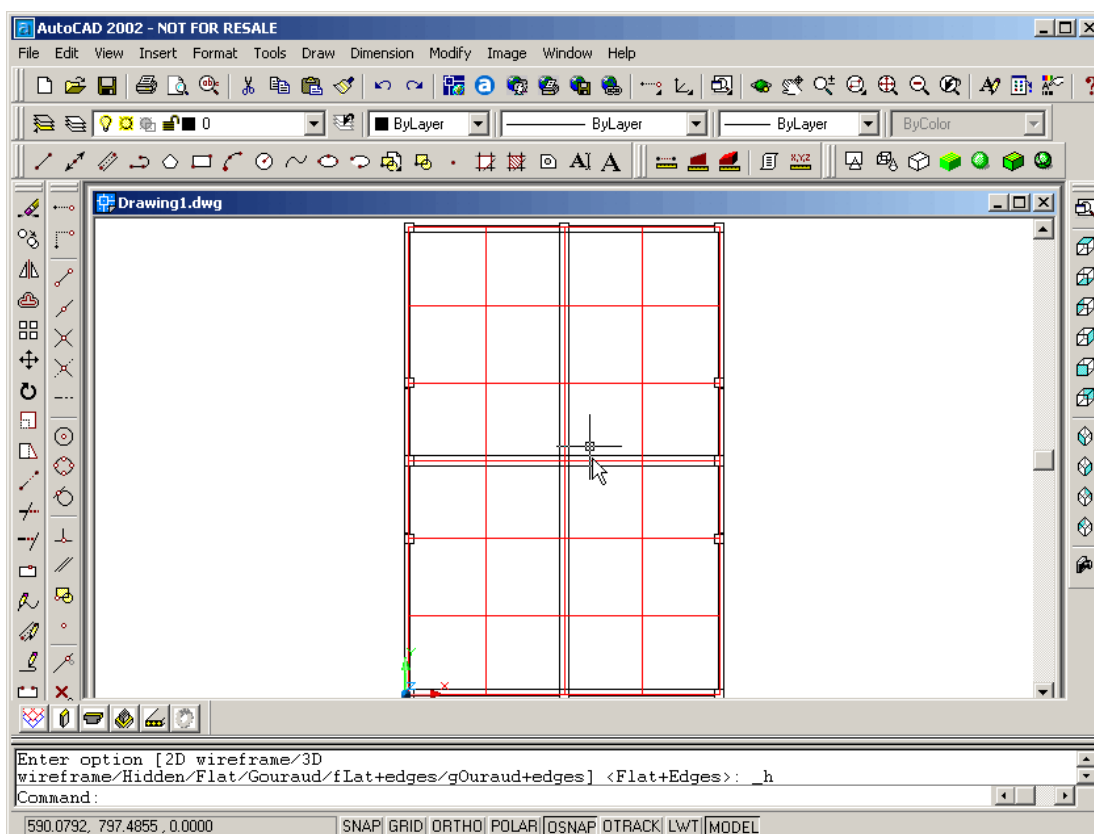


FIGURA A. 60 – Vista superior da estrutura da FIG. A.57.

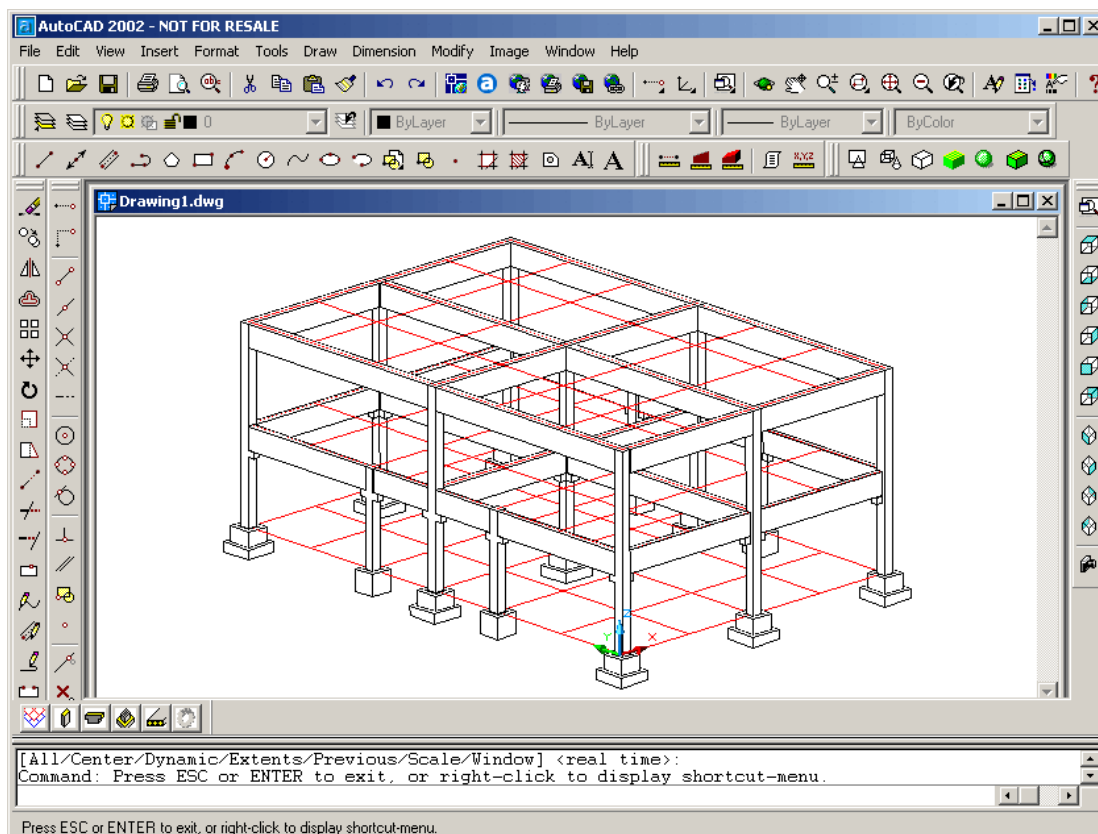


FIGURA A. 61 – Estrutura com pilares, vigas nos pavimentos 1 e 2 e blocos de fundação.

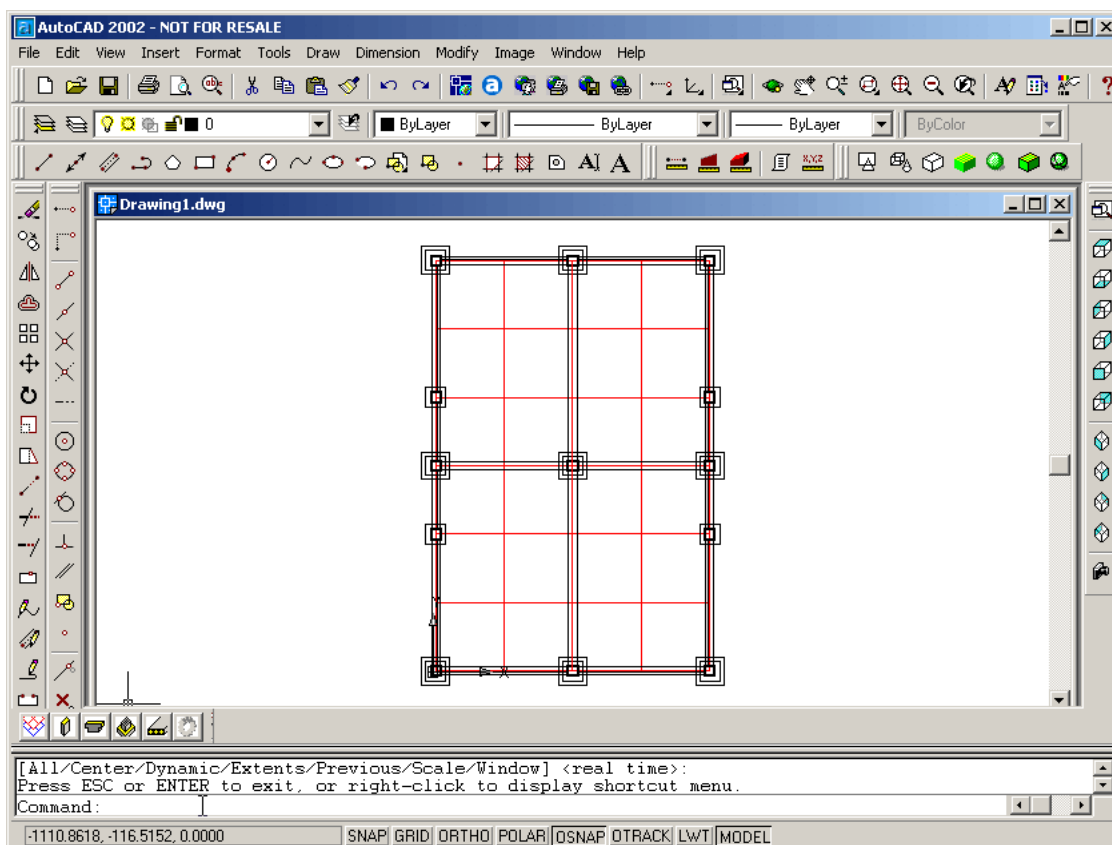
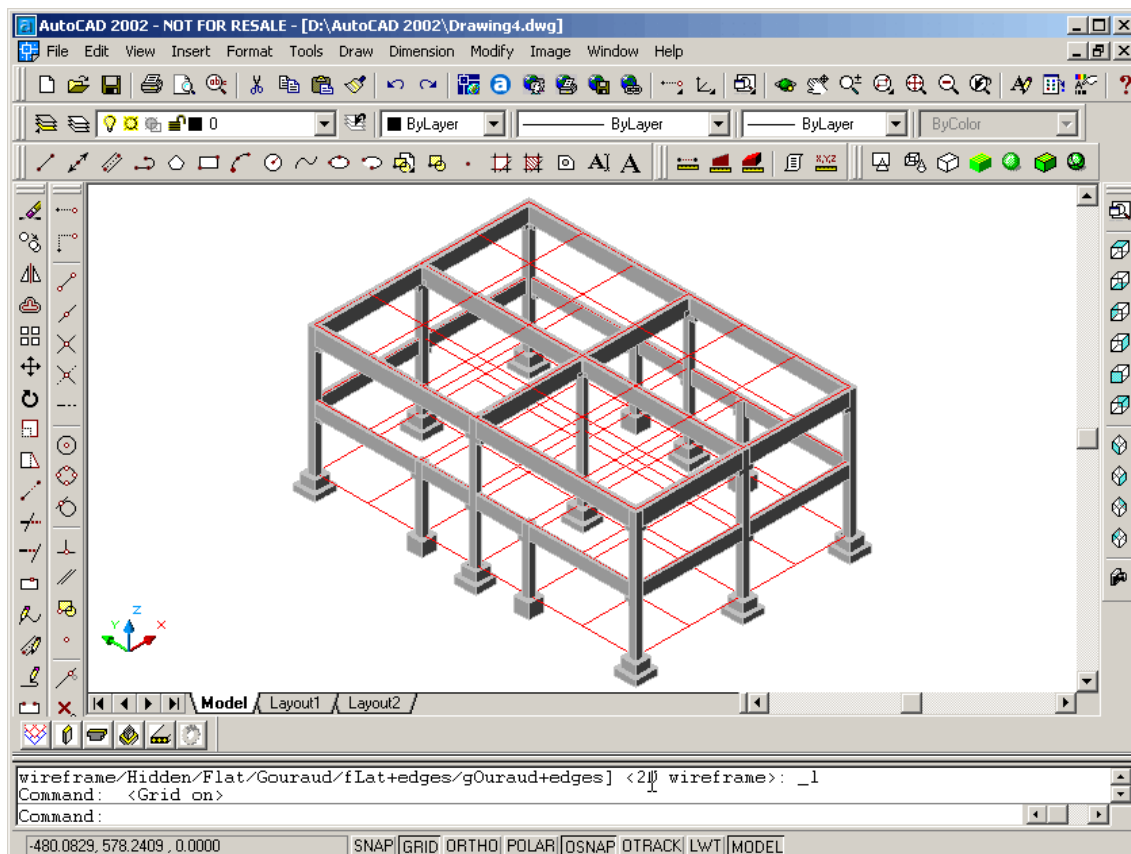
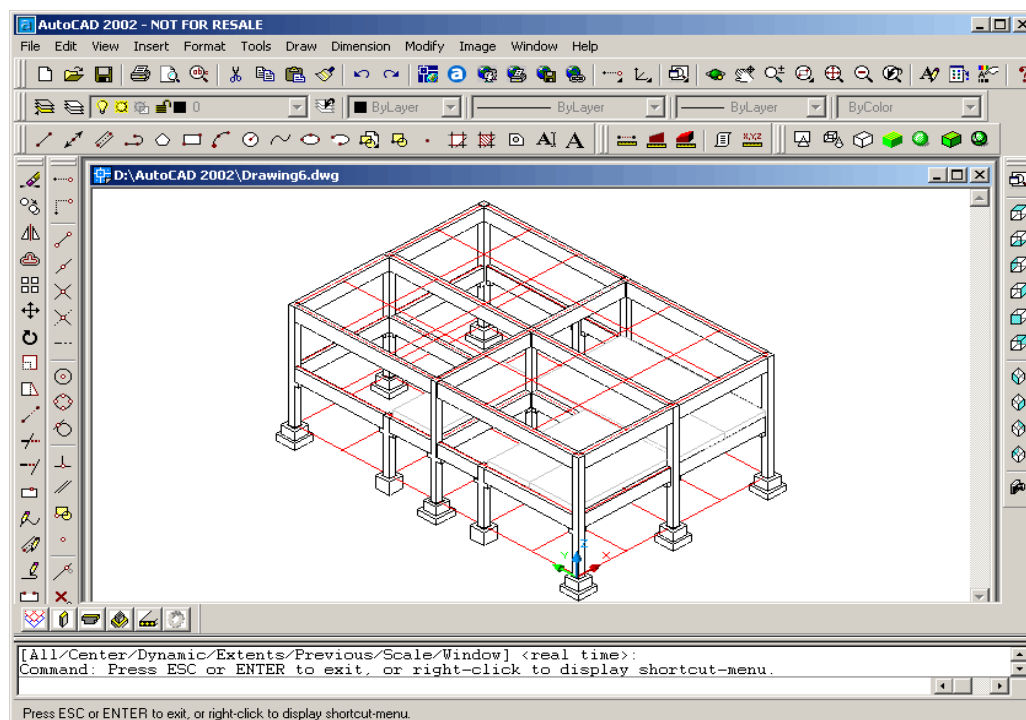


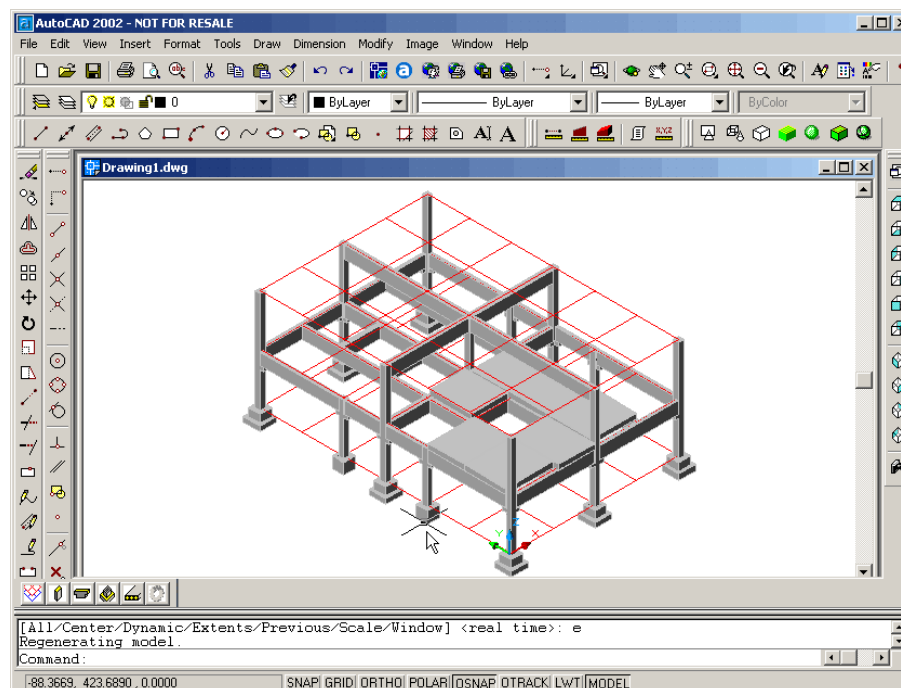
FIGURA A. 62 – Vista superior da estrutura mostrada na FIG. A.59.



**FIGURA A. 63** – Comando *Flat Shaded* do AutoCAD sobre estrutura gerada pelo PREMOLD.



**FIGURA A. 64** – Estrutura da FIG. A.59 com painéis de piso do tipo laje sobre vigas do primeiro pavimento.



**FIGURA A. 65** – Estrutura da FIG. A.63 sem as vigas de fechamento do segundo pavimento, com inserção de vigas secundárias e outra distribuição de painéis de piso.

## REFERÊNCIAS BIBLIOGRÁFICAS

- (ABNT, 1982) ABNT – Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *NBR 7191 -Execução de desenhos para obras de concreto simples ou armado (orig. NB16)*, 1982.
- (ABNT, 1985) ABNT – Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *NBR 9062 -Projeto e execução de estruturas de concreto pré-moldado – Procedimento (orig. NB949)*, 1985.
- (ABNT, 1987) ABNT – Associação Brasileira de Normas Técnicas, Norma Técnica Brasileira, *NBR 10068 - "Folha de desenho - Leiaut e dimensões"*, 1987.