

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA  
MECÂNICA**

**JULIO CÉSAR TICONA BENAVENTE**

**UM SISTEMA PARA O PROJETO E FABRICAÇÃO DE PEÇAS  
MECÂNICAS A DISTÂNCIA VIA INTERNET ADERENTE À  
NORMA ISO 14649 (STEP-NC)**

Tese submetida ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina para a obtenção do Grau de Doutor em Engenharia Mecânica.

Orientador: Prof. Ph.D. João Carlos Espíndola Ferreira.

Florianópolis

2011

Catálogo na fonte pela Biblioteca Universitária  
da  
Universidade Federal de Santa Catarina

B456s Benavente, Julio César Ticona

Um sistema para o projeto e fabricação de peças mecânicas a distância via internet aderente à norma ISO 14649 (STEP-NC) [tese] / Julio César Ticona Benavente ; orientador, João Carlos Espíndola Ferreira. - Florianópolis, SC, 2011.  
239 p.: grafs., tabs.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Mecânica.

Inclui referências

1. Engenharia mecânica. 2. Peças de máquinas. 3. Padrões de produção. 4. Processos de fabricação. 5. Projeto auxiliado por computador. I. Ferreira, João Carlos Espíndola. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

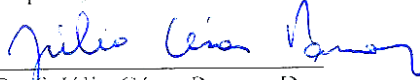
CDU 621

JULIO CÉSAR TICONA BENAVENTE

**UM SISTEMA PARA O PROJETO E FABRICAÇÃO DE PEÇAS  
MECÂNICAS A DISTÂNCIA VIA INTERNET ADERENTE À  
NORMA ISO 14649 (STEP-NC)**

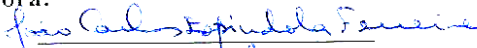
Esta Tese foi julgada adequada para obtenção do Título de Doutor em Engenharia Mecânica, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Mecânica.

Florianópolis, 16 de dezembro de 2011.

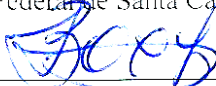


Prof. Júlio César Passos, Dr.  
Coordenador do Curso

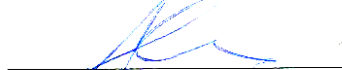
**Banca Examinadora:**



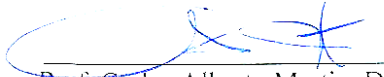
Prof. João Carlos Espíndola Ferreira, Ph.D. - Orientador  
Universidade Federal de Santa Catarina



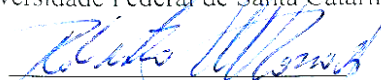
Prof. Lourival Boehs, Dr.  
Universidade Federal de Santa Catarina



Prof. Carlos Henrique Ahrens, Dr.  
Universidade Federal de Santa Catarina



Prof. Carlos Alberto Martin, Dr.  
Universidade Federal de Santa Catarina



Prof. Roberto Silvio Ubertino Rosso Junior, PhD.  
Universidade do Estado de Santa Catarina



Prof. Alberto José Álvares, Dr.  
Universidade de Brasília

Este trabalho está dedicado a Deus quem sempre tem sido fiel para comigo, aos meus pais Agustín e Gumercinda, pelo seu amor incondicional, aos meus irmãos César e Sonia e aos meus cunhados Moisés e Isabela.

## AGRADECIMENTOS

A meu orientador, o professor João Carlos Espíndola Ferreira, pela sua colaboração e paciência no desenvolvimento do presente trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro.

Aos bolsistas de iniciação científica do laboratório GRIMA/GRUCON/UFSC: Cássio Marcos Goulart, Victor Rolando Sanchez Jara, Victor Gomes de Oliveira, Felipe Santos Eberhardt, Ricardo Santos da Silva, Daniel Miranda Cruz e Guilherme Senger, pela assistência na parte técnica.

Aos meus colegas: Izabel Cristina Zattar, Walter Luís Mikos e Paulo Antonio Reaes, que através do exemplo de dedicação e trabalho, me incentivaram a continuar nesta jornada.

Aos meus colegas de laboratório: Nazareno de Oliveira Pacheco, Rafael Alberto López, e David Manuel Ochoa Gonzales com os quais trabalhei e compartilhei o laboratório.

Aos amigos da Igreja Adventista do Sétimo Dia Universitária, por terem me dado sua amizade e carinho.

## RESUMO

Com o rápido avanço da tecnologia da informação associada à tecnologia da fabricação, os ambientes de manufatura mudaram significativamente nas últimas décadas. A usinagem de alta velocidade, usinagem de ultra-precisão e usinagem de múltiplos eixos tem melhorado a produtividade e a qualidade da manufatura. Além disso, a tecnologia da internet introduziu a *e-manufatura* (isto é, projetado por qualquer um em qualquer lugar, fabricado por qualquer um em qualquer lugar). A chave para o sucesso da *e-manufatura* consiste em um fluxo de dados compatíveis e sem brechas na cadeia CAD-CAPP-CAM-CNC. Apesar do avanço nas tecnologias de manufatura, a linguagem que as máquinas de comando numérico utilizam nos dias de hoje (código G – norma ISO 6983) ainda baseia-se nas movimentações da extremidade das ferramentas e no status da máquina, fazendo com que muitas informações acerca do produto sejam perdidas, tornando difícil ou impossível o compartilhamento de tais informações em diferentes áreas da empresa. O padrão ISO 14649 (STEP-NC) fornece um modelo de informações de alto nível referentes ao produto, como por exemplo o uso de *features* para a modelagem de peças e operações de fabricação, e este padrão tem sido apontado como uma ferramenta adequada para superar os problemas acima mencionados. Assim, para um mercado onde é bastante comum a elevada variabilidade das peças sendo produzidas, a alteração nas *features* que compõem a peça é mais rápida e fácil do que a alteração do código G para a usinagem da peça, e desta forma capacita-se as empresas a responderem mais rapidamente à mudança no projeto das peças, além de contribuir para acelerar a tomada de decisões referentes à fabricação das peças. Uma outra vantagem do uso de STEP-NC é que a sua estruturação proporciona que organizações de manufatura venham a compartilhar informações de maneira transparente via Internet. Tendo em vista o contexto acima, no presente trabalho foi desenvolvido um protótipo de um sistema computacional CAD/CAPP/CAM voltado à manufatura remota de peças mecânicas de característica prismática utilizando a Internet como meio de comunicação, sendo completamente aderente ao padrão ISO 14649. No sistema desenvolvido foram disponibilizadas inúmeras *features* para o usuário remoto, incluindo não apenas superfícies com características prismáticas, mas também elementos curvos como fundos abaulados de cavidades. Um dos módulos do sistema verifica se a peça criada pelo usuário é válida do ponto de vista de fabricabilidade com as ferramentas disponíveis. Além disso, o sistema identifica se a peça pode ser

fabricada em uma máquina de comando numérico de três eixos, e no caso de impossibilidade o sistema informa o usuário o que causou a não-fabricabilidade. Finalmente, o sistema gera automaticamente o arquivo físico no formato STEP (padrão ISO 10303 parte 21), o qual é disponibilizado para o usuário via Internet, podendo ele efetuar uma simulação gráfica dos movimentos das ferramentas usinando a peça, tendo o arquivo físico como entrada. Até o momento não foi desenvolvido nenhum sistema computacional que disponibilize tal arquivo gratuitamente na Internet, e conseqüentemente o sistema desenvolvido neste trabalho contribui para uma maior disseminação do padrão STEP-NC, além do fato que o arquivo gerado poderá ser utilizado em máquinas CNC reais compatíveis com o formato STEP parte 21.

**Palavras-chave:** STEP-NC, *features*, peças prismáticas, padronização, manufatura assistida por computador, internet, projeto assistido por computador.

## ABSTRACT

With the rapid advancement of information technology associated with manufacturing technology, manufacturing environments have changed significantly in recent decades. High speed machining, ultra-precision machining, and multi-axis machining have improved manufacturing productivity and quality. In addition, Internet technology has introduced e-manufacturing (i.e., designed by anyone anywhere, manufactured by anyone anywhere). The key to the success of e-manufacturing consists of compatible data stream with no gaps in the CAD-CAPP-CAM-CNC chain. Despite advances in manufacturing technologies, the language that are used by numerical control machines today (i.e. G-code - ISO 6983 standard) is still based on the movements of the end of the tools and on the machine status, causing the loss of much information about the product, making it difficult or impossible to share such information in different areas of the company. The ISO 14649 (STEP-NC) standard provides a model of high-level information about the product, such as the use of features for the modeling parts and manufacturing operations, and this standard has been recommended as a suitable tool to overcome the aforementioned problems. Thus, for a market where it is quite common the high variability of parts being produced, the change in features that compose the part is faster and easier than changing the G code for machining the part, and thus companies are enabled to respond more quickly to the change in part design, besides helping to accelerate decision-making related to the manufacture of parts. Another advantage of using STEP-NC is that its structure enables manufacturing organizations to share information seamlessly via the Internet. Given the above context, in the present research work a prototype CAD/CAPP/CAM computer system was developed for the remote manufacture of mechanical prismatic mechanical parts using the Internet as a means of communication, completely compliant with the ISO 14649 standard. In the developed system many features are provided to the remote user, including not only surfaces with prismatic features, but also curved elements like general cavities. One of the modules of the system verifies whether the part created by the user is valid from the standpoint of manufacturability with the tools available. In addition, the system identifies whether the part can be manufactured in a numerical control machine with three axes, and if that is not feasible, the system informs the user the cause of non-manufacturability. Finally, the system generates automatically the physical STEP file (ISO 10303 Part 21),



which is provided to the user via the Internet, and he can perform a graphic simulation of the movement of the tools machining the part, having the physical file as input. So far no computer system was developed that provides such a file for free on the Internet, and therefore the system developed in this work contributes to a further spread of the STEP-NC standard, plus the fact that the generated file can be used in real CNC machines compatible with the STEP part 21 format.

**Keywords:** STEP-NC, features, prismatic parts, standardization, computer-aided manufacturing, internet, computer-aided design.

## LISTA DE FIGURAS

Figura 2.1. Evolução da capacidade de definição do produto.....	29
Figura 2.2 Troca de informações (a) com tradutores diretos (b) com interface neutra.....	31
Figura 2.3. Intercambio de dados usando IGES.....	34
Figura 2.4. Estrutura geral de um arquivo IGES.....	35
Figura 2.5. Arquitetura de três camadas PDES.....	37
Figura 2.6. Evolução dos padrões para intercâmbio de dados do produto. .....	38
Figura 2.7. Estrutura de alto nível do STEP.....	40
Figura 2.8. Resumo da estrutura do STEP. ....	41
Figura 2.9. Exemplo de um esquema em EXPRESS. ....	43
Figura 2.10. Diagrama EXPRESS-G para o esquema da Figura 2.9. ...	44
Figura 2.11. Arquitetura da especificação dos dados.....	47
Figura 2.12 Modelo de dados do padrão ISO 14649 em diagrama EXPRESS-G. ....	51
Figura 2.13. Linha de programa típica usando a norma ISO 6983. ....	53
Figura 2.14. (a) Fluxo unidirecional de informações. (b) fluxo bidirecional de informações com STEP-NC .....	55
Figura 2.15. (a) Utilização do padrão ISO 6983 (situação atual). (b) Utilização da nova interface STEP-NC.....	55
Figura 2.16. Descrição geral do modelo de dados ISO 14649.....	57
Figura 2.17. Estrutura do padrão ISO 14649. ....	58
Figura 2.18. Estrutura de um programa segundo o padrão ISO 14649. ....	59
Figura 2.19. Entidade “PROJECT” e seus atributos em EXPRESS-G. ....	60
Figura 2.20. Diagrama EXPRESS-G da entidade <i>executable</i> . ....	61
Figura 2.21. Diagrama EXPRESS-G do <i>workingstep</i> .....	62
Figura 2.22. Diagrama EXPRESS-G do <i>workplan</i> . ....	62
Figura 2.23. Entidade <i>machining workingstep</i> e seus atributos em um diagrama EXPRESS-G. ....	63
Figura 2.24. Entidade <i>in process geometry</i> .....	63
Figura 2.25. Entidade <i>manufacturing feature</i> em EXPRESS-G (Atributos e subtipos).....	64
Figura 2.26. <i>Operation</i> : Atributos e subtipos em EXPRESS-G. ....	65
Figura 2.27. Diagrama EXPRESS-G da entidade <i>machining operation</i> . .....	66
Figura 2.28. Diagrama EXPRESS-G da entidade <i>milling machining operation</i> . ....	67
Figura 2.29. <i>Milling type operation</i> em EXPRESS-G. ....	68

Figura 2.30. Operações do tipo furacão em um diagrama EXPRESS-G. .....	69
Figura 2.31. Diagrama EXPRESS-G da entidade <i>milling technology</i> .....	70
Figura 2.32. Entidade <i>milling machining function</i> em EXPRESS-G. ...	71
Figura 2.33. Entidade <i>toolpath</i> e seus atributos em EXPRESS-G. ....	71
Figura 2.34. Diagrama EXPRESS-G da entidade <i>milling cutting tool</i> ..	72
Figura 2.35. Diagrama EXPRESS-G da entidade <i>milling tool body</i> .....	74
Figura 2.36. Dimensões de diferentes tipos de ferramentas. ....	75
Figura 2.37. Entidade <i>Tool dimension</i> e seus atributos em EXPRESS-G. .....	76
Figura 3.1. Peça modelada em <i>wireframe</i> . ....	80
Figura 3.2. Modelagem usando superfícies. ....	81
Figura 3.3. Operações booleanas entre dois sólidos. ....	82
Figura 3.4. Exemplos de <i>features</i> . (a) chanfro, (b) raio de concordância, (c) furo, (d) cavidade retangular, (e) degrau, e (f) ranhura. ....	84
Figura 3.5. Criação de <i>features</i> utilizando projeto por <i>features</i> . ....	85
Figura 4.1. Evolução da internet até a utilização de gráficos 3D .....	88
Figura 5.1. Fabricação aderente a STEP-NC, distribuída. ....	97
Figura 5.2. Estrutura de dados de um sistema CAD/CAM que exporta dados aderentes a STEP-NC. ....	100
Figura 5.3. Estrutura de dados de um sistema que exporta/importa dados STEP-NC. ....	101
Figura 5.4. Sistema aderente a STEP interfaceado externamente ao ambiente CAD/CAM. ....	102
Figura 5.5. Um ambiente CAD/CAM aderente a STEP-NC e compartilhado internamente. ....	103
Figura 5.6. <i>Framework</i> para uma estrutura de dados interna aderente a STEP. ....	104
Figura 6.1. Modelo de informações do produto. ....	107
Figura 6.2. Modelo de informações do processo. ....	108
Figura 6.3. Modelo de informação de recursos. ....	109
Figura 6.4. Diagrama IDEF0 do sistema proposto. ....	112
Figura 6.5. Diagrama IDEF0 do Sistema Integrado, nível A0. ....	113
Figura 6.6. Módulo CAD modelado em um diagrama IDEF0. ....	115
Figura 6.7. Classe Projeto em UML. ....	116
Figura 6.8. Biblioteca de <i>features</i> modelada em UML. ....	117
Figura 6.9. Furo e sua <i>feature</i> precedente. ....	118
Figura 6.10. Diagrama UML das classes desenhadoras. ....	119
Figura 6.11. Diagrama IDEF0 do gerador da representação 3D da peça. .....	121
Figura 6.12. Diagrama IDEF0 do gerador de sólidos das <i>features</i> .....	123

Figura 6.13. Diagrama UML das classes que geram as <i>features</i> . .....	124
Figura 6.14. Operações de Sólidos.....	125
Figura 6.15. Atividade de planejamento do processo em um diagrama IDEF0.....	126
Figura 6.16. <i>Machining Workingstep</i> modelado em UML. ....	127
Figura 6.17. Diagrama IDEF0 da atividade “Gerar <i>Machining Workingsteps</i> ”.....	128
Figura 6.18. Furo com base plana e suas operações.....	129
Figura 6.19. Diagrama UML das operações de usinagem. ....	130
Figura 6.20. Diagrama UML das ferramentas. ....	133
Figura 6.21. Diagrama UML das condições de usinagem. ....	134
Figura 6.22. <i>Features</i> mapeadas em <i>machinig workingsteps</i> .....	135
Figura 6.23. Diagrama IDEF0 do determinador de pontos de apoio. ..	137
Figura 6.24. Arquitetura SDAI. ....	139
Figura 6.25. Diagrama IDEF0 da atividade que gera o arquivo neutro. ....	141
Figura 6.26. Diagrama IDEF0 da atividade CAM. ....	144
Figura 6.27. Diagrama IDEF0 da atividade de leitura de arquivo físico. ....	145
Figura 6.28. Diagrama IDEF0 da atividade pós-processar para ISO 6983. ....	147
Figura 6.29. Modelagem das trajetórias em UML. ....	149
Figura 6.30. Diagrama IDEF0 das atividades de simulação. ....	150
Figura 6.31. Diagrama UML do projeto de simulação. ....	151
Figura 6.32. Modelagem das movimentações em UML. ....	152
Figura 6.33. Tipos de movimentações: (a) Zig-Zag. (b) Circular concêntrica. (c) Retangular concêntrica.....	152
Figura 6.34. Modelagem da simulação em UML. ....	153
Figura 6.35. Arquitetura do protótipo proposto. ....	154
Figura 7.1. Criação de um novo projeto.....	155
Figura 7.2. Janela principal do aplicativo. ....	156
Figura 7.3. Sólidos que representam as <i>features</i> . (a), (b), (c) e (d), diferentes tipos de furos. (e) e (f), diferentes tipos de cavidades retangulares. (g) ranhura com perfil Quadrado U. (h) perfil genérico (curva de Bèzier).....	157
Figura 7.4. Classe responsável pela criação da cavidade retangular em B-rep. ....	158
Figura 7.5. Sólido em estágio de criação. ....	159
Figura 7.6. Sólido que representa a peça acabada.....	159
Figura 7.7. Interface de captura de dados do furo.....	160
Figura 7.8. Interface de captura de dados da ranhura.....	160

Figura 7.9. Raio máximo para uma cavidade retangular.....	161
Figura 7.10. Mensagem de erro devido à inconsistência na criação de uma cavidade retangular.....	162
Figura 7.11. Nova ferramenta criada pelo usuário. ....	163
Figura 7.12. Peça exemplo para a escolha automática de ferramentas.164	
Figura 7.13. Catálogo de ferramentas cadastradas no servidor. ....	165
Figura 7.14. Ferramentas identificadas pelo sistema como adequadas para a usinagem da peça da Figura 7.12.....	165
Figura 7.15. Peça exemplo para explicar como funciona o mapeamento para <i>machining workingsteps</i> . ....	167
Figura 7.16. Perfil genérico dividido em <i>machinings workingsteps</i> . ..	168
Figura 7.17. Divisão da curva em trechos côncavos e convexos. ....	168
Figura 7.18. Segmentação da curva em triângulos para calcular seu circuncentro.....	168
Figura 7.19. Resultado do mapeamento de uma ranhura com perfil genérico.....	169
Figura 7.20. Diâmetros calculados para uma curva de Bèzier dividida em 100 segmentos. ....	171
Figura 7.21. <i>Machining workingsteps</i> mapeadas a partir das <i>features</i> .172	
Figura 7.22. Mensagem de erro ao tentar criar os <i>machining workingsteps</i> para um furo com base cônica.....	173
Figura 7.23. Mensagem de advertência ao tentar remover a operação furação de centro.....	174
Figura 7.24. Mensagem de advertência de remoção de um <i>machining workingstep</i> . ....	175
Figura 7.25. Interface de edição dos parâmetros da operação.....	176
Figura 7.26. Interface de edição da ferramenta. ....	177
Figura 7.27. Interface de edição dos parâmetros tecnológicos.....	177
Figura 7.28. Interface para definição dos pontos de apoio.....	178
Figura 7.29. Arquivo neutro gerado. ....	179
Figura 7.30 Trecho do arquivo neutro em formato XML.....	180
Figura 7.31. Estrutura em níveis para leitura do arquivo físico. ....	182
Figura 7.32. Código G gerado para o CNC Siemens840di. ....	183
Figura 7.33. Simulação no software da máquina (Sinutrain).....	184
Figura 7.34. Áreas removidas por duas fresas de diferente diâmetro..	185
Figura 7.35. Simulador de trajetórias. ....	187

## LISTA DE TABELAS

Tabela 5.1. Resumo das pesquisas em STEP-NC. ....	97
<b>Tabela 5.1. Resumo das pesquisas em STEP-NC.</b> .....	98
<b>Tabela 5.1. Resumo das pesquisas em STEP-NC.</b> .....	99
Tabela 6.1. Precisões de usinagem e rugosidades obtidas através de diversos processos de usinagem para superfícies cilíndricas internas. ....	131
Tabela 6.2. Precisões de usinagem e rugosidades obtidas através de diversos processos de usinagem para superfícies planas .....	132
Tabela 6.3. <i>Machining workinsteps</i> e seus precedentes. ....	135
Tabela 6.4. Precedências das operações de usinagem para um furo com base plana. ....	136
Tabela 7.1. <i>Workplan</i> e <i>machinings workingsteps</i> resultantes do mapeamento do perfil Bèzier. ....	169

## LISTA DE ABREVIATURAS E SIGLAS

AAM	Modelo de Atividade da Aplicação.
AIM	Modelo Interpretado da Aplicação.
ARM	Modelo de Referência da Aplicação.
API	Interface de Programação da Aplicação
APs	Protocolos de Aplicação
B-rep	<i>Boundary Representation</i>
CNC	Controle Numérico Computadorizado
CAD	Projeto Assistido por Computador
CAM	Manufatura Assistida por Computador
CAPP	Planejamento do Processo Assistido por Computador
CSG	Geometria Construtiva de Sólidos
EDM	<i>Electrical discharge machining</i>
FTP	<i>File Transfer Protocol</i>
HTML	<i>Hypertext Markup Language</i>
IDEF	Métodos de Definição de Integração
IDEF0	Definição da Integração para a Modelagem Funcional
IGES	Especificação Inicial de Intercambio de Gráficos.
IP	Protocolo de Internet
ISO	Organização Internacional de Padrões
NC	Controle Numérico
NIST	Instituto Nacional de Padrões e Tecnologia
OMAC	Controlador de Arquitetura Modular Aberta.
PDES	Especificação de Intercâmbio de Dados do Produto
PHP	<i>Hypertext Preprocessor</i>
SDAI	Interface de Acesso de Dados Padrão.
SET	<i>Standart d'exchange et de Transfer</i>
STEP	Padrão para a Troca de Dados do Modelo do Produto
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
UML	Linguagem de Modelagem Unificada
VDA	<i>Flachenschnittstelle des Verbandes der deutschen Automobilindustrie</i>
VRML	Linguagem de Modelagem de Realidade Virtual
WWW	Rede Mundial
XML	<i>Extensible Markup Language</i>

## SUMÁRIO

<b>RESUMO</b> .....	<b>VI</b>
<b>ABSTRACT</b> .....	<b>VIII</b>
<b>LISTA DE FIGURAS</b> .....	<b>X</b>
<b>LISTA DE TABELAS</b> .....	<b>XIV</b>
<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	<b>XV</b>
<b>SUMÁRIO</b> .....	<b>XVI</b>
<b>1 INTRODUÇÃO</b> .....	<b>22</b>
<b>1.1 PROBLEMA DE PESQUISA</b> .....	<b>23</b>
<b>1.2 OBJETIVOS DO TRABALHO DE TESE</b> .....	<b>23</b>
1.2.1 Objetivo Geral.....	23
1.2.2 Objetivos Específicos.....	23
<b>1.3 DELIMITAÇÃO DO TRABALHO</b> .....	<b>24</b>
<b>1.4 JUSTIFICATIVA E CONTRIBUIÇÃO DA TESE</b> .....	<b>24</b>
<b>1.5 METODOLOGIA</b> .....	<b>25</b>
<b>1.6 ESTRUTURA DA TESE</b> .....	<b>26</b>
<b>2 COMPARTILHAMENTO DAS INFORMAÇÕES DO PRODUTO</b> .....	<b>28</b>
<b>2.1 EVOLUÇÃO DO COMPARTILHAMENTO DAS INFORMAÇÕES DO PRODUTO</b> .....	<b>28</b>
2.1.1 IGES (Especificação Inicial de Intercâmbio de Gráficos) ..	32
2.1.1.1 Descrição do padrão IGES.....	33



2.1.1.2	Formato e estrutura do arquivo neutro IGES .....	34
2.1.1.3	Limitações e desvantagens do IGES .....	35
2.1.2	PDES (Especificação de Intercâmbio de Dados do Produto) 36	
2.1.3	Outros esforços e contribuições para a implementação de arquivos neutros.....	38
<b>2.2</b>	<b>O PADRÃO ISO 10303 (STEP) .....</b>	<b>39</b>
2.2.1	Estrutura do Padrão ISO 10303.....	40
2.2.1.1	Métodos de descrição.....	41
2.2.1.1.1	A linguagem EXPRESS.....	42
2.2.1.2	Métodos de implementação.....	44
2.2.1.2.1	A linguagem XML.....	45
2.2.1.3	Testes de conformidade .....	46
2.2.1.4	Especificação dos dados.....	46
2.2.1.4.1	Recursos Integrados.....	47
2.2.1.4.2	Protocolos de Aplicação (AP).....	48
2.2.1.4.3	Construtores Interpretados da Aplicação (AICs) .....	49
2.2.2	O padrão STEP-NC.....	49
2.2.2.1	O padrão ISO 6983 .....	51
2.2.2.2	O padrão ISO 14649 .....	54
2.2.2.3	O modelo de dados do padrão ISO 14649.....	56
2.2.2.4	Estrutura e organização do padrão ISO 14649.....	57
2.2.2.4.1	ISO 14649 Parte 1: Descrição Geral e princípios fundamentais .....	58
2.2.2.4.2	ISO 14649 Parte 10: Dados gerais do processo .....	59
2.2.2.4.3	ISO 14649 Parte 11: Dados do processo para fresamento .....	66
2.2.2.4.4	ISO 14649 parte 111: Ferramentas para fresamento.	71
<b>3</b>	<b>PROJETO ASSISTIDO POR COMPUTADOR.....</b>	<b>77</b>
<b>3.1</b>	<b>ANTECEDENTES .....</b>	<b>77</b>
<b>3.2</b>	<b>CLASSIFICAÇÃO DOS SISTEMAS CAD .....</b>	<b>78</b>
<b>3.3</b>	<b>MODELAGEM GEOMÉTRICA .....</b>	<b>79</b>
3.3.1	Modelos de arame ( <i>wireframe</i> ) .....	79
3.3.2	Modelos de superfícies.....	80
3.3.3	Modelos sólidos .....	81

3.3.3.1	Geometria construtiva de sólidos (CSG) .....	82
3.3.3.2	Representação de fronteira (B-rep).....	82
3.3.4	Projeto por <i>features</i> .....	83
<b>4</b>	<b>TECNOLOGIA DA INFORMAÇÃO E MANUFATURA .....</b>	<b>86</b>
<b>4.1</b>	<b>MANUFATURA ELETRÔNICA.....</b>	<b>86</b>
<b>4.2</b>	<b>TELEMANUFATURA .....</b>	<b>86</b>
<b>4.3</b>	<b>MANUFATURA VIRTUAL .....</b>	<b>87</b>
<b>4.4</b>	<b>EVOLUÇÃO DAS TECNOLOGIAS BASEADAS EM REDE</b>	<b>87</b>
4.4.1	Java .....	88
4.4.2	Java 3D .....	89
4.4.3	A Linguagem de Modelagem de Realidade Virtual.....	90
4.4.4	3D Extensível (X3D) .....	90
<b>4.5</b>	<b>COMUNICAÇÃO DE DADOS EM SISTEMAS</b>	
<b>DISTRIBUIDOS .....</b>	<b>91</b>	
4.5.1	O modelo OSI (Interconexão de Sistemas Abertos) .....	91
4.5.2	Modelo Cliente servidor.....	92
<b>5</b>	<b>ESTRUTURAS DE DADOS E PESQUISAS RELACIONADAS</b>	
<b>AO PADRÃO STEP-NC .....</b>	<b>93</b>	
<b>5.1</b>	<b>PESQUISAS EM STEP-NC .....</b>	<b>93</b>
5.1.1	Controle convencional do CNC usando STEP-NC.....	93
5.1.2	Novo STEP-NC que possibilita o controle .....	94
5.1.3	STEP-NC que possibilita o controle inteligente .....	95
5.1.4	STEP-NC Colaborativo que possibilita a usinagem. ....	96
<b>5.2</b>	<b>ESTRUTURAS DE DADOS QUE SUPORTAM SISTEMAS</b>	
<b>CAD/CAM ADERENTES AO PADRÃO ISO 14649 .....</b>	<b>99</b>	
5.2.1	Sistema CAD/CAM que importa/exporta uma saída aderente	
ao padrão STEP-NC .....	99	
5.2.1.1	Sistema CAD/CAM que exporta dados aderentes à	
norma STEP-NC .....	100	

5.2.1.2	Sistema CAD/CAM que importa e exporta códigos aderentes à norma STEP-NC .....	100
5.2.2	Sistema CAD/CAM com estruturas de suporte de dados aderentes a STEP-NC. ....	101
5.2.2.1	Sistema CAD/CAM integrado com um sistema externo de suporte de dados aderente a STEP-NC.....	101
5.2.2.2	Ambiente CAD/CAM com estruturas de dados nativa e aderente a STEP-NC internas.....	102
5.2.3	Ambiente CAD/CAM com kernel baseado na estrutura de dados aderente a STEP-NC .....	103

## **6 MODELAGEM CONCEITUAL DO PROTÓTIPO DO SISTEMA INTEGRADO CAD/CAPP/CAM ADERENTE AO PADRÃO ISO 14649..... 105**

### **6.1 REQUISITOS PARA UMA ESTRUTURA DE DADOS CAD/CAPP/CAM INTEROPERÁVEL E EXECUTADA REMOTAMENTE ..... 105**

6.1.1	Uso de modelos de dados padronizados.....	106
6.1.1.1	Modelo de informação padrão do produto .....	106
6.1.1.2	Modelo de informação do processo .....	107
6.1.1.3	Modelo de informação dos recursos.....	108
6.1.2	Orientação a objetos .....	110
6.1.3	Independência de plataforma .....	110
6.1.4	Modularidade .....	110

### **6.2 ESPECIFICAÇÃO DO MODELO..... 111**

6.2.1	Modelagem funcional do protótipo .....	111
6.2.1.1	Atividade de projeto.....	114
6.2.1.1.1	Criação de um novo projeto.....	114
6.2.1.1.2	Modelagem da peça usando a biblioteca de features: 114	
6.2.1.1.3	Validação do modelo .....	118
6.2.1.1.4	Geração do modelo geométrico 2D .....	119
6.2.1.1.5	Geração do modelo gráfico 3D .....	120
6.2.1.2	Atividade de Planejamento do Processo .....	125
6.2.1.2.1	Gerar Machining workingsteps.....	127
6.2.1.2.2	Organização dos Machining workingsteps .....	135
6.2.1.2.3	Determinação dos pontos de fixação .....	136
6.2.1.2.4	Geração do plano de trabalho .....	138

6.2.1.2.5	Geração do arquivo neutro STEP-NC .....	138
6.2.1.3	Atividade CAM .....	143
6.2.1.3.1	Leitura do arquivo neutro .....	143
6.2.1.3.2	Geração do arquivo em formato ISO 6983 .....	146
6.2.1.3.3	Simulação de movimentações da ferramenta.....	149
6.2.2	Arquitetura do sistema proposto .....	153
<b>7</b>	<b>IMPLEMENTAÇÃO DO PROTÓTIPO DO SISTEMA CAD/CAPP/CAM .....</b>	<b>155</b>
<b>7.1</b>	<b>IMPLEMENTAÇÃO DO SISTEMA CAD .....</b>	<b>155</b>
7.1.1	Implementação dos modelos sólidos 3D.....	157
7.1.2	Interfaces gráficas de captura de dados.....	158
<b>7.2</b>	<b>IMPLEMENTAÇÃO DO SISTEMA CAPP .....</b>	<b>162</b>
7.2.1	Definição das ferramentas do projeto .....	162
7.2.2	Mapeamento para <i>machining workingsteps</i> .....	165
7.2.2.1	Remoção e edição de <i>machining workingsteps</i> .....	174
7.2.3	Pontos de apoio .....	178
7.2.4	Geração do arquivo neutro.....	179
<b>7.3</b>	<b>IMPLEMENTAÇÃO DO SISTEMA CAM.....</b>	<b>180</b>
7.3.1	Leitura de arquivos neutros.....	180
7.3.2	Geração do código G .....	182
7.3.3	Simulação da usinagem.....	185
7.3.4	Disponibilização para usuários remotos.....	188
<b>8</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>189</b>
<b>8.1</b>	<b>CONCLUSÕES .....</b>	<b>189</b>
<b>8.2</b>	<b>CONTRIBUIÇÕES.....</b>	<b>190</b>
<b>8.3</b>	<b>SUGESTÕES PARA FUTUROS TRABALHOS.....</b>	<b>191</b>
<b>9</b>	<b>REFERÊNCIAS.....</b>	<b>193</b>
	<b>APÊNDICE A – MAPEADORAS DE <i>WORKINGSTEPS</i> .....</b>	<b>206</b>

**APÊNDICE B – FLUXO DE INFORMAÇÕES BIDIRECIONAL  
217**

**APÊNDICE C – UTILIZAÇÃO DO SISTEMA.....223**

# 1 INTRODUÇÃO

Nas últimas décadas as prioridades econômicas da manufatura têm sofrido grandes mudanças. Anteriormente os produtos eram via de regra padronizados de baixo custo, e passou-se então ao uso dos equipamentos e estrutura da manufatura moderna no âmbito do conceito da produção sob demanda.

Este conceito tem sido adotado a fim de tirar vantagem das oportunidades da globalização econômica. O ambiente de negócios constantemente em mudança leva à necessidade de colaboração entre empresas que vai mais além dos limites geográficos. Os rápidos avanços nas tecnologias da informação e de manufatura podem dar o suporte necessário nestes ambientes globalizados (TIWARI; JHA; ANAND, 2010, YUSOF, 2008; ZAIDAT *et. al.*, 2005).

Em uma abordagem tradicional, projeto e manufatura são considerados separadamente, porém em um ambiente moderno e global, tal separação torna-se uma grande fraqueza que resulta em ciclos de produção mais lentos e mais custosos. Isto acontece devido ao fato que tanto a área de projeto como a de manufatura têm equipes próprias, sendo que os colaboradores da equipe de projeto se focam exclusivamente no design do produto, e então passam as informações para os engenheiros de fabricação os quais decidirão como fabricar o produto. Se no estágio de fabricação é detectado algum problema e há a necessidade de mudança em alguma parte do projeto, esta informação será passada de volta para a equipe de projeto. Atualmente, com o uso das tecnologias de computação e de comunicações nas indústrias de manufatura, os métodos acima mencionados são substituídos por sistemas CAD/CAM e a engenharia simultânea. Estes sistemas CAD/CAM reduzem significativamente a intervenção humana, resultando no aumento da produtividade, redução de custos e melhora na qualidade dos produtos.

As máquinas CNC desempenham um papel importante dentro da indústria de manufatura atual, e nas últimas décadas tem havido desenvolvimentos importantes na tecnologia do controle numérico (NC), porém autores como ZHANG *et. al.*, (2011), GARRIDO *et. al.*, (2009), YUSOF *et. al.* (2009), ROSSO (2005) apontam a problemática da perda das informações do produto associada à linguagem de programação que tais máquinas utilizam (código G e M, ISO 6983), já que esta linguagem ainda está baseada nas movimentações das ferramentas e no status da

máquina, o que torna-se um limitante do potencial da capacidade das máquinas-ferramenta. Além disso, este padrão dá liberdade para que o fabricante do CNC possa desenvolver a sua própria linguagem de programação (normalmente chamada de dialeto), e consequentemente perde-se a interoperabilidade.

## 1.1 PROBLEMA DE PESQUISA

O cenário apresentado acima traz à luz alguns motivos que levaram ao desenvolvimento do presente trabalho de tese, como por exemplo: (a) Como superar os obstáculos decorrentes da utilização do padrão ISO 6983 na programação de máquinas-ferramentas CNC? (b) Como conseguir a integração de dados do produto através da cadeia CAD-CAPP-CAM? (c) Como utilizar o padrão ISO 14649 de maneira a dar suporte ao projeto, planejamento e fabricação de peças prismáticas através da internet?

A hipótese deste trabalho é que o modelo de dados fornecido pelo padrão STEP-NC pode ser aplicado na manufatura CNC voltada para internet.

## 1.2 OBJETIVOS DO TRABALHO DE TESE

### 1.2.1 Objetivo Geral

Propor um modelo computacional de um sistema CAD/CAPP/CAM aderente à norma ISO 14649 para o projeto e fabricação de peças prismáticas e que use a internet como meio de comunicação.

### 1.2.2 Objetivos Específicos

- Realizar uma revisão bibliográfica dos assuntos relacionados com CAD, CAPP, CAM, CNC, e aos padrões internacionais ISO 10303 (conhecidos também como STEP).

- Apontar os modelos CAD/CAPP/CAM que sejam aderentes aos padrões STEP.
- Explorar o uso de *features* na modelagem de peças prismáticas e na fabricação.
- Projetar e implementar um protótipo de um sistema computacional que dê suporte ao projeto de peças prismáticas baseado no modelo proposto.
- Disponibilizar o protótipo do sistema computacional para usuários remotos através da internet.

### 1.3 DELIMITAÇÃO DO TRABALHO

O presente trabalho de tese foca-se no problema do fluxo de informações entre os módulos CAD, CAPP e CAM para peças prismáticas, e como os padrões STEP podem ser usados para alcançar a integração entre tais módulos.

A abordagem de projeto por *features* implica na implementação de uma biblioteca com *features* presentes no padrão ISO 14649, parte 10, de onde foram implementadas uma grande quantidade, porém não todas.

Deve-se notar que a grande maioria dos sistemas CNC atuais não possuem a capacidade de interpretar e executar arquivos físicos contendo informações em formato STEP, parte 21 ou parte 28 (XML), portanto deve-se fazer o pós-processamento de tais arquivos para a linguagem específica do controlador da máquina-ferramenta onde será executada a usinagem. Não faz parte do presente trabalho de tese o desenvolvimento de módulos que capacitem ao controlador NC “entender” o arquivo físico em formato STEP.

### 1.4 JUSTIFICATIVA E CONTRIBUIÇÃO DA TESE

Uma justificativa para o desenvolvimento do presente trabalho de tese encontra-se nos problemas causados pelas informações incompatíveis entre diferentes sistemas. XU *et. al.*, (2006) reportam um estudo que foi feito na indústria automotiva dos Estados Unidos relacionado aos gastos decorrentes da ausência de interoperabilidade entre sistemas. Segundo tal estudo, têm sido gastos dez bilhões de



dólares por ano para superar as barreiras de informação e da falta de interoperabilidade entre sistemas CAD, CAPP e CAM. Outros nove bilhões de dólares foram gastos para consertar e substituir arquivos inúteis.

Outra justificativa advém dos novos paradigmas para sistemas CAD/CAPP/CAM. HENDERSON-SELLERS e GIORGINI, (2005) sustentam que em ambientes de manufatura caracterizados pelas novas estruturas organizacionais onde as distâncias geográficas não sejam um obstáculo, onde seja possível responder às rápidas mudanças de requisitos, é recomendável o uso de novas abordagens baseadas na disponibilidade de tecnologia de informação e comunicação.

ALVARES (2005) diz que a tecnologia da internet oferece um novo paradigma para o desenvolvimento de ambientes CAD/CAPP/CAM integrados. Estes ambientes globais são centrados em rede e distribuídos, tendo navegadores web como *front-end* com o sistema CAD/CAPP/CAM, e a linguagem Java para permitir a independência da plataforma computacional utilizada pelo cliente remoto. Tais sistemas oferecem um novo paradigma a ser aplicado na indústria e na academia.

O modelo proposto neste trabalho encaixa-se dentro deste paradigma, contribuindo para o desenvolvimento de sistemas de manufatura eletrônica, e inovando no aspecto de ser um sistema totalmente aderente ao padrão ISO 14649, bem como sendo disponibilizado para usuários remotos por meio da internet, ou seja, ele pode ser utilizado por instituições acadêmicas e por qualquer usuário que tenha interesse no assunto.

Atualmente existem sistemas comerciais correlatos ao proposto nesta tese (por exemplo, o STEPTools - [www.steptools.com](http://www.steptools.com)), porém tais sistemas não podem ser executados via internet, os seus aplicativos devem ser instalados diretamente no computador do usuário, e suas licenças são relativamente caras.

## 1.5 METODOLOGIA

Segundo CERVO, BERVIAN, SILVA (2007), SILVA e MENEZES (2001), uma pesquisa científica se pode classificar a partir de quatro dimensões:

- Do ponto de vista da sua natureza, a pesquisa pode ser básica ou aplicada. A pesquisa descrita nesta tese apresenta características de pesquisa aplicada, pois busca usar o padrão ISO 14649 (STEP-NC) em um protótipo computacional CAD/CAPP/CAM que possa ser executado remotamente. Porém, a aplicação dos resultados deste trabalho na usinagem de peças em máquinas CNC não foi feita, pois protótipos de CNCs que “entendam” arquivos físicos aderentes ao padrão STEP-NC ainda estão em fase de desenvolvimento.
- Do ponto de vista da forma de abordagem do problema, a pesquisa pode ser qualitativa ou quantitativa. A pesquisa aqui apresentada é quantitativa, pois se trabalhou com assuntos objetivos ou quantificáveis.
- Do ponto de vista de seus objetivos, as pesquisas se classificam em: pesquisa exploratória, pesquisa descritiva e pesquisa explicativa. Em relação a esta classificação, a pesquisa descrita aqui apresenta características de pesquisa exploratória porque busca trazer à luz os problemas decorrentes da utilização do padrão ISO 6983, e porque também se torna necessário um levantamento bibliográfico, porém não será necessário o uso de entrevistas.
- Do ponto de vista dos procedimentos técnicos, as pesquisas podem ser: bibliográfica, documental, experimental, levantamento, estudo de caso, pesquisa *expost-facto*, pesquisa-ação, e pesquisa participante. Com respeito a esta classificação, a pesquisa aqui apresentada tem características de pesquisa bibliográfica, pois se apresenta uma revisão bibliográfica sobre o tema de pesquisa.

## 1.6 ESTRUTURA DA TESE

Com o intuito de alcançar os objetivos propostos neste trabalho de tese, e para ter uma sequência lógica, a mesma está dividida em sete capítulos.

O capítulo 1 introduz o tema desta pesquisa, são apresentados os objetivos e as justificativas do trabalho, os tipos de pesquisas, e como o

presente trabalho se enquadra nessa classificação, e finalmente é apresentada a estrutura da tese.

Os capítulos 2, 3 e 4 apresentam a revisão da literatura concernente aos conceitos que fundamentam esta pesquisa.

O capítulo 5 descreve as estruturas baseadas em STEP que dão suporte a sistemas CAD/CAPP/CAM.

No capítulo 6 é proposto o modelo de um sistema computacional aderente ao padrão STEP-NC, e são especificados os requisitos de tal protótipo.

O capítulo 7 trata acerca da implementação do protótipo especificado no capítulo 6.

O capítulo 8 apresenta as conclusões e as recomendações para trabalhos futuros.

## **2 COMPARTILHAMENTO DAS INFORMAÇÕES DO PRODUTO**

Este capítulo tem como finalidade apresentar o fundamento teórico para o tema proposto, especificamente no assunto dos mecanismos para representar as informações associadas ao produto. O mesmo começa com a descrição do processo de evolução do compartilhamento das informações do produto. São apresentados os primeiros esforços feitos neste campo até os padrões atuais ISO 10303 (conhecido também como STEP). É apresentado também o padrão ISO 6983 (códigos G e M), e são mostradas as suas limitações. Depois, é descrito o padrão ISO 14649 (STEP-NC), que busca superar as deficiências do padrão ISO 6983.

### **2.1 EVOLUÇÃO DO COMPARTILHAMENTO DAS INFORMAÇÕES DO PRODUTO**

Antes do advento da revolução industrial, o trabalho de engenharia era definido pelo modelo físico de um produto a ser fabricado. Para fabricar determinado componente, dever-se-ia assegurar que as suas dimensões correspondessem ao modelo da peça através da utilização de calibradores, por meio dos quais se transferia as medidas de um para o outro. As empresas de manufatura fabricavam um tipo de produto específico e não componentes genéricos de diversos tipos. Antes de 1800, um modelo físico definia ou descrevia o produto, e com a invenção dos desenhos de engenharia tempos depois pode-se ter descrições mais precisas e detalhadas dos produtos. O conceito do desenho mecânico se prolongou por quase duzentos anos (NIST, 1999).

Para se desenvolver produtos de qualidade, os processos de fabricação estavam associados ao método de descrição dos produtos. O desenho tornou-se a saída do processo de projeto e a entrada para o processo de fabricação. Tais desenhos foram convertidos em planos de processos, e estes últimos por sua vez se converteram em programas ou procedimentos para as operações de fabricação. Desta maneira cada processo tem a sua própria visão acerca dos dados do produto. Estas diferentes visões têm dificultado a disponibilização do conhecimento sobre os diferentes processos aos projetistas. Nas empresas de manufatura atuais, todos os processos do ciclo de vida de um produto

não são mais realizados pelo mesmo grupo de pessoas, e de fato os processos são distribuídos através de uma rede de fábricas.

Com o advento da era do computador e da informação há algumas décadas, os desenhos criados manualmente foram substituídos por desenhos digitais, criados com ajuda de ferramentas CAD (Projeto Assistido por Computador). Os desenhos criados com estas ferramentas têm uma vantagem significativa com relação aos desenhos em papel tais como maior produtividade, fácil revisão e arquivamento. A Figura 2.1 apresenta este processo de evolução das definições do produto.

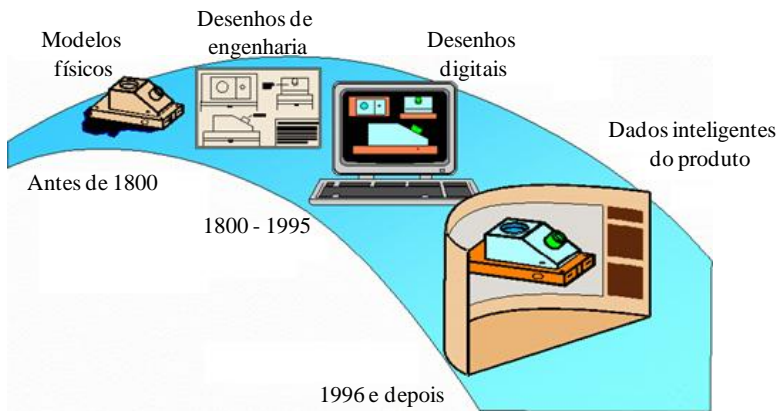


Figura 2.1. Evolução da capacidade de definição do produto.  
Fonte: National Institute of Standards and Technology (1999).

As ferramentas CAD abriram novas oportunidades como, por exemplo, permitir que as instruções de manufatura sejam derivadas automaticamente e que sejam executadas diretamente a partir do desenho. Entretanto, as ferramentas CAD e de manufatura (CAM) proliferaram para satisfazer as crescentes, complexas e diversas necessidades da engenharia, fazendo com que os formatos de cada ferramenta usada para capturar e armazenar dados também fossem diferentes (NIST, 1999). Enquanto desenhos em papel podem ser marcados com uma caneta e interpretados por qualquer um, um modelo de produto que não pode ser interpretado pela ferramenta CAD torna-se inútil. Para que as organizações possam compartilhar os projetos através de várias ferramentas CAD e CAM, estes projetos devem estar formatados de tal maneira que cada ferramenta possa reconhecê-los. Em uma época onde os grandes fabricantes frequentemente formam

empresas conjuntas para aproveitar as oportunidades de negócio, e onde os sócios em uma cadeia de suprimentos são chamados a oferecer uma gama cada vez mais complexa de serviços, este requisito está se tornando cada vez mais importante. Porém, este ainda é um grande problema para as empresas.

A maioria das empresas encontra dificuldade em pôr em vigor o uso de um conjunto de ferramentas CAD/CAM comuns dentro das suas próprias organizações, quanto mais através da múltipla cadeia de suprimentos e entre as empresas sócias. Devido à lacuna presente causada pelas ferramentas computacionais não compatíveis, é necessário um arquivo com formato neutro de intercâmbio de dados. Para resolver este problema, duas maneiras de intercâmbio foram propostas:

- Tradução direta, onde para cada dois sistemas deveria haver também dois tradutores.
- Tradução indireta, utilizando um arquivo intermediário, também chamado de arquivo neutro.

A Figura 2.2 ilustra como ambas as soluções funcionam. A tradução direta converte os dados diretamente em um só estágio. Estes tradutores são escritos tipicamente por empresas que se especializam em conversão de bases de dados CAD/CAM. Os tradutores diretos são considerados como programas dedicados, dois dos quais interligam dois sistemas como indicado pelas setas em ambas as direções na Figura 2.2(a). Já nos tradutores indiretos, cada sistema tem seu próprio par de processadores para transferir os dados de/para o arquivo com formato neutro. O tradutor que transfere os dados a partir do formato da base de dados de um determinado sistema para o formato neutro é chamado de pré-processador, ao passo que o tradutor que faz a transferência em sentido oposto é conhecido como pós-processador.

Cada uma das formas de traduzir tem suas vantagens e desvantagens. Os tradutores diretos fornecem uma solução satisfatória quando somente um número pequeno de sistemas está envolvido, porém à medida que o número de sistemas cresce, o número de programas tradutores torna-se proibitivo. A tradução direta tem a vantagem de que os arquivos gerados são menores e o tempo de processamento é menor. Contudo, o principal inconveniente é quando o número de sistemas CAx envolvidos aumenta, podendo o número de tradutores necessários ser tão alto que a solução torna-se muito cara.

Em geral, se os dados de um modelo devem ser transferidos entre todos os possíveis pares de  $n$  sistemas CAD/CAM, o número  $N$  de

tradutores será determinado pela equação (2.1). A adição de um novo sistema aos já existentes  $n$  sistemas implica em  $2n$  tradutores adicionais.

$$N = 2 \times \binom{n}{2} = n \times (n - 1) \quad (2.1)$$

Onde 
$$\binom{n}{2} = \frac{n!}{2!(n-2)!}$$

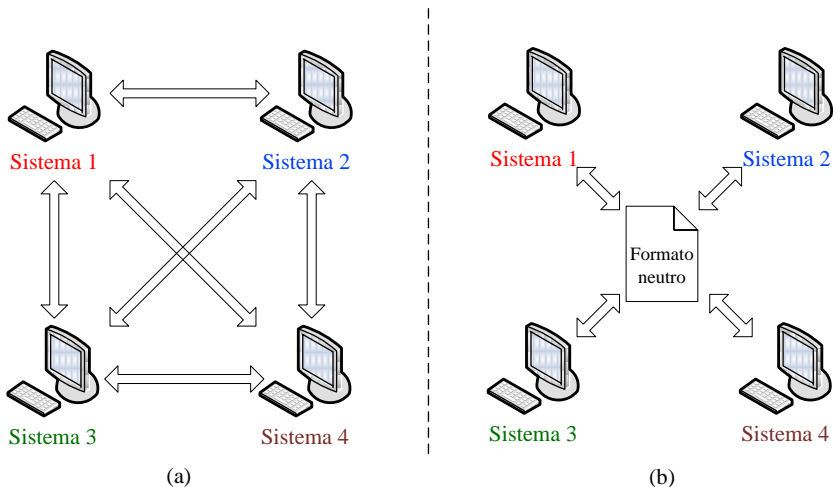


Figura 2.2 Troca de informações (a) com tradutores diretos (b) com interface neutra.

Fonte: National Institute of Standards and Technology (1999).

Por outro lado, os tradutores indiretos não sofrem incremento acentuado no número de programas a serem escritos em caso de adição de um novo sistema como no caso do método de tradução direta. Para o caso de  $n$  sistemas, o número total de tradutores indiretos que devem ser implementados é dado pela equação (2.2).

$$N = 2 \times n \quad (2.2)$$

A equação (2.2) aplica-se para  $n \geq 3$ . Para  $n = 3$ , as equações 2.1 e 2.3 resultam no mesmo número de tradutores, porém quando  $n > 3$ , o número de tradutores baseados na filosofia do uso de arquivo neutro é menor do que no baseado na tradução direta. Além disto, o método de tradução indireta proporciona comunicação estável entre os sistemas

CAD/CAM, protege contra a obsolescência dos sistemas, e elimina a dependência de um único fornecedor.

Em se tratando do desenvolvimento de soluções que usam a filosofia da utilização de arquivos neutros, várias técnicas foram propostas, e algumas delas são apresentadas a seguir.

### 2.1.1 IGES (Especificação Inicial de Intercâmbio de Gráficos)

Em 1979, as empresas desenvolvedoras de sistemas CAD se uniram à comunidade de usuários para criar o primeiro padrão nacional para intercâmbio de dados de sistemas CAD. Os sistemas CAD eram poucos, e somente havia alguns produtos com significativa penetração no mercado. Além disso, neste estágio inicial, os usuários não podiam compartilhar os dados destas ferramentas CAD com as suas próprias bases de dados desenvolvidas internamente (NIST, 1999).

Apesar da necessidade de intercâmbio de dados ser intuitiva do ponto de vista do usuário, ela era uma proposta ameaçadora para as empresas desenvolvedoras de sistemas CAD, os quais temiam o fato de que compartilhar a estrutura de suas bases de dados seria o equivalente a dar de graça a sua vantagem competitiva. Mesmo assim, um time multidisciplinar foi criado que incluía representantes de empresas como a Boeing e a General Electric. Uma sigla foi sugerida:

- I – (*Interim*) Provisório, para sugerir que não iria substituir o trabalho da ANSI.
- G – (*Graphics*) Gráfico, não geométrico, para reconhecer que acadêmicos poderiam vir com descrições matemáticas superiores.
- E – (*Exchange*) Intercâmbio, para sugerir que não iria ditar a maneira como os vendedores deveriam implementar as suas bases de dados.
- S – (*Specification*) Especificação, para não se impor como um padrão.

Posteriormente mudou-se o termo *Interim* para *Initial*, e ficou assim até hoje.

Depois de duas revisões, a equipe IGES apresentou o seu primeiro projeto em 1980, o qual continha a geometria, dados gráficos e anotações.



### 2.1.1.1 Descrição do padrão IGES

IGES define uma base de dados neutra na forma de um formato de arquivo que descreve um “modelo IGES” dos dados da modelagem de um determinado produto. O modelo IGES pode ser lido e interpretado por sistemas CAD/CAM não similares, portanto, os dados do produto podem ser trocados entre tais sistemas. IGES descreve: (a) as entidades de informação possíveis para serem usadas na construção do modelo IGES; (b) os parâmetros necessários para a definição das entidades do modelo; (c) as relações e associações entre tais entidades do modelo.

Como a maioria dos sistemas CAD/CAM, um modelo IGES baseia-se no conceito de entidades. Portanto a unidade de informação fundamental no modelo e no arquivo IGES é a entidade. Todos os dados que definem o produto são expressos como uma lista de entidades pré-definidas (ZEID, 1991), as quais são categorizadas em geométricas e não geométricas.

As entidades geométricas representam a definição da forma do produto incluindo curvas e superfícies. As relações que podem existir entre várias entidades são incluídas como parâmetros. Por exemplo, uma superfície B-spline pode ter uma curva B-spline como um dos seus parâmetros.

As entidades não geométricas proporcionam funções do modelo para enriquecer sua representação e incluem anotações e entidades de estrutura.

A Figura 2.3 mostra a troca de dados usando IGES. O sistema fonte é a origem ou sistema CAD/CAM que envia os dados, enquanto o sistema alvo é o receptor.

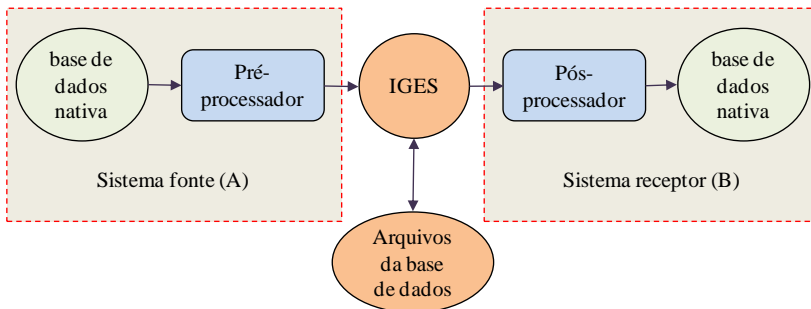


Figura 2.3. Intercâmbio de dados usando IGES.  
 Fonte: Adaptado de ZEID, 1991.

### 2.1.1.2 Formato e estrutura do arquivo neutro IGES

Um arquivo IGES está formado por seis seções as quais devem aparecer na seguinte ordem: seção *Flag* (opcional), seção início (*Start*), seção global (*Global*), seção acesso de diretório (*Directory Entry*), seção dados de parâmetros (*Parameter Data*) e seção fim (*Terminate section*) como apresentado na Figura 2.4. Os caracteres de identificação, também chamados de códigos de seção são: S, G, D, P e T (na Figura 2.4 aparecem entre parênteses, à exceção da seção *Flag*).

A seção **global** inclui a informação descrevendo o pré-processador e a informação que será necessária pelo pós-processador para interpretar o arquivo.

A seção **acesso de diretório** é uma lista de todas as entidades definidas no arquivo IGES junto a certos atributos associados a eles. Alguns dos atributos das entidades especificados nesta seção são: tipo de linha, número de camada, matriz de transformação, largura e cor da linha.

A seção **dados dos parâmetros** contém os dados definindo cada entidade listada na seção acesso de diretório. Por exemplo, uma entidade linha reta é definida pelas seis coordenadas de seus dois pontos.



Figura 2.4. Estrutura geral de um arquivo IGES.

Fonte: Adaptado de ZEID, 1991.

A seção **fim** contém um único registro que especifica o número de registros em cada uma das seções precedentes a fim de checar seu uso correto.

### 2.1.1.3 Limitações e desvantagens do IGES

Assim que o IGES começou a ser utilizado pela crescente comunidade de usuários, foram identificadas diversas limitações, que depois de um longo tempo contribuíram para a criação de outro padrão, o qual poderia vir a substituí-lo. Uma comissão avaliou o IGES, identificando as deficiências do mesmo (NIST, 1999), e algumas dessas deficiências são apresentadas a seguir:

- Tamanho dos arquivos/Tempo de processamento: o IGES foi duramente criticado por causa dos arquivos muito grandes, que

resultavam em horas ou até dias para serem analisados (levando em conta a velocidade dos computadores daqueles anos).

- Perda da informação durante o intercâmbio;
- Falta de disciplina técnica: isto porque percebeu-se que o IGES foi desenvolvido sem uma rigorosa disciplina técnica.
- Sistema de documentação automatizado: o IGES foi visto como um método para o intercâmbio de desenhos de engenharia, porém não com a capacidade de descrever os dados completos do produto (incluindo informação administrativa) para permitir uma automação mais sofisticada, o que poderia reduzir ou eliminar a intervenção humana ao traduzir.
- Subconjuntos: as empresas desenvolvedoras selecionavam e implementavam apenas parte do IGES, e por causa disto a troca de dados entre dois sistemas se fazia impossível sem previamente concordar-se no conteúdo que seria trocado.
- Teste de processadores: não existia nenhum mecanismo para testar os processadores ou resolver os erros entre dois processadores (WILSON, 1987).

### **2.1.2 PDES (Especificação de Intercâmbio de Dados do Produto)**

É um formato de intercâmbio para os dados do produto em suporte à automação industrial. Os “dados do produto” são interpretados de maneira que sejam mais genéricos que os “dados de definição do produto”, os quais formam o núcleo da filosofia IGES. Os “dados do produto” abrangem dados relevantes para o ciclo de vida inteiro de um produto, que incluem dados de projeto, manufatura, precisão, qualidade, testes e suporte.

A troca de dados PDES é feita de acordo com os “modelos de disciplina” ou “modelos mentais”. Exemplos disto são: modelos de disciplina de produtos mecânicos, de produtos elétricos, etc. Desta maneira, o conceito de modelos de disciplina tornam o PDES suficientemente flexível para acomodar futuros modelos e áreas de aplicação, quando estas estejam disponíveis.

A metodologia PDES envolve uma arquitetura de três camadas, a qual é apresentada na Figura 2.5.

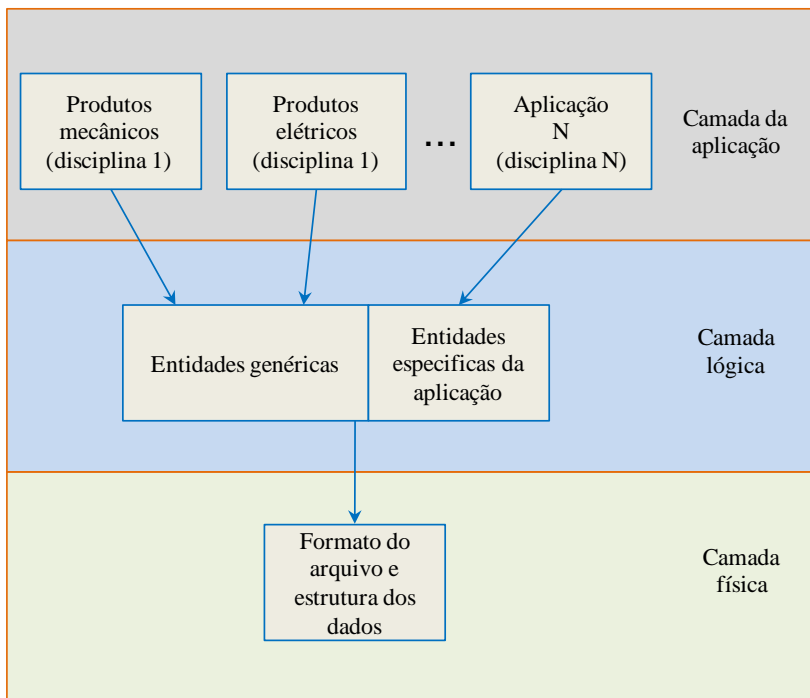


Figura 2.5. Arquitetura de três camadas PDES.

Fonte: Adaptado de ZEID (1991).

A camada de aplicação é a interface entre o usuário e o PDES. Ela contém todas as descrições e informações das diversas áreas de aplicação. Estas descrições e aplicações são expressas formalmente dentro do PDES através das técnicas de modelagem da informação (conhecidas também por “modelos de referência”).

A camada lógica tem o propósito de fornecer a descrição consistente, e independente, dos construtores dos dados que contém a informação a ser trocada. São identificados os construtores da aplicação genéricos e os específicos.

A camada física ocupa-se das estruturas e com os formatos dos dados do arquivo de intercâmbio propriamente dito. O seu principal objetivo é estabelecer e manter a eficiência no tamanho do arquivo e no tempo de processamento para evitar problemas parecidos aos experimentados usando-se o IGES.

### 2.1.3 Outros esforços e contribuições para a implementação de arquivos neutros

Houve outros esforços além do IGES e do PDES tratados nos itens 2.1.1 e 2.1.2 respectivamente, com o propósito de resolver o problema da troca de dados, alguns dos quais são: SET (*Standart d'exchange et de Transfer*) e VDA FS (*Verband der Deutschen Automobilindustrie Flächenchnittstelle*), os quais evoluíram em várias versões, como apresentado na Figura 2.6.

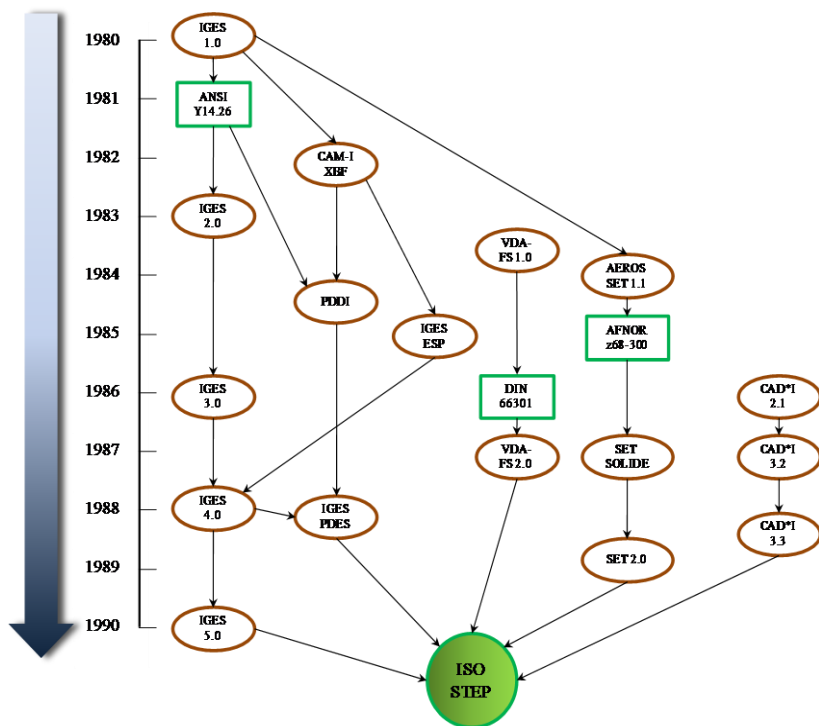


Figura 2.6. Evolução dos padrões para intercâmbio de dados do produto.  
Fonte: MOURA (2003).

## 2.2 O PADRÃO ISO 10303 (STEP)

Em julho de 1984 foi realizada a primeira reunião do subcomitê ISO TC 184/SC4 no Instituto Nacional de Padrões e Tecnologia (NIST). Participaram representantes de países tais como Canadá, França, Alemanha, Suíça, Inglaterra, e Estados Unidos. O motivo desta reunião foi criar um padrão internacional que permitisse capturar a informação do modelo computadorizado do produto de uma forma neutra sem perda da integridade, através do ciclo de vida do produto.

STEP é o acrônimo de “*Standard for the Exchange of Product Model Data*”, e descreve a coleção de documentos padrão publicados sob a norma ISO 10303.

A referência inicial do STEP foi ter a coleção de todos os dados do modelo do produto. A idéia de modelo de produto foi derivada da utilização dos sistemas de projeto assistido por computador (CAD). A segunda referência foi uma troca filosófica de “dados do produto” (implicando que qualquer quantidade de dados do produto poderia ser intercambiada) (NIST, 1999).

STEP fornece uma representação da informação do produto junto com os mecanismos e definições necessários para permitir que os dados do produto sejam intercambiados. O intercâmbio é entre ambientes e sistemas computacionais diferentes associados ao ciclo de vida completo de um produto, incluindo projeto, manufatura, utilização, manutenção e descarte. A informação do produto gerada durante esse processo pode ser usada para diferentes propósitos. Isto pode envolver diversos sistemas de computadores, inclusive alguns podem estar localizados em organizações diferentes. Para suportar tais usos, as organizações devem ser capazes de representar a informação do produto de uma forma interpretável, e é necessário conservar tal informação completa e consistente quando intercambiada entre diferentes sistemas (SCRA, 2006).

O objetivo global deste padrão é fornecer um mecanismo capaz de descrever os dados do produto ao longo do seu ciclo de vida, independentemente de qualquer sistema particular. A natureza desta descrição o torna conveniente não somente para intercâmbio de arquivos neutros, mas também como uma base para a implementação e compartilhamento de bases de dados do produto e para o seu arquivamento. O objetivo final é uma base de dados da informação do produto integrada que seja acessível e útil a todos os recursos necessários para suportar um produto além do seu ciclo de vida.

### 2.2.1 Estrutura do Padrão ISO 10303

O padrão STEP está organizado em uma série de partes, cada uma publicada separadamente. Estas partes incluem uma das seguintes séries: métodos de descrição, recursos integrados, protocolos de aplicação, conjuntos de testes abstratos, métodos de implementação, e testes de conformidade (SCRA, 2006 e LOFFREDO, 2000).

LOFFREDO (2000) divide a estrutura em duas partes: Infraestrutura e Modelos de informação, como pode-se ver na Figura 2.7.

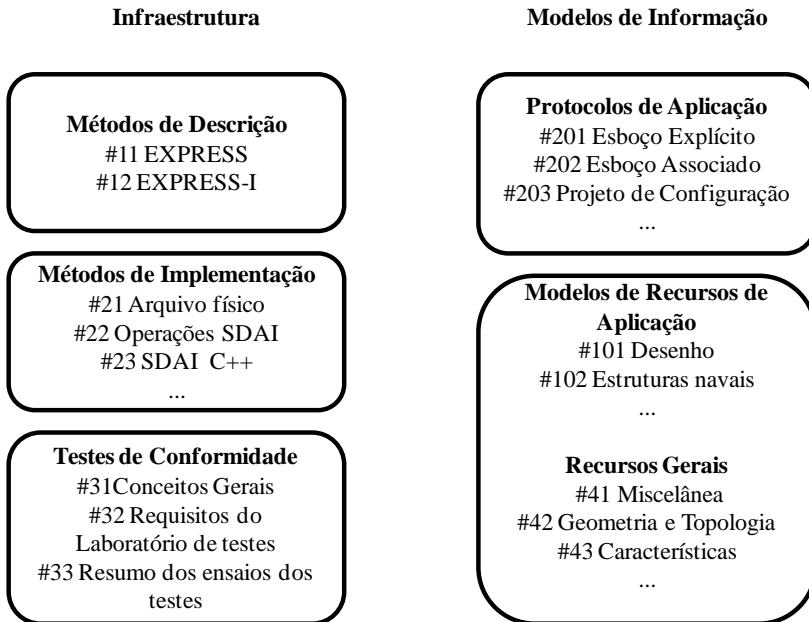


Figura 2.7. Estrutura de alto nível do STEP

Fonte: Adaptado de LOFFREDO (2000).

O National Institute of Standards and Technology (NIST) apresenta a arquitetura do STEP como mostrado na Figura 2.8, a qual está dividida em quatro áreas fundamentais:

- Métodos de descrição,



- Métodos de implementação,
- Testes de conformidade e
- Especificações dos dados.

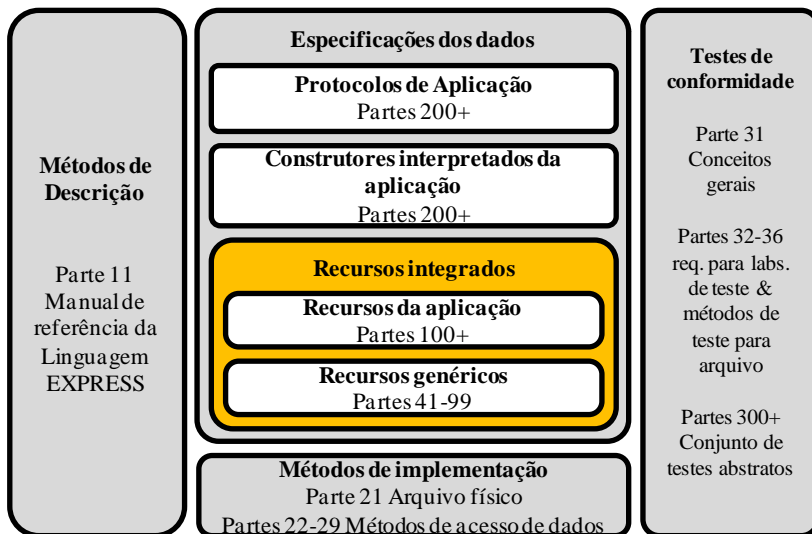


Figura 2.8. Resumo da estrutura do STEP.  
Fonte: Adaptado de NIST (1999).

A seguir se tratará estas partes com mais detalhe.

### 2.2.1.1 Métodos de descrição

São mecanismos comuns para especificar os dados para descrever os construtores dos dados do STEP. Estes métodos de descrição incluem a linguagem de especificação formal de dados desenvolvida dentro do STEP, conhecida como EXPRESS (ISO 10303-11, 1994). Outro método de descrição inclui a forma gráfica da linguagem EXPRESS (conhecido também como EXPRESS-G).

Foram avaliadas as linguagens existentes para sua utilização dentro do STEP, porém nenhuma delas satisfaz todos os requisitos do padrão.

### 2.2.1.1.1 A linguagem EXPRESS

O STEP usa uma linguagem formal EXPRESS para especificar a informação do produto a ser representado. O uso de uma linguagem formal permite a precisão e a consistência da representação e facilita o desenvolvimento das implementações.

A linguagem EXPRESS está definida na parte 11 do padrão (ISO10303-1, 1994, ISO10303-11, 1994). Diferentemente das linguagens de base de dados tais como SQL (ISO/IEC 9075, 1992), ou as linguagens de programação tais como o C (ISO/IEC 9899, 1990), EXPRESS não mistura as tarefas de modelagem da informação com as de programação ou com as tarefas de projeto de bases de dados, e não é específico para um sistema particular de base de dados ou de programação. Ela fornece a sintaxe e a semântica para representar a informação em um modo uniforme, preciso e compacto. A representação em EXPRESS pode ser feita de duas formas:

- Como uma linguagem formal: que usa uma notação léxica e uma sintaxe definida por uma gramática própria e;
- Como uma representação gráfica: que fornece uma ilustração bastante compacta, rica e amigável, chamada EXPRESS-G.

A Figura 2.9 apresenta um exemplo de um esquema em EXPRESS, e na Figura 2.10 se apresenta o seu correspondente diagrama em EXPRESS-G, onde as linhas mais delgadas representam atributos, e a linha mais larga representa uma relação de classe e subclasse (segundo este diagrama, fabricante é subclasse de proprietário), e as caixas representam classes.

É preciso destacar que todos os dados encontrados em um esquema EXPRESS podem ser representados em um equivalente diagrama EXPRESS-G, porém nem todo diagrama EXPRESS-G pode ser transformado em esquema EXPRESS, pois há informações que não aparecem no esquema gráfico.

Os construtores usados na definição de modelos de dados em EXPRESS, segundo SCHENCK e WILSON (1994), são:

- SCHEMAS: subdivisão funcional de um modelo que permite a reutilização de informações entre diferentes modelos.
- TYPE: descreve os tipos primitivos de dados (inteiro, real, booleano, string)

- ENTITY: são as unidades básicas de informação que compõem o *schema*.
- SUBTYPE, SUPERTYPE: estabelecem as relações hierárquicas entre diferentes entidades.
- FUNCTION, PROCEDURE, RULE: adicionam restrições ao modelo de dados.

```

ENTITY proprietario SUPERTYPE OF (fabricante);
    nome_proprietario : STRING;
UNIQUE
    nome_proprietario;
END_ENTITY;

ENTITY fabricante SUBTYPE OF (proprietario);
END_ENTITY;

ENTITY modelo_carro;
    nome_modelo      :STRING;
UNIQUE
nome_modelo;
END_ENTITY;

ENTITY car;
    fabricante                :fabricante;
    numero_serie              :NUMBER;
    designacao_modelo        :modelo_carro;
    ano_modelo                :NUMBER;
UNIQUE
    numero_serie, fabricante;
END_ENTITY;

```

Figura 2.9. Exemplo de um esquema em EXPRESS.

Fonte: Adaptado de NIST, (1999).

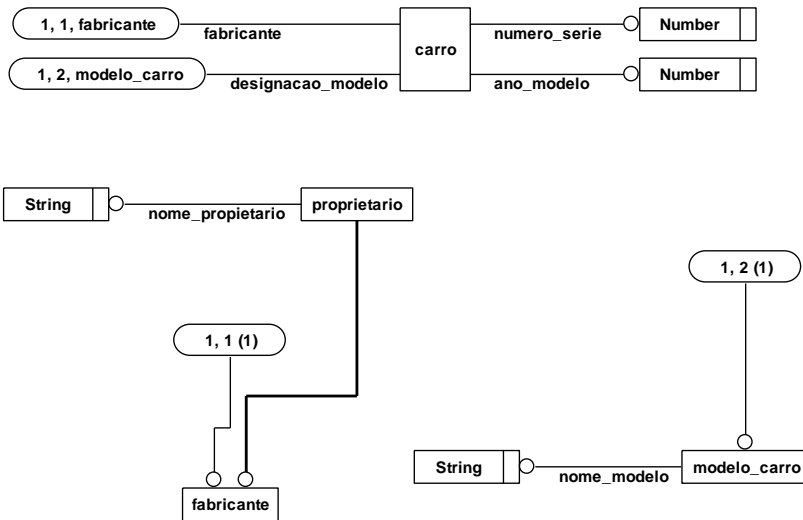


Figura 2.10. Diagrama EXPRESS-G para o esquema da Figura 2.9.

Fonte: Criado pelo autor.

A linguagem EXPRESS adota vários conceitos da orientação a objetos (porém há alguns conceitos que diferem da programação orientada a objetos) e está estruturada em esquemas que representam o modelo do produto. Um esquema consiste em entidades, que são os principais objetos e tipos de dados em que se apoiam as definições destas entidades. Dentro das entidades estão encapsulados atributos e vínculos que restringem os valores dos atributos (MOURA, 2003).

### 2.2.1.2 Métodos de implementação

São técnicas de implementação padrão para as estruturas de informação especificadas apenas pelos protocolos de aplicação. Esta série inclui a estrutura de intercâmbio do arquivo físico (ISO 10303-21, 1994), a Interface de Acesso de Dados Padrão (SDAI) (ISO 10303-22, 1998), e suas linguagens de ligação (ISO 10303-23 (linguagem C++), ISO 10303-24 (linguagem C), ISO 10303-27 (Java), ISO 10303-28 (XML)).

### 2.2.1.2.1 A linguagem XML

Desde meados da década de 2000, tem-se visto um grande crescimento da Web, e aliado a este grande crescimento tem-se observado as limitações da linguagem HTML (HyperText Markup Language) (DAUM e METER, 2002).

Em resposta à crescente demanda por extensões, bem como a necessidade de interoperabilidade, houve um esforço inicial de estender a linguagem HTML. Entretanto, essa solução apenas foi um paliativo. Como resultado, fez-se um esforço coordenado pelo W3C (WWW Consortium, [www.w3c.org](http://www.w3c.org)) visando oferecer uma nova linguagem que pudesse satisfazer as necessidades de interoperabilidade, escalabilidade, e flexibilidade, permitindo a sua fácil extensão (DE OLIVEIRA, 2004).

Surgiu então a linguagem XML (eXtensible Markup Language), que se caracteriza por fornecer independência de dados bem como separar o conteúdo da apresentação. Um programa em XML abrange a descrição de dados, tornando-se possível o seu processamento por uma aplicação. Esta linguagem tem sido cada vez mais utilizada por desenvolvedores de aplicativos devido ao suporte que oferece tanto à interoperabilidade quanto à funcionalidade da Web. XML se baseia em texto, o que permite a qualquer pessoa escrever um código em XML, sendo facilmente compreensível pelas pessoas quanto manipulável pelos computadores (MOULTIS, KIRK, 2000).

XML é uma linguagem simples, muito flexível, tendo sido originalmente projetada para enfrentar os grandes desafios da publicidade eletrônica de grande escala. Porém a XML tem também um papel importante no compartilhamento de uma ampla variedade de dados na Web e em outros lugares (W3C, 2011).

A norma ISO 10303-28 (ISO 10303-28, 2002), também conhecida como STEP-XML, especifica o uso da linguagem XML para representar esquemas EXPRESS e os dados que são manipulados através destes esquemas EXPRESS. Este é um método alternativo para a troca de dados dos arquivos STEP.

Para a representação dos dados correspondentes a um esquema EXPRESS, esta parte da norma ISO 10303 tem duas grandes abordagens. Uma abordagem é para especificar um único conjunto de declarações de “marcas” (*tags*) (isto é, definições em XML) que são independentes do esquema EXPRESS, e podem representar dados para qualquer esquema. Esta abordagem denomina-se vinculação tardia (*late binding*). A segunda abordagem é para especificar os resultados da

geração do conjunto de declarações de marcas que são dependentes do esquema EXPRESS. Esta abordagem se chama de vinculação antecipada (*early binding*).

### 2.2.1.3 Testes de conformidade

Os testes de conformidade estão cobertos em duas partes do padrão STEP:

- Estrutura e metodologia de testes de conformidade e
- Testes abstratos.

A estrutura e metodologia de testes de conformidade descreve como deve ser realizado o teste das implementações das várias partes do STEP. Os métodos de teste de conformidade estão padronizados nas séries ISO 10303-30.

Os testes abstratos contêm o conjunto de casos necessários para testar a conformidade de uma implementação de um protocolo de aplicação do STEP. Cada caso de teste abstrato especifica os dados de entrada a serem fornecidos à implementação sob teste, junto com a informação sobre como avaliar as capacidades de tal implementação.

### 2.2.1.4 Especificação dos dados

O último grande componente da arquitetura STEP é a especificação dos dados. Há quatro séries de especificação de dados na estrutura do STEP, embora conceitualmente haja três tipos primários de especificação de dados:

- Recursos integrados,
- Protocolos de aplicação e
- Construtores interpretados da aplicação.

Todas as especificações dos dados são documentadas usando os métodos de descrição. A Figura 2.11 apresenta a arquitetura das especificações dos dados.

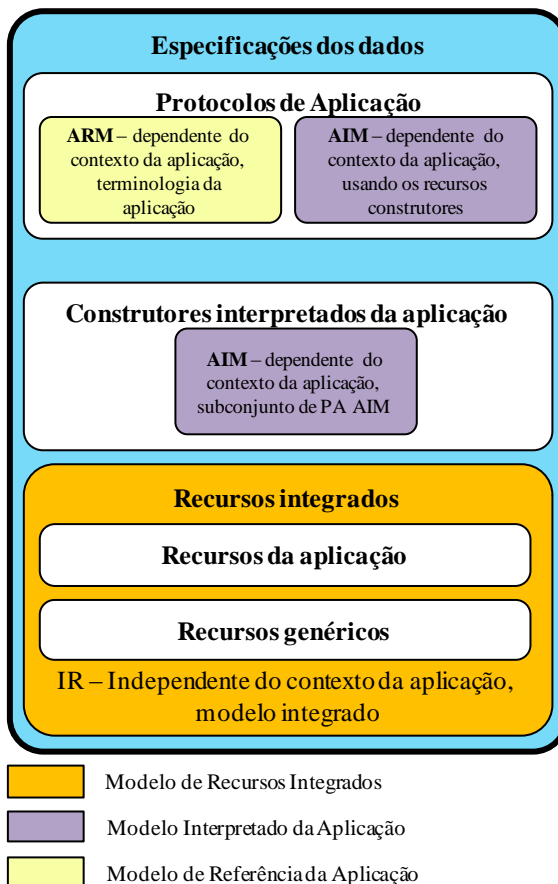


Figura 2.11. Arquitetura da especificação dos dados.  
 Fonte: Adaptado de NIST (1999).

#### 2.2.1.4.1 Recursos Integrados

Os recursos integrados constituem um único modelo conceitual para os dados do produto. Os construtores dentro dos recursos integrados são os elementos semânticos básicos usados para a descrição de qualquer produto em qualquer etapa do ciclo de vida do produto. Embora os recursos integrados sejam usados como a base para o desenvolvimento dos protocolos de aplicação, eles não se destinam à implementação direta. Eles definem os componentes reutilizáveis

destinados a serem combinados e refinados para satisfazer uma necessidade específica.

Os recursos integrados são formados por duas partes: os recursos integrados genéricos, e os recursos integrados da aplicação. As duas partes têm funções similares: elas são especificações de dados padrão independentes do contexto da aplicação que suportam o desenvolvimento consistente dos protocolos de aplicação (AP) através de vários contextos de aplicação.

Exemplos de construtores de recursos genéricos incluem ponto cartesiano, data e produto. Estes construtores poderiam ser usados potencialmente por qualquer aplicação. Os recursos genéricos integrados estão padronizados nas séries ISO 10303-40. Os recursos genéricos cobrem:

- Descrição do produto e suporte (ISO 10303-41).
- Representação geométrica e topológica (ISO 10303-42).
- Estruturas de representação (ISO 10303-43).
- Configuração da estrutura do produto (ISO 10303-44).
- Materiais (ISO 10303-45).
- Apresentação visual (ISO 10303-46).
- Variação de tolerâncias de forma (ISO 10303-47)
- Estrutura e propriedades do processo (ISO 10303-49).

#### 2.2.1.4.2 *Protocolos de Aplicação (AP)*

São as especificações de dados do STEP que podem ser implementadas. APs incluem um modelo de informação em EXPRESS que satisfaz as necessidades dos dados do produto para um determinado contexto de aplicação. APs podem ser implementados usando um ou mais métodos de implementação. Eles são os componentes centrais da arquitetura STEP, a qual é projetada principalmente para suportar e facilitar o desenvolvimento dos Protocolos de Aplicação. Os elementos de um protocolo de aplicação são:

- Modelo de Atividade da Aplicação (AAM): é um modelo de função que descreve as atividades e processos de um domínio de contexto da aplicação determinado. Ele é um requerimento.
- Modelo de Referência da Aplicação (ARM): é um modelo de informação que descreve os requisitos da informação e as restrições para uma área do contexto de



aplicação. O modelo usa terminologia e regras específicas de determinada aplicação que são familiares aos especialistas na área de aplicação.

- Modelo Interpretado da Aplicação (AIM): é um modelo de informação que descreve as estruturas de dados STEP requeridas pela equivalência funcional com os contextos de aplicação AAM e ARM.
- Classes de Conformidade: são descrições das populações de arquivos, os mesmos que servem para definir o uso dos Protocolos de Aplicação.

Os Protocolos de Aplicação estão padronizados nas séries ISO 10303-200.

#### 2.2.1.4.3 *Construtores Interpretados da Aplicação (AICs)*

São especificações de dados que satisfazem uma necessidade específica dos dados do produto que surge em mais de um contexto de aplicação.

Um construtor interpretado da aplicação especifica as estruturas e semânticas dos dados a serem usados no intercâmbio de dados do produto comuns a dois ou mais protocolos de aplicação. Protocolos de Aplicação com requisitos de informação similar são comparados sistematicamente para determinar a equivalência funcional. Os AICs estão padronizados na série ISO 10303-500.

### 2.2.2 **O padrão STEP-NC**

Desde o projeto e construção da primeira máquina controlada numericamente (NC), a manufatura automática tem sido um sonho que os engenheiros têm buscado tornar realidade. Durante os últimos 50 anos, as máquinas têm evoluído desde as simples máquinas com controladores que não tinham memória, controladas por instruções em uma fita perfurada, para as altamente sofisticadas estações de trabalho multi-processos atuais controladas numericamente por computador (CNC). Estas estações têm capacidades tais como controle de múltiplos eixos, controle adaptativo, compensação de desgaste, bem como a execução de múltiplos processos de usinagem (p.ex. corte a laser, torneamento, fresamento, retificação, eletroerosão a fio). Porém tais

capacidades têm dificultado significativamente a tarefa de programação, fazendo com que ferramentas de software *off-line* para CAD/CAM sejam necessárias para a rápida e eficiente geração e verificação do código NC (XU e NEWMAN, 2006).

Embora as máquinas tenham mudado radicalmente ao longo dos anos, a linguagem padrão de programação de baixo nível tem permanecido basicamente igual à utilizada nas primeiras máquinas, linguagem essa que se baseia na norma ISO 6983 (ISO 6983-1, 1982), também conhecido como código G e M.

Desenvolvedores e usuários de sistemas CAD, CAPP, CAM e CNC têm procurado uma linguagem comum para tais sistemas, a qual venha a integrar e traduzir o conhecimento de cada um desses estágios. Embora existam muitas ferramentas CAM para dar suporte à manufatura, utilizando-se as máquinas de controle numérico, o problema da portabilidade e da interoperabilidade de um sistema para outro ainda é uma das questões chave que limitam o uso amplo dessas ferramentas. Algumas soluções foram propostas, visando uma maneira de intercâmbio de dados padrão, tais como SET, VDA, e IGES (NIST, 1999), entretanto elas não foram totalmente convenientes para as necessidades da indústria CAD/CAPP/CAM.

O padrão ISO conhecido como STEP-NC foi desenvolvido com a finalidade de fornecer um modelo de dados para um novo conjunto de controladores CNC inteligentes. Ao seu Modelo de Referência da Aplicação (ARM) tem sido dado o número ISO 14649 (ISO 14649-1, 2003), o mesmo está direcionado especificamente à programação NC, buscando satisfazer o objetivo de um controlador CNC em um nível mais elevado, com a geração do programa tornando-se mais rápida e consistente. Em conjunto com o ARM está o Modelo Interpretado da Aplicação (AIM) do STEP-NC, ISO 10303 parte 238 (ISO/DIS 10303-238, 2003).

A principal diferença entre o ARM (ISO 14649) e o AIM (ISO 10303-238) é o grau que eles usam a arquitetura técnica e os métodos de representação do padrão STEP (FEENEY *et. al.*, 2003).

O Modelo de Referência da Aplicação do STEP-NC (ISO 14649) fornece um modelo de dados orientado a objetos para CNCs com uma interface de dados estruturada e detalhada que incorpora a programação baseada em *features* a serem usinadas, os tipos de ferramentas usadas, as operações a serem realizadas (*Workingsteps*), e o plano de processo (*Workplan*). A Figura 2.12 apresenta este modelo de maneira simplificada na linguagem EXPRESS-G.

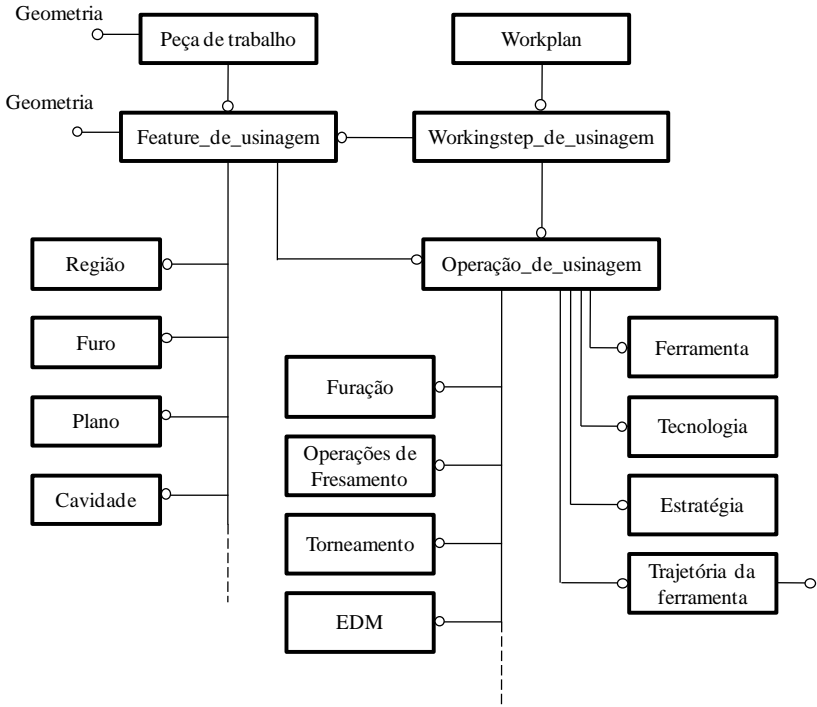


Figura 2.12 Modelo de dados do padrão ISO 14649 em diagrama EXPRESS-G. Fonte: Adaptado de ISO 14649-1, (2002).

### 2.2.2.1 O padrão ISO 6983

O padrão ISO 6983 descreve o formato das instruções do programa para máquinas de controle numérico. É um formato genérico de programação, ou seja, não é um formato para um tipo de máquina específica. Este padrão é muito flexível, o qual não garante a interoperabilidade entre máquinas NC. O padrão ISO 6983 está dividido em três partes:

- Parte 1: Foi publicada em 1982, e a última versão em 2009. Especifica os requisitos e fornece recomendações para um formato dos dados para posicionamento e movimentação de máquinas NC. Esta parte ajuda na coordenação do projeto do sistema a fim de minimizar a diversidade de programas

manuscritos requeridos. Promove a uniformidade das técnicas de programação e promove a intercambiabilidade de programas entre máquinas controladas numericamente do mesmo tipo por classificação, processo, função, tamanho e precisão (ISO 6983-1, 2009).

- Parte 2: Especifica a codificação e a elaboração das funções preparatórias (G) e miscelâneas (M) (classes 1 a 8). É apenas um rascunho (ISO 6983-2, 1982);
- Parte 3: especifica a codificação das classes 1 a 9 das funções miscelâneas. Esta parte também não foi publicada oficialmente, apenas permanece como rascunho.

Um exemplo de uma linha de comando típica de código G se apresenta na Figura 2.13.

A programação de máquinas NC usando o padrão ISO 6983 resulta em programas extensos e difíceis de manipular. Mudanças e correções de última hora de programas complexos são difíceis de gerenciar no chão de fábrica, e o controle da execução do programa na máquina é muito limitado (MAEDER *et. al.*, 2002).

A fim de simplificar as atividades de programação e para fazer com que os programas NC sejam mais compactos, a norma ISO 6983 permite um conjunto especial de código G chamado de ciclos fixos, os quais são funções pré-programadas, como por exemplo ciclos para furação profunda com quebra de cavaco (G83) (também conhecida como estratégia “pica-pau”), entrando e saindo para facilitar a quebra e remoção de cavacos em operações de furação profunda (ROSSO JUNIOR, 2005).

Assim como o ciclo fixo G83, existem outros ciclos implementados por cada fabricante.

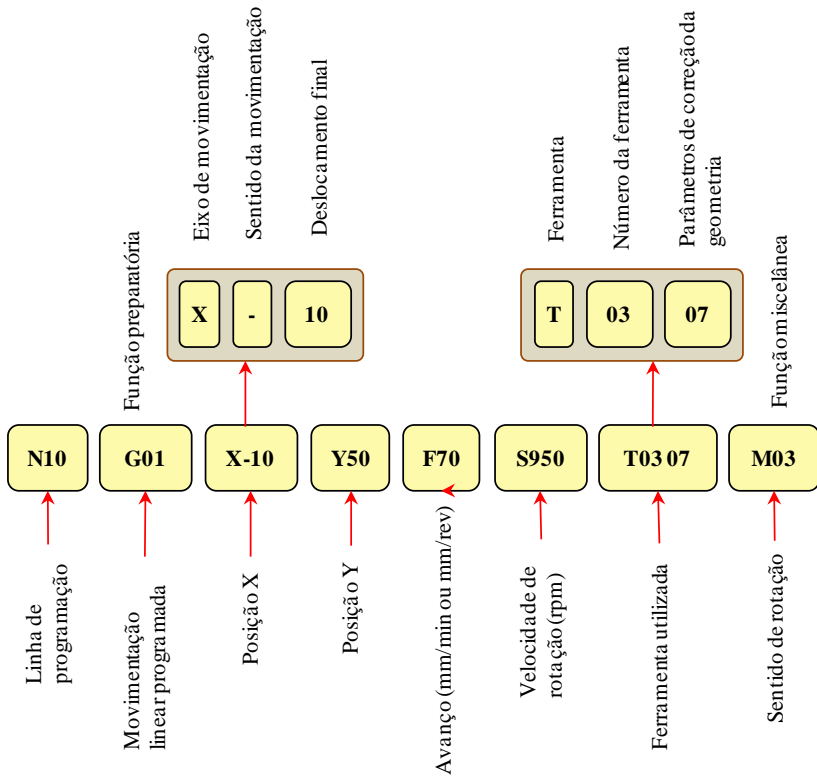


Figura 2.13. Linha de programa típica usando a norma ISO 6983.  
Fonte: Criado pelo autor.

Outra característica da norma ISO 6983 é que ela permite a implementação de subprogramas e macros. Subprogramas são conjuntos de linhas de código que são usadas várias vezes em um ou mais programas principais. A implementação destes subprogramas é específico de cada fabricante e tem diferenças entre dois controladores. (ROSSO JUNIOR, 2005).

Estas variações de programação dos ciclos fixos e dos subprogramas, que segundo MAEDER *et. al.*, (2002) são mais de 5000 “dialetos”, resulta em problemas de interoperabilidade entre máquinas CNC. Portanto, é necessária uma abordagem padronizada que permita a interoperabilidade na manufatura CNC, e é por esta razão que os padrões relacionados à fabricação evoluíram (ROSSO JUNIOR, 2005).

### 2.2.2.2 O padrão ISO 14649

As empresas de manufatura moderna são constituídas por instalações espalhadas ao redor do mundo. Tais empresas possuem equipamentos de centenas de fabricantes diferentes. Grandes quantidades de informações devem ser trocadas entre as diferentes instalações e entre as máquinas. Atualmente pode se transferir informação através de redes globais com segurança, graças aos padrões de comunicação digital.

O padrão ISO 14649 é um novo modelo de intercâmbio de dados entre sistemas CAD/CAM e máquinas CNC. Ele busca solucionar as limitações do padrão ISO 6983 por meio da especificação dos processos de usinagem ao invés da movimentação da ferramenta da máquina, usando o conceito de orientação a objetos dos *Workingsteps*. (ISO 14649-1, 2002). Os *Workingsteps* correspondem às *features* de usinagem e aos parâmetros do processo associados de alto nível.

A Figura 2.14 mostra que ao usar a norma ISO 6983, o fluxo de dados é unidirecional, porém o padrão ISO 14649 dá suporte ao fluxo bidirecional, ou seja, as informações geométricas e de usinagem podem ser transferidas entre o sistema CAD/CAM e o controlador STEP-NC. Também pode ser visto na Figura 2.14 (b) que a informação das trajetórias de movimentação é opcional, e idealmente deveria ser gerada para a máquina que irá efetuar a usinagem através do seu controlador aderente ao padrão STEP-NC (XU e NEWMAN, 2006). A Figura 2.15 mostra que a utilização do padrão ISO 6983 implica em uma linguagem de baixo nível, enquanto a programação usando o padrão ISO 14649 é de alto nível.

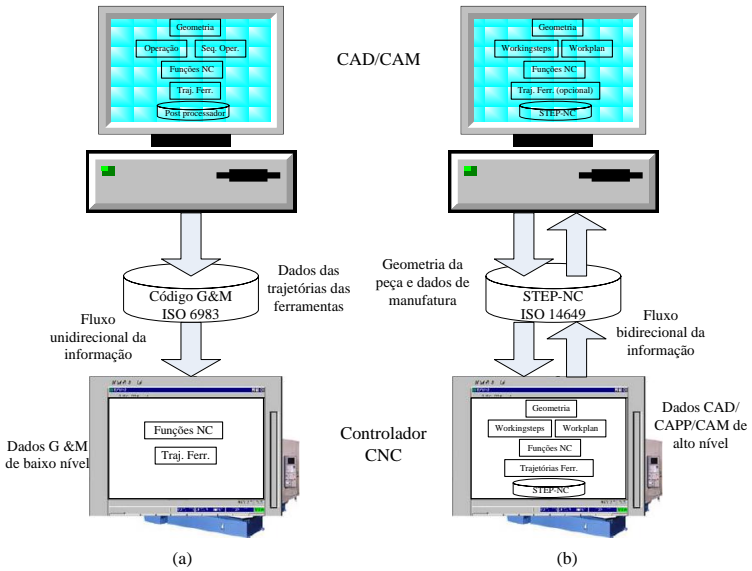


Figura 2.14. (a) Fluxo unidirecional de informações. (b) fluxo bidirecional de informações com STEP-NC  
 Fonte: adaptado de ALLEN *et. al.*, (2003).

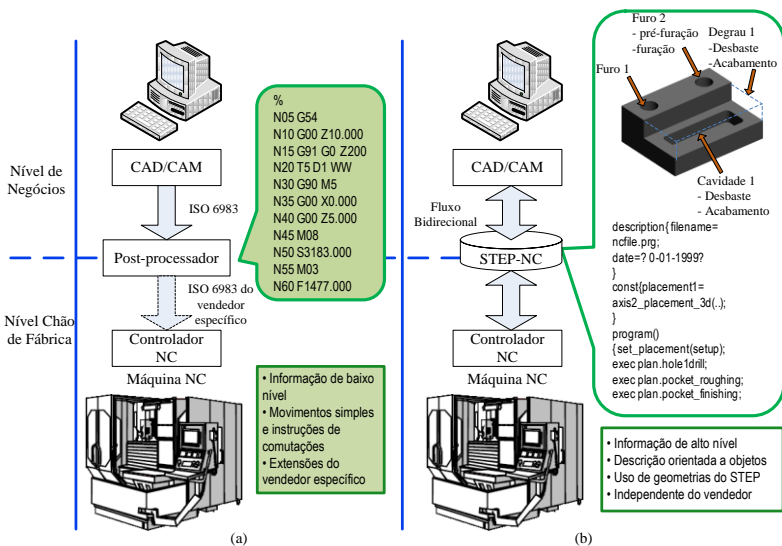


Figura 2.15. (a) Utilização do padrão ISO 6983 (situação atual). (b) Utilização da nova interface STEP-NC  
 Fonte: Adaptado de XU e MAO (2004).

Segundo a abordagem da norma ISO 14649, os CNCs seriam os responsáveis por traduzir os *Workingsteps* em movimentações dos eixos da máquina e em operações das ferramentas.

Uma outra vantagem importante do padrão ISO 14649 é que ele usa os modelos de dados já existentes e definidos no padrão ISO 10303 (STEP).

### 2.2.2.3 O modelo de dados do padrão ISO 14649

O modelo de dados do padrão ISO 14649 contém os dados geométricos, dados das *features* de usinagem e dados do processo.

Os dados geométricos normalmente são originados a partir de sistemas CAD, e são descritos no protocolo de aplicação 203 da norma ISO 10303. Esta inclui toda a informação necessária para definir a geometria de uma peça acabada (ISO 14649-1, 2002).

Por sua vez, os dados das *features* de usinagem se originam tipicamente em sistemas CAM. Os dados do processo de usinagem também se originam no sistema CAM, e definem os parâmetros tecnológicos a serem usados durante o processo de corte, que incluem a velocidade de corte, velocidade de rotação da ferramenta e as descrições do ferramental requerido para executar cada uma das operações de usinagem (ISO 14649-1, 2002).

Durante o desenvolvimento do padrão ISO 14649, foram feitos esforços para harmonizar as definições das *features* com as que já existiam no padrão ISO 10303 AP – 224, porém há diferenças entre ambos (ISO/DIS 10303-238, 2003).

A Figura 2.16 apresenta um diagrama com a descrição geral do modelo de dados do padrão ISO 14649.



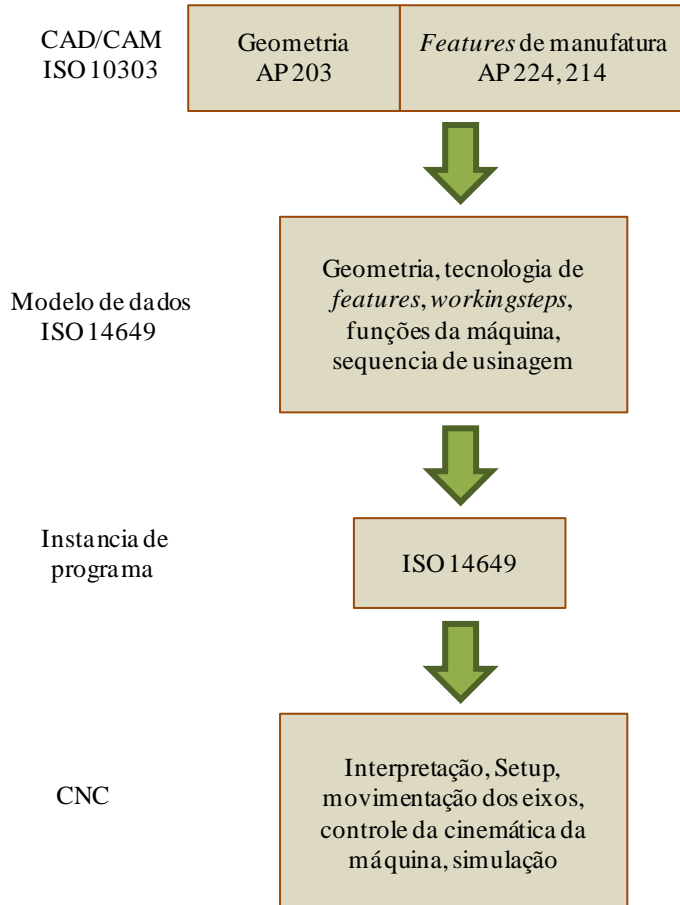


Figura 2.16. Descrição geral do modelo de dados ISO 14649.  
 Fonte: Adaptado de ISO 14649-1 (2002).

#### 2.2.2.4 Estrutura e organização do padrão ISO 14649

O padrão ISO 14649 possui uma estrutura típica do padrão STEP, onde há uma norma geral de diretrizes e muitas partes as quais descrevem cada ramo do processo, conforme apresentado na Figura 2.17.

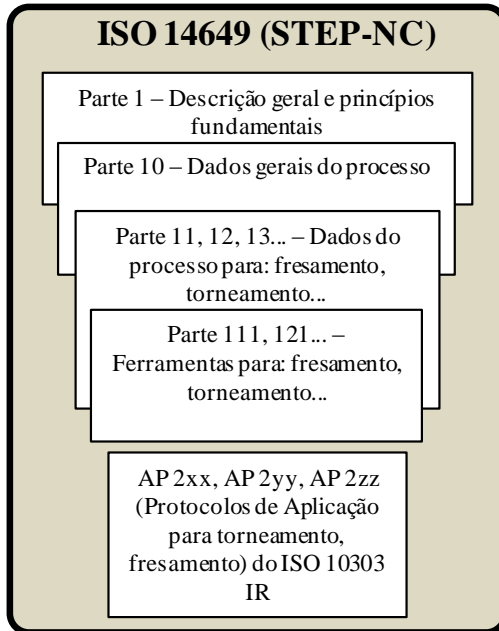


Figura 2.17. Estrutura do padrão ISO 14649.

Fonte: Adaptado de ROSSO JUNIOR (2005).

#### 2.2.2.4.1 ISO 14649 Parte 1: Descrição Geral e princípios fundamentais

A parte 1 da norma fornece uma introdução e uma descrição geral do modelo de dados para controladores NC, e também explica as vantagens do padrão e seus princípios básicos, baseado nos conceitos dos dados do produto (ISO 14649-1, 2002). Esta parte divide a estrutura geral de um arquivo físico em duas seções: cabeçalho e dados, e está em concordância com o padrão ISO 10303-21.

A primeira seção tem a palavra chave “HEADER”, e nesta parte encontram-se informações gerais e comentários referentes ao programa da peça, tais como: o nome do arquivo, autor, data de criação, organização, etc.

A segunda e principal seção do arquivo do programa é marcada com a palavra chave “DATA”, e contém todas as informações acerca da geometria, *features* e tarefas de manufatura. O conteúdo desta parte está dividido em três partes: o *Workplan* e executáveis com a descrição da

sua tecnologia, as *features* de manufatura, e descrição da geometria. A Figura 2.18 apresenta a estrutura genérica de um programa em ISO 14649.

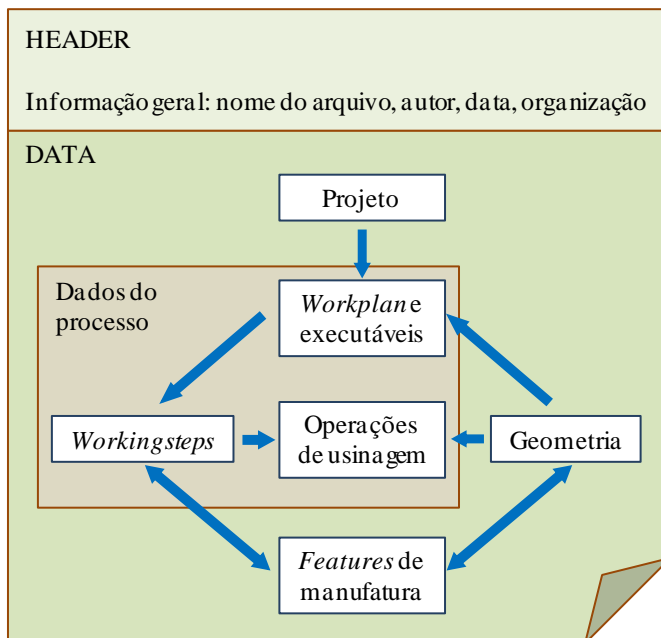


Figura 2.18. Estrutura de um programa segundo o padrão ISO 14649.  
Fonte: Adaptado de ISO 14649-1, (2002).

#### 2.2.2.4.2 ISO 14649 Parte 10: Dados gerais do processo

A parte 10 do padrão especifica os dados do processo necessários para a programação NC dentro das tecnologias de usinagem. Estes elementos de dados descrevem a interface entre um controlador numérico computadorizado e o sistema de programação (que pode corresponder ao sistema CAM ou ao sistema de programação do chão de fábrica). O programa gerado inclui informações geométricas e tecnológicas (ISO 14649-10, 2002).

Esta parte também fornece as estruturas de controle para a sequência de execução de um programa, principalmente a sequência de *workingsteps* e as funções de máquina associadas.

Nesta parte também é definido o “MACHINING\_SCHEMA”, que contém definições de tipos de dados que são relevantes para os diferentes tipos de tecnologias (fresamento, torneamento, etc). Além disso, inclui a definição de peça de trabalho (*workpiece*), o catálogo de *features*, as mesmas que podem ser referenciadas por diferentes tecnologias, os executáveis e a base para a definição de uma operação.

O *machining\_schema* usa *features* de manufatura de maneira similar à norma ISO 10303 AP-224, e ISO 10303 AP-214. A descrição dos dados do processo é feita usando-se a linguagem EXPRESS definida na parte 11 da norma ISO 10303, e a codificação dos dados usando a parte 21 da mesma norma.

A seguir são descritas as entidades mais importantes definidas nesta parte da norma ISO 14649.

### (a) Projeto

A entidade “PROJECT” na seção “DATA” é o ponto de início na execução de um programa de uma peça. Deve haver exatamente uma entidade deste tipo (project). Esta instância deve conter um *Workplan* principal, o mesmo que contém os subconjuntos sequenciados de executáveis, e pode também incluir informações de peças a serem usinadas (ISO 14649-1, 2002, p. 6).

A Figura 2.19 apresenta esta entidade e seus atributos em um diagrama EXPRESS-G.

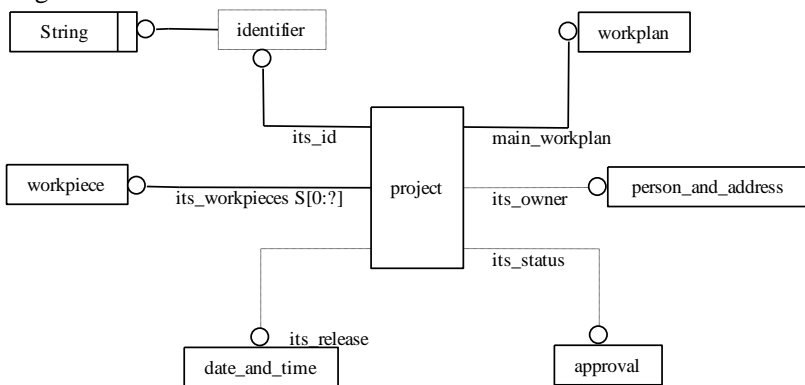


Figura 2.19. Entidade “PROJECT” e seus atributos em EXPRESS-G.

Fonte: Adaptado de ISO 14649-10 (2002).

### (b) Executáveis

É a entidade base de todos os objetos executáveis. Existem três tipos de executáveis: o *workingstep*, a estrutura de programa e a função NC, como pode-se observar na Figura 2.20. Elas iniciam ações em uma máquina e devem ser arrumadas em uma ordem definida.

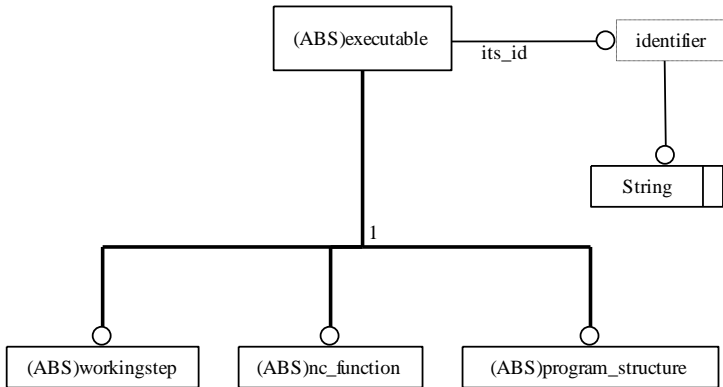


Figura 2.20. Diagrama EXPRESS-G da entidade *executable*.

Fonte: Adaptado de ISO 14649-10 (2002).

### (c) *Workingstep*

Os *workingsteps* representam o principal bloco de construção das tarefas de manufatura, e cada *workingstep* descreve uma única operação de usinagem usando uma ferramenta. Os *workingsteps* podem ser de três tipos: movimentação rápida, prova de toque, ou “*machining workingsteps*”. O *workingstep* é único em um programa, replicá-lo em um *workplan* implicará em uma ação exatamente igual na máquina. (ISO 14649-1, 2002, p.50).

A Figura 2.21 apresenta esta entidade em um diagrama.

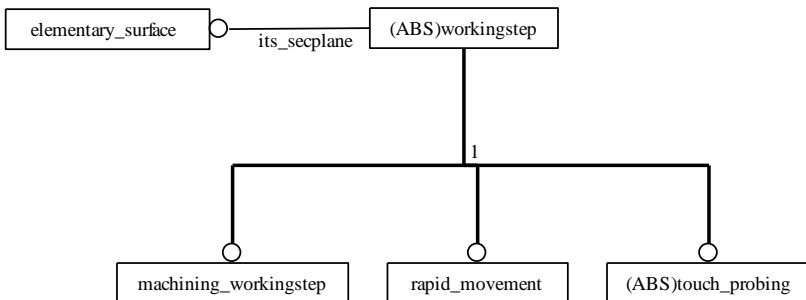


Figura 2.21. Diagrama EXPRESS-G do *workingstep*.

Fonte: Adaptado de ISO 14649-10 (2002).

#### (d) *Workplan*

Um *workplan* pode conter diversos *workingsteps* e funções NC em uma ordem linear. Também serve como um atributo de nível mais alto da entidade *project* (ISO 14649-10, 2002, p. 58).

A Figura 2.22 apresenta um diagrama desta entidade e seus atributos em EXPRESS-G.

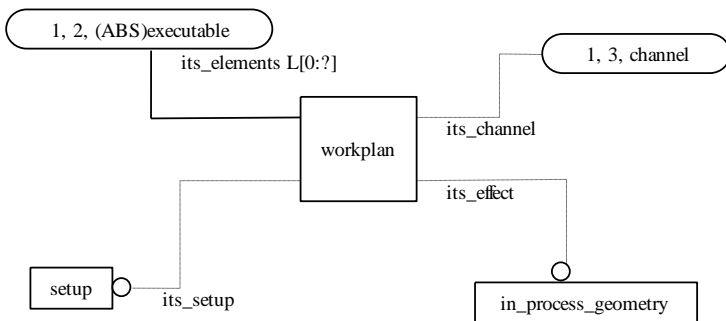


Figura 2.22. Diagrama EXPRESS-G do *workplan*.

Fonte: Adaptado de ISO 14649-10 (2002).

#### (e) *Machining workingstep*

Os *machining workingsteps* representam os processos de usinagem para uma área específica da peça de trabalho. O *machining workinstep* sempre está associado a uma *feature*, não podendo existir independente dela. Um exemplo de um

*machining workingstep* é uma operação de desbaste de uma cavidade. Elas especificam a associação entre uma *feature* e uma operação. A Figura 2.23 apresenta um diagrama desta entidade e seus atributos (ISO 14649-10, 2002, p. 51).

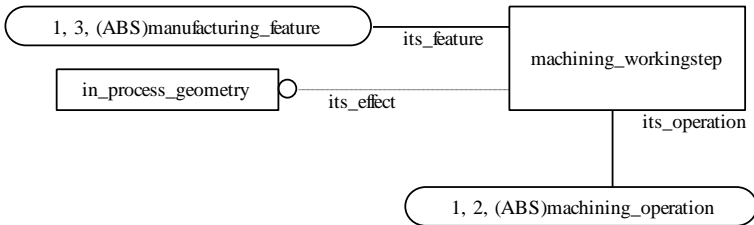


Figura 2.23. Entidade *machining workingstep* e seus atributos em um diagrama EXPRESS-G.

Fonte: Adaptado de ISO 14649-10 (2002).

#### (f) *Geometria em processo*

Esta entidade é útil para rotinas de verificação adicionais. O sistema CAM pode usar esta informação para descrever o efeito previsto de uma operação de usinagem sobre uma *feature*. A Figura 2.24 mostra esta entidade em um diagrama EXPRESS-G.

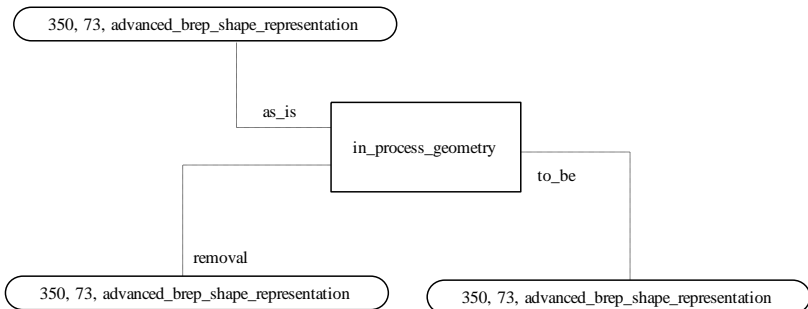


Figura 2.24. Entidade *in process geometry*.

Fonte: Adaptado de ISO 14649-10 (2002).

#### (g) *Manufacturing feature*

Esta entidade é o supertipo de todas as *features* de manufatura. Quando se trata de manufatura 2½D,

as *features* serão do subtipo *two5D\_manufacturing\_feature* e podem ser furos, cavidades, degraus, ranhuras, etc. Se for considerada a usinagem de forma livre, usam-se regiões (entidade “*region*”). (ISO 14649-10, 2002, p. 8).

A Figura 2.25 apresenta um diagrama desta entidade, seus atributos e subclasses.

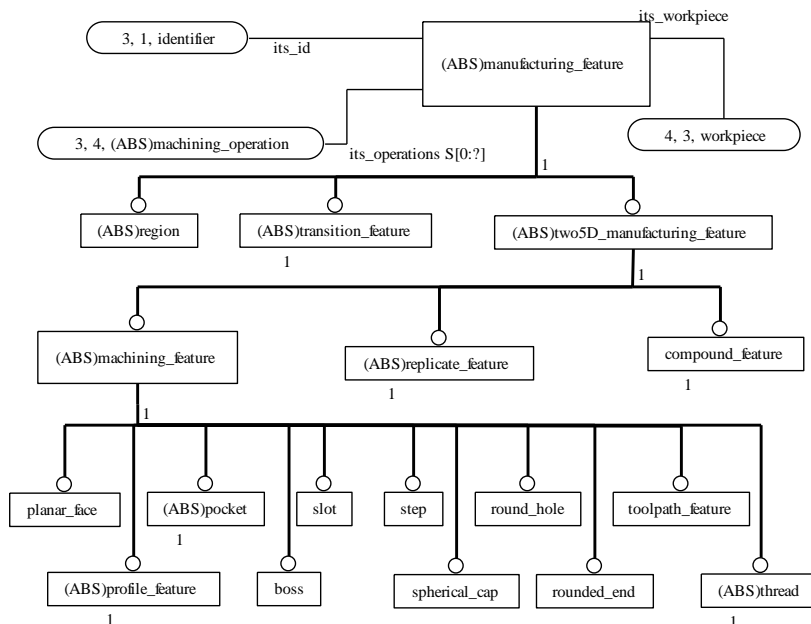


Figura 2.25. Entidade *manufacturing feature* em EXPRESS-G (Atributos e subtipos).

Fonte: ISO 14649-10 (2002).

As *manufacturing features* descrevem características como por exemplo as propriedades geométricas. Elas não contêm nenhuma instrução de como usinar uma peça, sendo que tal informação relacionada ao processo está contida unicamente nas operações (vide na Figura 2.25 que esta entidade possui um atributo “*its\_operations*”).



(h) *Operações (operation)*

As operações são usadas para especificar o conteúdo de um *workingstep*. Podem ser de três tipos: operação de usinagem (*machining\_operation*); movimentação rápida (*rapid\_movement*), e prova de toque (*touch\_probing*) como se vê na Figura 2.26. Uma operação não pode executar-se por si mesma. Para que sua interpretação não tenha ambiguidade, às vezes ela não requer a presença de uma *feature*, porém a operação mais típica (*machining\_operation*) necessita ser associada a uma *feature*, uma vez que operações se baseiam na informação geométrica fornecida pela *feature*.

Uma operação pode estar associada a nenhuma, ou a múltiplas *features*. Portanto, a mesma operação pode ser executada em diferentes lugares da peça. Segundo a norma ISO 14649-10 (2002), isto foi pensado com a finalidade de reutilizar os dados de uma operação em diferentes *features*. Por exemplo, na usinagem de  $n$  furos iguais em diferentes posições da peça, eles não podem compartilhar os mesmos *workingsteps*, porém as operações podem ser compartilhadas, reduzindo-se assim o volume de dados a serem trocados (ISO 14649-10, 2002).

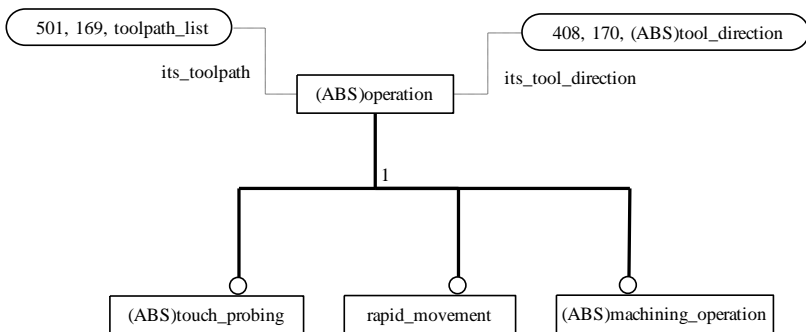


Figura 2.26. *Operation*: Atributos e subtipos em EXPRESS-G.

Fonte: ISO 14649-10 (2002).

Todas as operações têm como opção a especificação das trajetórias da ferramenta de maneira explícita. Para operações de usinagem, os controladores numéricos deveriam especificar os ciclos para gerar as suas próprias trajetórias. Entretanto, para controladores antigos, os quais não são capazes de gerar as suas próprias trajetórias, ou

por razões tecnológicas que requerem a execução de trajetórias exatamente determinadas, pode se usar o atributo *its\_toolpath*.

(i) *Operação* de usinagem (*machining\_operation*)

Esta entidade define o processo de usinagem para uma determinada área da peça de trabalho. Ela especifica a ferramenta a ser utilizada e um conjunto de parâmetros tecnológicos (ISO 14649-10, 2002, p. 68).

A Figura 2.27 apresenta esta entidade e seus atributos.

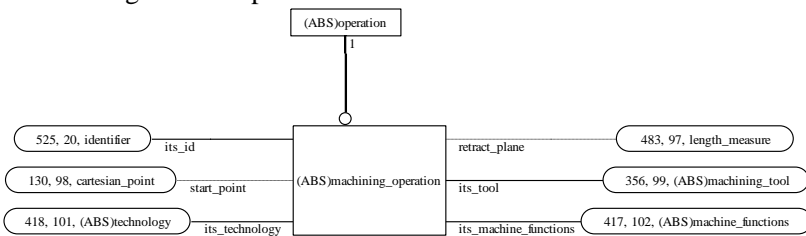


Figura 2.27. Diagrama EXPRESS-G da entidade *machining\_operation*.

Fonte: ISO 14649-10 (2002).

As operações de usinagem específicas para processos de fresamento são definidas na parte 11 da norma ISO 14649.

### 2.2.2.4.3 ISO 14649 Parte 11: Dados do processo para fresamento

Esta parte especifica os elementos de dados específicos para cada tecnologia necessários para os processos de fresamento. Junto com a parte 10, descreve a interface entre um controlador NC e seu sistema de programação (sistema CAM ou sistema de programação do chão de fábrica) para fresamento. Esta parte não inclui qualquer outra tecnologia como torneamento, EDM (eletro-erosão), etc., abrangendo apenas processos de fresamento e furação.

Esta parte também define o “MILLING\_SCHEMA”, o mesmo que define tipos de dados que representam os processos de fresamento e furação.

A seguir se apresentará algumas das mais importantes entidades definidas nesta parte.

(a) Operação de fresamento e furação (*milling machining operation*)

Esta é a classe base para todas as operações descritas na parte 11, sendo subtipo da entidade *machining\_operation* definida na parte 10, e se divide em: operações de fresamento e operações de furação. A Figura 2.28 apresenta um diagrama EXPRESS-G desta entidade.

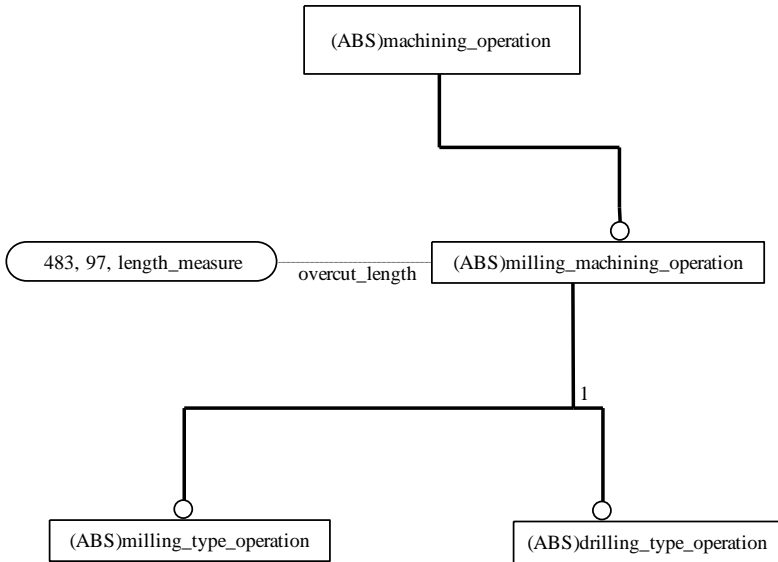


Figura 2.28. Diagrama EXPRESS-G da entidade *milling machining operation*.  
Fonte: Adaptado de ISO 14649-11 (2002).

(b) Operação do tipo fresamento (*milling type operation*)

Esta é a classe base para todas as operações de fresamento, e inclui todos os atributos necessários para descrever a tecnologia e a estratégia de tais processos. Existem dois tipos de operações de usinagem: desbaste e acabamento. As operações de desbaste removem todo o material da peça deixando sobrematerial (*allowance*) na base ou na lateral dependendo do tipo de *feature*. A operação de acabamento remove este sobrematerial (*allowance*) até a superfície final da *feature* em uma ou várias passadas (ISO 14649 – 11, 2002). A Figura 2.29 apresenta esta entidade com seus atributos e subtipos (operações de forma livre e operações em duas dimensões e meia).

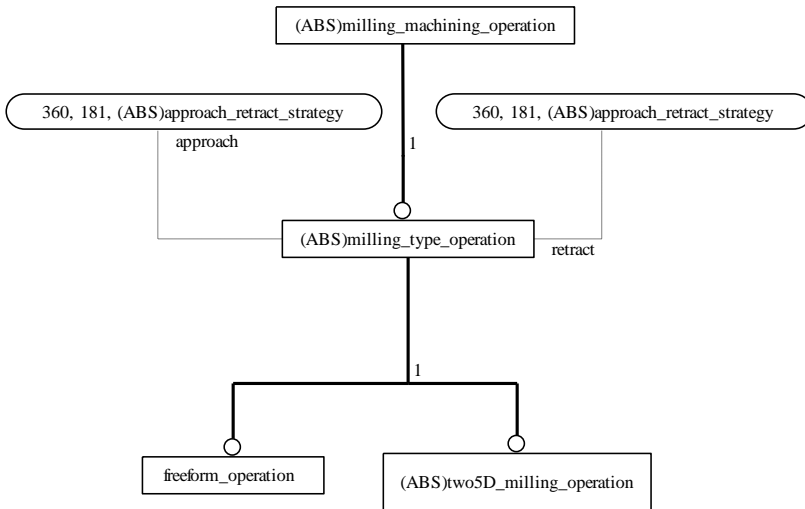


Figura 2.29. *Milling type operation* em EXPRESS-G.

Fonte: Adaptado de ISO 14649 – 11 (2002).

(c) *Operação do tipo furação (drilling type operation)*

Esta é a classe base para todas as operações relacionadas à furação, alargamento, etc. de furos. Ela fornece todos os atributos necessários para descrever a tecnologia e a estratégia para as operações deste tipo. Em caso de pré-furos ou furos feitos por fundição que requerem usinagem, pode-se especificar a profundidade de furação e a ferramenta adequada, além de poder-se especificar o diâmetro prévio (atributo *previous\_diameter*). A Figura 2.30 apresenta um diagrama desta entidade, onde também pode ser visto que ela possui cinco subtipos.

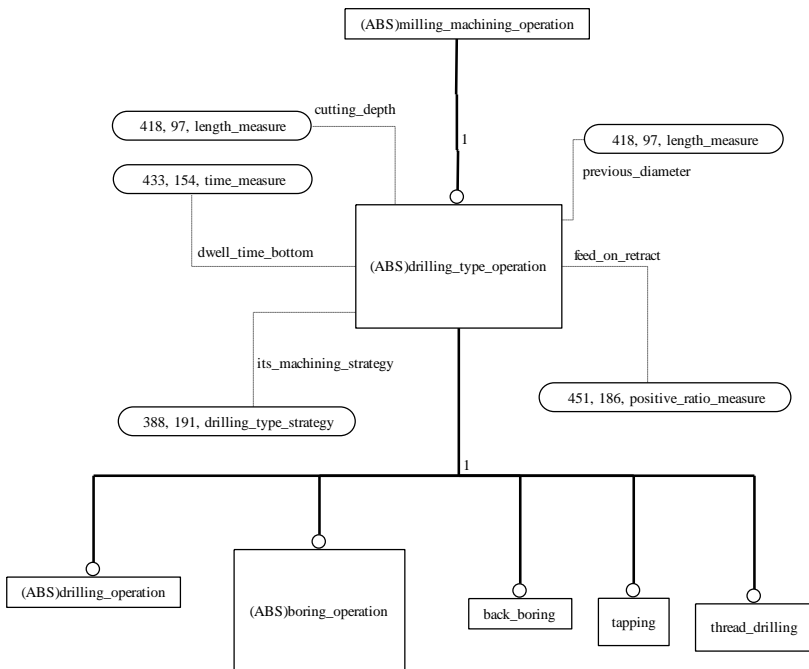


Figura 2.30. Operações do tipo furação em um diagrama EXPRESS-G.  
 Fonte: Adaptado de ISO 14649 – 11 (2002).

(d) *Tecnologia em fresamento (milling technology)*

Esta entidade define os parâmetros tecnológicos das operações de fresamento e de furação (velocidade de corte, velocidade de rotação, avanço, etc.). A Figura 2.31 apresenta esta entidade e seus atributos em um diagrama EXPRESS-G.

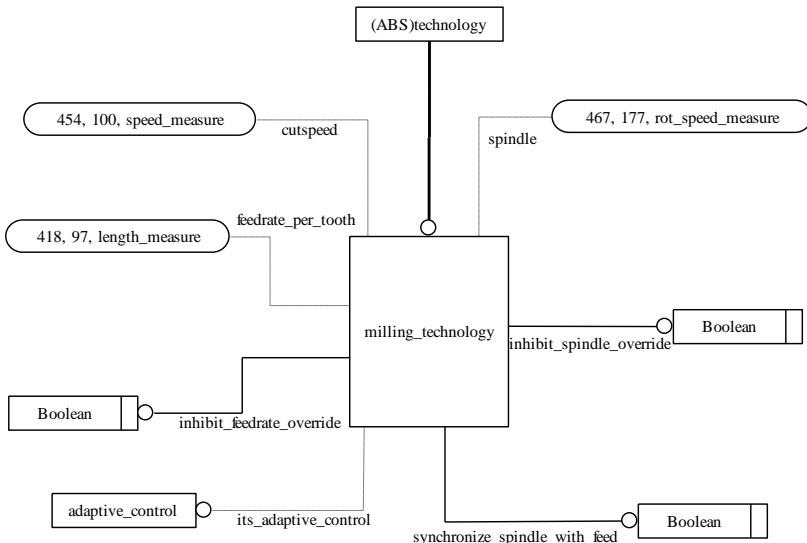


Figura 2.31. Diagrama EXPRESS-G da entidade *milling tecnologia*.  
 Fonte: Adaptado de ISO 14649 – 11 (2002).

### (e) Funções de máquina de fresamento

Descreve o estado de várias funções da máquina (refrigeração, remoção de cavacos, etc.) a serem aplicadas durante o tempo de uma operação. Esta entidade é subtipo da entidade *technology* definida na parte 10 da norma ISO 14649 (ISO 14649-11, 2002). A Figura 2.32 apresenta um diagrama desta entidade e seus atributos.

### (f) Trajetórias

As entidades do tipo *toolpath* descrevem movimentos da ferramenta ou dos eixos da máquina. As entidades deste tipo representam a informação de mais baixo nível da norma ISO 14649 (ISO 14649-10, 2002, p. 70).

A Figura 2.33 apresenta um diagrama EXPRESS-G desta entidade.

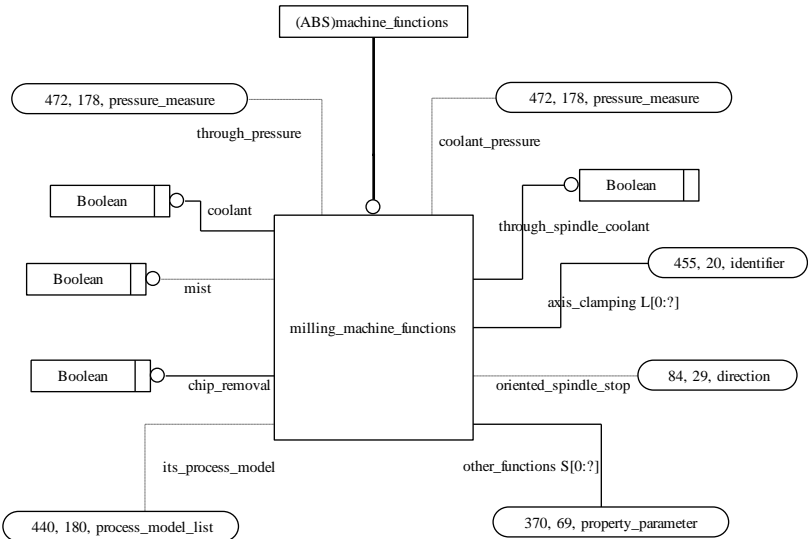


Figura 2.32. Entidade *milling machining function* em EXPRESS-G.  
Fonte: Adaptado de ISO 14649 – 11 (2002).

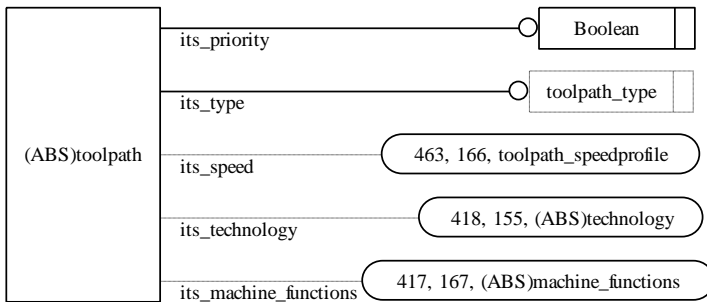


Figura 2.33. Entidade *toolpath* e seus atributos em EXPRESS-G.  
Fonte: Adaptado de ISO 14649 – 10 (2002).

#### 2.2.2.4.4 ISO 14649 parte 111: Ferramentas para fresamento

A parte 111 da norma ISO 14649 especifica elementos de dados necessários para representar ferramentas para fresamento. Esta parte também trabalha em conjunto com a parte 11 da mesma norma descrita na seção anterior. Esses elementos de dados podem ser usados como critério para selecionar uma ou várias ferramentas para fresamento ou para furação.

Nesta parte define-se o “MILLING\_TOOL\_SCHEMA”, que fornece um esquema básico de ferramentas incluindo apenas as informações mais importantes. Este esquema foi criado para dar ao controlador a informação suficiente para selecionar a ferramenta especificada no programa NC (ISO 14649 – 111, 2002). Como apontado em seções anteriores, na norma ISO 6983 a ferramenta é definida apenas com um número (por exemplo, T1), e nenhuma informação adicional relacionada ao tipo ou geometria da ferramenta é dada.

Os dados das ferramentas que estão inclusos no `milling_tool_schema` são: tipo, geometria e a vida esperada da ferramenta. A seguir são apresentadas as principais entidades definidas neste esquema.

#### (a) Ferramenta de corte para fresamento

Esta entidade descreve a informação necessária para descrever ferramentas de corte para operações de fresamento, incluindo tipos de ferramentas como alargadores, brocas, fresas, etc. Esta entidade é subtipo da entidade genérica `cutting_tool` definida na parte 10 da norma ISO 14649 (ISO 14649 – 111, 2002). A Figura 2.34 apresenta um diagrama desta entidade e seus atributos.

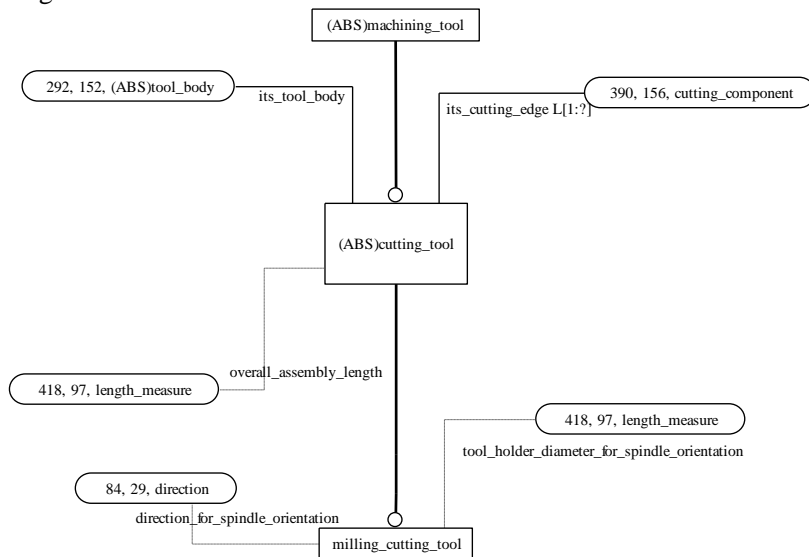


Figura 2.34. Diagrama EXPRESS-G da entidade *milling cutting tool*.

Fonte: Adaptado de ISO 14649 – 111 (2002).



*Milling cutting tool* herda os atributos da sua classe pai (*cutting tool*).

(b) Corpo da ferramenta de fresamento (*milling tool body*)

Esta é a classe base para todos os tipos de corpos de ferramenta usados em processos de fresamento. É subtipo da entidade *tool body* definida na parte 10. A Figura 2.35 mostra um diagrama desta entidade com seus atributos, supertipo (*tool body*), e seus subtipos.

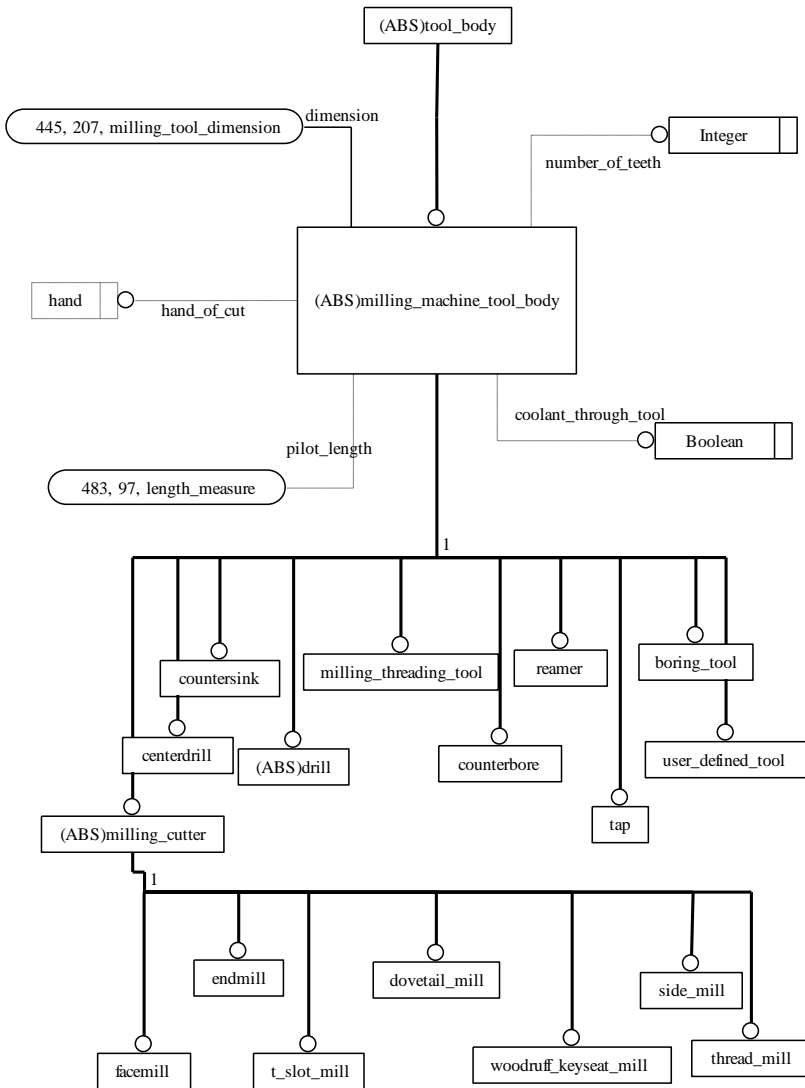


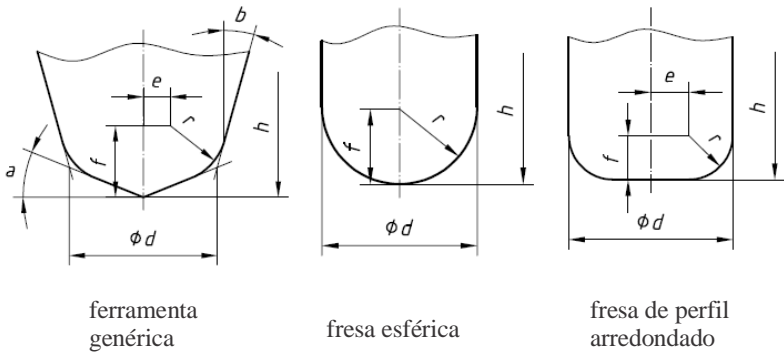
Figura 2.35. Diagrama EXPRESS-G da entidade *milling tool body*.

Fonte: Adaptado de ISO 14649 – 111, (2002).

### (c) Dimensões da ferramenta de fresamento

Esta entidade descreve as dimensões do corpo da ferramenta. A Figura 2.36 descreve uma ferramenta genérica, uma fresa esférica e uma

de perfil arredondado, e a Figura 2.37 apresenta um diagrama em EXPRESS-G desta entidade.



Onde:

$d$ : diâmetro

$r$ : raio de canto

$e$ : centro da borda horizontal

$f$ : centro da borda vertical

$a$ : ângulo da ponta da ferramenta

$b$ : ângulo de circunferência da ferramenta

$h$ : comprimento da aresta de corte

Figura 2.36. Dimensões de diferentes tipos de ferramentas.

Fonte: ISO 14649 – 111 (2002).

Todas estas dimensões aparecem como atributos desta entidade, como se pode ver na Figura 2.37.

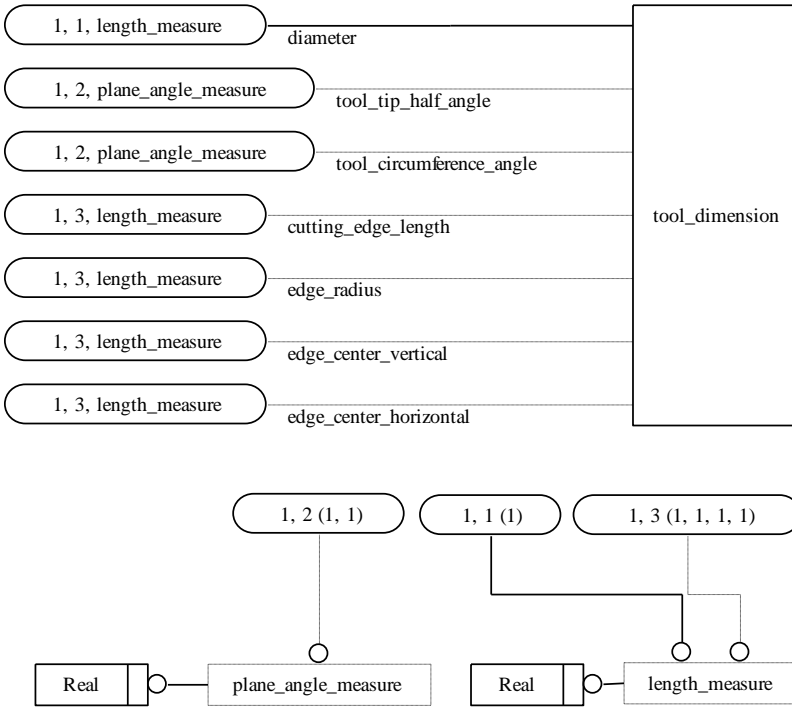


Figura 2.37. Entidade *Tool dimension* e seus atributos em EXPRESS-G.  
 Fonte: Adaptado de ISO 14649 – 111 (2002).

### 3 PROJETO ASSISTIDO POR COMPUTADOR

Este capítulo tem como objetivo apresentar conceitos e definições importantes relacionados ao projeto assistido por computador (CAD), assim como também as suas aplicações.

#### 3.1 ANTECEDENTES

Na década de 1950 iniciaram as primeiras aplicações de computadores auxiliando as tarefas de engenharia. Os sistemas daquela geração eram limitados, os quais representavam e manipulavam apenas entidades simples em duas dimensões usando terminais monocromáticos (SOUZA e ULBRICH, 2009). As vantagens sobre os desenhos em papel eram notórias:

- Possibilidade de enviar ou receber os desenhos por processos eletrônicos e comunicação de dados.
- Melhor gerenciamento dos desenhos.
- Precisão do dimensionamento.
- Rapidez na recuperação, modificação, atualização e armazenagem dos desenhos.

A principal desvantagem de tais sistemas era o seu elevado custo, o qual limitava seu uso. Entretanto, o crescente desenvolvimento de sistemas CAD os deixou economicamente mais acessíveis, e o emprego da prancheta para desenhos em papel reduziu drasticamente.

Até o início da década de 1990 se desenvolveram sistemas CAD de pequeno porte (*low-end*) dando suporte apenas ao desenho 2D, enquanto sistemas mais robustos (*high-end*) com CAD em 3D, CAM e outros CAx eram usados apenas quando modelos tridimensionais de produtos eram necessários, pois demandavam o uso de computadores de altíssimo desempenho (*workstations*). As interfaces eram pouco amigáveis, baseadas em comandos introduzidos através do teclado.

Após anos de contínuo desenvolvimento tecnológico, os sistemas CAD passaram a oferecer uma ampla gama de opções, desde simples estações de trabalho de baixo custo, a sofisticados sistemas CAD exigindo computadores potentes capazes de representar realisticamente qualquer forma geométrica por mais complexa que esta seja.

Na atualidade existe uma ampla variedade de soluções CAD para diversas aplicações como:

- Engenharia mecânica: projeto de configuração de peças mecânicas, projeto de peças, modelagem de moldes etc.
- Engenharia elétrica: para desenho de circuitos, leiaute de placas eletrônicas.
- Arquitetura: projetos arquitetônicos de qualquer tipo.
- Engenharia civil: edificações, projeto de tubulações, projetos elétricos, etc.

### 3.2 CLASSIFICAÇÃO DOS SISTEMAS CAD

SOUZA e ULBRICH (2009) afirmam que não existe nenhum padrão para classificar os sistemas CAD, porém eles classificam os sistemas CAD em relação à sua possibilidade de aplicação.

- Sistemas CAD de pequeno porte (low-end): são os sistemas CAD usados para representar objetos e formas geométricas em 2D. Usam entidades geométricas simples como retas, círculos e curvas genéricas. Estas entidades são de baixa ordem devido à sua simplicidade matemática de representação.
- Sistemas CAD de médio porte (middle-end): têm como característica principal a capacidade de representação geométrica em 3D de forma realística. Este tipo de sistema CAD pode gerar objetos com informações acerca da superfície, propriedades mecânicas, centro de gravidade, volume, etc.
- Sistemas CAD de grande porte (high-end): são sistemas compostos por vários módulos dentro do mesmo software, podendo abranger CAD, CAM, CAI, CAE, etc. Cada módulo pode ter sub-módulos, por exemplo, o módulo CAM pode ter um sub-módulo para programação CNC para torneamento e outro para fresamento de três eixos.

### 3.3 MODELAGEM GEOMÉTRICA

Segundo ZEID (1991), os modelos geométricos de um objeto e suas bases de dados associadas podem ser de três tipos: modelos de arame (*wireframe*), de superfícies e modelos sólidos.

#### 3.3.1 Modelos de arame (*wireframe*)

Este é o modelo geométrico mais simples que pode ser usado para representar matematicamente no computador. A palavra *wireframe* está associada ao fato de que pode-se imaginar um arame que é dobrado para resultar nas arestas do objeto que gerará o modelo (ZEID, 1991).

Tipicamente um modelo de arame consiste inteiramente em pontos, linhas, arcos, círculos, e curvas. Os primeiros sistemas (desenvolvidos nos anos 1960) usavam esta técnica, e eram estritamente em duas dimensões, e foram projetados para o desenho. Estes modelos continham informações suficientes para realizar a tarefa de programação NC. Os usuários deviam construir as vistas desejadas das geometrias independentemente devido à lacuna de centralização e de associatividade das bases de dados destes sistemas CAD. Após alguns anos se superou este problema e conseguiu-se efetuar a modelagem tridimensional de objetos usando modelos de arame. A Figura 3.1 apresenta uma peça modelada usando esta técnica.

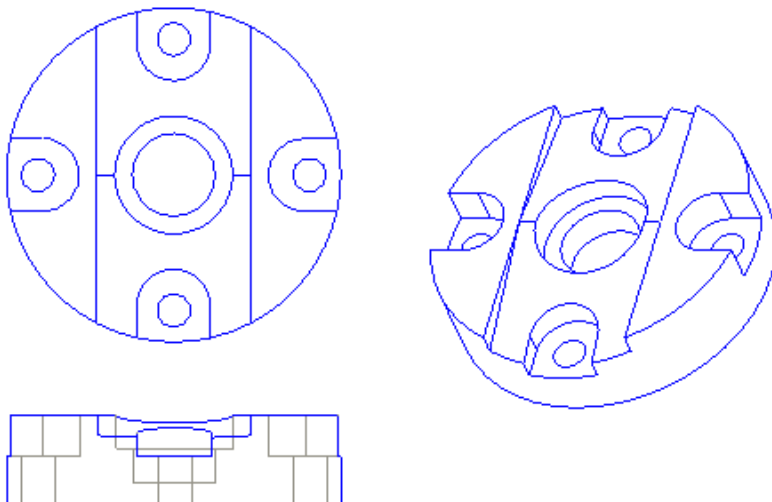


Figura 3.1. Peça modelada em *wireframe*.

Fonte: Criado pelo autor.

A maior vantagem deste tipo de modelo é a sua simplicidade de construção, não requerendo um tempo elevado para o processamento nem tanta memória do computador quando comparado à modelagem por superfícies ou por sólidos. As desvantagens que esta técnica de modelagem tem são: estes modelos são geralmente representações ambíguas de objetos reais, e deve-se confiar fortemente na interpretação humana.

A criação dos objetos usando esta técnica é feita usando entidades analíticas ou sintéticas. Entidades analíticas são: pontos, linhas, arcos, círculos, elipses, parábolas, hipérbolas. As entidades sintéticas incluem os vários tipos de *splines* (ZEID, 1991).

### 3.3.2 Modelos de superfícies

Um modelo de superfície de um objeto é uma representação mais completa e menos ambígua que um modelo de arame. Tipicamente um modelo de superfície consiste de entidades *wireframe* que formam a base para criar entidades de superfície. A descrição da superfície é



usualmente abordada como uma extensão da representação *wireframe*. A Figura 3.2 apresenta uma lanterna modelada usando superfícies.

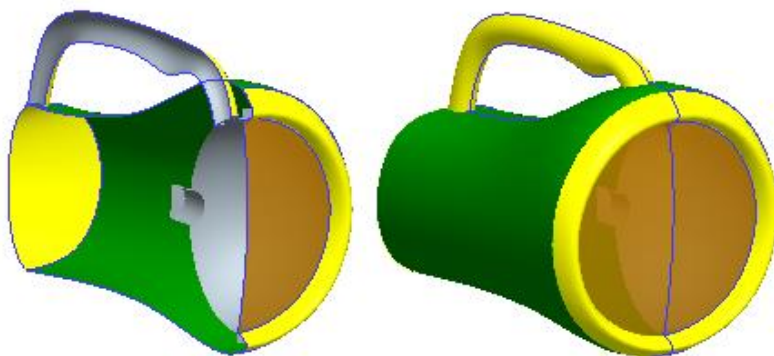


Figura 3.2. Modelagem usando superfícies.

Fonte: Criado pelo autor.

A modelagem por superfícies se desenvolveu rapidamente devido às limitações e inconvenientes apresentados pela modelagem em *wireframe*. Em geral pode-se extrair o modelo *wireframe* a partir do modelo de superfícies. As bases de dados dos modelos de superfície são centralizadas e associativas, portanto manipular entidades de superfície em uma vista possibilita a geração automática das outras vistas. A criação de objetos usando esta técnica é feita usando-se entidades de superfície analíticas e sintéticas. As entidades analíticas são: superfícies planas, superfícies de revolução, etc. As entidades sintéticas são: superfícies *splines* de Hermite, superfícies B-spline, superfícies de Gordon, etc. (ZEID, 1991).

### 3.3.3 Modelos sólidos

Um modelo sólido de um objeto é uma representação mais completa que um modelo de superfície. Ele contém informação topológica que potencialmente permite a automação e a integração funcional (ZEID, 1991).

Segundo FAGALI e ULBRICH (2009), existem várias técnicas computacionais para a modelagem de sólidos, como por exemplo, *octree*, *quadtree*, decomposição celular, porém as mais usadas são a

geometria construtiva de sólidos (CSG) e a representação de fronteira (B-rep), e estas últimas são detalhadas a seguir.

### 3.3.3.1 Geometria construtiva de sólidos (CSG)

Este método foi proposto por Voelker e Requicha da universidade de Rochester em 1972 (FAGALI E ULBRICH, 2009). Objetos sólidos são construídos com a combinação de sólidos primitivos simples como cubos, cilindros, cones, esferas, usando operações booleanas de adição, subtração e interseção entre tais objetos sólidos. A Figura 3.3 mostra duas primitivas e as operações booleanas possíveis entre elas.

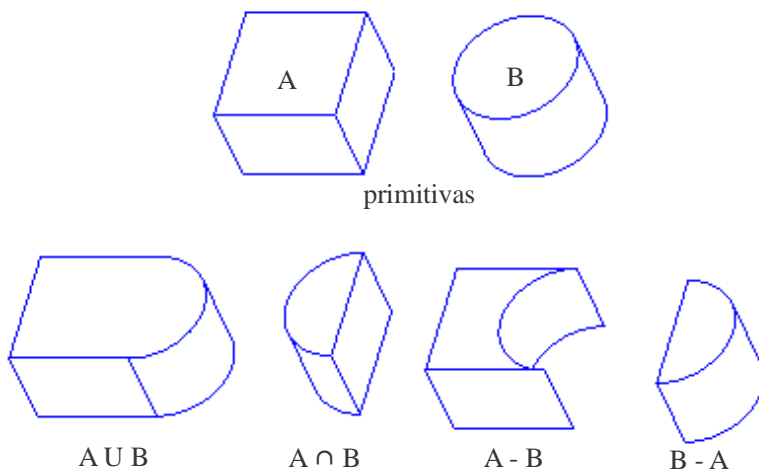


Figura 3.3. Operações booleanas entre dois sólidos.

Fonte: ZEID (1991).

### 3.3.3.2 Representação de fronteira (B-rep<sup>1</sup>)

Este método é um dos mais amplamente usados para criar modelos de objetos físicos. Um modelo B-rep ou modelo de fronteira é baseado na noção topológica de que um objeto físico está limitado por

<sup>1</sup> B-rep: Boundary representation, Este método foi proposto por Ian Braid da universidade de Cambridge.

um conjunto de faces. Estas faces são regiões ou subconjuntos de superfícies fechadas e orientáveis. Uma superfície fechada é contínua, sem lacunas, enquanto uma superfície orientável é uma superfície na qual é possível distinguir dois lados por meio do uso da direção da superfície normal (ZEID, 1991).

Cada face está limitada por arestas e cada aresta é limitada por vértices. Portanto, topologicamente, um modelo B-rep de um objeto é composto por faces, arestas e vértices relacionados de tal maneira que assegurem a consistência topológica do modelo.

Os sólidos criados por este método devem ser validados por uma série de regras matemáticas, e as principais regras para um sólido convexo sem furos são:

- Cada face do sólido deve constituir um único contorno (*loop*) de arestas.
- Cada aresta deve possuir dois vértices nas suas extremidades.
- Cada aresta deve unir exatamente duas faces.
- Pelo menos três arestas devem se encontrar em um vértice.

Existe uma regra que relaciona o número de arestas com o número de vértices e faces. Esta regra foi proposta pelo matemático suíço Leonhard Euler (3.1).

$$F - E + V - L = 2(B - G) \quad (3.1)$$

onde:

*F*: número de faces.

*E*: número de arestas.

*V*: número de vértices.

*L*: número de loops internos (furos)

*B*: número de sólidos

*G*: número de anéis.

### 3.3.4 Projeto por *features*

SHAH e MÄNTYLÄ (1995) definem *features* como sendo formas genéricas de um produto às quais os engenheiros podem associar certos atributos e certo conhecimento útil para raciocinar acerca do funcionamento do produto. As *features* encapsulam aspectos da

engenharia, e são aplicáveis na definição do produto e no planejamento do processo para a sua fabricação. Como exemplos de *features* tem-se: furo, degrau, chanfro, cavidade (ver Figura 3.4).

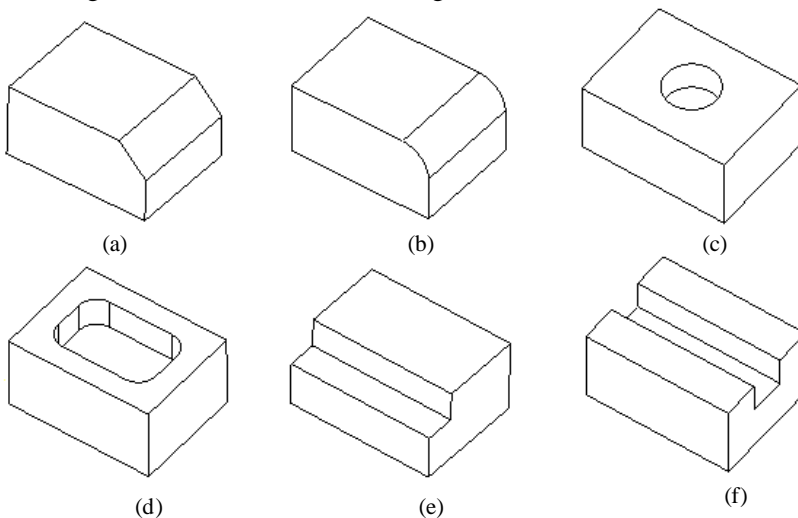


Figura 3.4. Exemplos de *features*. (a) chanfro, (b) raio de concordância, (c) furo, (d) cavidade retangular, (e) degrau, e (f) ranhura.

Fonte: Criado pelo autor.

Através de *features* pode-se compor uma estrutura de dados para representar uma peça ou uma montagem em termos de seus constituintes, onde cada *feature* tem uma identidade, e a forma pode ser expressa em termos de:

- Parâmetros dimensionais;
- Enumeração geométrica;
- Relações e entidades topológicas;
- Etapas de construção para produzir a geometria da *feature*.

Em se tratando de projeto por *features*, SHAH E MÄNTYLÄ (1995) apontam duas metodologias de projeto que são comumente utilizadas:

- *Destruction by Machining Features*, também conhecida como *Destructive Solid Geometry* ou *Deforming Solid Geometry (DSG)*, onde a modelagem se inicia com um modelo da matéria-prima (peça bruta). O modelo da peça acabada é criado subtraindo-se da peça bruta *features* que correspondem ao

material removido por operações de usinagem. A vantagem deste método é que as *features* de usinagem estão diretamente disponíveis no modelo da peça, não sendo necessário o reconhecimento ou mapeamento de *features* de manufatura.

- Síntese por *Features* de Projeto, onde o modelo pode ser construído tanto pela adição quanto pela subtração de *features*, não sendo necessário iniciar com um modelo de peça bruta.

A Figura 3.5 apresenta um diagrama que mostra o funcionamento da técnica de projeto por *features*, em que o usuário interage com o modelador através do uso da biblioteca de *features* disponibilizada pelo sistema. Logo, o modelo em *features* é usado para criar o modelo geométrico da peça.

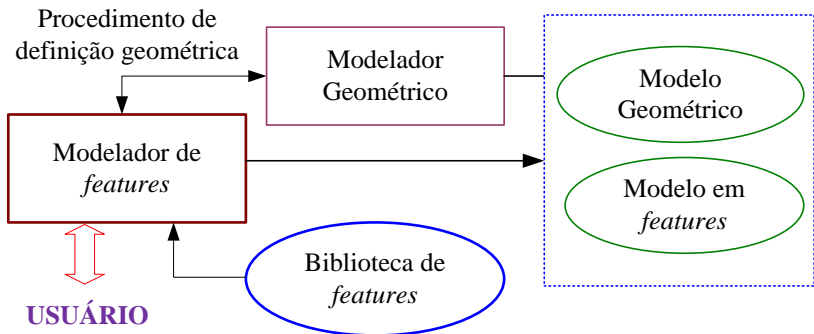


Figura 3.5. Criação de *features* utilizando projeto por *features*.  
Fonte: Adaptado de SHAH e MÄNTYLÄ (1995).

## 4 TECNOLOGIA DA INFORMAÇÃO E MANUFATURA

Este capítulo apresenta conceitos importantes relacionados à tecnologia da informação (TI) e tecnologias relacionadas, e como elas podem ser utilizadas no contexto da manufatura.

### 4.1 MANUFATURA ELETRÔNICA

MOLINA e RAMÍREZ (2006) definem a manufatura eletrônica como o conjunto de tecnologias da informação que permite às empresas realizar a manufatura sob demanda através da integração das aplicações de negócios eletrônicos (*e-business*).

KOC *et. al.*, (2002) e LEE (2003) definem *e-manufacturing* como uma metodologia que possibilita a integração das operações de manufatura com os objetivos funcionais da empresa através do uso da internet.

Na manufatura, o gerenciamento, a manipulação e o uso eficiente das informações são essenciais para seu crescimento e vitalidade. Uma batalha competitiva está sendo travada nos bastidores das empresas que procuram aproveitar a web para usos que vão além da simples publicação de informações. A web está sendo usada para integrar dados, informações e conhecimentos entre todos os atores da cadeia de suprimentos (fornecedores, a própria empresa e os clientes). Esta integração das informações a diferentes níveis da fábrica (usando a tecnologia da web) é essencial para oferecer novos serviços de informação aos fornecedores e clientes para alcançar a manufatura sob demanda (*e-manufacturing*). A manufatura eletrônica contribui para vincular o desenvolvimento do produto e a cadeia de suprimentos, a qual contém inúmeras informações importantes sobre o ciclo de vida do produto e as capacidades dos fornecedores (TIWARI *et. al.*, 2010).

### 4.2 TELEMANUFATURA

MALEK *et. al.*, (1998) definem telemanufatura como a atividade onde uma empresa usa serviços que são oferecidos através das redes de

comunicação para realizar, em tempo real, as operações e processos necessários para o projeto e a produção de itens.

### 4.3 MANUFATURA VIRTUAL

OFFODILE e MALEK, (2002) definem a manufatura virtual como o uso da tecnologia da informação e da simulação computacional para modelar os processos de manufatura do mundo real com a finalidade de analisá-los e entendê-los. Diferente da simulação de eventos discretos clássica, modelos de animação tridimensionais complexos são usados para projetar o ambiente de manufatura real. Máquinas, células de manufatura, peças e equipamentos podem ser projetados e avaliados na tela do computador antes que os verdadeiros equipamentos ou produtos sejam construídos, economizando-se tempo, esforço, materiais e custos.

### 4.4 EVOLUÇÃO DAS TECNOLOGIAS BASEADAS EM REDE

Em 1980, alguns pesquisadores se focaram no uso da rede para compartilhar ideias e distribuir software, modelos e até mesmo imagens. Esta nova habilidade de comunicar instantaneamente quebraria as barreiras que rodeavam os acadêmicos e outros pesquisadores. Em 1982 o termo “internet” foi usado pela primeira vez, e o protocolo TCP/IP tornou-se a linguagem universal da internet.

Desde o início dos anos 1980 até os anos 1990, a internet experimentou um crescimento explosivo. Em 1987 o número de usuários era de dez mil, em 1989 este número excedia os cem mil, e em 1992 excedia um milhão de usuários da internet (TORNINCASA e CHIRONE, 2002). A Figura 4.1 apresenta a evolução da Internet, e particularmente a parte gráfica no decorrer dos anos.

Os *hiperlinks* eram realizados através da incorporação dos métodos de acesso nas páginas web usando a linguagem HTML (*Hypertext Markup Language*). O HTML permitiu aos usuários fazer conexões entre *sites* da internet, e tudo o que o usuário devia fazer era clicar no link para ser transferido através da rede para página (*site*) desejada. Os navegadores de internet lêem o código fonte HTML e convertem-no em texto legível ao usuário. Os primeiros navegadores

eram baseados em texto, e não havia gráficos. Logo depois apareceram os navegadores que podiam suportar gráficos, como o Mosaic lançado em 1993.

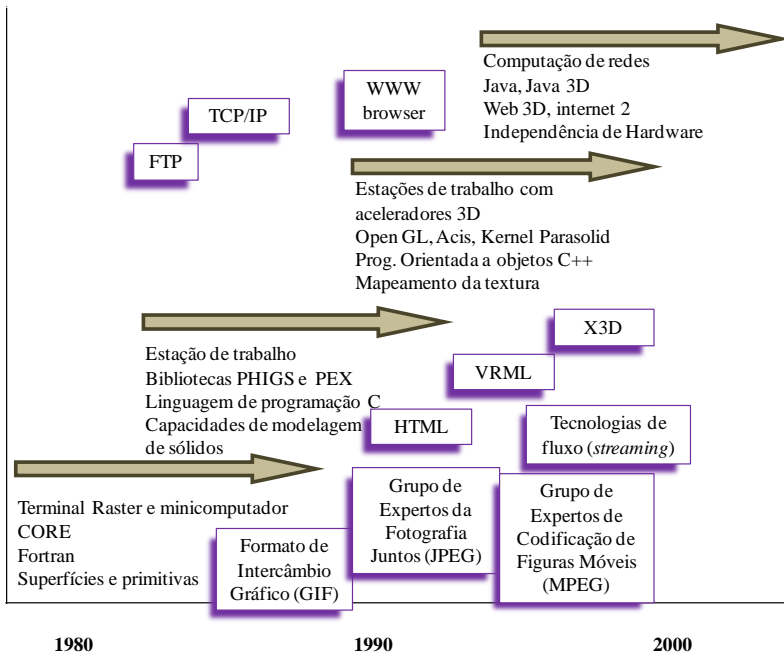


Figura 4.1. Evolução da internet até a utilização de gráficos 3D  
Fonte: Adaptado de TORNINCASA e CHIRONE, (2002).

#### 4.4.1 Java

Uma das grandes revoluções no mundo do software aconteceu quando a linguagem Java foi anunciada em 1995 pela empresa Sun Microsystems, a qual poderia ser utilizada em sites produzidos na *World Wide Web*. O que chamava a atenção era o fato de que ela podia ser portátil para outros sistemas operacionais. O tamanho também foi um fator importante, pois depois de instalada a máquina virtual no computador, os aplicativos em Java (*Applets*) eram pequenos, e mesmo com a reduzida velocidade de transmissão naquela época, os aplicativos eram visualizados com certa rapidez. Com o tempo a linguagem Java



evoluiu e se transformou em uma linguagem robusta e complexa (GONÇALVES, 2006).

Nos dias de hoje, a linguagem Java é utilizada por grandes bancos, pois fornece extrema segurança, além de grandes empresas que se responsabilizam pelo tráfego de uma grande quantidade de dados entre outras empresas, uma vez que Java proporciona estabilidade e portabilidade. Segundo (HORSTMANN e CORNELL, 2001), algumas aplicações do Java podem ser encontradas em: servidores *web*, bancos de dados, computadores de grande porte, telefones móveis e sistemas de cartão de crédito. A linguagem Java pode ser executada em várias plataformas: Microsoft Windows, Linux, Sun Solaris e MacOs.

Com os aplicativos (*applets*) escritos em Java, é possível o desenvolvimento de páginas *web* incluindo animações, gráficos, cálculos, aplicações distribuídas e comunicação. No entanto, o maior destaque desta linguagem é a possibilidade de criar páginas *web* altamente interativas. Segundo DECEMBER (1996), a linguagem Java transforma a *web* em um sistema de distribuição de software, onde o usuário tem “coisas para fazer” e não apenas “lugares para onde ir”. Java tem mudado o comportamento dos usuários da *web*, os quais têm deixado de apenas “surfear” para também jogar, interagir e aprender nos novos ambientes interativos. Esta linguagem também tem permitido melhorar o suporte a cursos a distância via internet através do uso de simuladores, os quais auxiliam no aumento do envolvimento dos alunos nas sessões de aprendizagem remota, ajudando-os a aprender através da experimentação e visualização (ÁLVARES e FERREIRA, 2003).

#### 4.4.2 Java 3D

A API (Interface de Programação da Aplicação) Java 3D é uma interface para escrever programas, mostrar e interagir com gráficos tridimensionais (BOUVIER, 2000). A API fornece uma biblioteca de alto nível para criar e manipular geometrias 3D e estruturas para renderizá-las. Java 3D fornece funções para a criação, visualização e animação de aplicações gráficas 3D. O programador trabalha com construções de alto nível para criar e manipular objetos geométricos 3D. Estes objetos geométricos residem em um universo virtual.

Um programa em Java 3D cria instâncias de objetos Java 3D e os coloca na estrutura de dados do grafo de cena. Um grafo de cena é um arranjo de objetos 3D em uma estrutura de árvore que especifica

completamente o conteúdo do universo virtual, e como este deve ser renderizado. Os programas em Java 3D podem ser executados como aplicativos independentes ou em navegadores (como *applets*).

#### 4.4.3 A Linguagem de Modelagem de Realidade Virtual

VRML é uma linguagem de descrição de simulações interativas com vários participantes comunicando-se através da Internet (ÁLVARES e FERREIRA, 2003).

A primeira versão (VRML 1.0) foi concebida em 1994, e foi projetada seguindo três requisitos principais: independência de plataforma, extensibilidade, e rapidez mesmo em conexões com pequena largura de banda (BELL et al, 2011). As principais características da versão 2.0 são:

- **Interatividade:** em um mundo interativo, espera-se, por exemplo, poder abrir portas, andar entre os móveis em um apartamento virtual, trocar objetos de lugar entre outras coisas. No entanto, este recurso resulta no aumento significativo do tamanho dos arquivos;
- **Comportamentos:** na versão 1.1 a capacidade de animação era ainda muito limitada, enquanto na versão 2.0 podem ser criados comportamentos, tal que os objetos possam ter um pouco de “vida”, através de alguns movimentos, e também ações que afetem outros objetos.
- **Som:** o som é uma característica chave na criação de um bom ambiente imersivo. Considera-se que uma pequena melhora na qualidade do som é percebida mais significativamente pelo usuário do que uma pequena melhora na qualidade da imagem.

#### 4.4.4 3D Extensível (X3D)

É um padrão de software que define o conteúdo 3D baseado em *web* integrado com multimedia. X3D é destinado ao uso de uma variedade de dispositivos de hardware em uma ampla gama de áreas tais como engenharia, visualizações científicas, apresentações multimedia,

aplicações educacionais e de entretenimento, páginas *web*, e mundos virtuais compartilhados (ISO/IEC 19775-2, 2010).

Entre suas principais características estão:

- É código livre, ou seja, licenças não são necessárias.
- Suporta XML, isto faz com que os dados 3D sejam expostos mais facilmente na *web* e em aplicações distribuídas.
- Os objetos 3D podem ser manipulados em linguagens de programação como C, C++ e Java.

#### 4.5 COMUNICAÇÃO DE DADOS EM SISTEMAS DISTRIBUIDOS

No mundo dos computadores, *networking* é a prática de acoplar dois ou mais dispositivos de computadores com a finalidade de compartilhar dados ou recursos. Redes são construídas com a combinação de *software* e de *hardware*.

As redes podem ser categorizadas de diferentes maneiras. Uma delas as classifica de acordo à área geográfica que elas abrangem, enquanto outra se baseia na topologia ou nos tipos de protocolos que elas suportam.

##### 4.5.1 O modelo OSI (Interconexão de Sistemas Abertos)

Quando as primeiras redes apareceram, cada fabricante de computador utilizava uma arquitetura própria de rede, o que impedia que computadores diferentes pudessem ser conectados à mesma rede. A Organização de Padrões Internacionais (ISO) criou um modelo chamado OSI para descrever as camadas definidas em um sistema operacional de redes (BLANCHARD, 2011). O propósito das camadas é fornecer funções definidas para melhorar a conectividade entre os computadores fabricados por diferentes empresas. Cada camada tem uma entrada e uma saída padrão definida. O modelo OSI tem sete camadas:

- Camada de aplicação (camada superior)
- Camada de apresentação
- Camada de sessão

- Camada de transporte
- Camada de rede
- Camada de ligação de dados e
- Camada física (camada inferior)

#### **4.5.2 Modelo Cliente servidor**

É um modelo geral e se ajusta ao que na indústria se conhece como “processar e cooperar” (IBM, 1994). Refere-se ao conceito de dividir o trabalho de processamento de dados entre um computador cliente e um computador servidor.

Servidor é um computador que fornece recursos para usuários da rede. Clientes os são computadores que acessam ou utilizam os recursos oferecidos pelo servidor. Tais recursos podem ser de hardware, por exemplo, impressoras ou de software, por exemplo banco de dados, páginas *web*, etc.

## **5 ESTRUTURAS DE DADOS E PESQUISAS RELACIONADAS AO PADRÃO STEP-NC**

Este capítulo tem como finalidade apresentar a pesquisa relacionada ao padrão STEP-NC, citando-se trabalhos desenvolvidos nesta área e estruturas de dados que suportam a manufatura usando o padrão.

### **5.1 PESQUISAS EM STEP-NC**

Em se tratando da pesquisa em STEP-NC, segundo XU e NEWMAN (2006), existem quatro tipos de trabalhos de pesquisa:

1. Controle convencional do CNC usando STEP-NC;
2. Novo STEP-NC que possibilita o controle;
3. STEP-NC que possibilita o controle inteligente e
4. STEP-NC colaborativo que possibilita a usinagem.

O grau de adaptabilidade aumenta do tipo um ao tipo quatro. A seguir se tratará com mais detalhe cada um deles.

#### **5.1.1 Controle convencional do CNC usando STEP-NC**

Este tipo de investigação marcou o começo dos esforços de pesquisa relacionados com o padrão STEP-NC. As principais linhas de pesquisa têm a ver com o desenvolvimento de “tradutores” que possam ler arquivos STEP AP-203 ou AP-224, acrescentar informações de manufatura, e convertê-los em um formato de código G que o CNC da máquina possa entender. Os tradutores são similares aos “pós-processadores” usados em muitos sistemas CAD/CAM. A única diferença é que os sistemas CAD/CAM e CNC são agora interoperáveis no sentido de que as informações aderentes ao padrão STEP podem ser usadas em diferentes níveis do sistema de manufatura. Também as informações de projeto que podem ser embutidas no arquivo STEP-NC ficam disponíveis para os sistemas CNC (XU e NEWMAN, 2006).

O projeto SuperModel está dentro desse tipo de trabalho de pesquisa, o qual está conduzindo esforços para o desenvolvimento de software e bases de dados para um sistema de projeto e manufatura. As bases de dados contêm todas as informações necessárias para construir a geometria de uma peça, dados do planejamento do processo como as *features* e as estratégias de usinagem, seleção das ferramentas, custos, etc. O projeto SuperModel está sendo desenvolvido pela empresa de software STEP Tools (<http://www.steptools.com>).

Numa primeira etapa, o “STEP-NC Machine Explorer” foi desenvolvido, o mesmo que pode ler dados de diversos softwares de CAD (Unigraphics, ProEngineer, Catia, etc.) em formato STEP AP-203.

Na segunda etapa, o sistema desenvolvido pela equipe da empresa STEP tools efetua a conversão dos dados de projeto do arquivo STEP AP-203 para o AP-238 (AIM do padrão ISO 14649) através de um plug-in desenvolvido para o Mastercam (<http://www.mastercam.com/>). A utilização desses módulos requer licenças comerciais.

### **5.1.2 Novo STEP-NC que possibilita o controle**

Diversas equipes de pesquisa têm usado controladores de arquitetura aberta (OMAC) para inserir dentro deles processadores que são capazes de interpretar o arquivo STEP-NC (XU e NEWMAN, 2006).

A maioria dos trabalhos realizados nos Estados Unidos encontra-se nesta categoria de pesquisa, onde o principal foco tem sido o desenvolvimento do controle CNC que possibilite o uso do STEP-NC, sendo utilizado o controlador Siemens 840D (WECK, 2003). Busca-se assim permitir que o arquivo físico STEP-NC seja integrado diretamente com o controlador, com a visualização das *features* e dos seus *workingsteps* associados em uma versão aderente ao padrão STEP-NC do seu sistema CAM ShopMill. Desenvolvimentos de programação em paralelo a este trabalho foram realizados no instituto WZL, na Universidade de Aachen, Alemanha, com o sistema de programação de chão de fábrica incorporando o “WZL Mill”, um sistema de programação aderente a STEP-NC para o processos de fresamento. Aplicações comerciais na Europa com sistemas CATIA e OpenMind têm sido apresentados pela Volvo e Daimler Chrysler (WECK, 2003), ilustrando a capacidade de incorporar o padrão dentro dos produtos

CAD/CAM e exportar o arquivo em STEP-NC para o controlador Siemens 840D.

Um módulo de software protótipo chamado “STEPTurn” foi desenvolvido no Instituto ISW Stuttgart (HEUSINGER *et. al.*, 2006), utilizando-se um controlador Siemens 840 em um centro de usinagem Boehringer NG200. O STEPTurn pode importar a geometria CAD e as *features* de usinagem, definir as estratégias e tecnologias de usinagem, e gerar o arquivo físico STEP-NC.

PACHECO *et. al.*, (2011) implementaram uma máquina com controlador livre de código G para processos de furação. Tal controlador é capaz de ler um arquivo em formato ISO 10303-21, e executá-lo.

### 5.1.3 STEP-NC que possibilita o controle inteligente

O controle inteligente em uma máquina CNC ainda não foi realizado verdadeiramente. A principal razão é que a informação disponível para a máquina CNC (código G) é de baixo nível, com a qual somente uma mínima quantidade de trabalho de otimização pode ser realizado em tempo real. Com o STEP-NC, a informação do projeto e do planejamento do processo está disponível para as máquinas CNC ou seus controladores, para desempenho de alto nível. Dentre estas atividades tem-se: *setup* automático da peça, geração automática das trajetórias das ferramentas, status da precisão da usinagem, realimentação dos resultados, verificação completa e antecipada da usinagem a fim de evitar colisões, otimização da sequência dos *workingsteps*, controle adaptativo e inspeção da peça na máquina (XU e NEWMAN, 2006).

Pesquisadores do Laboratório de Pesquisa Nacional para a Tecnologia STEP-NC (NRL-SNT) na PosTECH, Coréia do Sul, têm desenvolvido um sistema de controle autônomo STEP-NC baseado em *features* para um sistema de manufatura virtual de arquitetura aberta (SUH *et. al.*, 2002, SUH *et. al.*, 2003a, SUH *et. al.*, 2003b).

No Laboratório de Sistemas de Manufatura da Universidade de Auckland (Nova Zelândia) desenvolveu-se uma máquina CNC aderente à norma STEP, buscando implementar um cenário de usinagem sem a utilização do código G (XU, 2004). Esse trabalho de pesquisa consiste de duas partes: readaptação de uma máquina CNC existente e o desenvolvimento de um conversor NC aderente a STEP (denominado

STEPcNC). O conversor STEPcNC pode entender e processar códigos em formato STEP-NC, e fazer a interface com o controlador CNC.

Nos anos recentes foram publicados trabalhos como o de LAN *et. al.*, (2008), que propuseram um arcabouço conceitual para um controlador CNC inteligente baseado em multi-agentes para uma máquina CNC, onde a arquitetura está composta por diversos agentes, entre os quais: agente interpretador, agente de planejamento do processo, agente de tomada de decisões. Tais agentes cooperam para realizar as várias funções do controlador inteligente STEP-NC da maneira desejada.

Outro trabalho é o de ZHAO *et. al.*, (2008), que propuseram um modelo de dados baseado no padrão STEP-NC para inspeções *on-line* (na máquina). Nesse trabalho, os autores apresentam um arcabouço para a usinagem de malha fechada possibilitando o uso de STEP-NC com o objetivo de conseguir uma cadeia completa de projeto assistido por computador, planejamento do processo, fabricação e inspeção.

VICHARE *et. al.*, (2007) apresentaram uma metodologia para modelar o aspecto funcional das máquinas CNC com o objetivo de estabelecer um procedimento universal para representar os recursos de máquinas. Esta informação pode então ser mapeada dentro do modelo do produto para gerar planos de processo específicos para cada máquina. Já que o modelo de dados STEP-NC pode suportar dados de operações de usinagem, tecnologias, ferramentas e geometrias do produto, porém não suporta a representação de máquinas, esses autores propõem um modelo que preencha esta lacuna através do fornecimento de dados específicos na forma de esquemas EXPRESS.

#### **5.1.4 STEP-NC Colaborativo que possibilita a usinagem.**

XU e NEWMAN, (2006) apontam à manufatura colaborativa e distribuída baseada na *web* como a última meta da usinagem usando-se STEP-NC (Figura 5.1). Isto permitiria projetar em qualquer lugar e fabricar em qualquer lugar. Isto pode-se conseguir já que um programa STEP-NC pode separar a informação de manufatura “genérica” (ou seja, o que fazer), da informação de manufatura específica (ou seja, como fazer) que é nativa a cada máquina específica. Portanto, um programa genérico STEP-NC pode ser elaborado independentemente da máquina, o que é vantajoso em relação ao código G convencional, o qual é sempre gerado para uma máquina CNC em particular. Para que este tipo de



programa STEP-NC seja implementado em um sistema nativo CNC, o conhecimento de manufatura nativo (de uma determinada máquina) deve ser acrescentado (WANG e XU, 2004).

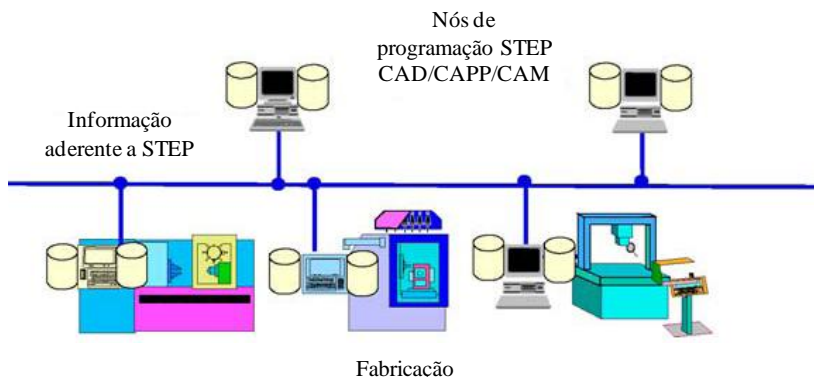


Figura 5.1. Fabricação aderente a STEP-NC, distribuída.

Fonte: Adaptado de NEWMAN, (2004).

Nos anos recentes tem havido a tendência ao uso de XML (ISO 10303-28) em lugar do tradicional arquivo em formato ISO 10303-21 para representar a informação STEP-NC. A razão para isto é que XML permite o processamento de informações em um cenário de manufatura eletrônica (*e-Manufacturing*) (XU e NEWMAN, 2006). Desta forma, as máquinas CNC podem compartilhar as informações com outros departamentos dentro ou fora da empresa por meio da Intranet/Internet.

Na Tabela 5.1 apresenta-se um resumo dos trabalhos relacionados a este campo de pesquisa, que foi extraída de NEWMAN *et. al.*, (2008).

**Tabela 5.1. Resumo das pesquisas em STEP-NC.**

(continua)

Trabalho de pesquisa	Tecnologia			Padrão		Armazenagem de dados	
	Fresam	Torn.	Outro	AP-238	ISO 14649	XML	Texto
Suh <i>et. al.</i> , (1995)	✓						
Hardwick, (2002)	✓			✓			
Venkatesh <i>et. al.</i> , (2005)	✓			✓			

**Tabela 5.2. Resumo das pesquisas em STEP-NC.**

(continuação)

Trabalho de pesquisa	Tecnologia			Padrão		Armazenagem de dados	
	Fresam	Torn.	Outro	AP-238	ISO 14649	XML	Texto
Hardwick e Loffredo (2006)	✓			✓			✓
Suh and Cheon, (2002)	✓				✓		
Suh <i>et. al.</i> (2002a)	✓				✓		
Suh <i>et. al.</i> (2002b)	✓				✓		
Suh <i>et. al.</i> (2003)	✓				✓		
Newman <i>et. al.</i> (2003)	✓				✓		✓
Lee and Bang (2003)	✓				✓		✓
Lee <i>et. al.</i> (2006)	✓				✓	✓	
Allen <i>et. al.</i> (2005)	✓				✓	✓	
Fichtner <i>et. al.</i> (2006)	✓				✓		✓
Nassehi <i>et. al.</i> (2006a)	✓			✓			
Nassehi <i>et. al.</i> (2006b)	✓			✓			✓
Nassehi <i>et. al.</i> (2006c)	✓			✓			✓
Liu <i>et. al.</i> (2006)	✓			✓			✓
Xu <i>et. al.</i> (2006)	✓				✓		✓
Amaitik and Engin Kilic (2007)	✓				✓	✓	
Zhao <i>et. al.</i> (2008)			✓		✓		✓
Rosso <i>et. al.</i> (2004)		✓			✓		✓
Xu and Wang (2004)		✓			✓		✓
Xu (2006)		✓			✓		✓

**Tabela 5.3. Resumo das pesquisas em STEP-NC.**

(conclusão)

Trabalho de pesquisa	Tecnologia			Padrão		Armazenagem de dados	
	Fresam	Torn.	Outro	AP-238	ISO 14649	XML	Texto
Chen <i>et al.</i> (2005)		✓		✓			
Suh <i>et al.</i> (2006)		✓			✓	✓	✓
Choi <i>et al.</i> (2006)		✓			✓	✓	
Heusinger <i>et. al.</i> , (2006)		✓			✓		✓
Yusof (2006)		✓			✓		✓
Shin <i>et al.</i> (2007)		✓			✓		✓
Zhang <i>et. al.</i> , (2011)		✓			✓		✓

FONTE: NEWMAN *et. al.*, (2008).

## 5.2 ESTRUTURAS DE DADOS QUE SUPORTAM SISTEMAS CAD/CAM ADERENTES AO PADRÃO ISO 14649

ROSSO JUNIOR *et. al.*, (2002) sustêm que existem diversas opções para interpretar e implementar sistemas CAD/CAM usando o padrão STEP-NC. Eles definiram três principais estruturas de dados, que são descritas a seguir.

### 5.2.1 Sistema CAD/CAM que importa/exporta uma saída aderente ao padrão STEP-NC

É um sistema CAD/CAM que tem a capacidade de gerar uma saída aderente à norma STEP-NC, e tem dois subtipos. O primeiro é capaz de somente exportar o código STEP-NC, enquanto o segundo pode importar e exportar.

### 5.2.1.1 Sistema CAD/CAM que exporta dados aderentes à norma STEP-NC

Um sistema CAD/CAM com esta estrutura de dados usa sua própria representação de *features* e suas estratégias de fabricação nativas para o projeto e manufatura de componentes. A geração da saída em ISO 14649 é criada pelo mapeamento das informações do sistema CAD/CAM nativo através de um pós-processador que gere a saída ISO 14649. O problema principal é que a informação da tecnologia de manufatura armazenada deve ser convertida no formato ISO 14649 através do pós-processador. A Figura 5.2 apresenta esta estrutura de dados.

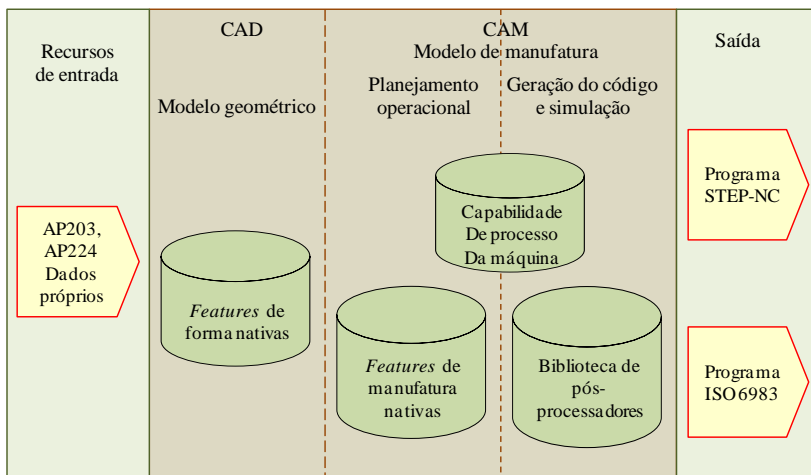


Figura 5.2. Estrutura de dados de um sistema CAD/CAM que exporta dados aderentes a STEP-NC.

Fonte: Adaptado de ROSSO JUNIOR (2005).

### 5.2.1.2 Sistema CAD/CAM que importa e exporta códigos aderentes à norma STEP-NC

Sistemas CAD/CAM com esta estrutura de dados são capazes de importar e gerar uma saída aderente ao padrão STEP-NC. Os dados importados aderentes à norma são traduzidos em estruturas de dados

geométricos e de manufatura nativos do sistema CAD/CAM usando um pós-processador reverso. A saída é gerada da mesma maneira que os sistemas descritos no item anterior. A Figura 5.3 apresenta um diagrama deste tipo de estrutura de dados.

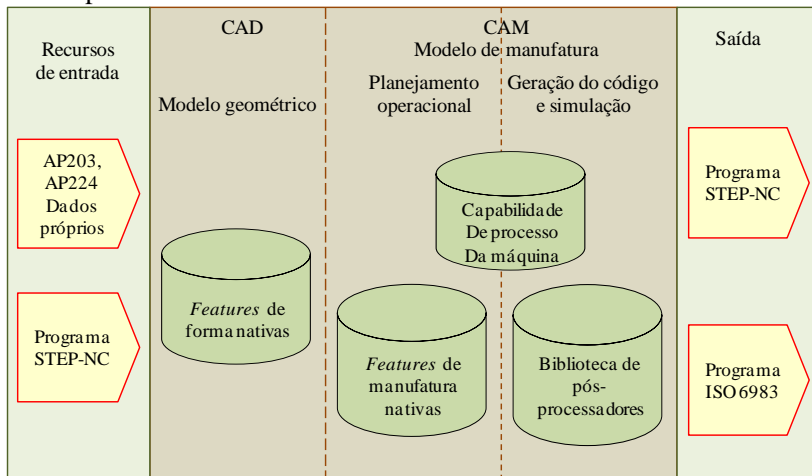


Figura 5.3. Estrutura de dados de um sistema que exporta/importa dados STEP-NC.

Fonte: Adaptado de ROSSO JUNIOR (2005).

## 5.2.2 Sistema CAD/CAM com estruturas de suporte de dados aderentes a STEP-NC.

Estes sistemas são classificados desta maneira no caso do sistema CAD/CAM possuir uma estrutura externa ou interna de suporte a dados STEP-NC.

### 5.2.2.1 Sistema CAD/CAM integrado com um sistema externo de suporte de dados aderente a STEP-NC

O sistema CAD/CAM está integrado a um sistema de software externo para fornecer o suporte do gerenciamento de dados aderentes a STEP-NC. A Figura 5.4 mostra o sistema CAD/CAM funcionando independente do sistema de gerenciamento de dados aderente a STEP.

Este não somente importa informações e gera o código STEP, mas também interpreta as rotinas geométricas e de manufatura nativas. Sistemas CAD/CAM com esta estrutura têm maior controle para configurar a saída comparado aos apresentados anteriormente (seção 5.2.1).

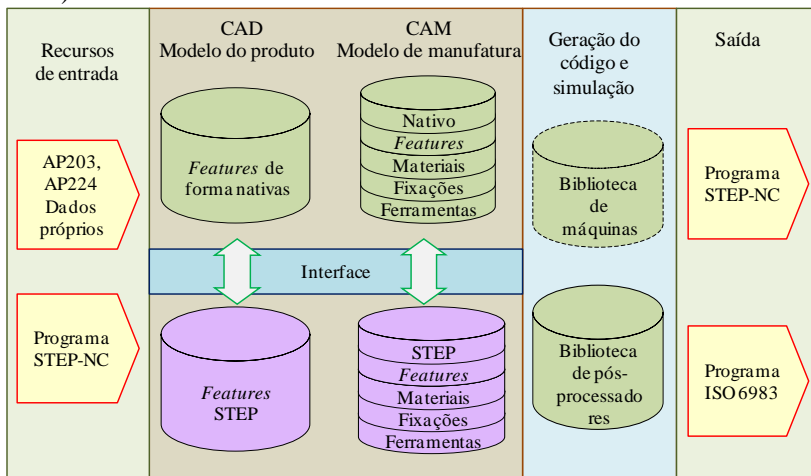


Figura 5.4. Sistema aderente a STEP interfaceado externamente ao ambiente CAD/CAM.

Fonte: Adaptado de ROSSO JUNIOR (2005).

### 5.2.2.2 Ambiente CAD/CAM com estruturas de dados nativa e aderente a STEP-NC internas

Representa o sistema CAD/CAM que tem uma representação interna dual dos dados geométricos e de manufatura, em formato nativo do sistema CAD/CAM e em formato ISO 14649. Portanto, no uso operacional deste sistema, ambos os conjuntos de dados são atualizados em tempo real antes de gerar a saída STEP-NC através do pós-processador ou de ter as estruturas de informação STEP em um ambiente de software externo. Nestes sistemas o pós-processamento não é requerido. A Figura 5.5 mostra esta estrutura de dados.

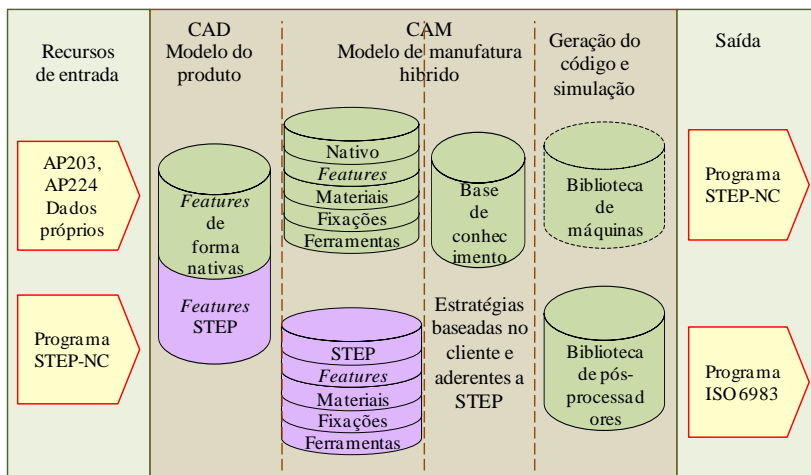


Figura 5.5. Um ambiente CAD/CAM aderente a STEP-NC e compartilhado internamente.

Fonte: Adaptado de ROSSO JUNIOR (2005).

### 5.2.3 Ambiente CAD/CAM com kernel baseado na estrutura de dados aderente a STEP-NC

Este é o mais alto nível de aderência ao padrão. O sistema CAD/CAM usa os padrões ISO 14649 e ISO 10303-224 para a representação dos modelos de dados geométricos e de manufatura. ROSSO JUNIOR (2005) aponta para esta estrutura de dados como a base para novos desenvolvimentos de sistemas CAD/CAM. Esta estrutura também tornará a geração do código STEP-NC direta e mais simples. A Figura 5.6 apresenta um arcabouço com esta estrutura de dados.

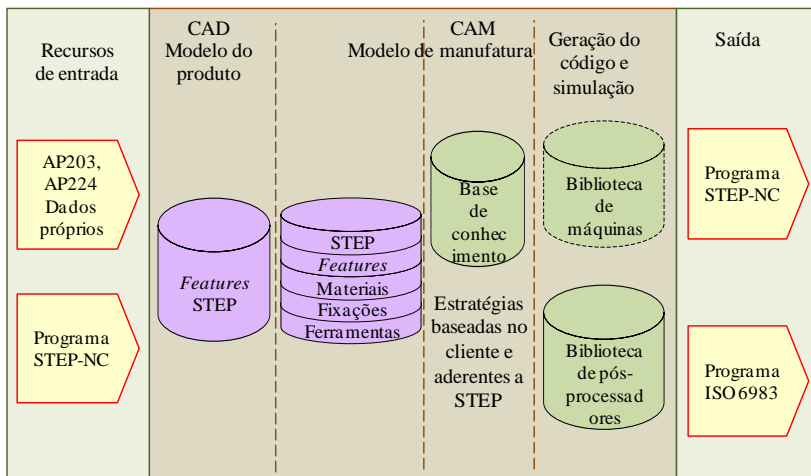


Figura 5.6. *Framework* para uma estrutura de dados interna aderente a STEP.  
 Fonte: Adaptado de ROSSO JUNIOR, (2005).



## **6 MODELAGEM CONCEITUAL DO PROTÓTIPO DO SISTEMA INTEGRADO CAD/CAPP/CAM ADERENTE AO PADRÃO ISO 14649**

No âmbito das metodologias de projeto existem vários modelos de desenvolvimento, e EVBUOMWAN *et. al.* (1996) apresentaram as várias filosofias e tendências das mesmas. As metodologias têm o propósito de prescrever, de maneira sistematizada, como o processo de projeto deveria ser realizado, apontando para a melhor maneira de realizar o projeto. Estas metodologias têm alguns pontos em comum, os quais foram adotados no presente trabalho de pesquisa. Os principais passos são:

- Reconhecimento das necessidades;
- Especificação dos requisitos;
- Formulação de um conceito;
- Elaboração do projeto detalhado, e
- Produção (implementação).

Em relação ao reconhecimento das necessidades, estas foram identificadas no capítulo 1, e tais necessidades motivaram o desenvolvimento do presente trabalho de tese.

No presente capítulo será feita a especificação dos requisitos e a especificação do modelo conceitual.

### **6.1 REQUISITOS PARA UMA ESTRUTURA DE DADOS CAD/CAPP/CAM INTEROPERÁVEL E EXECUTADA REMOTAMENTE**

Os inconvenientes provocados pela forma na qual os sistemas CAD/CAM tradicionais trocam informações foram descritos no capítulo 1, e tais problemas motivaram a busca de um conjunto de requisitos de projeto para uma estrutura de dados interoperável. Tal estrutura de dados deveria garantir que as informações sejam mantidas estáveis e integras. Segundo XU *et. al.*, (2006), sistemas de manufatura futuros devem possuir algumas características importantes:

- Fluxo de informações sem lacunas;
- Utilização de *features* no modelo de manufatura;

- Autonomia do CNC;
- Devem ser baseados em rede, distribuídos;
- Devem ser modulares;
- Devem ser portáteis e adaptativos.

Tomando como base estas características, e levando em consideração os objetivos da presente tese apresentados no capítulo 1, pode-se adotar um conjunto de requisitos, os quais serão apresentados a seguir.

### **6.1.1 Uso de modelos de dados padronizados**

Sistemas proprietários não permitem qualquer desenvolvimento adicional. Novas versões do sistema devem ser atualizadas pelos criadores originais. Por outro lado, sistemas padronizados têm o benefício da quantidade maior de desenvolvedores, e novos desenvolvedores poderiam contribuir e estender tal estrutura de dados.

Para o desenvolvimento da presente estrutura de dados, será necessária a utilização de vários modelos de informação padronizados:

#### **6.1.1.1 Modelo de informação padrão do produto**

Para se ter um fluxo de dados homogêneo, o produto deve ser definido de maneira que as informações possam ser lidas e usadas pelos outros módulos. Este modelo deve conter informações acerca da representação geométrica do produto, *features* e tolerâncias como mostrado na Figura 6.1.

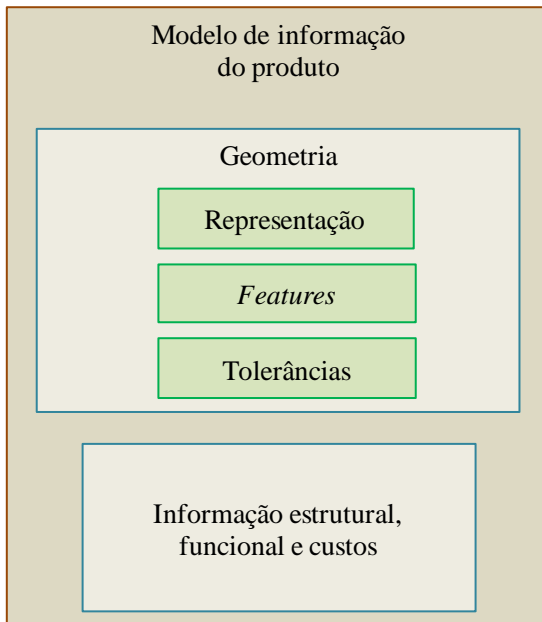


Figura 6.1. Modelo de informações do produto.

Fonte: Adaptado de NASSEHI, (2007).

Adicionalmente, o modelo também pode conter informações acerca da função que o produto desempenhará, e seus custos.

#### 6.1.1.2 Modelo de informação do processo

Este modelo contém as informações relacionadas ao plano do processo para a fabricação do produto, especificamente o plano de usinagem e o plano de inspeção da peça. Adicionalmente, ele contém os elementos de fixação como mostrado na Figura 6.2.

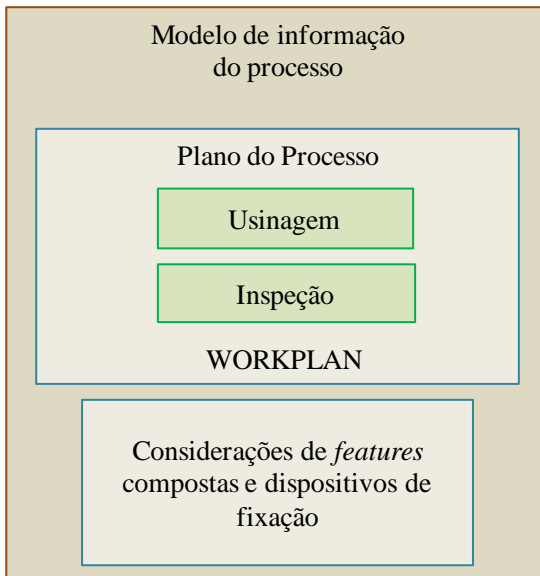


Figura 6.2. Modelo de informações do processo.  
Fonte: Adaptado de NASSEHI, (2007).

### 6.1.1.3 Modelo de informação dos recursos

Este modelo contém as informações relacionadas aos recursos disponíveis para a manufatura do produto como, por exemplo, máquinas e ferramentas (Figura 6.3).

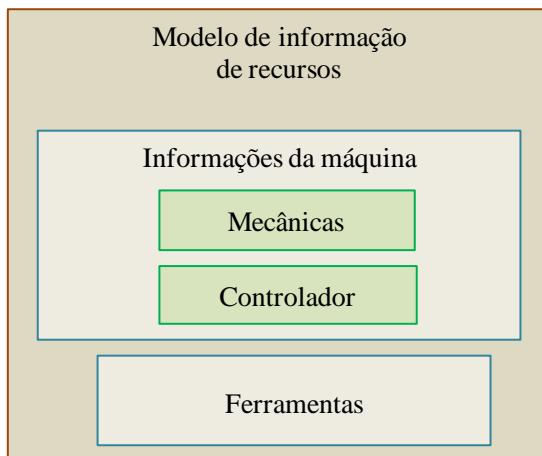


Figura 6.3. Modelo de informação de recursos.  
Fonte: Adaptado de NASSEHI, (2007).

Como visto no item 2.2.2.2, o padrão ISO 14649 facilita a modelagem dos processos de manufatura, pois ele fornece um modelo de dados para representar planos de processo, ele também abrange a representação das geometrias e das *features*, além de possuir as entidades necessárias para modelar o processo de manufatura. Como nesta pesquisa o interesse está nos processos de fabricação de peças prismáticas, o uso deste padrão torna-se adequado.

Em relação ao modelo de recursos, o padrão ISO 14649 dá suporte à modelagem de ferramentas, porém no momento da elaboração da presente tese, ainda não suporta a modelagem de máquinas nem a controladores.

Outra razão para a utilização do ARM do padrão STEP-NC (ISO 14649) no presente trabalho de tese, é que sua implementação é menos complexa do que a do AIM (ISO 10303-238). Diferenças entre as duas implementações, suas vantagens e desvantagens podem ser encontradas em FEENEY *et. al.*, (2003).

Para finalizar esta seção, as informações acerca dos modelos do produto, do processo e de manufatura, devem ser transferidas de maneira que possam ser entendidas, e portanto uma linguagem de transferência padrão deve ser usada para conseguir isto. A parte 21 da norma ISO 10303, e a parte 28 da mesma norma descritas na seção 2.2.1.2 serão usadas no presente trabalho de pesquisa.

### **6.1.2 Orientação a objetos**

A programação orientada a objetos é o paradigma dominante na atualidade, tendo substituído as técnicas de programação estruturais desenvolvidas nos anos 1970 (HORSTMANN e CORNELL, 2008).

Diferentemente das técnicas estruturadas em que os programadores procuram um conjunto de procedimentos ou algoritmos para resolver o problema, em um programa orientado a objetos, cada objeto tem uma funcionalidade específica, a mesma que é exposta aos seus usuários, e sua implementação permanece escondida.

O modelo de dados fornecido pela norma ISO 14649 é muito semelhante aos conceitos da programação orientada a objetos, onde as entidades definidas na norma podem ser abstraídas como sendo classes de onde objetos são instanciados. Portanto, o uso da técnica de programação orientada a objetos é a mais adequada no desenvolvimento da estrutura de dados a ser implementada neste trabalho.

### **6.1.3 Independência de plataforma**

Como uma das características da proposta do presente trabalho de tese é que o sistema possa ser executado através da internet, a independência de plataforma computacional torna-se imperativa. Os diferentes usuários remotos do sistema devem ter a liberdade de executar o sistema em seu sistema operacional nativo.

Em relação a este requisito e juntamente com o último apresentado, a linguagem Java (seção 4.4.1) ajusta-se a tais requisitos.

### **6.1.4 Modularidade**

Para se tirar maior vantagem da orientação a objetos, as tarefas devem ser divididas em pequenos grupos ou módulos e sub-módulos que trabalhem independentemente, facilitando o teste individual de cada um dos componentes do módulo.

## 6.2 ESPECIFICAÇÃO DO MODELO

No contexto de projeto do arcabouço, uma série de atividades, tarefas ou funções devem ser definidas usando-se ferramentas específicas. Isto com a finalidade de facilitar a implementação, detectar e depurar erros de maneira eficiente.

### 6.2.1 Modelagem funcional do protótipo

A modelagem funcional tem por objetivo traduzir os requisitos qualitativos e quantitativos do projeto em funções ou tarefas agrupadas em diferentes níveis e hierarquias (BLANCHARD *et. al.*, 1995). Esta tradução é um processo iterativo que se realiza com o uso de linguagens de modelagem gráfica tais como os diagramas de fluxo funcionais. Estas metodologias de modelagem, particularmente os métodos de definição integrados (IDEF) são de grande importância, devido à sua capacidade para facilitar a análise (HEISIG *et. al.*, 2010). A abordagem funcional assegura que a definição de todo o sistema seja completamente reconhecida e tratada.

Um modelo IDEF0 pode ser decomposto para mostrar informações mais detalhadas.

No presente trabalho, a modelagem do sistema será feita usando a ferramenta IDEF0.

As informações necessárias para que o sistema possa realizar as suas funções são:

- O cliente ou usuário remoto: com informações associadas a ele (nome, endereço, instituição, etc.).
- Projeto: o que conterà todas as informações gerais acerca do projeto (nome do projeto, peça de trabalho, material, etc.).
- *Features*: que contém informações acerca da topologia da peça.

A Figura 6.4 apresenta um diagrama do sistema no nível mais alto, enquanto a Figura 6.5 mostra a decomposição do sistema em suas três principais atividades, a saber: atividade de projeto (CAD, A1), atividade de planejamento do processo (CAPP, A2) e atividade de manufatura (CAM, A3). Serão descritas a seguir cada uma delas.

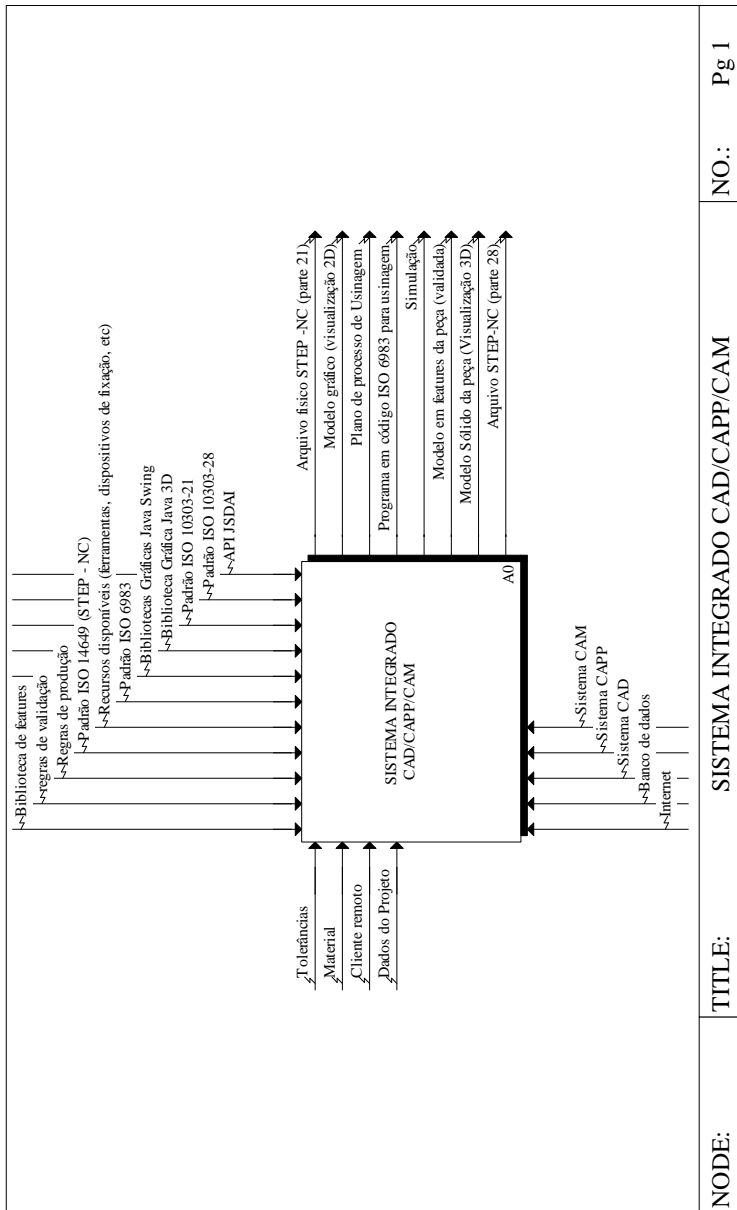


Figura 6.4. Diagrama IDEF0 do sistema proposto.  
 Fonte: Criado pelo autor.



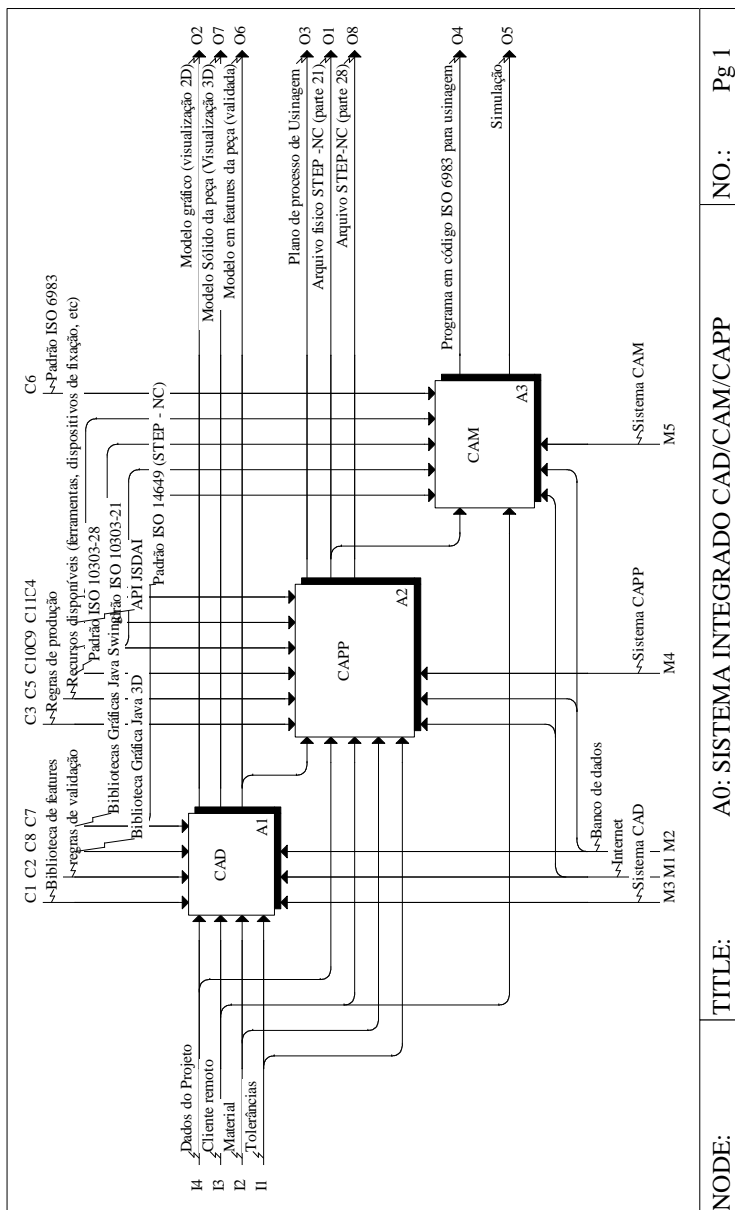


Figura 6.5. Diagrama IDEF0 do Sistema Integrado, nível A0.

Fonte: Criado pelo autor.

### 6.2.1.1 Atividade de projeto

Esta atividade deveria dar suporte ao projeto de peças prismáticas. Será utilizada a abordagem de modelagem por *features* DSG proposta por SHAH e MÄNTYLÄ (1995), descrita no item 3.3.4. O usuário instanciará as *features* disponibilizadas na biblioteca. A Figura 6.6 apresenta um diagrama IDEF0 desta atividade, e as suas etapas são descritas a seguir.

#### 6.2.1.1.1 Criação de um novo projeto

Para a criação de um novo projeto, certas informações são essenciais, como por exemplo, dados acerca do projetista (cliente remoto) os dados de material da peça, tolerâncias da mesma, e suas dimensões. A Figura 6.7 apresenta em um diagrama UML<sup>2</sup> os seus atributos principais.

#### 6.2.1.1.2 Modelagem da peça usando a biblioteca de *features*:

A modelagem é baseada nas entidades *machining\_features* definidas na norma ISO 14649 parte 10 (ver também item 2.2.2.4.2). Na norma, estas entidades estão definidas na linguagem EXPRESS, portanto devem ser mapeadas para classes Java. A Figura 6.8 mostra em um diagrama UML as classes Java que compõem a biblioteca de *features* do sistema.

---

<sup>2</sup> Unified Modeling Language™ é uma linguagem de modelagem orientada a objetos no âmbito da engenharia de software. Ela fornece uma forma de visualizar a arquitetura dos sistemas de software (OMG, 2011).

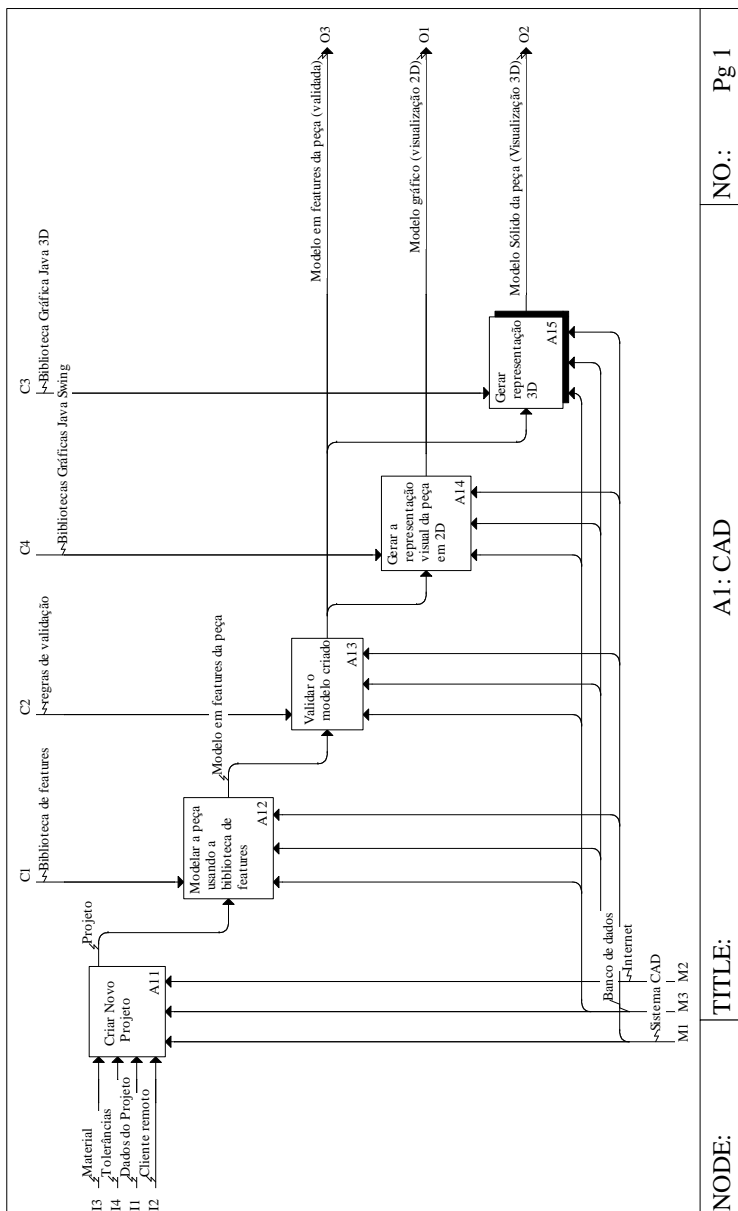


Figura 6.6. Módulo CAD modelado em um diagrama IDEF0.  
 Fonte: Criado pelo autor.

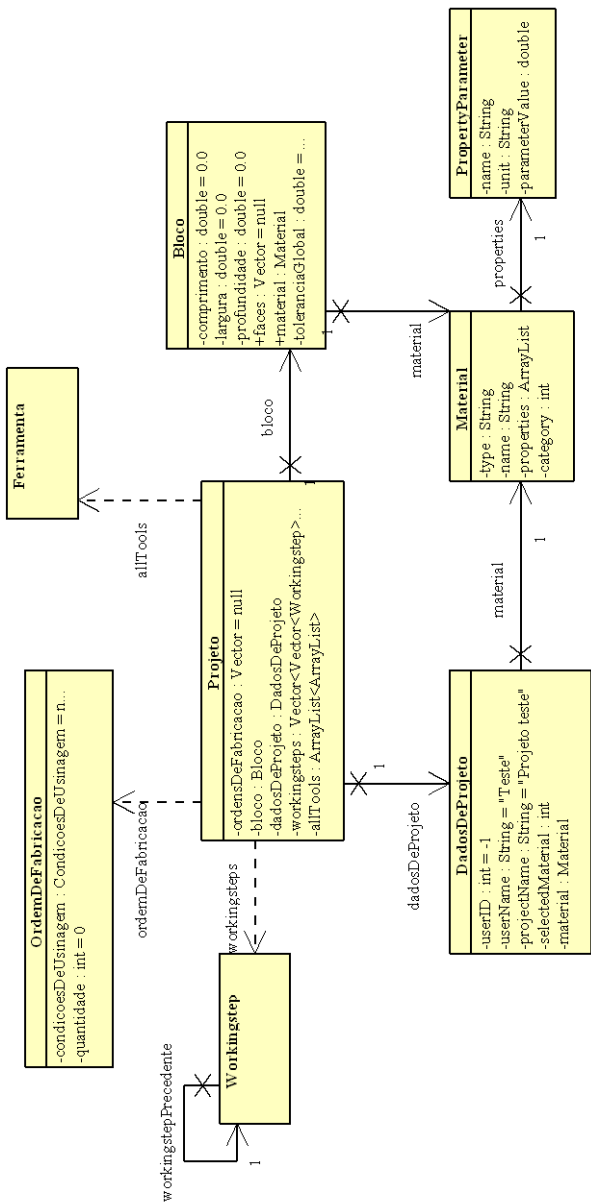


Figura 6.7. Classe Projeto em UML.  
 Fonte: Criado pelo autor.

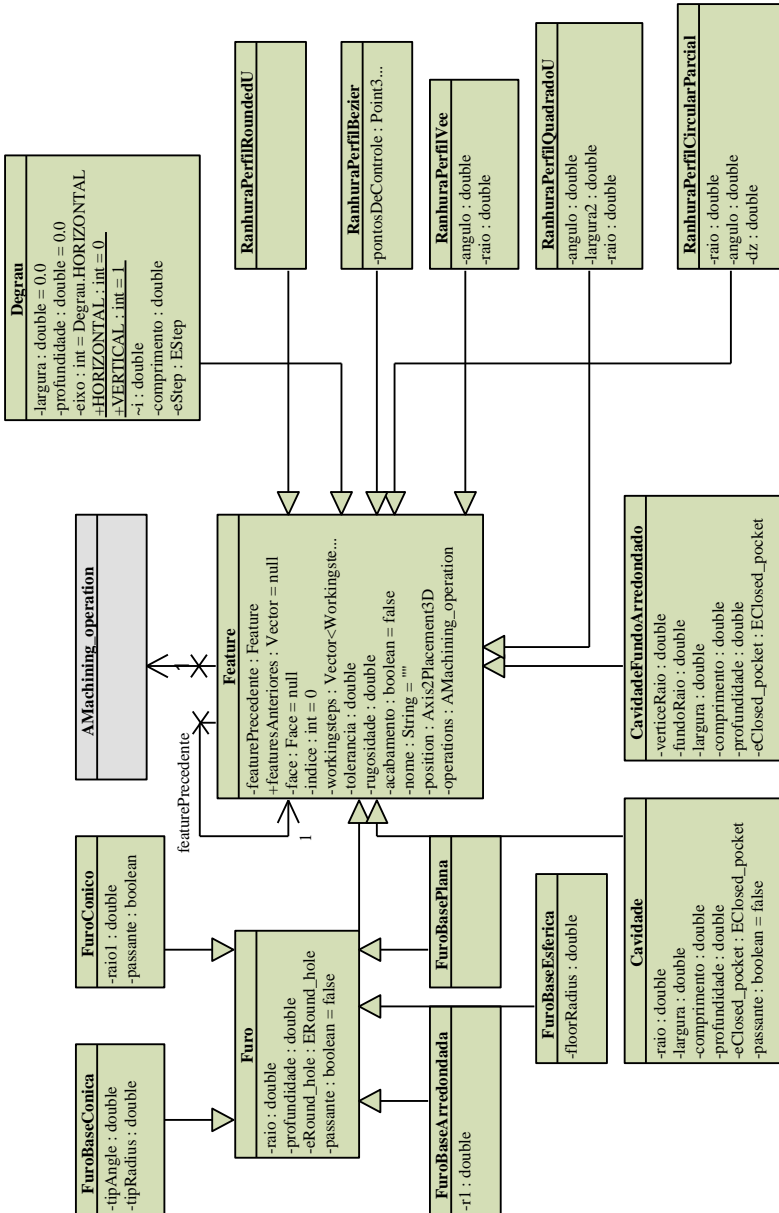


Figura 6.8. Biblioteca de *features* modelada em UML.  
Fonte: Criado pelo autor.

### 6.2.1.1.3 Validação do modelo

Antes da geração dos modelos gráficos em 2D ou em 3D, é necessário verificar se os dados fornecidos pelo usuário do sistema são consistentes. Esta atividade é essencial, pois deve garantir não apenas que o modelo sólido seja válido, mas também que o modelo possa ser obtido por operações de usinagem. Para conseguir isto, algumas regras de validação devem ser consideradas. Por exemplo, não é válida uma *feature* instanciada em uma posição cujas coordenadas estejam fora da peça. Também não será válida uma *feature* cuja profundidade exceda a profundidade máxima do bloco (para que seja válida, deveria ser especificado explicitamente que tal *feature* é passante). Não pode ser instanciada uma *feature* cuja posição indique que ela está dentro de material ainda sem remover da peça (ou seja, como se a peça tivesse vazia por dentro).

Outra importante regra é que qualquer nova *feature* instanciada não deve estar dentro do volume ocupado por outra, por exemplo, um furo instanciado na mesma posição que outro. É por esta razão que a classe *Feature* possui um atributo da mesma classe chamado *featurePrecedente* (ver Figura 6.8) que apontará para a *feature* na qual ela está inserida. Por exemplo, na Figura 6.9, a *featurePrecedente* do Furo é a Cavidade. Sem a cavidade, o furo é inválido. Isto também se aplica no momento de remover *features* da peça. A eliminação de uma *feature* pode causar a remoção de todas as suas *features* associadas.

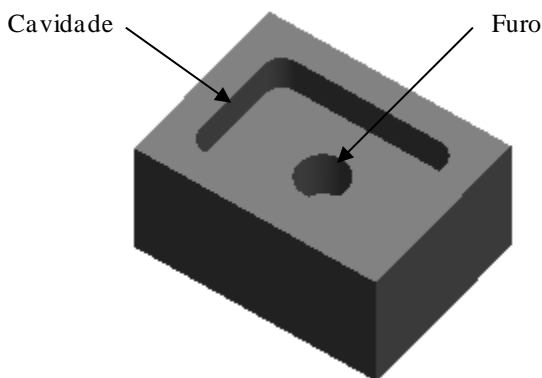


Figura 6.9. Furo e sua *feature* precedente.

Fonte: Criado pelo autor.

### 6.2.1.1.4 Geração do modelo geométrico 2D

Com os dados advindos do módulo validador, uma estrutura de classes especializadas no desenho 2D deve criar a representação da peça. A biblioteca Java Swing será usada aqui, pois ela possui classes com entidades gráficas prontas (incluindo-se círculos, elipses, retângulos, polígonos, diferentes tipos de linhas, etc.).

Basicamente duas classes estarão encarregadas do desenho (Figura 6.10):

- Uma delas desenhará as faces: a face principal, ou face visualizada, as faces laterais, e a face oposta.
- Outra deverá desenhar as *features* presentes em tais faces.

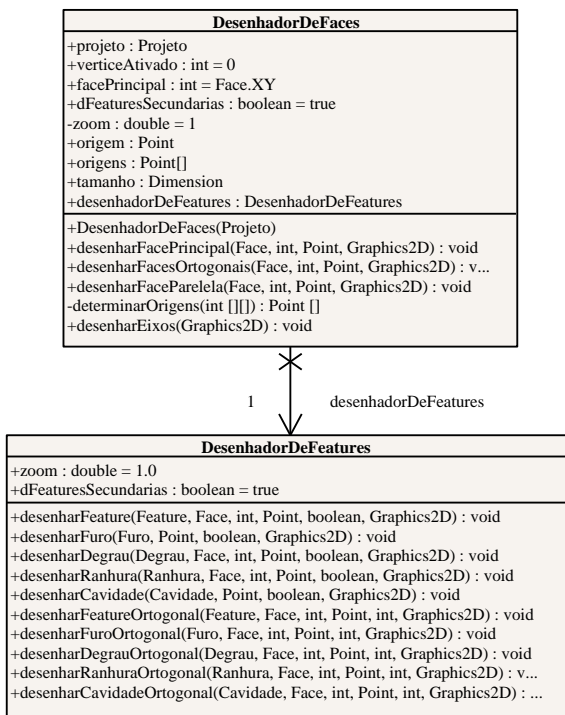


Figura 6.10. Diagrama UML das classes desenadoras.

Fonte: Criado pelo autor.

### 6.2.1.1.5 Geração do modelo gráfico 3D

As abordagens de representação de sólidos apresentadas no item 3.3.3 são capazes de criar modelos sólidos eficientemente. No presente trabalho se usará uma abordagem híbrida, ou seja, as abordagens B-rep e CSG serão usadas. Três tarefas importantes devem ser realizadas: a criação do sólido que representará a peça bruta, a criação do sólido que representará a *feature*, e a criação da peça acabada. A peça acabada será o resultado de operações booleanas entre o sólido da peça bruta e o sólido da *feature*. A Figura 6.11 apresenta um diagrama IDEF0 deste módulo mostrando estas três atividades.

#### (a) Criação da peça bruta

Neste trabalho uma entidade primitiva paralelepípedo representará a peça bruta (sendo esta uma peça prismática). Como a sua geometria é simples, pode se criar diretamente o seu modelo sólido em CSG (ver item 3.3.3.1) a partir das dimensões da peça bruta.



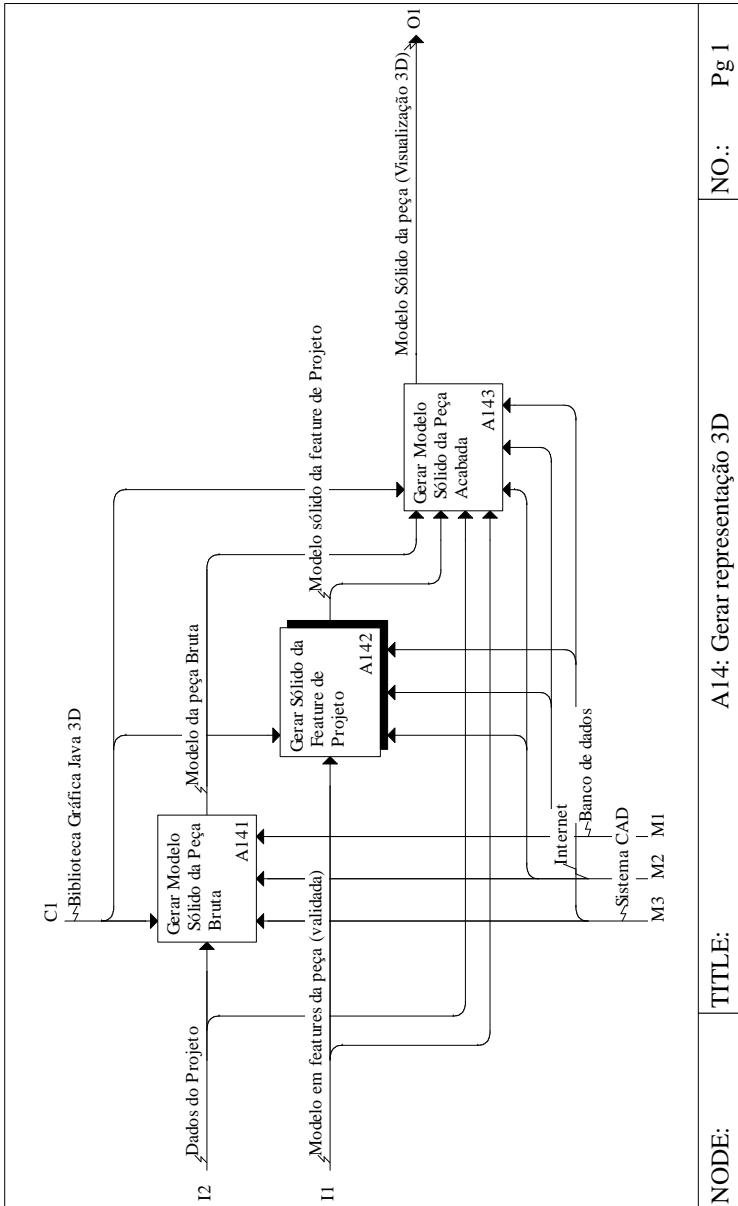


Figura 6.11. Diagrama IDEF0 do gerador da representação 3D da peça.  
 Fonte: Criado pelo autor.

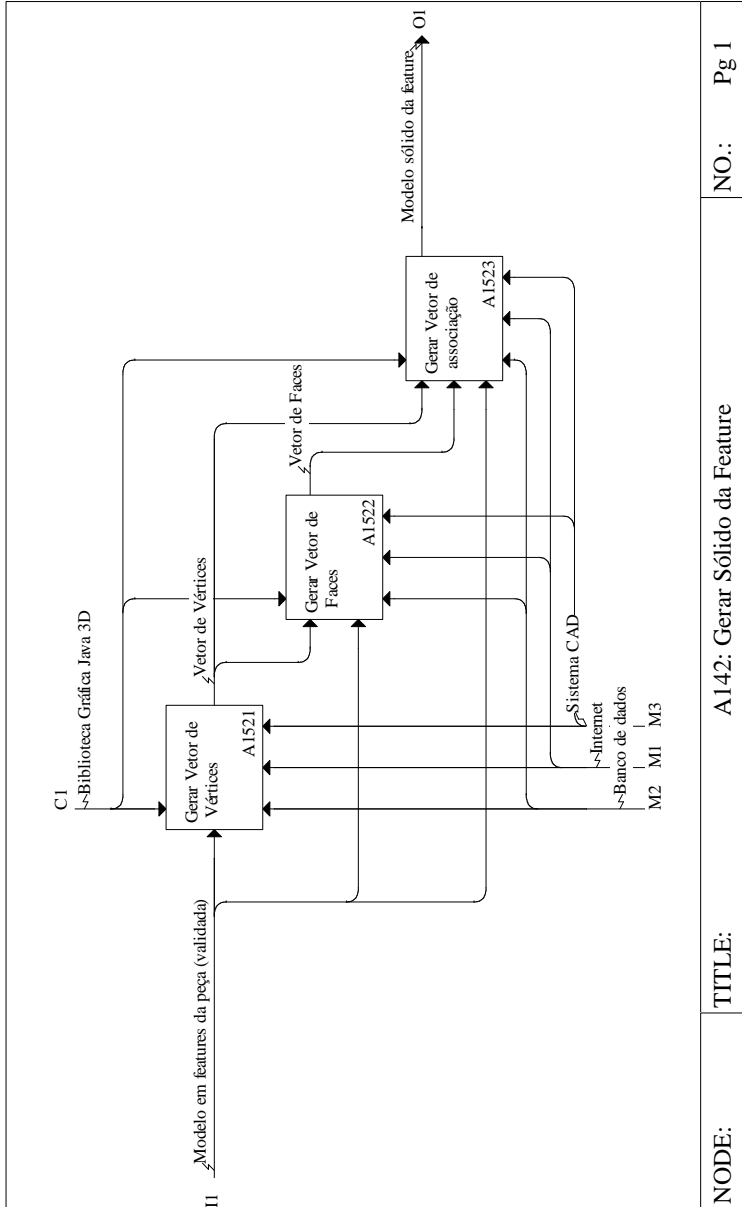
(b) Criação do modelo sólido das *features*

As *features* serão representadas usando-se a técnica B-rep descrita no item 3.3.3.2. Isto implica na geração de vetores de vértices, vetores de faces e vetores de arestas. A Figura 6.12 apresenta este módulo e as suas atividades em um diagrama IDEF0, enquanto a Figura 6.13 apresenta as classes que compõem esta atividade modeladas em UML.

Estas classes devem ter métodos capazes de: (a) calcular os vetores de vértices; (b) criar as arestas a partir da associação de pares de vértices; (c) criar as faces pela associação de arestas.

(c) Geração do modelo sólido da peça acabada

Para a geração do modelo sólido da peça acabada, a abordagem CSG será utilizada. Operações booleanas entre o sólido da peça bruta e os sólidos das *features* devem ser realizadas, portanto é necessário transformar o sólido da *feature* para CSG. O sólido da peça bruta não necessita ser transformado, pois ele é criado diretamente em CSG. Na Figura 6.14 o retângulo “Operações de sólidos” é o responsável por transformar sólidos de B-rep para CSG.



Pg 1

NO.:

A142: Gerar Sólido da Feature

TITLE:

NODE:

Figura 6.12. Diagrama IDEF0 do gerador de sólidos das *features*.  
 Fonte: Criado pelo autor.

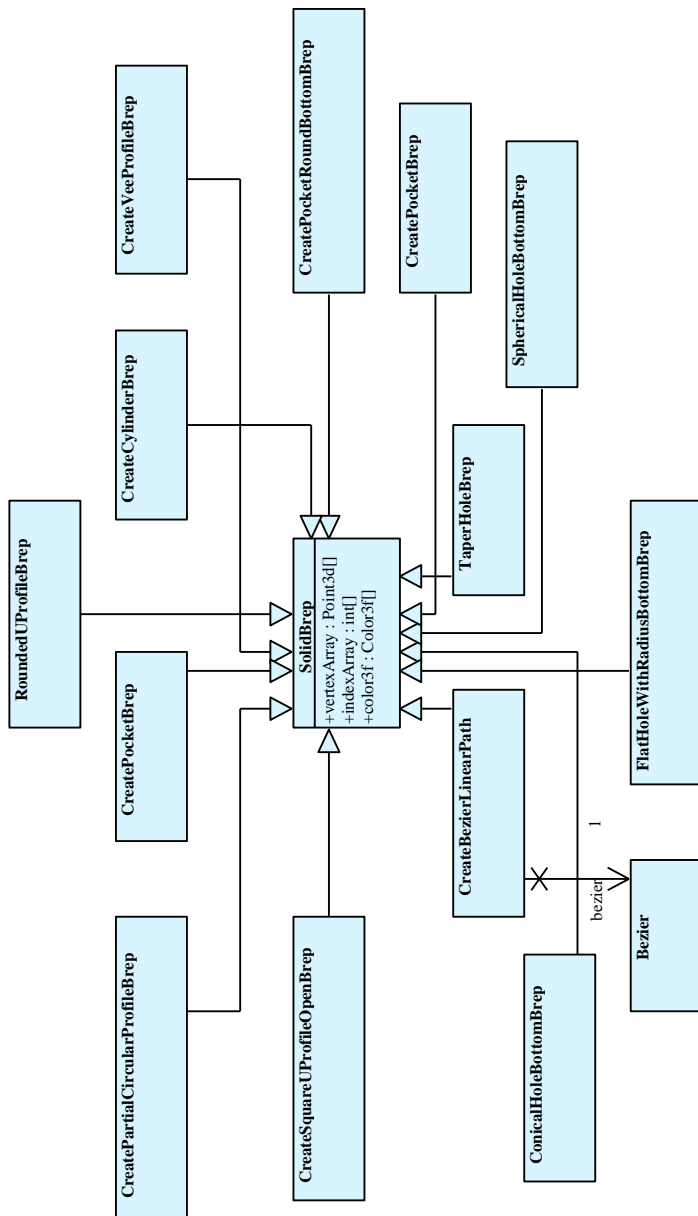


Figura 6.13. Diagrama UML das classes que geram as *features*.  
Fonte: Criado pelo autor.

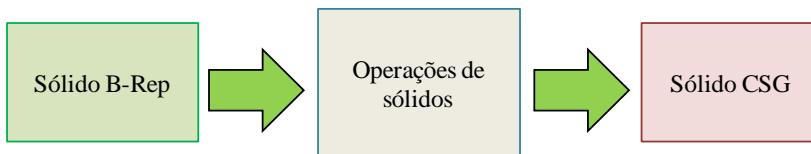


Figura 6.14. Operações de Sólidos.

Fonte: Criado pelo autor.

### 6.2.1.2 Atividade de Planejamento do Processo

Esta atividade tem por objetivo dar suporte ao planejamento do processo da peça projetada pelo usuário remoto. O resultado desta atividade é o plano do processo. Várias tarefas devem ser realizadas para conseguir tal objetivo (ver Figura 6.15):

- Gerar *Machining workingsteps* a partir das features;
- Organizar os *Machining workingsteps*;
- Determinar os pontos de apoio;
- Gerar o plano do processo; e
- Gerar o arquivo neutro.

A seguir serão vistos cada um deles em detalhe.

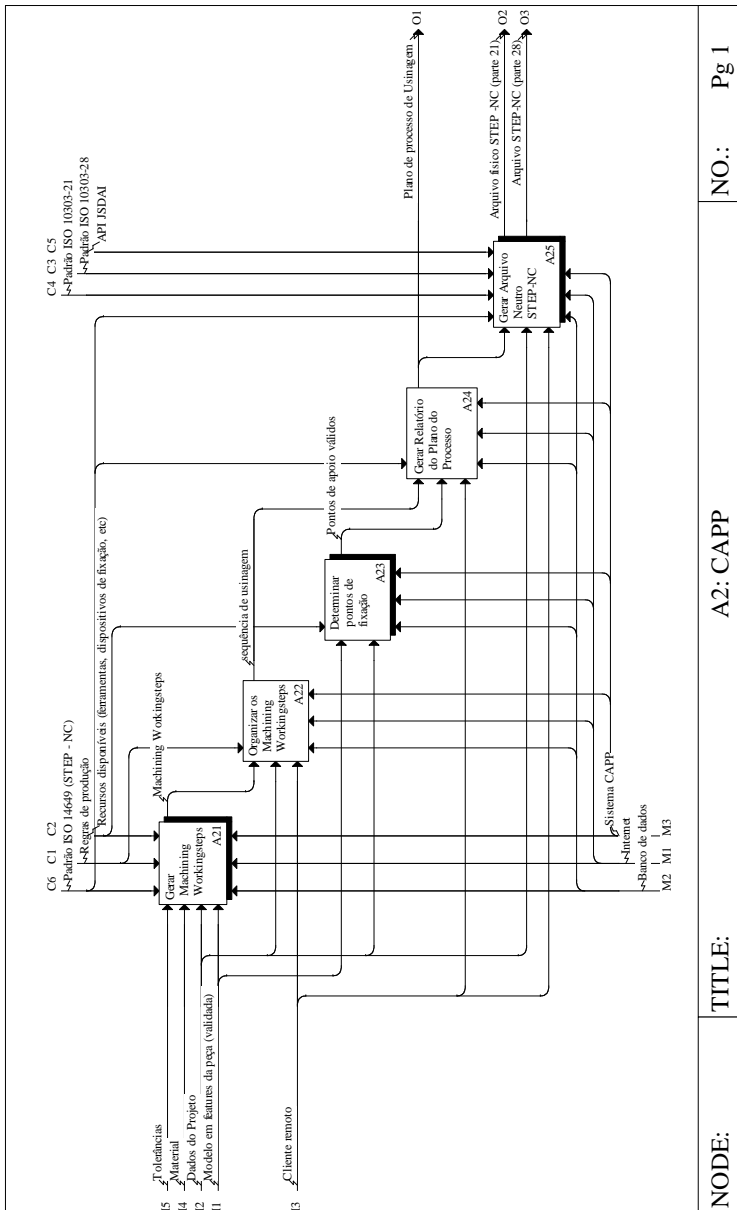


Figura 6.15. Atividade de planejamento do processo em um diagrama IDEF0.  
 Fonte: Criado pelo autor.

### 6.2.1.2.1 Gerar *Machining workingsteps*

Basicamente um *machining workingstep* é composto por: uma única operação de usinagem, uma única ferramenta e suas condições de usinagem. Além disso, o *machining workingstep* tem uma *feature*. A Figura 6.16 apresenta um diagrama UML do *machining workingstep*.

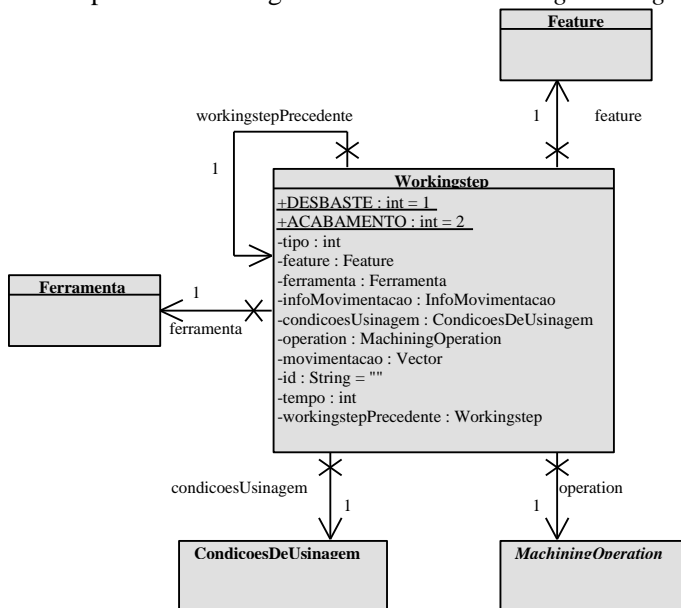
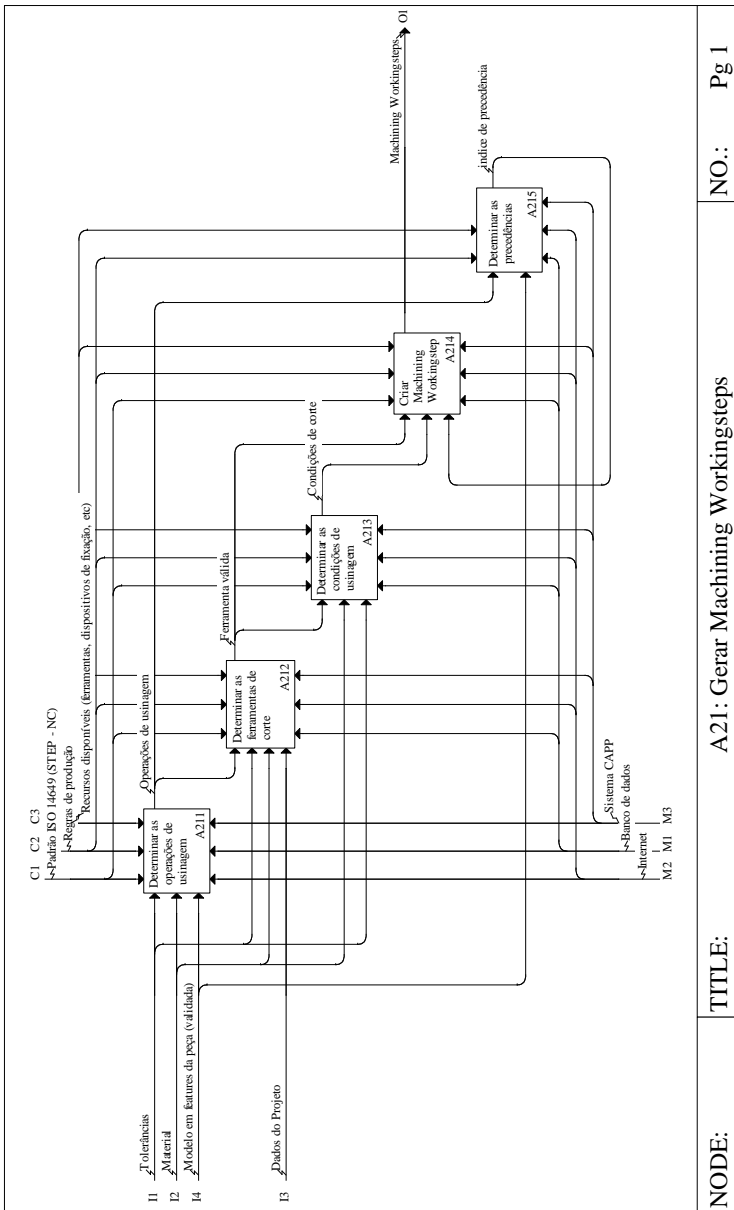


Figura 6.16. *Machining Workingstep* modelado em UML.

Fonte: Criado pelo autor.

Esta tarefa consiste basicamente em criar vetores (*arrays*) de *machining workingsteps* e associá-los a uma *feature*. A execução destes *machinings workingsteps* resultará na remoção do material da *feature*. A criação dos *arrays* de *machining workingsteps* deve ser feita por classes mapeadoras.

A Figura 6.17 apresenta um diagrama IDEF0 desta atividade, onde se pode ver que esta tarefa envolve outras cinco sub-tarefas, as quais serão descritas a seguir.



NO.: Pg 1

A21: Gerar Machining Workingsteps

TITLE:

NODE:

Figura 6.17. Diagrama IDEFO da atividade “Gerar Machining Workingsteps”.  
 Fonte: Criado pelo autor.



## (a) Determinação das operações de usinagem

Para se obter a forma de cada *feature*, processos de usinagem devem ser efetuados em determinadas regiões da peça. Por exemplo, em se tratando de furos com base plana (ver Figura 6.18), operações de furação não são suficientes, pois operações de furação com broca helicoidal deixam o perfil resultante da movimentação da broca, que é uma ponta cônica. Nesse caso, será necessário uma segunda operação (possivelmente de fresamento com fresa de topo com perfil adequado) ou de alargamento com ferramenta com perfil adequado.

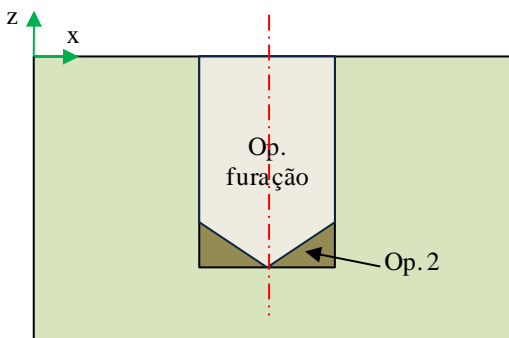


Figura 6.18. Furo com base plana e suas operações.  
Fonte: Criado pelo autor.

A modelagem em UML das operações de usinagem adotadas no presente trabalho segue a estrutura sugerida pela norma (vide 2.2.2.4.3). Basicamente há uma classe geral (*MachiningOperation*) que é a classe mãe das operações de usinagem, que incluem: furação de centro, furação, alargamento, mandrilamento, fresamento de superfície plana de desbaste e de acabamento, fresamento lateral e de base de desbaste e de acabamento, e fresamento de forma livre. A Figura 6.19 apresenta um diagrama UML das operações de usinagem.

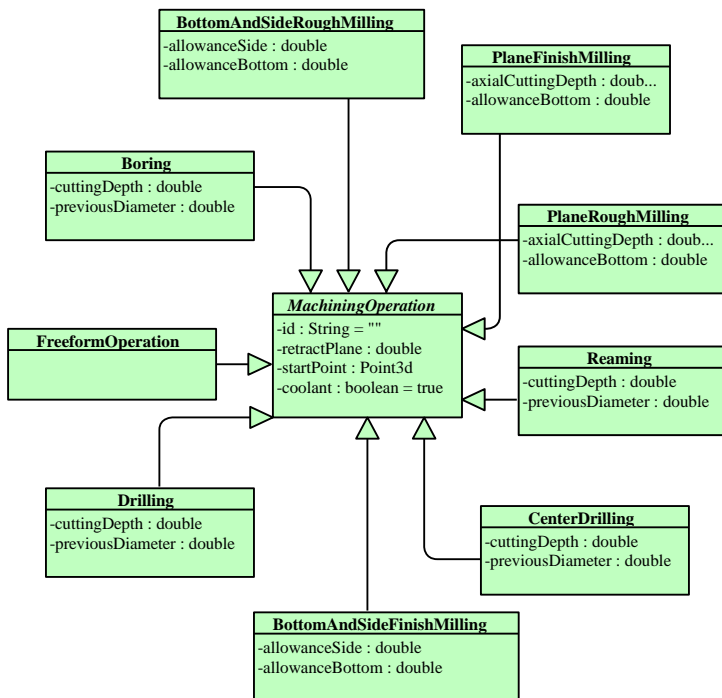


Figura 6.19. Diagrama UML das operações de usinagem.

Fonte: Criado pelo autor.

Dados relacionados à textura e às tolerâncias devem ser considerados no momento de determinar as operações de usinagem, pois sabe-se que alguns processos de usinagem podem dar melhor qualidade à superfície do que outros. Por exemplo, para o furo da Figura 6.18, poder-se-ia ainda adicionar a operação de alargamento ou de mandrilamento se as exigências de qualidade das superfícies de projeto assim o requeressem.

As

Tabela

6.1

e

Tabela 6.2 apresentam os processos de usinagem com os quais podem ser obtidas precisões desejadas(WANG e LI, 1991). Tais tabelas junto com as especificações do fabricante das ferramentas podem ser usadas para selecionar os processos a fim de obter a geometria e a qualidade de cada *feature*.

**Tabela 6.1. Precisões de usinagem e rugosidades obtidas através de diversos processos de usinagem para superfícies cilíndricas internas.**

Processo		Classe de Tolerância IT	Rugosidade Ra (mm)
Furação		11-13	5-80
Rebaixamento		10-11	1,25-20
Alargamento	Desbaste	8-9	1,25-5
	Semi-acabamento	7-8	0,63-1,25
	Acabamento	6-7	0,16-0,63
Mandrilamento	Desbaste	12-13	5-20
	Semi-acabamento	10-11	2,5-10
	Acabamento	7-9	0,63-2,5
	de Precisão	5-7	0,16-0,63
Brochamento	Semi-acabamento	9-10	0,32-2,5
	Acabamento	6-9	0,16-0,63
Retificação	Desbaste	7-9	0,63-2,5
	Semi-acabamento	6-7	0,16-0,63
	Acabamento	5-6	0,08-0,16
Brunimento	Semi-acabamento	6-7	0,16-1,25
	Acabamento	4-6	0,04-0,32
Lapidação	Semi-acabamento	5-6	0,04-0,63
	Acabamento	3-5	0,008-0,08
Super-acabamento		3-5	0,008-0,16

FONTE: WANG e LI (1991).

**Tabela 6.2. Preciões de usinagem e rugosidades obtidas através de diversos processos de usinagem para superfícies planas**

Processo		Classe de Tolerância IT	Rugosidade Ra (mm)
Fresamento	Desbaste	11-13	5-20
	Semi-acabamento	8-11	1,25-10
	Acabamento	6-8	0,32-1,25
Faceamento	Desbaste	12-13	10-80
	Semi-acabamento	10-11	2,5-10
	Acabamento	7-9	1,25-2,5
Aplainamento	Desbaste	11-13	5-20
	Semi-acabamento	8-11	2,5-10
	Acabamento	6-8	0,63-5
Brochamento	Semi-acabamento	10-11	0,63-2,5
	Acabamento	6-9	0,16-0,63
Retificação	Desbaste	7-9	0,63-2,5
	Semi-acabamento	6-7	0,16-0,63
	Acabamento	5-6	0,08-0,16
Lapidação	Semi-acabamento	5-6	0,04-0,63
	Acabamento	3-5	0,008-0,08
Super-acabamento		3-5	0,008-0,16

FONTE: WANG e LI (1991).

Em resumo, em esta atividade, cada tipo de *feature* de usinagem resulta em um conjunto de operações de usinagem.

#### (b) Determinação das ferramentas de corte

Nesta etapa, associa-se uma única ferramenta de corte a uma operação de usinagem. A Figura 6.20 mostra a modelagem da ferramenta em um diagrama UML.

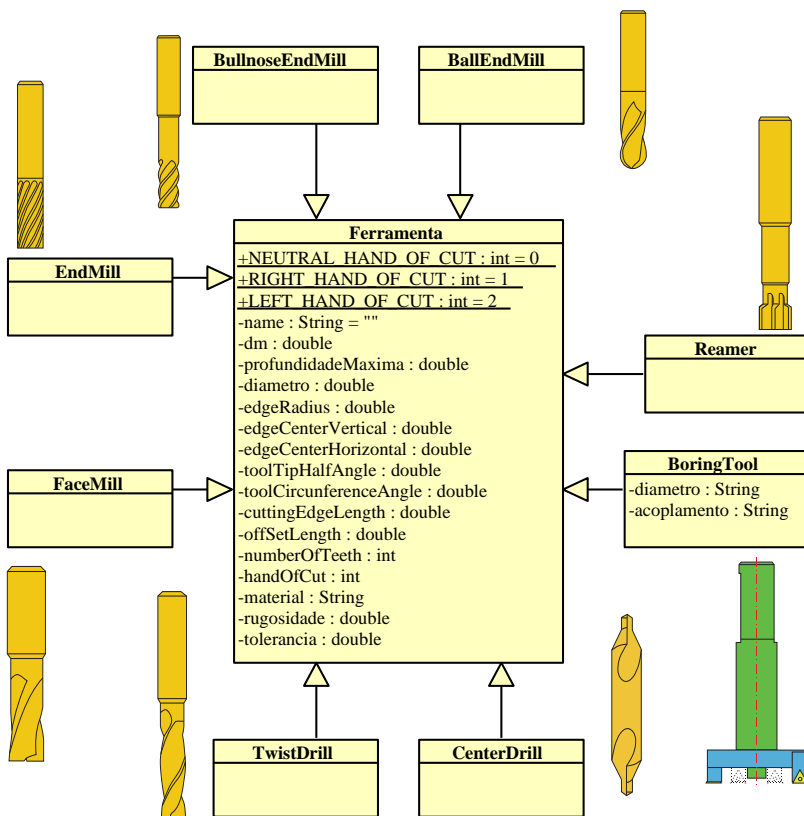


Figura 6.20. Diagrama UML das ferramentas.

Fonte: Criado pelo autor.

### (c) Determinação das condições de usinagem

Neste estágio as condições de usinagem são determinadas segundo o tipo de operação. Os fabricantes das ferramentas normalmente informam os dados de corte recomendados para se obter bons resultados. A Figura 6.21 apresenta um diagrama UML da classe condições de corte.

CondiçõesDeUsinagem
-vc : double
-vf : double
-ap : double
-ae : double
-af : double
-f : double
-n : double
+ferramenta : Ferramenta

Figura 6.21. Diagrama UML das condições de usinagem.

Fonte: Criado pelo autor.

(d) Criação dos *Machining workingsteps*.

Como as operações de usinagem, as ferramentas e as suas condições de corte associadas já foram determinadas, pode-se então criar o *machining workingstep*, os quais serão utilizados em atividades que serão descritas posteriormente.

(e) Determinação das precedências

Um *machining workingstep* deve apontar para outro *machining workingstep*, definindo-o como seu precedente. Se não houver nenhum *machining workingstep* precedente, este atributo será definido como nulo. A Figura 6.22 apresenta a peça da Figura 6.9 mapeada em vários *machining workingsteps*, e percebe-se que cada um destes *machining workingsteps* aponta para algum outro como seu precedente. A Tabela 6.3 apresenta os *machining workingsteps* e seus precedentes para a peça mapeada da Figura 6.22.

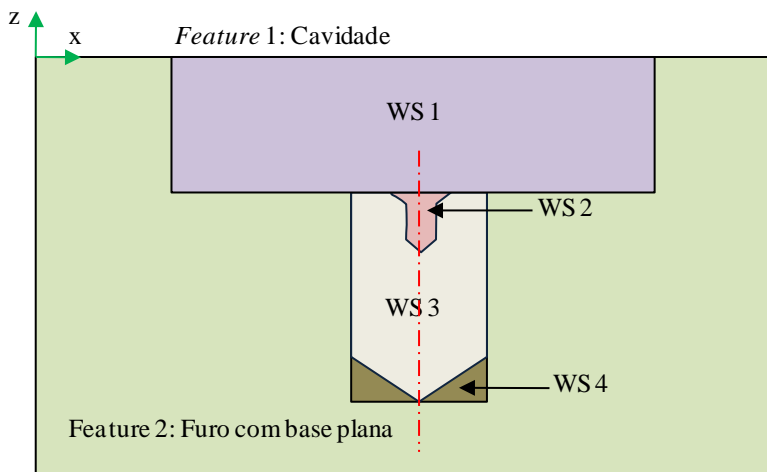


Figura 6.22. *Features* mapeadas em *machinig workingsteps*.

Fonte: Criado pelo autor.

**Tabela 6.3. *Machinig workinsteps* e seus precedentes.**

<i>Feature</i>	<i>Machinig workinsteps</i>	<b>M. WS precedente</b>
Cavidade	WS 1 (Fresamento desbaste)	Nulo
Furo com base plana	WS 2 (Furação de centro)	WS 1
	WS 3 (Furação)	WS 2
	WS 4 (Fresamento desbaste)	WS 3

FONTE: Criado pelo autor.

#### 6.2.1.2.2 *Organização dos Machinig workingsteps*

Esta atividade tem por objetivo dar suporte ao planejamento das seqüências de usinagem, ou seja, definir a ordem de execução dos *machinig workingsteps*. Para tal, usam-se algumas regras de produção:

- (a) De maneira geral, *machinig workingsteps* de desbaste devem ser os primeiros a serem executados.
- (b) De maneira específica, *machinig workingsteps* de desbaste que pertencem a *features* que têm precedência maior (vide item 6.2.1.1.3) devem ser executadas em primeiro lugar. No

exemplo da Figura 6.22, os *machining workingsteps* da cavidade deveriam ser executadas antes que os do furo.

Existem várias regras específicas para cada tipo de *feature*. A Tabela 6.4 apresenta os *machinings workingsteps* e os seus índices de preferência para um furo com base plana. O *machining workingstep* que tem índice de preferência 1 deveria ser executado em primeiro lugar.

**Tabela 6.4. Precedências das operações de usinagem para um furo com base plana.**

<i>Feature</i>	<i>Machining workingsteps</i> necessários	Índice de preferência
Furo com base plana	Furação de centro	1
	Furação	2
	Fresamento lateral e de base desbaste	3
	Alargamento (opcional, depende dos requisitos de projeto)	4
	Mandrilamento (opcional, depende dos requisitos de projeto)	5
	Fresamento de base acabamento (opcional, depende dos requisitos de projeto)	6

FONTE: Criado pelo autor.

#### 6.2.1.2.3 Determinação dos pontos de fixação

Esta atividade tem por finalidade assistir na determinação dos lugares da peça onde podem ser colocados os apoios, de maneira a evitar colisão de ferramentas com tais pontos. A Figura 6.23 mostra o diagrama IDEF0 desta atividade.

##### (a) Extração das *features*

Esta atividade deve mapear o projeto e extrair do mesmo, as *features* presentes na peça.



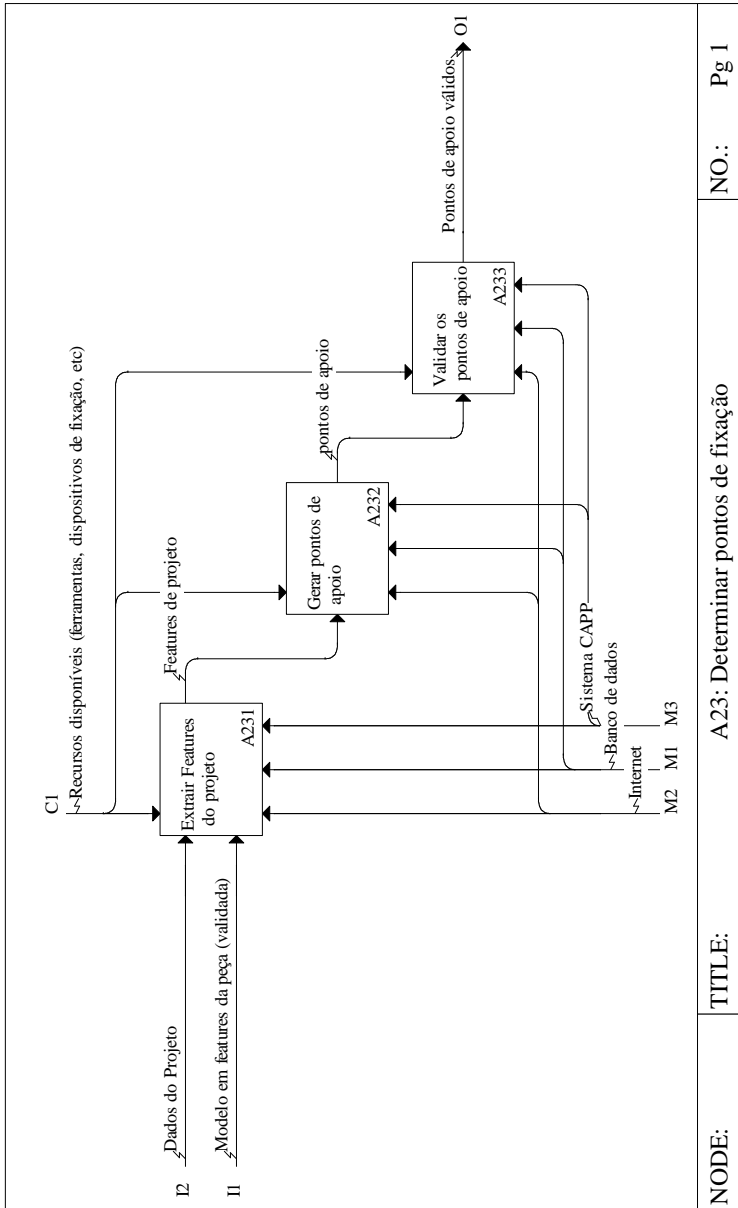


Figura 6.23. Diagrama IDEF0 do determinador de pontos de apoio.  
 Fonte: Criado pelo autor.

### (b) Geração dos pontos de apoio

Nesta atividade são gerados os pontos de apoio de maneira a dar estabilidade à peça na base.

### (c) Validação

Os pontos de apoio gerados devem ser validados seguindo a seguinte condição: *Features* que são passantes (furos e cavidades passantes) devem ser consideradas, pois os apoios não deveriam ser colocados na mesma posição de tais *features* para evitar que a ferramenta colida com a fixação.

#### 6.2.1.2.4 Geração do plano de trabalho

Uma vez determinados os *machining workingsteps*, as suas sequências de execução e os pontos de fixação da peça, pode se gerar um plano de trabalho (*workplan*). Dados acerca do cliente remoto, acerca da peça, material, etc. também são usados nesta atividade.

#### 6.2.1.2.5 Geração do arquivo neutro STEP-NC

O objetivo desta atividade consiste em obter um arquivo físico STEP-NC que contenha as entidades que representam o modelo de dados criado usando classes na linguagem Java. O modelo de dados fornecido pela norma ISO 14649 é definido usando-se a linguagem EXPRESS. Portanto é necessário fazer uma ponte entre EXPRESS e a linguagem de programação Java.

A parte 22 da norma ISO 10303 define a interface padrão de acesso de dados (ISO 10303-22, 1998), enquanto a parte 27 da mesma norma (ISO 10303-27, 2000) define a interface de ligação com a linguagem Java.

SDAI (*Standard Data Access Interface*) fornece uma interface de programação da aplicação para dados descritos pelo modelo de informação EXPRESS (NIST, 1999). SDAI tem por objetivo acessar e manipular instâncias de dados criadas de acordo a um esquema EXPRESS.

Conceitualmente, SDAI consiste de duas partes: os esquemas SDAI e o mapeamento do EXPRESS para os construtores da linguagem desejada. Os esquemas SDAI especificam os construtores necessários

para o funcionamento independente dos dados que estão sendo acessados.

Um modelo de informação em EXPRESS representa SDAI de duas maneiras diferentes:

- Pode ser representado através do dicionário de dados e,
- Pode ser representado diretamente na estrutura de dados disponível na linguagem de programação.

O dicionário captura o núcleo da estrutura de dados do esquema em EXPRESS junto com um número limitado de restrições explícitas. O mapeamento de EXPRESS para uma linguagem de programação particular depende da própria linguagem de programação.

A Figura 6.24 apresenta a arquitetura SDAI, onde são mostrados os seus principais componentes, que são os seguintes:

- Sessão SDAI
- Dicionário SDAI
- Repositório SDAI
- Modelo SDAI

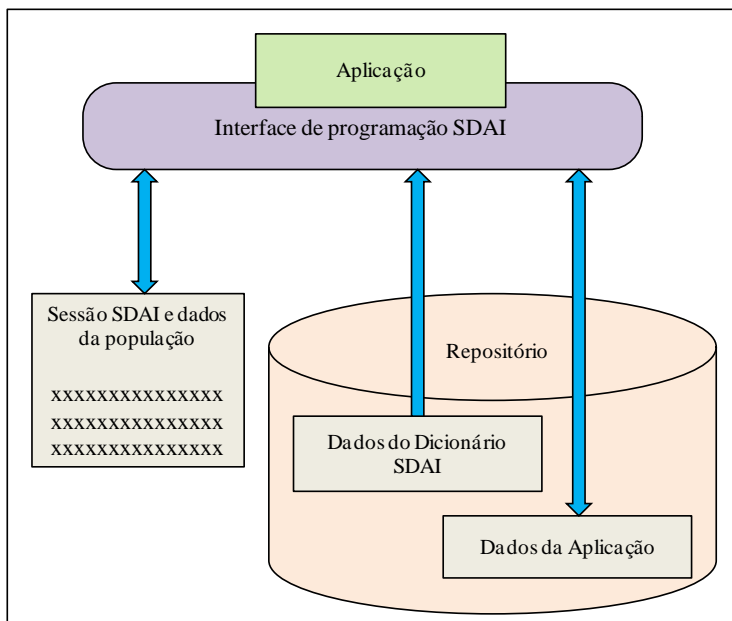


Figura 6.24. Arquitetura SDAI.  
Fonte: Adaptado de NIST (1999).

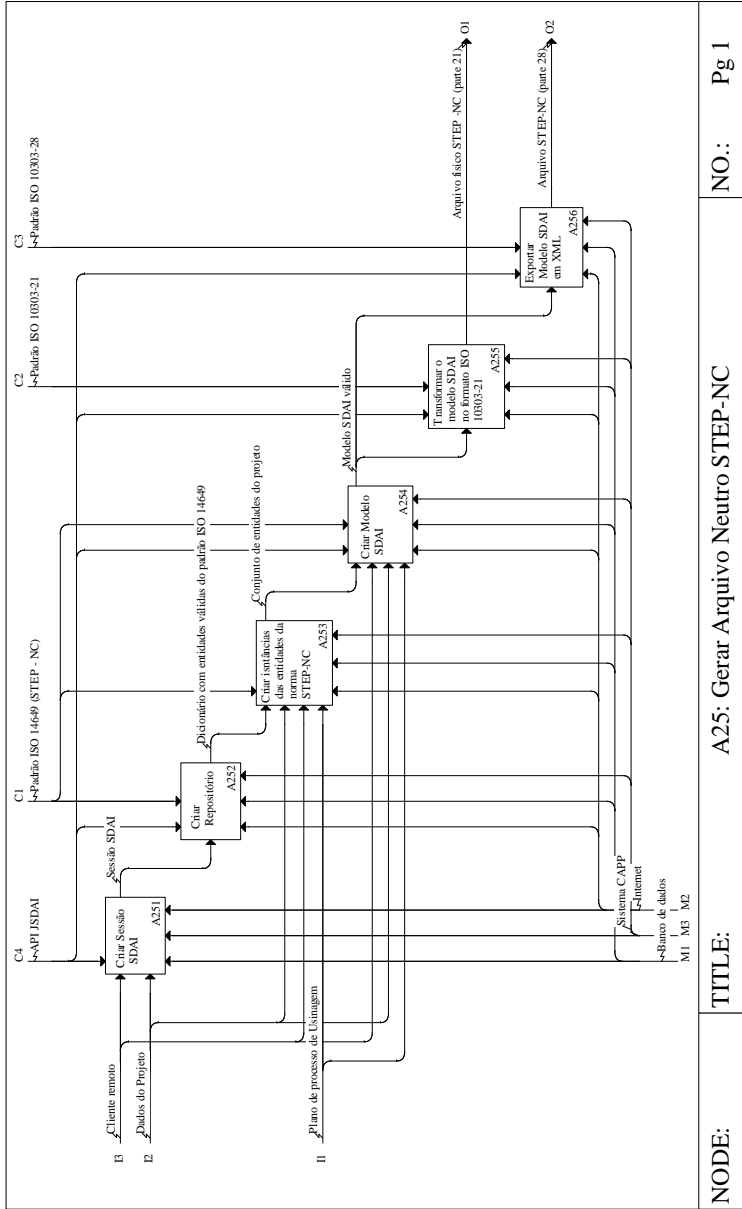
Para atingir o objetivo desta atividade, devem ser realizadas as seguintes tarefas, as quais são mostradas no diagrama IDEF0 na Figura 6.25:

- Criação de uma sessão SDAI;
- Criação do repositório;
- Criação das entidades da norma STEP-NC;
- Criação do modelo SDAI;
- Exportar modelo SDAI em ISO 10303-21 e
- Exportar modelo SDAI em XML.

Além disto, a biblioteca JSDAI<sup>3</sup> será usada nesta atividade para ajudar na ligação entre as linguagens EXPRESS e Java.

---

<sup>3</sup> JSDAI ([www.jsdai.net](http://www.jsdai.net)) é uma biblioteca em Java que é útil para a manipulação de dados orientados a objetos definidos por um modelo de dados baseado em EXPRESS.



NODE: A25: Gerar Arquivo Neutro STEP-NC NO.: Pg 1

Figura 6.25. Diagrama IDEF0 da atividade que gera o arquivo neutro.  
 Fonte: Criado pelo autor.

(a) Criação da sessão SDAI.

Nesta atividade cria-se a sessão SDAI, a qual é usada para controlar o acesso ao ambiente SDAI e às transações de leitura/escrita de um único usuário. O acesso é feito apenas através de uma sessão SDAI. É permitida somente uma sessão por vez, ou seja, só um usuário por vez poderá acessar ao repositório.

(b) Criação do repositório

Uma vez criada a sessão SDAI, é necessário criar o repositório SDAI, o qual é o depósito físico onde se armazenam os modelos SDAI e instâncias dos esquemas. Este repositório pode ser criado em uma base de dados remota ou no computador do próprio cliente remoto.

Um dicionário contendo todas as instâncias válidas do esquema EXPRESS da norma ISO 14649 também é criado e gravado fisicamente no repositório.

(c) Criação das instâncias das entidades segundo a norma ISO 14649

Nesta atividade, o modelo de dados da peça criado usando-se classes e objetos Java deve ser representado por meio de instâncias válidas da norma ISO 14649.

(d) Criação do modelo SDAI

Nesta atividade, o modelo SDAI é criado a partir do modelo computacional em Java. O modelo SDAI é uma subdivisão do repositório e contém instâncias das entidades de acordo com um esquema particular em EXPRESS. No caso particular desta atividade, este modelo SDAI deverá conter instâncias válidas do MACHINING\_SCHEMA (ISO 14649-10), do MILLING\_SCHEMA (ISO 14649-11), e do MILLING\_TOOL\_SCHEMA (ISO 14649-111) que representem o modelo computacional de uma peça com suas *features*, suas informações de manufatura, e informações adicionais.

### (e) Mapeamento do modelo SDAI para ISO 10303-21 e XML

Cada instância das entidades presentes no modelo SDAI, com ajuda da biblioteca JSDAI, é mapeada e transformada no formato ISO 10303 – 21, e em XML.

#### 6.2.1.3 Atividade CAM

A proposta do padrão STEP-NC é que controladores inteligentes leiam o arquivo físico, interpretem os dados, e executem a usinagem. Porém, ainda existem em grande número os controladores convencionais, os quais suportam apenas instruções em formato ISO 6983 (também conhecido como código G). Portanto será necessário o pós-processamento do arquivo físico para um controlador específico.

O arquivo neutro deve ser lido e interpretado de maneira a possibilitar a fabricação da peça modelada. Para tal, a biblioteca JSDAI será usada, pois ela possibilita a leitura dos parâmetros solicitados a partir de um arquivo neutro em formato ISO 10303 – 21 ou em XML.

A Figura 6.26 apresenta em um diagrama IDEF0 a atividade CAM, onde também podem ser vistas as tarefas que devem ser efetuadas:

- Leitura do arquivo físico;
- Pós-processamento para ISO 6983 e
- Simulação da movimentação

A seguir se verá em detalhes cada uma dessas tarefas.

##### 6.2.1.3.1 *Leitura do arquivo neutro*

O objetivo desta atividade é extrair os parâmetros desejados do arquivo neutro e criar objetos Java que representem o modelo descrito no arquivo neutro. A Figura 6.27 mostra em um diagrama IDEF0 as sub-atividades para a realização da leitura do arquivo.

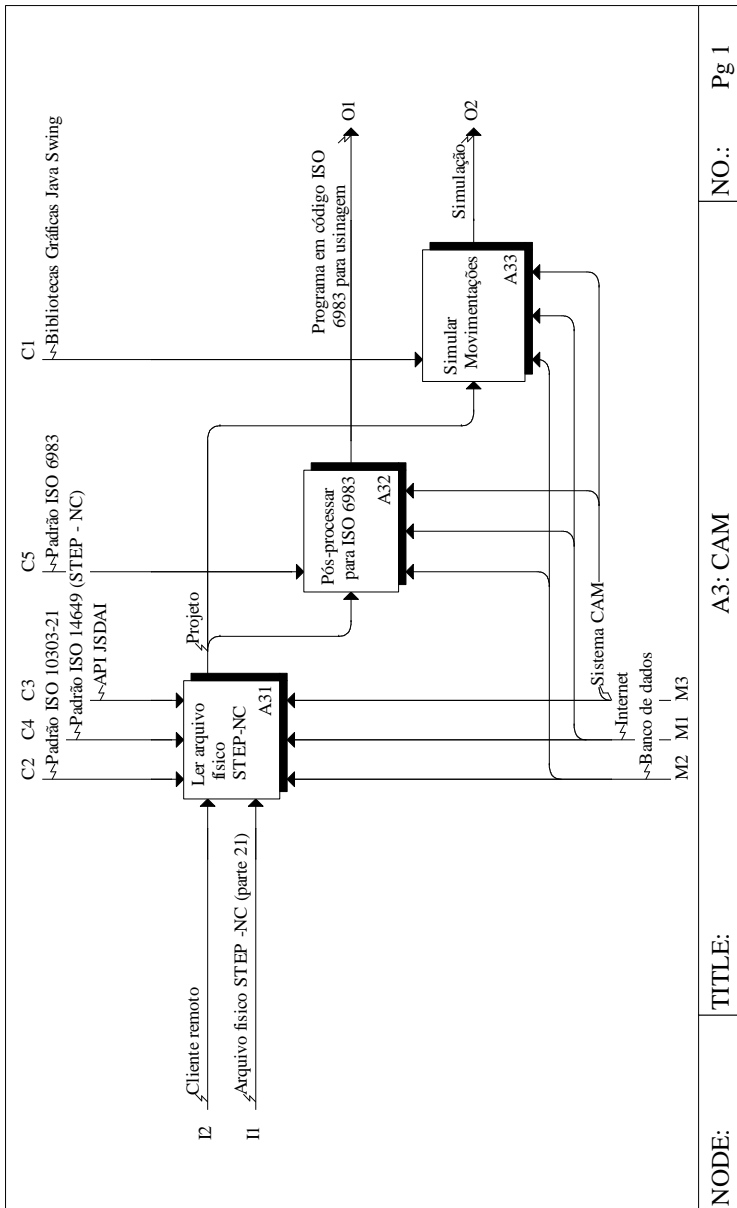


Figura 6.26. Diagrama IDEF0 da atividade CAM.  
 Fonte: Criado pelo autor.



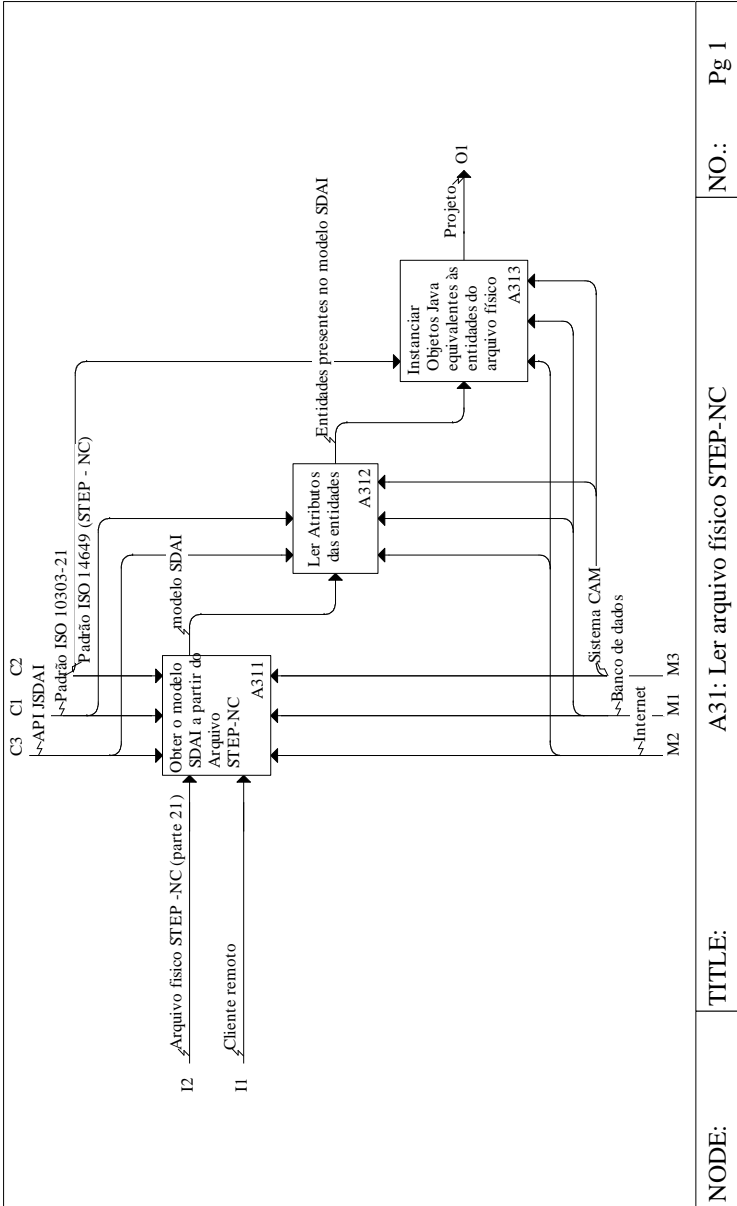


Figura 6.27. Diagrama IDEF0 da atividade de leitura de arquivo físico.  
 Fonte: Criado pelo autor.

(a) Obtenção do modelo SDAI

O modelo SDAI deve ser reconstruído a partir do arquivo físico. Isto é feito com ajuda do dicionário SDAI, onde estão cadastradas todas as entidades válidas que pertencem aos esquemas usados (MACHINING\_SCHEMA, MILLING\_SCHEMA e MILLING\_TOOL\_SCHEMA).

(b) Leitura dos atributos

Com ajuda da API JSDAI, podem ser obtidos os parâmetros das entidades desejadas, e tais dados serão usados para criar objetos Java.

(c) Criação dos objetos Java que representam as entidades do arquivo físico

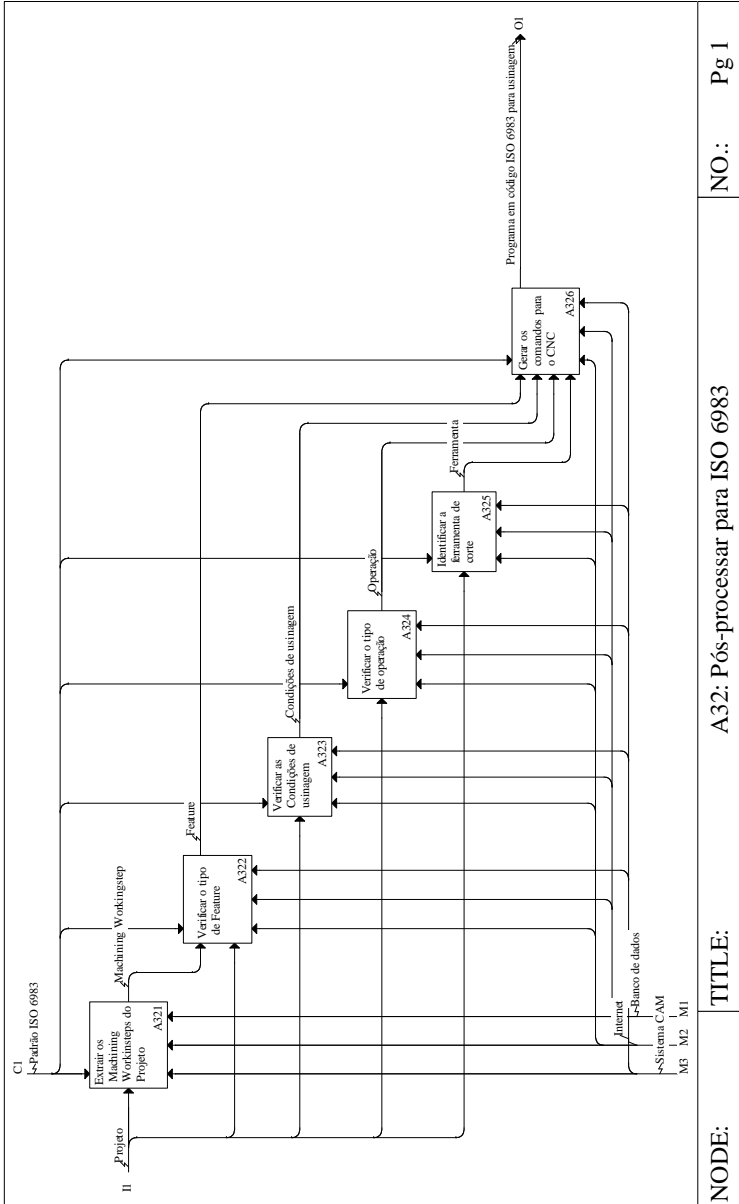
Os parâmetros obtidos na atividade anterior são usados aqui para instanciar objetos Java, os quais são equivalentes às entidades presentes no arquivo físico. Cada entidade presente no modelo SDAI é mapeada para criar objetos Java.

### 6.2.1.3.2 Geração do arquivo em formato ISO 6983

Esta atividade visa a obtenção de um arquivo que possa ser entendido por um controlador CNC convencional. Portanto, cada um dos objetos Java deverá ser mapeado em linhas de instruções a serem executadas. A Figura 6.28 apresenta um diagrama IDEF0, onde podem-se ver as suas sub-atividades.

(a) Extração dos *Machining Workingsteps*

Nesta atividade, são extraídos os *machining workingsteps* pertencentes à peça projetada e arranjadas em vetores.



NO.: Pg 1  
 TITLE: A32: Pós-processar para ISO 6983  
 NODE:

Figura 6.28. Diagrama IDEFO da atividade pós-processar para ISO 6983. Fonte: Criado pelo autor.

(b) Extração das *features*

Uma vez que cada *machining workingstep* aponta para uma *feature* (ver Figura 6.16), tal *feature* pode ser extraída. A obtenção da *feature* é útil para saber a forma final que deve ser alcançada através dos processos de usinagem, uma vez que vários *machining workingsteps* podem remover diversos volumes da mesma *feature*.

(c) Extração das condições de corte

Nesta sub-atividade são extraídos os parâmetros de usinagem a partir do *machining workingstep*. As condições de corte são importantes para a construção das linhas de comando do código G (ver item 2.2.2.1 e Figura 2.13).

(d) Extração da operação de usinagem

Esta informação é útil para ajustar as funções miscelâneas do comando NC (ligar ou desligar fluido de corte, etc.).

(e) Identificação da ferramenta de corte

A informação acerca da ferramenta de corte é essencial para corrigir ou compensar as trajetórias (normalmente geradas em relação ao centro da ferramenta, sem considerar o diâmetro da mesma).

(f) Geração dos comandos em formato ISO 6983

No arquivo neutro gerado no âmbito do STEP-NC, não é obrigatório que as trajetórias das ferramentas sejam especificadas, pois esta geração ficaria por conta do controlador inteligente. Porém, no caso de controladores convencionais, é necessária a geração das trajetórias. A Figura 6.29 apresenta a modelagem das trajetórias em UML.

As informações das trajetórias, em conjunto com as informações coletadas nas atividades anteriores, são usadas nesta atividade para gerar as linhas de comando em formato ISO 6983.

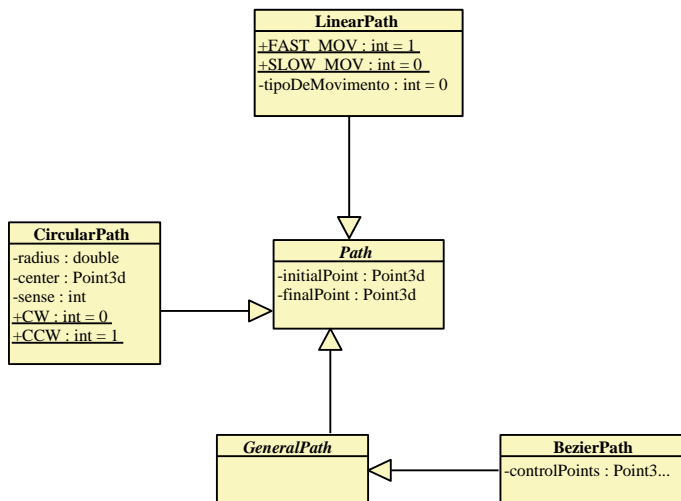


Figura 6.29. Modelagem das trajetórias em UML.

Fonte: Criado pelo autor.

### 6.2.1.3.3 Simulação de movimentações da ferramenta

Os dados do arquivo físico são lidos por meio da atividade de leitura, descrita no item 6.2.1.3.1, e transformados em objetos Java. Estes objetos Java por sua vez devem ser traduzidos em vetores de movimentações. A Figura 6.30 apresenta um diagrama IDEF0 das tarefas que devem ser realizadas nesta atividade.

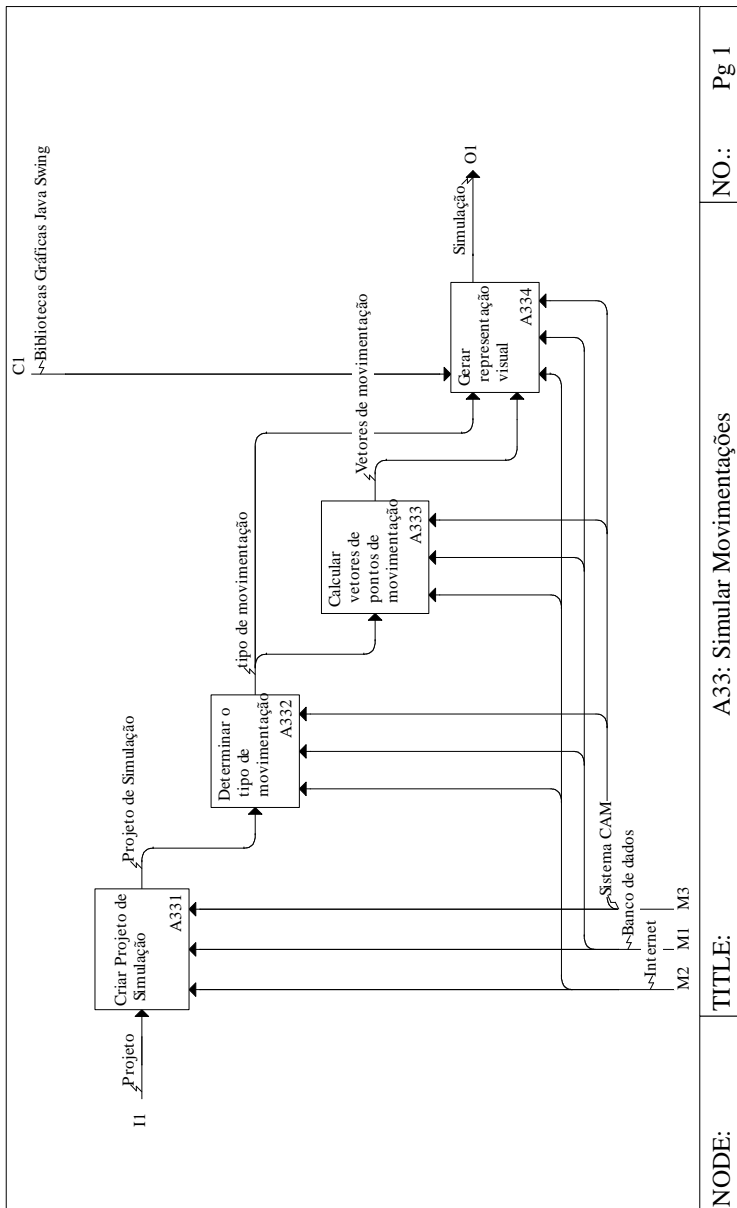


Figura 6.30. Diagrama IDEFO das atividades de simulação.  
 Fonte: Criado pelo autor.

## (a) Criação do projeto de simulação

Um projeto de simulação contém todos os dados necessários para desenhar as movimentações, que incluem o bloco (matéria-prima) da peça, as *features* e os *machining workingsteps*. A Figura 6.31 mostra um diagrama UML do projeto de simulação.

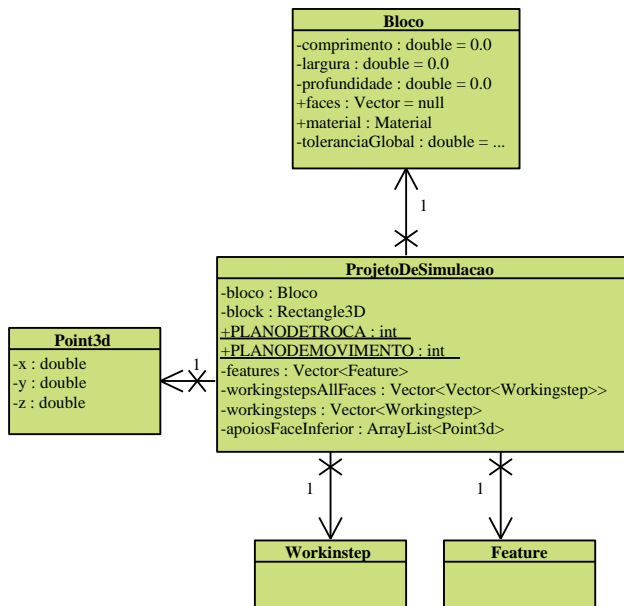


Figura 6.31. Diagrama UML do projeto de simulação.  
Fonte: Criado pelo autor.

## (b) Determinação do tipo de movimentação

Segundo o tipo de operação, e dependendo do tipo de *feature*, as movimentações serão diferentes. Por exemplo, a operação de desbaste para uma ranhura deveria ter uma movimentação em zig-zag, enquanto uma operação de furação de centro é feita por meio de uma movimentação linear desde um ponto no plano de segurança até a profundidade especificada na operação. A Figura 6.32 apresenta a modelagem das movimentações em um diagrama UML, e a Figura 6.33 mostra os tipos de movimentações.

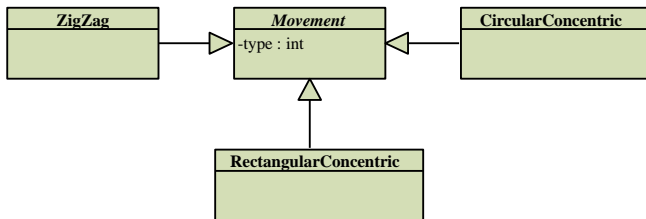


Figura 6.32. Modelagem das movimentações em UML.

Fonte: Criado pelo autor.

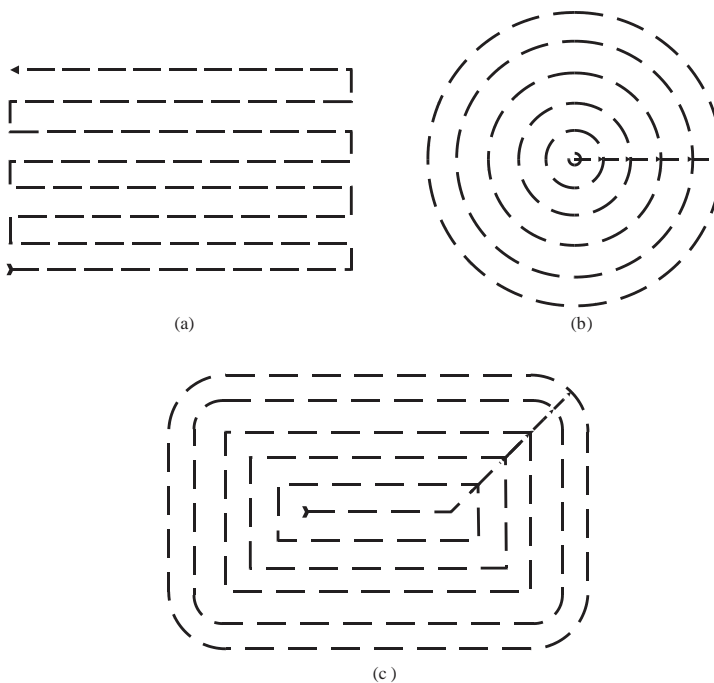


Figura 6.33. Tipos de movimentações: (a) Zig-Zag. (b) Circular concêntrica. (c) Retangular concêntrica.

Fonte: Criado pelo autor.

### (c) Cálculo dos vetores de movimentação

Nesta atividade, os pontos por onde a ferramenta deverá passar são calculados e armazenados em vetores, e cada *machining workingstep* deverá ter um vetor de movimentação (ver Figura 6.16).



## (d) Geração da representação visual

Cada ponto no vetor de movimentação do *machining workingstep* deve ser desenhado, uma animação da ferramenta deverá aparecer em duas vistas. Para isto, desenhadores de vista lateral (*LateralToolPanel*), de vista frontal (*ToolPanel*) devem mostrar ao usuário as trajetórias. A Figura 6.34 apresenta um diagrama UML do gerador de representação visual, onde “*SimulationPanel*” controla os eventos de início da simulação (*StartAction*), pausa (*StopAction*), e velocidade da mesma (*SpeedAction*). A classe “*ProjetoDeSimulacao*” contém as informações necessárias para realizar a simulação (vetores de movimentação, tamanho das ferramentas, tamanho do bloco, etc.).

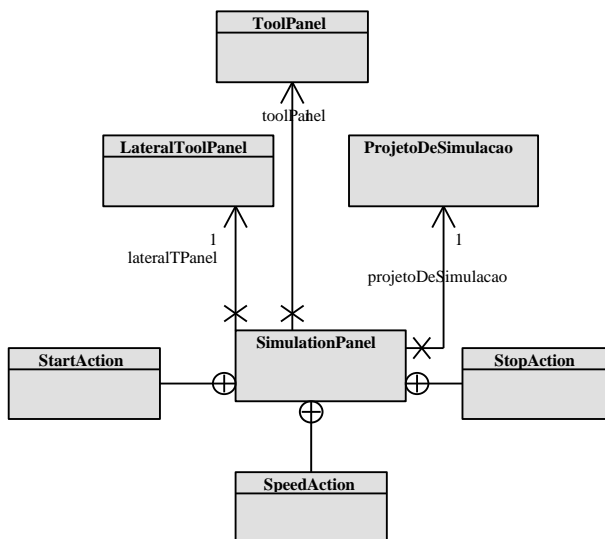


Figura 6.34. Modelagem da simulação em UML.

Fonte: Criado pelo autor.

## 6.2.2 Arquitetura do sistema proposto

A arquitetura do sistema usará o modelo cliente-servidor descrito na seção 4.5.2, onde o próprio aplicativo às vezes funciona como cliente fazendo requisição dos dados acerca das condições de usinagem das ferramentas, ou de ferramentas cadastradas no banco de dados do servidor. O cliente remoto poderá enviar um pedido para fabricação, ou

apenas salvar o projeto no servidor. A Figura 6.35 apresenta a arquitetura do sistema.

A implementação do protótipo do sistema CAD/CAPP/CAM para o projeto, planejamento do processo e fabricação de peças via internet, aderente a STEP-NC, será descrito no próximo capítulo.

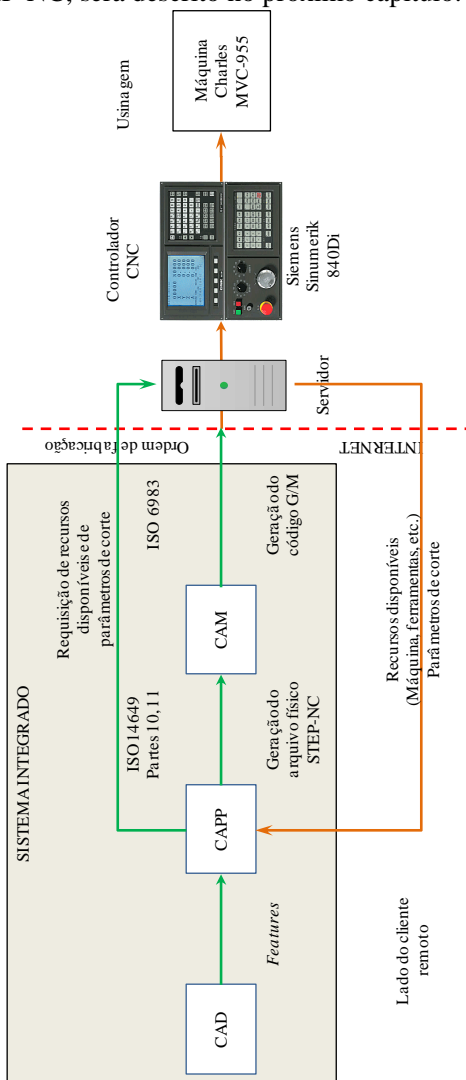


Figura 6.35. Arquitetura do protótipo proposto.

Fonte: Criado pelo autor.

## 7 IMPLEMENTAÇÃO DO PROTÓTIPO DO SISTEMA CAD/CAPP/CAM

Este capítulo apresentará a implementação do protótipo proposto e modelado no capítulo anterior. Serão apresentadas algumas janelas capturadas da execução do protótipo para demonstrar a funcionalidade do mesmo.

### 7.1 IMPLEMENTAÇÃO DO SISTEMA CAD

Os dados necessários para a representação do modelo gráfico 2D e do modelo sólido 3D vêm a partir da criação de um novo projeto. Para o sistema CAD são necessárias apenas as dimensões da peça bruta. Outros dados como material da peça, tolerâncias, dados do usuário, etc. são úteis em outros módulos.

A Figura 7.1 apresenta a janela onde são capturados os dados iniciais para a criação de um novo projeto.

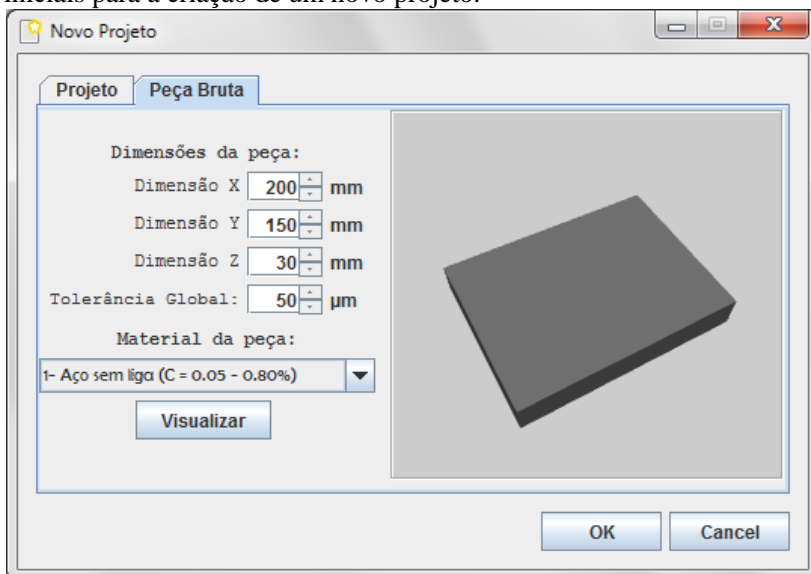


Figura 7.1. Criação de um novo projeto.

Fonte: Criado pelo autor.

A Figura 7.2 mostra a janela principal do programa, a qual está dividida em três campos principais:

- No lado esquerdo são mostradas as informações de dados de projeto (nome, dados da peça bruta), as *features* presentes na peça, os *machining workingsteps* e o plano de trabalho para a usinagem da peça (*workplan*).
- No lado direito é mostrada a visualização da peça em 2D,
- O lado inferior contém um campo onde o programa mostra mensagens que informam ao usuário o status do programa.

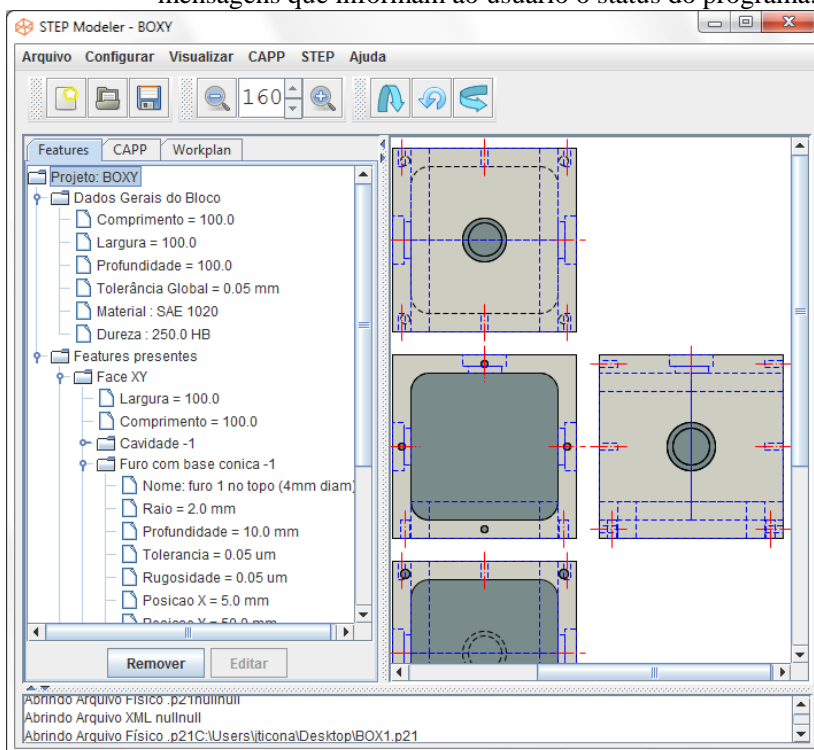


Figura 7.2. Janela principal do aplicativo.

Fonte: Criado pelo autor.

### 7.1.1 Implementação dos modelos sólidos 3D

Conforme mencionado no item 6.2.1.1.5, o modelo sólido é criado por uma abordagem híbrida, onde as *features* são criadas usando-se a técnica B-rep, enquanto o modelo final é representado usando-se CSG.

A classe Shape3D da API Java 3D fornece métodos que são capazes de criar superfícies usando-se triângulos ou quadriláteros. Neste trabalho um sólido é constituído por uma malha de triângulos, e a Figura 7.3 mostra alguns sólidos que representam as *features* implementadas no presente trabalho de tese. Ou seja, estes sólidos representam os volumes que deverão ser removidos mediante processos de usinagem.

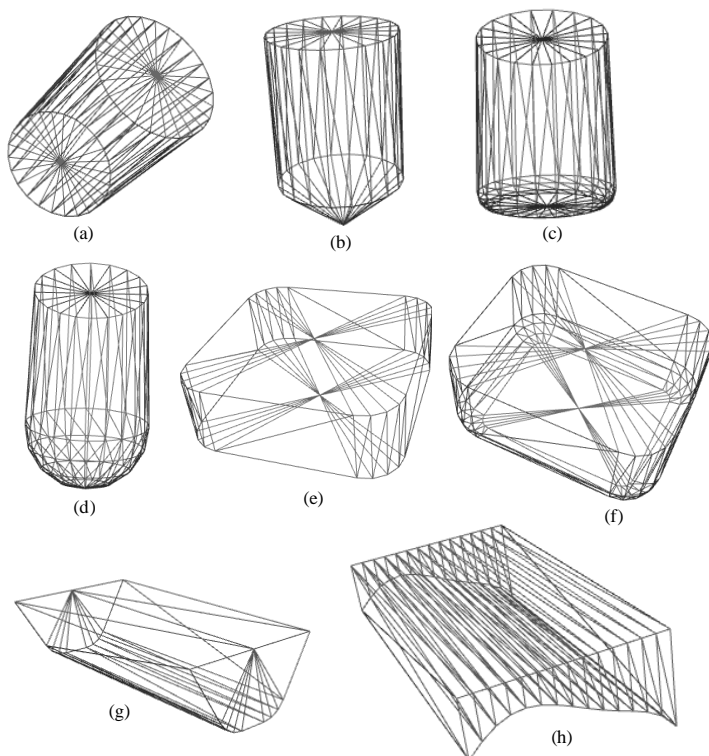


Figura 7.3. Sólidos que representam as *features*. (a), (b), (c) e (d), diferentes tipos de furos. (e) e (f), diferentes tipos de cavidades retangulares. (g) ranhura com perfil Quadrado U. (h) perfil genérico (curva de Bèzier).

Fonte: Criado pelo autor.

Existe uma classe geradora de sólido B-rep para cada *feature*. Estas classes têm três funções básicas:

- Calcular as coordenadas dos vértices que formarão as faces do sólido;
- Determinar quais pares de vértices formam uma aresta, e
- Determinar quais três arestas formam uma face.

Desta maneira, várias faces (triângulos) são formadas para constituir o sólido da *feature*. A Figura 7.4 apresenta um diagrama UML da classe responsável pela criação do sólido B-rep da *feature* cavidade retangular, onde se vê que em função dos parâmetros dimensionais da *feature* (comprimento, largura, raio de concordância, etc.) são criados os vetores (*arrays*) necessários para a criação do sólido.

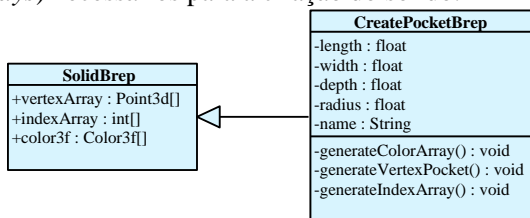


Figura 7.4. Classe responsável pela criação da cavidade retangular em B-rep.  
Fonte: Criado pelo autor.

O modelo sólido final é formado mediante operações booleanas de subtração do modelo da peça bruta e do modelo da *feature*. O resultado pode ver-se na Figura 7.5 e na Figura 7.6.

### 7.1.2 Interfaces gráficas de captura de dados

Os parâmetros das *features* são capturados através de interfaces especializadas para cada tipo de *feature*. A Figura 7.7 apresenta a interface de captura de dados da *feature* “furo” e a Figura 7.8 a interface da ranhura.

As interfaces têm duas tarefas básicas:

- Capturar e armazenar os dados e
- Verificar que os dados sejam consistentes

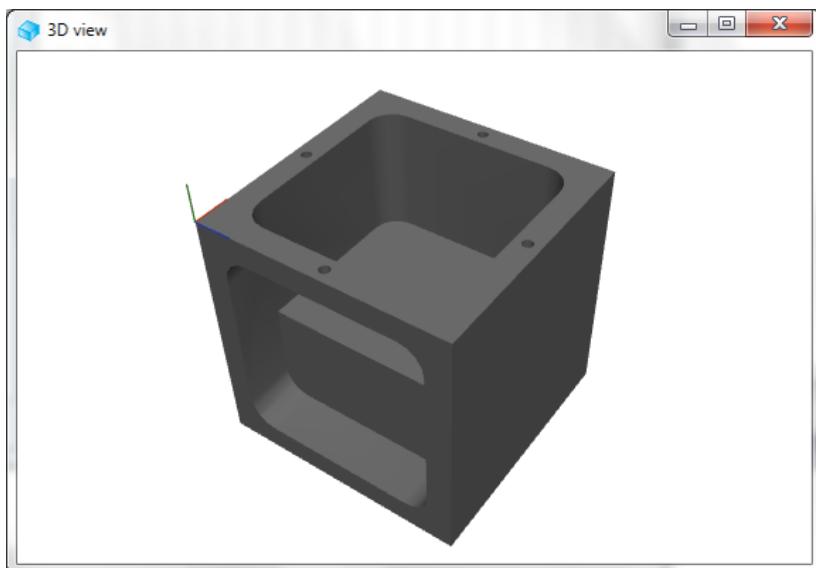


Figura 7.5. Sólido em estágio de criação.  
Fonte: Criado pelo autor.

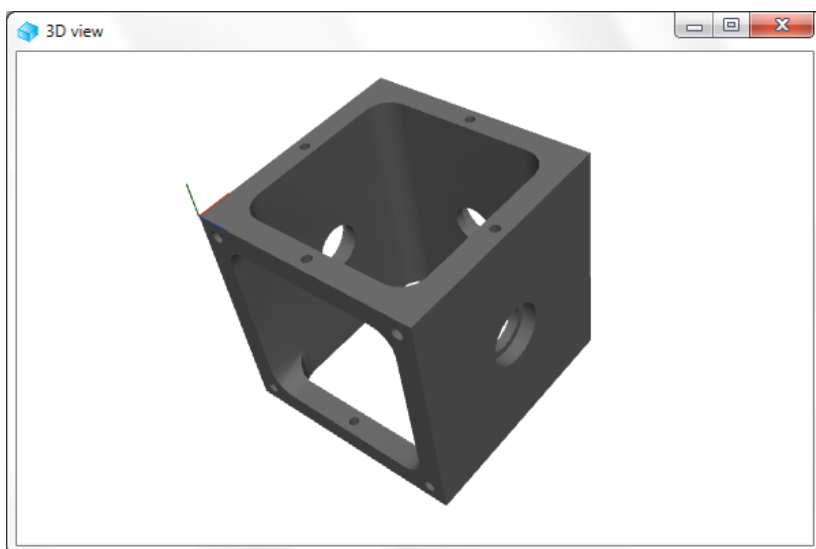


Figura 7.6. Sólido que representa a peça acabada.  
Fonte: Criado pelo autor.

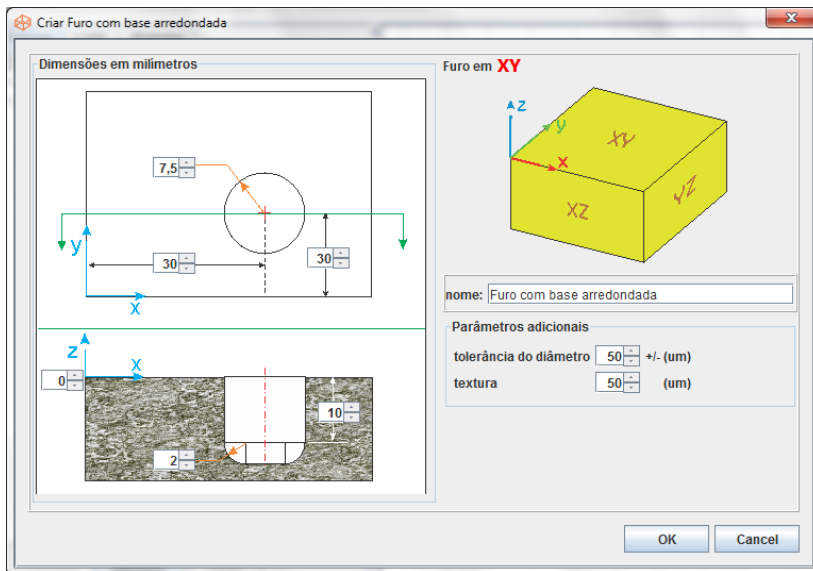


Figura 7.7. Interface de captura de dados do furo.

Fonte: Criado pelo autor.

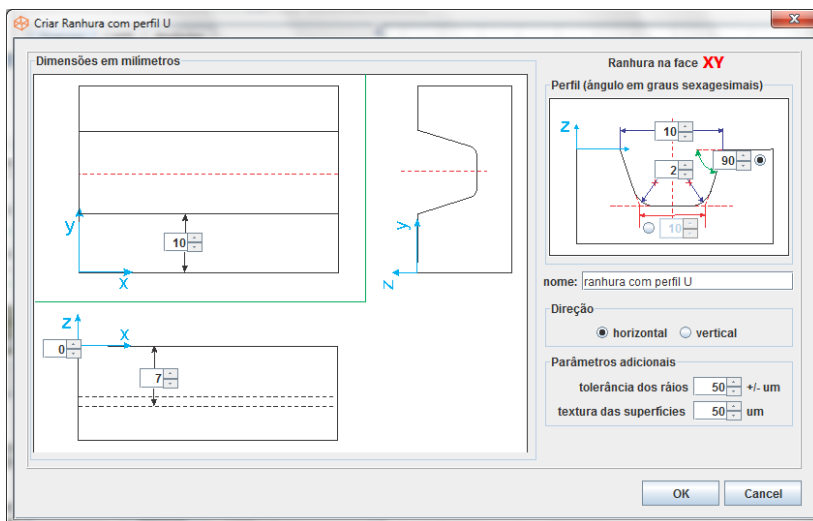


Figura 7.8. Interface de captura de dados da ranhura.

Fonte: Criado pelo autor.



Estas interfaces estão divididas em três campos principais: no lado esquerdo é mostrada uma representação 2D onde se pode inserir os parâmetros (por exemplo, localização do plano de corte para visualização da peça seccionada em vista lateral), à direita na parte superior mostra-se a face na qual a *feature* será aplicada, à direita na parte inferior são especificados dados adicionais (por exemplo, nome da *feature*, tolerâncias, textura das superfícies).

Qualquer inconsistência na introdução dos parâmetros será notificada, não sendo possível a criação da nova *feature*. Por exemplo, a criação de uma cavidade retangular com raio de concordância maior do que duas vezes o menor lado (ver Figura 7.9), resultaria em um sólido inválido. A Figura 7.10 apresenta a mensagem de erro indicando a faixa de valores nos quais o raio de concordância da cavidade é válido.

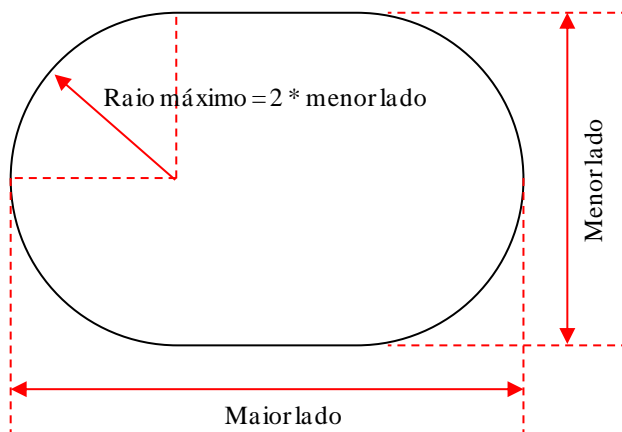


Figura 7.9. Raio máximo para uma cavidade retangular.  
Fonte: Criado pelo autor.

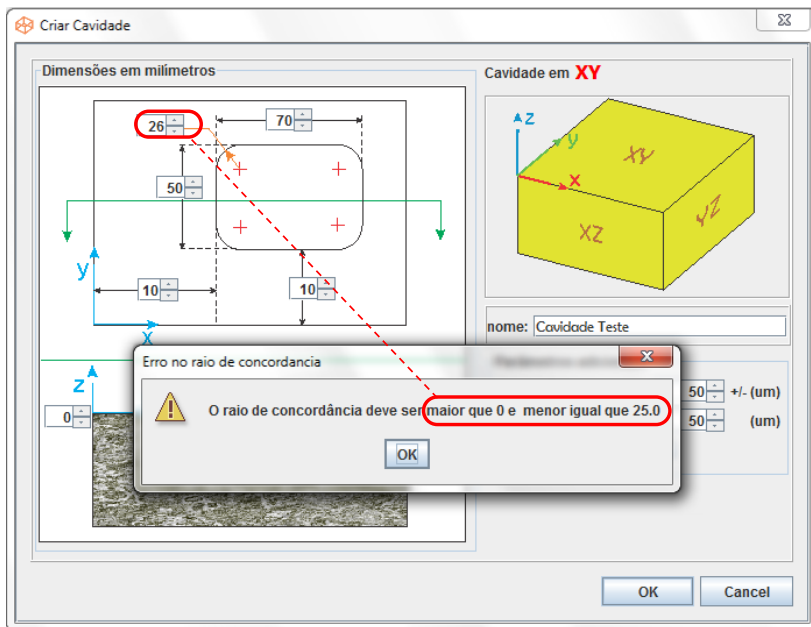


Figura 7.10. Mensagem de erro devido à inconsistência na criação de uma cavidade retangular.

Fonte: Criado pelo autor.

## 7.2 IMPLEMENTAÇÃO DO SISTEMA CAPP

As informações (*features* e à peça bruta, dados do usuário) foram capturadas e armazenadas na estrutura de dados aderente à norma STEP-NC pelo módulo CAD. Estas informações estão disponíveis e podem ser acessadas pelos demais módulos. O sistema CAPP adicionará informações relacionadas ao processo de manufatura.

### 7.2.1 Definição das ferramentas do projeto

Um conjunto de ferramentas deverá ser definido antes de criar-se o plano do processo. Tal grupo de ferramentas é associado ao projeto e pode ser armazenado para ser usado em projetos posteriores que o usuário venha a realizar.

As ferramentas podem ser criadas pelo próprio usuário (supondo que ele conta com um conjunto particular de ferramentas, Figura 7.11), ou podem ser carregadas a partir de uma base de ferramentas no servidor (Figura 7.13).

O sistema é capaz de identificar quais ferramentas do catálogo serão necessárias para a usinagem da peça modelada, e pode adicioná-las automaticamente ao projeto (ver Figura 7.13, botão “*add automatically*”).

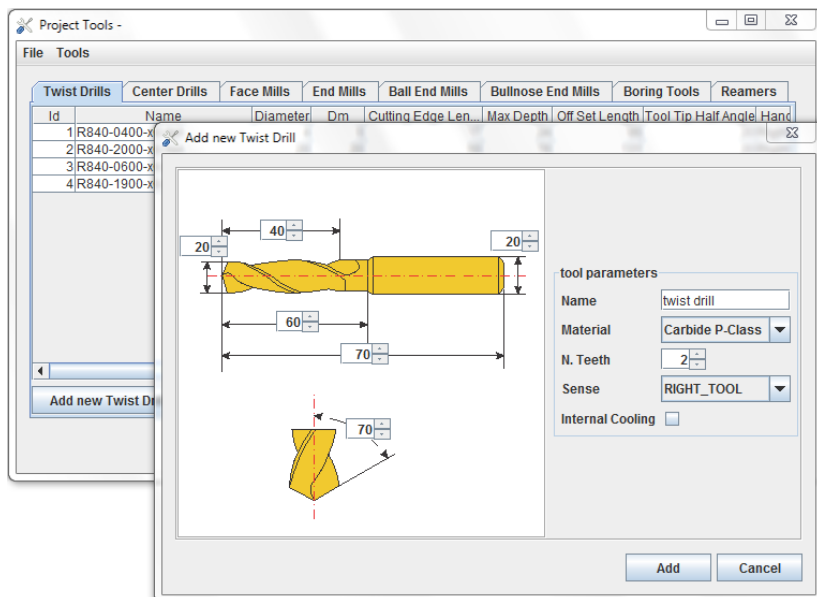


Figura 7.11. Nova ferramenta criada pelo usuário.

Fonte: Criado pelo autor.

A identificação de tais ferramentas é feita da seguinte maneira:

- (a) É provável que mais de uma ferramenta no catálogo possa remover o material de uma *feature*, portanto, do grupo de todas as ferramentas, são selecionadas apenas ferramentas que são capazes de usinar toda a *feature*, ou parte dela. Este grupo de ferramentas chama-se de “ferramentas candidatas” e salvam-se temporariamente em um vetor.
- (b) Do grupo das “ferramentas candidatas”, a que tenha maior diâmetro será escolhida, pois esta efetuará a remoção de material em menor tempo.

- (c) Se ainda houver algum material que não possa remover a primeira ferramenta, o programa novamente procura no vetor de ferramentas candidatas uma ferramenta que consiga remover tal material.
- (d) Os procedimentos descritos em (a), (b) e (c) são repetidos para cada *feature* presente na peça.

A modo de exemplo, considere-se a cavidade com fundo plano da Figura 7.12. As fresas (*face mills*) disponíveis no catálogo (Figura 7.13) possuem diâmetros desde 2mm até 20 mm. Nem todas as fresas possuem profundidade máxima que possa alcançar até 40 mm de profundidade (ver Figura 7.13), portanto estas ferramentas são descartadas para a usinagem desta cavidade. O grupo de “candidatas” estará formado pelas fresas com ID maior ou igual a 26 na tabela de fresas (Figura 7.13).

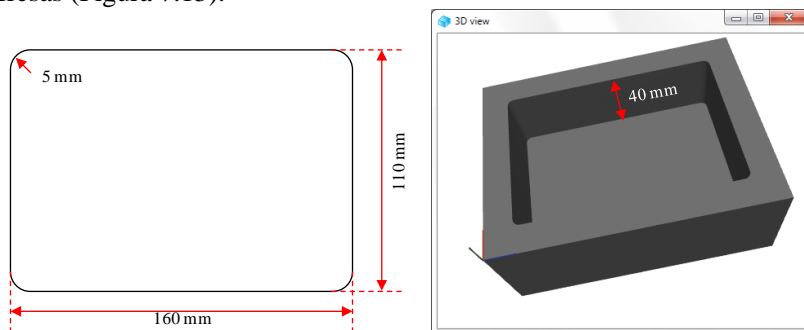


Figura 7.12. Peça exemplo para a escolha automática de ferramentas.

Fonte: Criado pelo autor.

A Figura 7.14 apresenta a tabela com as ferramentas selecionadas pelo sistema, podendo-se perceber que das ferramentas candidatas do catálogo, foram selecionadas apenas duas: a fresa de maior diâmetro cadastrado (20 mm) e a fresa que tem diâmetro igual duas vezes o raio de concordância da cavidade (10 mm).

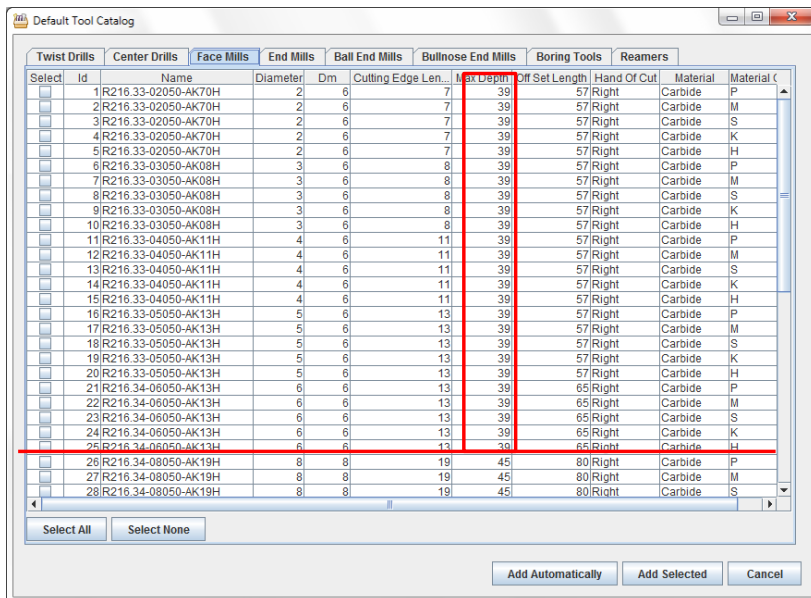


Figura 7.13. Catálogo de ferramentas cadastradas no servidor.

Fonte: Criado pelo autor.

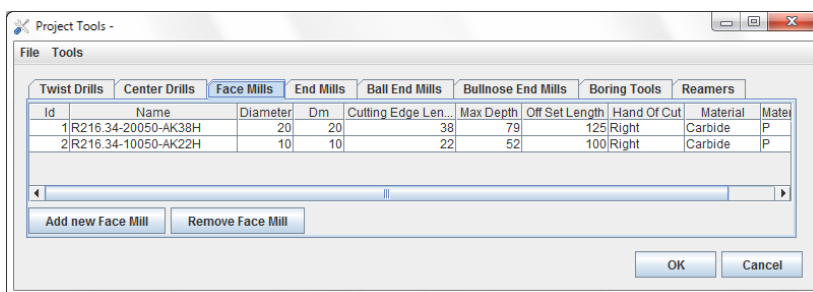


Figura 7.14. Ferramentas identificadas pelo sistema como adequadas para a usinagem da peça da Figura 7.12.

Fonte: Criado pelo autor.

## 7.2.2 Mapeamento para *machining workingsteps*

Cada *feature* é mapeada em um ou mais *machining workingsteps*, respeitando-se as regras de manufatura apresentadas na seção 6.2.1.2.1.

Como mencionado nessa seção, os principais passos para se obter um *machining workingstep* são:

- Determinação da sua operação de usinagem,
- Determinação da sua ferramenta de corte,
- Determinação dos seus parâmetros de corte.
- Determinar seu *machining workingstep* precedente.

Como exemplo considera-se uma peça com uma *feature* com perfil curva de Bèzier (Figura 7.15). As operações de usinagem serão:

- Fresamento com fresa de topo que caiba dentro dos 60 mm do perfil, ferramenta esta que poderá remover uma parte do material. As condições de usinagem são consultadas via internet à base de dados no servidor. Como este seria o primeiro *machining workingstep* a ser criado, não aponta para nenhum outro para defini-lo como seu precedente. Com estes dados pode-se criar o *machining workingstep* M. WS 1 da Figura 7.16.
- Uma segunda operação de fresamento com fresa de topo será necessária para remover o material restante, e esta segunda fresa tem um diâmetro menor para poder remover material nos lugares onde a primeira não consegue. Este diâmetro é igual ou menor que o diâmetro da fresa esférica necessária para a terceira operação. Seu *machining workingstep* precedente será M. WS 1. Com estes dados cria-se o segundo *machining workingstep* (M. WS 2, na Figura 7.16).
- Finalmente, uma operação de fresamento com fresa esférica será necessária para remover as arestas deixadas pelas operações anteriores (M. WS 3, na Figura 7.16). Seu *machining workingstep* precedente será M. WS2.

O raio da fresa esférica deve caber dentro da menor curvatura côncava do perfil.

O programa calcula a derivada ao longo da curva e detecta onde a curva é côncava e onde é convexa (Figura 7.17) por meio da análise da derivada.

O raio da fresa esférica que cabe dentro das partes côncavas da curva é calculado da seguinte maneira:

- (a) A curva é segmentada em um número finito de pontos (Figura 7.18).

- (b) Agrupam-se de três em três pontos para formar triângulos (Figura 7.18).
- (c) Calcula-se o circuncentro<sup>4</sup> dos triângulos.
- (d) O menor raio nos trechos côncavos garante que a fresa esférica poderá usinar toda a superfície.

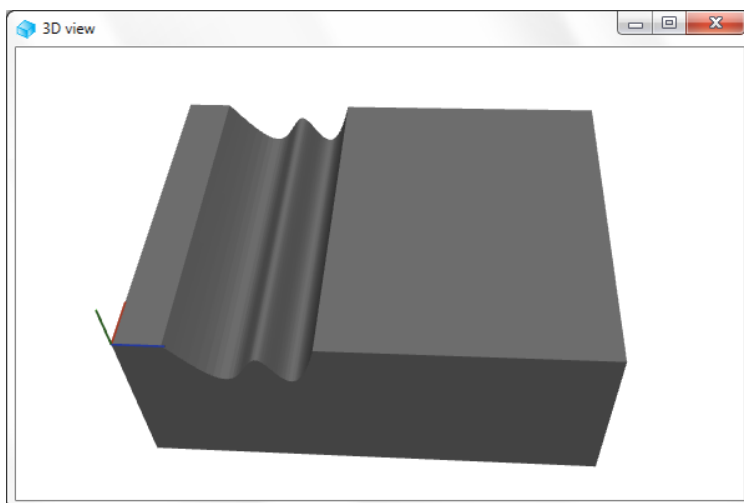
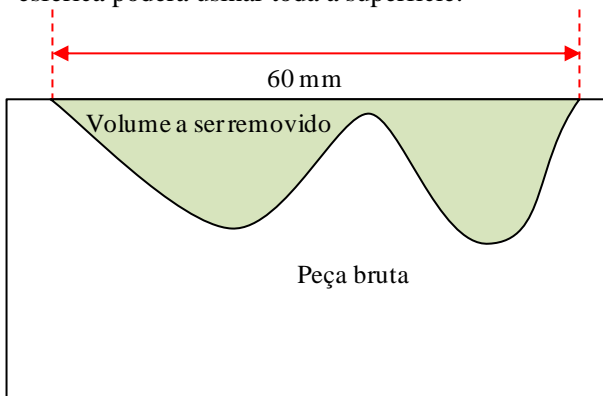


Figura 7.15. Peça exemplo para explicar como funciona o mapeamento para *machining workingsteps*.

Fonte: Criado pelo autor.

---

<sup>4</sup> Centro da circunferência circunscrita ao triângulo.

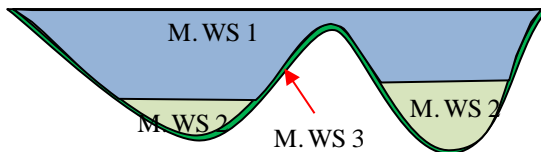


Figura 7.16. Perfil genérico dividido em *machinings workingsteps*.  
Fonte: Criado pelo autor.

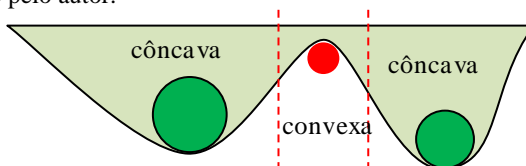


Figura 7.17. Divisão da curva em trechos côncavos e convexos.  
Fonte: Criado pelo autor.

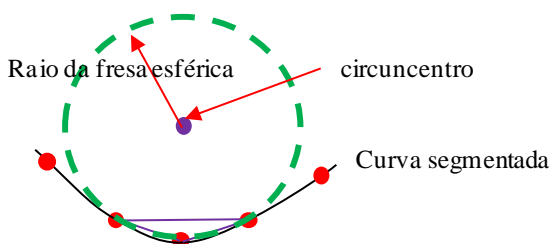


Figura 7.18. Segmentação da curva em triângulos para calcular seu circuncentro.  
Fonte: Criado pelo autor.

A Figura 7.19 e a Tabela 7.1 apresentam o resultado do mapeamento de *machining workingsteps* para este perfil de Bèzier. Como o primeiro *machining workingstep* (M. WS 1) não possui nenhum outro *machining workingstep* como seu precedente, isto indica que ele será o primeiro a ser executado no *workplan*. Como M. WS 3 exige que primeiro seja executado M.WS 2 (ver na Tabela 7.1), então M. WS 3 ficará por último, e M. WS 2 em segundo lugar.



**Tabela 7.1. Workplan e machinings workingsteps resultantes do mapeamento do perfil Bèzier.**

Ordem de execução no Workplan	Machining workingstep	Operação	Tipo de ferramenta	Machining workingstep Precedente
1	M. WS 1	Fresamento desbaste	Face Mill 20 mm	Nulo
2	M. WS 2	Fresamento desbaste	Face Mill 5 mm	M. WS 1
3	M. WS 3	Fresamento desbaste	Ball End Mill 5 mm	M. WS 2

FONTE: Criado pelo autor.

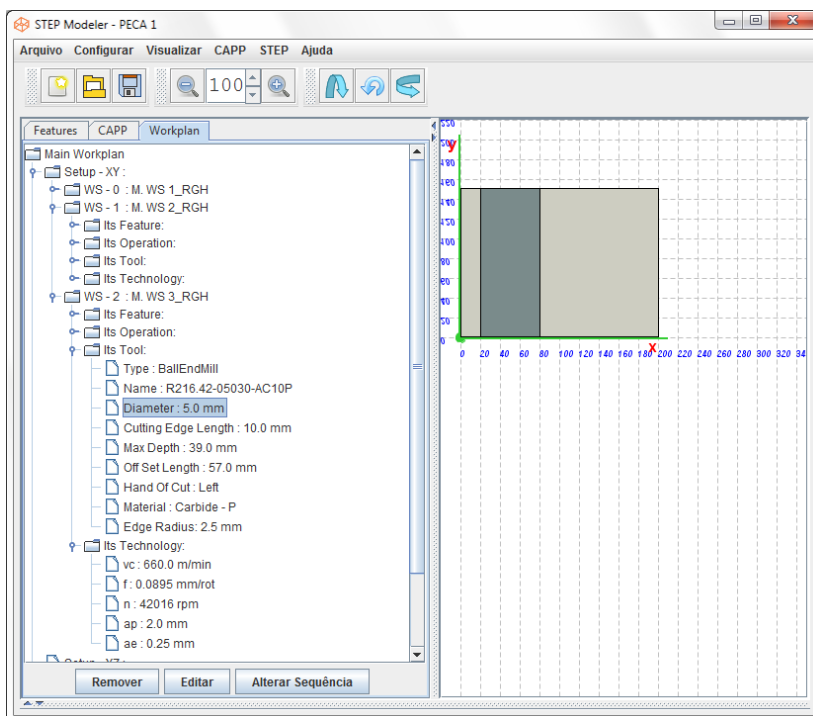


Figura 7.19. Resultado do mapeamento de uma ranhura com perfil genérico.  
Fonte: Criado pelo autor.

A Figura 7.20 apresenta os diâmetros calculados para cada segmento de curva (divisão em 100 segmentos). Deve-se notar que o menor valor (marcado em vermelho) é igual a 3,83 mm, porém o programa escolheu uma fresa de maior diâmetro (5 mm), isto por que o valor de 3,83 caiu em uma parte onde a curva é convexa, portanto o programa o desconsiderou. Entretanto o programa detectou que no segmento 81 da curva (marcado em verde na Figura 7.20) há um valor de 5.7 mm que está na região côncava, logo, a menor fresa esférica que pode usinar esse perfil deverá ser de 5.7 mm de diâmetro.

Outras *features* (cavidades com base arredondada, furos com base cônica, ranhuras com perfil “Vee” ranhuras com perfil parcial circular, etc.) resultam em vetores de *machining workingsteps* que, ao serem executados, removerão material e deixarão a peça com a forma desejada. Estes mapeadores podem ser vistos no apêndice A.

Considerando-se uma peça com várias *features* diferentes (por exemplo, a peça da Figura 7.6), todas as suas *features* foram mapeadas em vários *machining workingsteps* como se vê na Figura 7.21, onde se mostra em uma estrutura de árvore cada *machining workingstep* com sua operação, sua ferramenta, seus parâmetros de corte, e a *feature* à qual ele pertence.

```

0 diametroTmp = 1365.692169788676
1 diametroTmp = 783.3756458801212
2 diametroTmp = 519.4628592756465
3 diametroTmp = 368.788113121891
4 diametroTmp = 271.7121088116812
5 diametroTmp = 204.4986723121882
6 diametroTmp = 155.80224190475607
7 diametroTmp = 119.4879391201687
8 diametroTmp = 91.92272211871398
9 diametroTmp = 70.79586257367406
10 diametroTmp = 54.54909640373552
11 diametroTmp = 42.0789495239915
12 diametroTmp = 32.57174623044444
13 diametroTmp = 25.408091388821102
14 diametroTmp = 20.10600094009092
15 diametroTmp = 16.286682648531
16 diametroTmp = 13.65410910826326
17 diametroTmp = 11.982982380247606
18 diametroTmp = 11.11276700663998
19 diametroTmp = 10.934356071471235
20 diametroTmp = 11.39858505768809
21 diametroTmp = 12.49384379089827
22 diametroTmp = 14.246057045034432
23 diametroTmp = 16.711966943206292
24 diametroTmp = 19.978733516124628
25 diametroTmp = 24.171280961166573
26 diametroTmp = 29.471846407032515
27 diametroTmp = 36.16070405606566
28 diametroTmp = 44.69860863015416
29 diametroTmp = 55.90252736518818
30 diametroTmp = 71.3603413698579
31 diametroTmp = 94.57168987774197
32 diametroTmp = 134.91299241940555
33 diametroTmp = 228.34011908987156
34 diametroTmp = 753.3470369390689
35 diametroTmp = 525.316386433472
36 diametroTmp = 181.83259705529775
37 diametroTmp = 103.17722492563665
38 diametroTmp = 67.63572678640755
39 diametroTmp = 47.187862189653906
40 diametroTmp = 33.91533451207075
41 diametroTmp = 24.715528528055707
42 diametroTmp = 18.116165192664383
43 diametroTmp = 13.31816186012149
44 diametroTmp = 9.839162514911283
45 diametroTmp = 7.36118554401498
46 diametroTmp = 5.660170297905065
47 diametroTmp = 4.573331204023879
48 diametroTmp = 3.8863141066573013
49 diametroTmp = 3.8311444636442396
50 diametroTmp = 4.088300561211411
51 diametroTmp = 4.787306287088663
52 diametroTmp = 6.003630881126561
53 diametroTmp = 7.854391534566256
54 diametroTmp = 10.497679959516926
55 diametroTmp = 14.140238144581549
56 diametroTmp = 19.058963581592646
57 diametroTmp = 25.64638023272858
58 diametroTmp = 34.50305972060364
59 diametroTmp = 46.634042977873996
60 diametroTmp = 63.90754657700858
61 diametroTmp = 90.29201175632127
62 diametroTmp = 136.00979075186038
63 diametroTmp = 237.81315822348557
64 diametroTmp = 702.1778171747656
65 diametroTmp = 863.5278147507241
66 diametroTmp = 271.18850549644543
67 diametroTmp = 158.6402710101728
68 diametroTmp = 109.24475596574287
69 diametroTmp = 80.58655223486457
70 diametroTmp = 61.41533452482557
71 diametroTmp = 47.49719159683763
72 diametroTmp = 36.90097647655485
73 diametroTmp = 28.627995928034487
74 diametroTmp = 22.10993926785115
75 diametroTmp = 16.994204576798197
76 diametroTmp = 13.041786485634223
77 diametroTmp = 10.075704725217456
78 diametroTmp = 7.955596210271658
79 diametroTmp = 6.568581003442338
80 diametroTmp = 5.832404877258679
81 diametroTmp = 5.708043229122911
82 diametroTmp = 6.216464887577651
83 diametroTmp = 7.45129955439019
84 diametroTmp = 9.58132661272452
85 diametroTmp = 12.843786596232654
86 diametroTmp = 17.534054609967658
87 diametroTmp = 23.996133702116587
88 diametroTmp = 32.615549991125675
89 diametroTmp = 43.814611188758384
90 diametroTmp = 58.049677662483
91 diametroTmp = 75.81034489810989
92 diametroTmp = 97.62084538623559
93 diametroTmp = 124.04446274966179
94 diametroTmp = 155.69237269386264
95 diametroTmp = 193.23922702988366
96 diametroTmp = 237.44922579504347
97 diametroTmp = 289.2188246652645
98 diametroTmp = 349.6464574958618

```

Figura 7.20. Diâmetros calculados para uma curva de Bèzier dividida em 100 segmentos.

Fonte: Criado pelo autor.

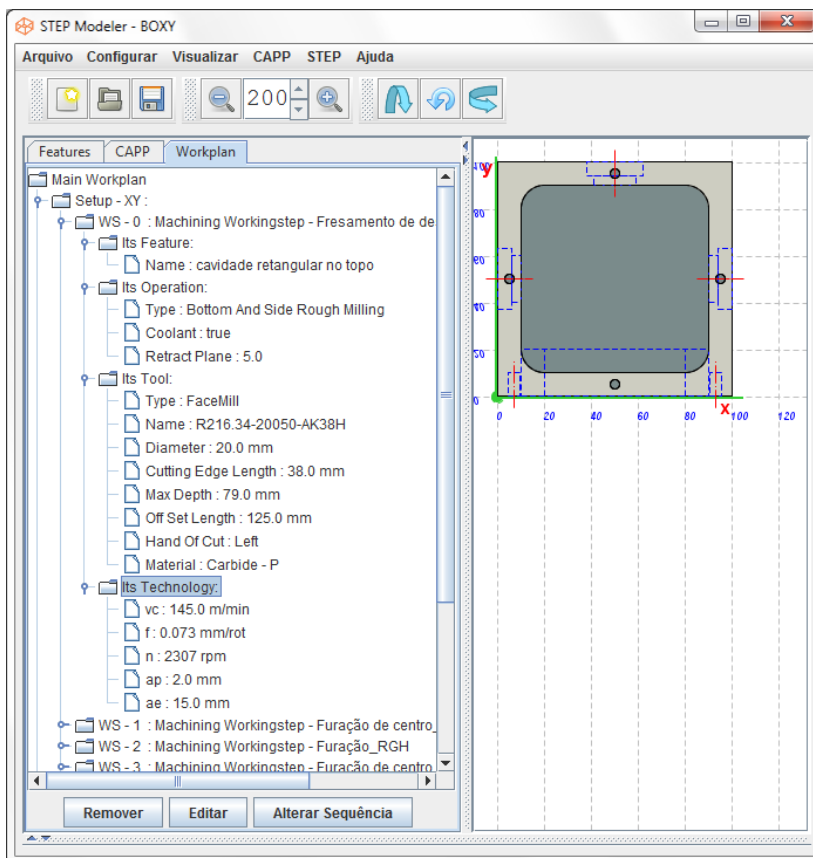


Figura 7.21. *Machining workingsteps* mapeadas a partir das *features*.

Fonte: Criado pelo autor.

Como mencionado anteriormente, os parâmetros de usinagem são lidos de uma tabela onde o fabricante fornece os parâmetros recomendados. Esta tabela se encontra em um banco de dados no servidor. O programa faz consulta via internet ao banco de dados e retorna os valores dos dados de corte para serem usados pelas classes que realizam o mapeamento. Entretanto, pode acontecer que não seja possível mapear uma *feature* devido a não haverem recursos disponíveis para a sua fabricação. Por exemplo, devido à falta de brocas com diâmetro ou ângulo de ponta adequado, o programa acusou um erro indicando qual a *feature* que não pôde ser mapeada e o motivo (Figura 7.22).

Outras causas podem ser: incompatibilidade do material a ser usinado com o da ferramenta, características geométricas (furos profundos com brocas que não conseguem penetrar até a profundidade especificada na *feature*), etc.

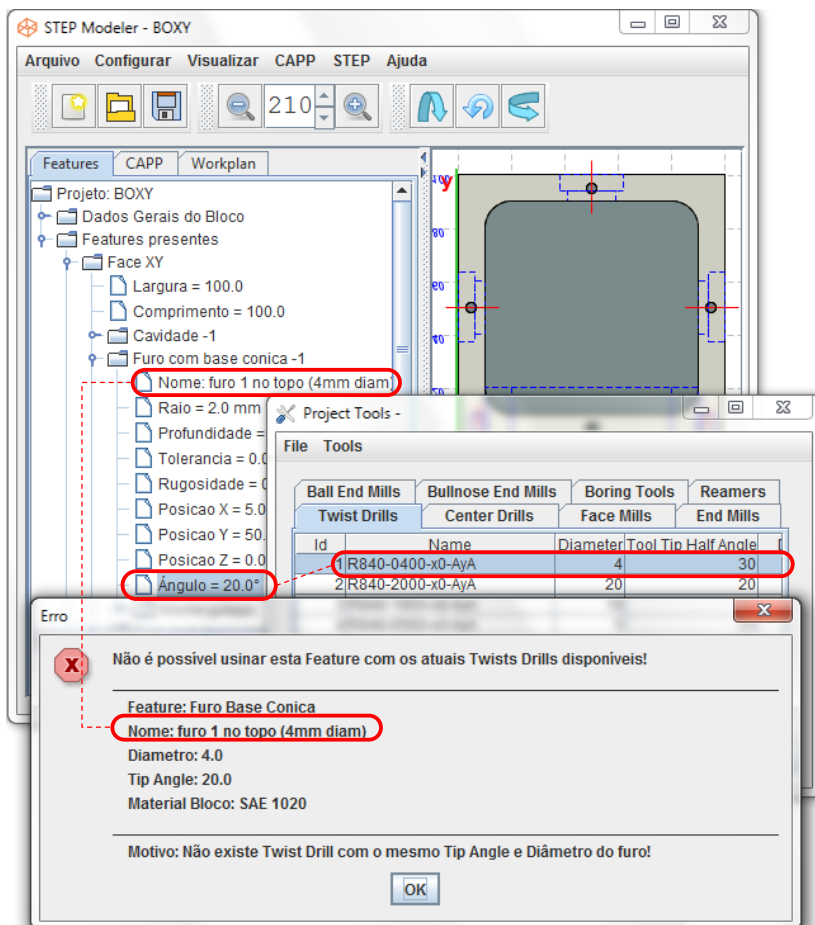


Figura 7.22. Mensagem de erro ao tentar criar os *machining workingsteps* para um furo com base cônica.

Fonte: Criado pelo autor.

### 7.2.2.1 Remoção e edição de *machining workingsteps*

O botão “remover” na parte inferior foi implementado com a finalidade de possibilitar ao usuário remover algum *machining workingstep* que talvez não seja obrigatório. Por exemplo, para operações de furação com diâmetros pequenos, onde a precisão não é tão elevada, poderia ser eliminado o *machining workingstep* da furação de centro. O sistema enviará uma mensagem de advertência ao usuário, como se vê na Figura 7.23.

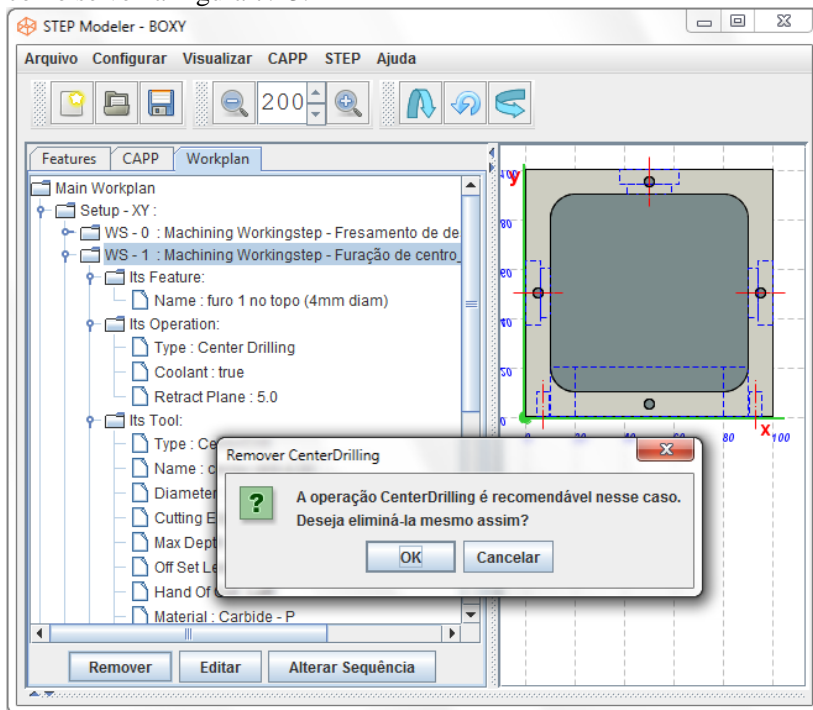


Figura 7.23. Mensagem de advertência ao tentar remover a operação furação de centro.

Fonte: Criado pelo autor.

O programa verifica a consistência das operações que não foram removidas, e se não for consistente, uma mensagem de advertência aparecerá. Por exemplo, na peça da Figura 7.15, a remoção do *machining workingstep* “M. WS 2” (desbaste com fresa de topo) deixaria inconsistente o *machining workingstep* “M. WS 3” (fresamento

com fresa esférica). Portanto, este último deveria ser removido junto (ver Figura 7.24).

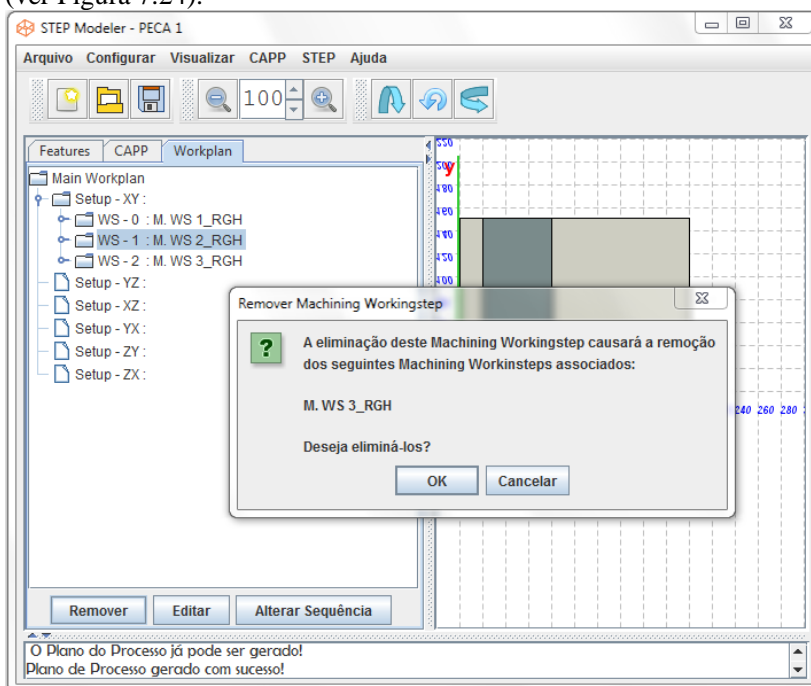


Figura 7.24. Mensagem de advertência de remoção de um *machining workingstep*.

Fonte: Criado pelo autor.

O botão “editar” possibilita que certos parâmetros do *machining workingstep* (parâmetros relacionados à sua operação de usinagem, à sua ferramenta, e à sua tecnologia de usinagem) sejam editados. A Figura 7.25 apresenta a interface de edição da operação do *machining workingstep* chamado “Machining Workingstep – Furação de centro”.

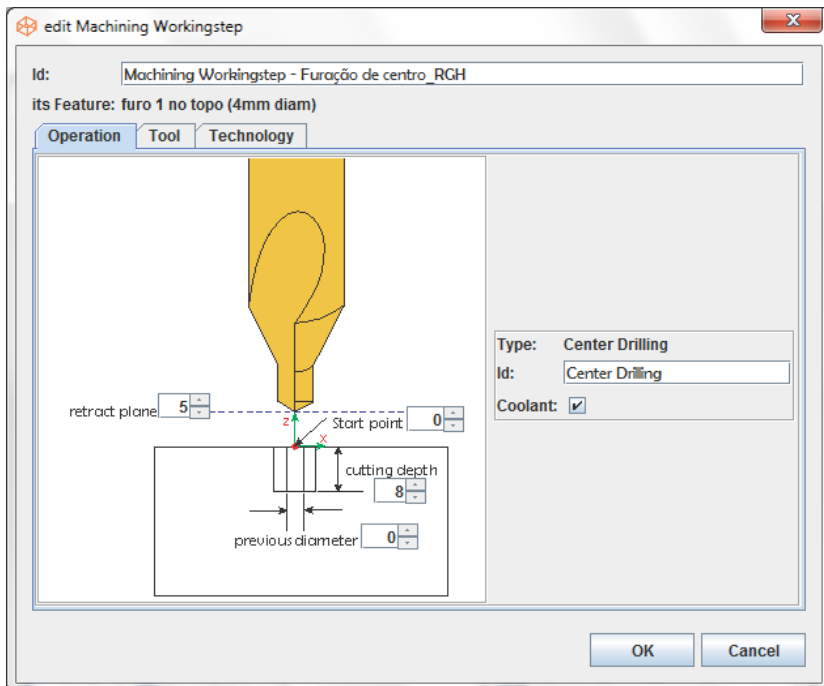


Figura 7.25. Interface de edição dos parâmetros da operação.  
Fonte: Criado pelo autor.

A Figura 7.26 apresenta a interface de edição da ferramenta, e a Figura 7.27 mostra a interface de edição dos parâmetros de corte associados a este *machining workingstep*.



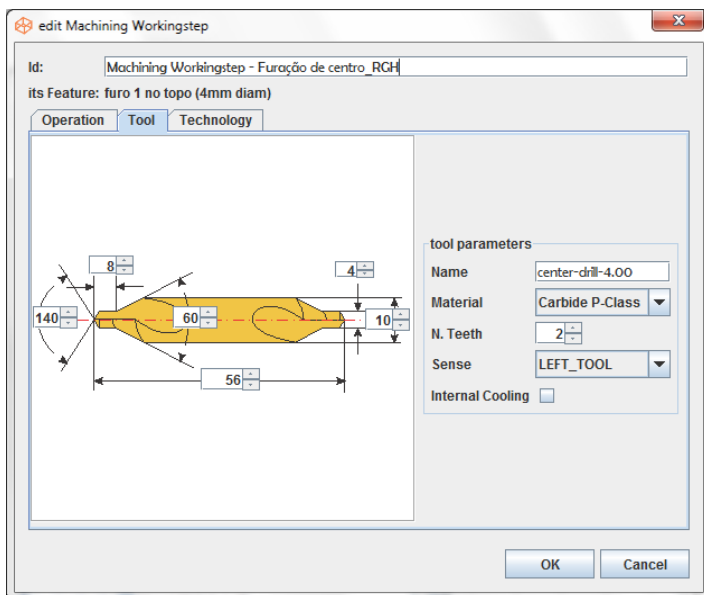


Figura 7.26. Interface de edição da ferramenta.

Fonte: Criado pelo autor.

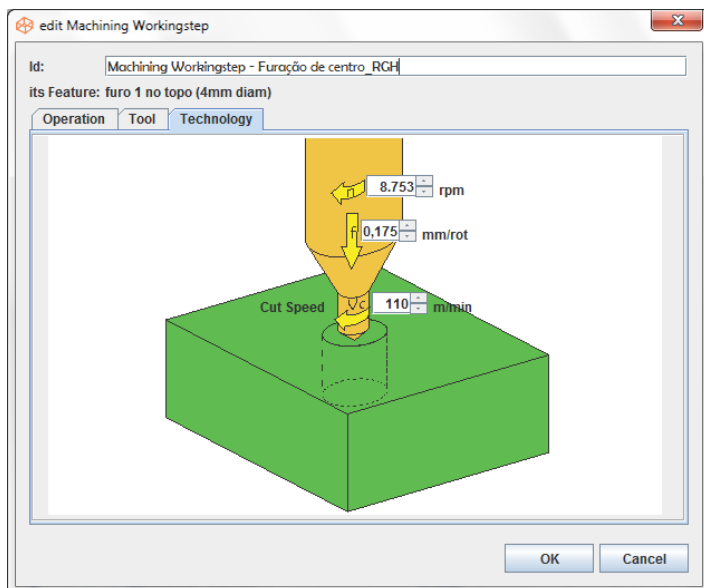


Figura 7.27. Interface de edição dos parâmetros tecnológicos

Fonte: Criado pelo autor.

O botão “alterar sequência” (ver Figura 7.23) permite alterar a ordem de execução dos *machining workingsteps* no *workplan*.

### 7.2.3 Pontos de apoio

O usuário tem a possibilidade de definir os pontos de apoio. A Figura 7.28 apresenta a interface de definição dos pontos de apoio. Na versão atual, apenas podem ser inseridos manualmente nas coordenadas desejadas. Primeiramente o usuário deve definir o número de pontos de apoio, aparecerá uma tabela com número de linhas igual ao número de pontos desejados. Na tabela pode-se digitar as coordenadas dos pontos de apoio. A representação gráfica dos pontos aparece em azul celeste e com linhas tracejadas, e desta maneira o usuário pode saber se os apoios estão na posição desejada.

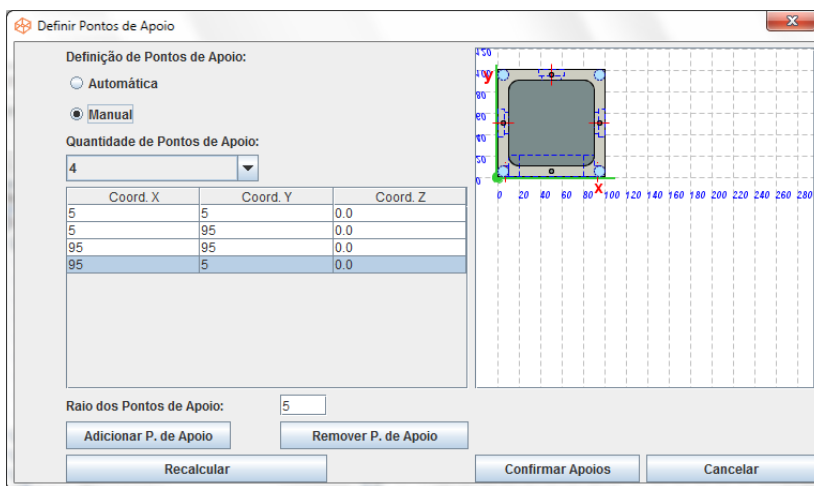
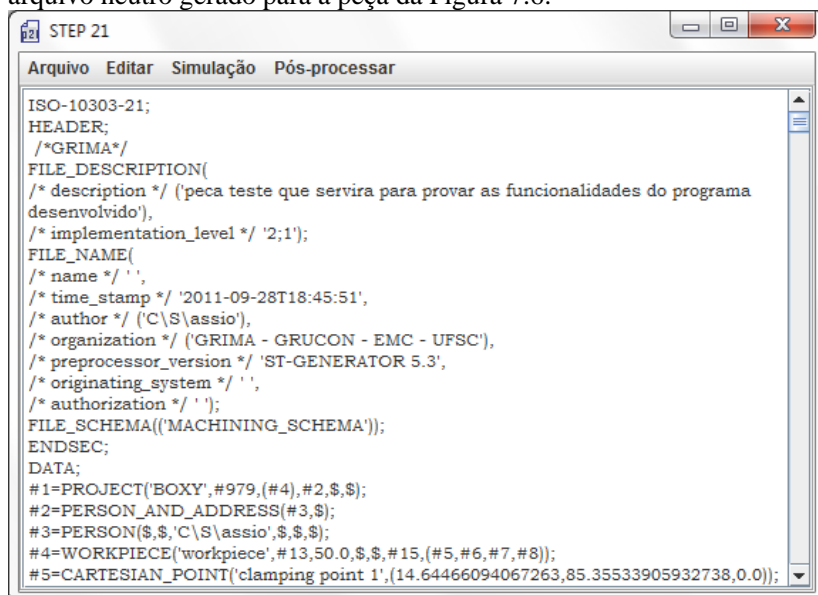


Figura 7.28. Interface para definição dos pontos de apoio.  
Fonte: Criado pelo autor.

## 7.2.4 Geração do arquivo neutro

Uma vez que as informações de projeto e de manufatura já foram determinadas, pode-se gerar o arquivo neutro. A Figura 7.29 apresenta o arquivo neutro gerado para a peça da Figura 7.6.



```

STEP 21
Arquivo Editar Simulação Pós-processar
ISO-10303-21;
HEADER;
/*GRIMA*/
FILE_DESCRIPTION(
/* description */ ('peca teste que servira para provar as funcionalidades do programa
desenvolvido'),
/* implementation_level */ '2;1');
FILE_NAME(
/* name */ '',
/* time_stamp */ '2011-09-28T18:45:51',
/* author */ ('C\S\assio'),
/* organization */ ('GRIMA - GRUCON - EMC - UFSC'),
/* preprocessor_version */ 'ST-GENERATOR 5.3',
/* originating_system */ '',
/* authorization */ '');
FILE_SCHEMA(('MACHINING_SCHEMA'));
ENDSEC;
DATA;
#1=PROJECT('BOXY',#979,(#4),#2,$,$);
#2=PERSON_AND_ADDRESS(#3,$);
#3=PERSON($,$,'C\S\assio',$,$,$);
#4=WORKPIECE('workpiece',#13,50.0,$,$,#15,(#5,#6,#7,#8));
#5=CARTESIAN_POINT('clamping point 1',(14.644466094067263,85.35533905932738,0.0));
  
```

Figura 7.29. Arquivo neutro gerado.

Fonte: Criado pelo autor.

Pode-se também exportar o arquivo neutro em XML, e visualizá-lo em qualquer navegador de internet. A Figura 7.30 apresenta um trecho do arquivo em XML exportado a partir do arquivo neutro da Figura 7.29.

```

<iso_10303_28 representation_category="LB" version="PDT5">
  <iso_10303_28_header>
    <document_name>peca teste que servira para provar as funcionalidades do
programa desenvolvido</document_name>
    <time_stamp>2011-09-28T17:32:20</time_stamp>
    <author>Cássio</author>
    <originating_organization>GRIMA - GRUCON - EMC -
UFSC</originating_organization>
    <authorization> </authorization>
    <originating_system> </originating_system>
    <preprocessor_version>ST-GENERATOR 5.3</preprocessor_version>
  </iso_10303_28_header>
  ...

```

Figura 7.30 Trecho do arquivo neutro em formato XML.

Fonte: Criado pelo autor.

## 7.3 IMPLEMENTAÇÃO DO SISTEMA CAM

### 7.3.1 Leitura de arquivos neutros

A leitura do arquivo físico é uma tarefa delicada, pois em se tratando de um conjunto de várias centenas de entidades diferentes, as quais recebem diferentes tipos de argumentos para serem instanciadas, é fácil cometer erros.

As Metodologias Ágeis<sup>5</sup> de desenvolvimento de software propõem a implementação de testes em diferentes níveis ou estágios para garantir a robustez do sistema. Os principais testes, segundo JENKINS (2008) são:

- Testes de nível unitário: estágio mais baixo de desenvolvimento, no qual os componentes discretos do produto final são testados independentemente antes de serem montados em unidades maiores. Tipicamente são usados os “testes de utilização” os quais simulam o contexto em que a unidade deve ser integrada. O teste de utilização fornece um

<sup>5</sup> Metodologias ágeis: Os conceitos de desenvolvimento ágil do produto emergiram nos anos 1980 com os fabricantes de automóveis japoneses. (AGUANNO *et. al.*, 2005). Estas metodologias visam o desenvolvimento de software capaz de tratar com as necessidades de velocidade e de flexibilidade, usando os recursos disponíveis da maneira mais eficiente (HIGHSMITH, 2002).

número de entradas conhecidas e mede as saídas da unidade sob teste. Estas saídas são comparadas com os valores esperados para, desta maneira, detectar a existência de algum problema.

- Testes de nível de integração: testes onde as funções de baixo nível são integradas em unidades maiores. Diferentemente dos testes unitários, as unidades são testadas em grupos. Isto é, o foco não é mais em cada unidade, mas nas interações existentes entre elas.
- Testes de nível de sistema: estes testes representam o teste global no produto de software montado. O teste de nível de sistema é particularmente importante, devido a que somente neste estágio é que toda a complexidade do produto está presente.

Neste trabalho foi implementada uma estrutura em níveis ou camadas (Figura 7.31), onde o nível unitário é constituído por classes que realizam funções de baixo nível, lendo atributos das entidades (valores numéricos, cadeias de caracteres, ou ainda outras entidades) presentes no arquivo físico.

Na camada de integração, são agrupadas as funções unitárias para compor um leitor de uma entidade inteira.

Por fim, na camada de sistema, tem-se a leitura completa do arquivo físico.

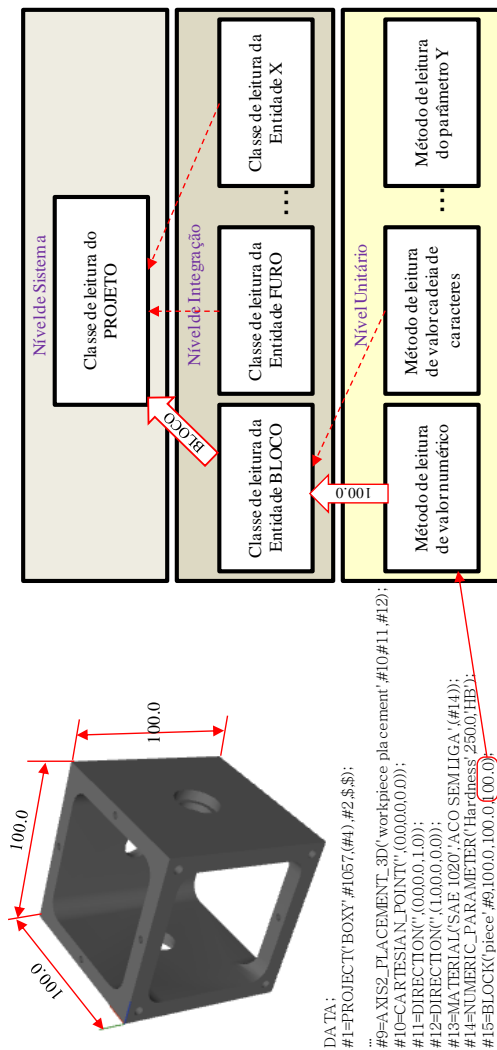


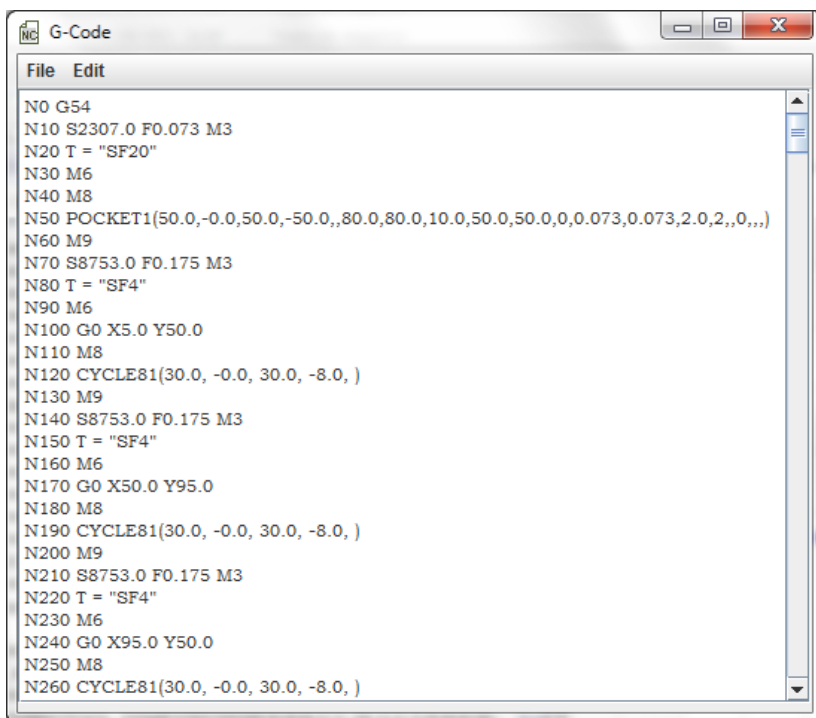
Figura 7.31. Estrutura em níveis para leitura do arquivo físico.

Fonte: Criado pelo autor.

### 7.3.2 Geração do código G

A geração do código G foi feita para uma máquina de três eixos com comando CNC Siemens 840di, disponível no laboratório USICON/EMC/UFSC. O comando possui alguns ciclos fixos para

usinagem de cavidades retangulares e circulares, e para usinagem de furos. A Figura 7.32 apresenta o código gerado pelo programa a partir do arquivo neutro da Figura 7.29.

The image shows a window titled "G-Code" with a menu bar containing "File" and "Edit". The main text area contains the following G-code:

```
N0 G54
N10 S2307.0 F0.073 M3
N20 T = "SF20"
N30 M6
N40 M8
N50 POCKET1(50.0,-0.0,50.0,-50.0,,80.0,80.0,10.0,50.0,50.0,0.0,0.073,0.073,2.0,2,,0,,)
N60 M9
N70 S8753.0 F0.175 M3
N80 T = "SF4"
N90 M6
N100 G0 X5.0 Y50.0
N110 M8
N120 CYCLE81(30.0, -0.0, 30.0, -8.0, )
N130 M9
N140 S8753.0 F0.175 M3
N150 T = "SF4"
N160 M6
N170 G0 X50.0 Y95.0
N180 M8
N190 CYCLE81(30.0, -0.0, 30.0, -8.0, )
N200 M9
N210 S8753.0 F0.175 M3
N220 T = "SF4"
N230 M6
N240 G0 X95.0 Y50.0
N250 M8
N260 CYCLE81(30.0, -0.0, 30.0, -8.0, )
```

Figura 7.32. Código G gerado para o CNC Siemens840di.

Fonte: Criado pelo autor.

A Figura 7.33 apresenta o resultado simulado no software do controlador Siemens, denominado Sinutrain.



Figura 7.33. Simulação no software da máquina (SinuTrain)

Fonte: Criado pelo autor.

O ciclo fixo “POCKET1” (ver Figura 7.32, linha N50) gera trajetórias para usinagem de cavidades retangulares utilizando somente uma fresa. Porém, às vezes é necessária a utilização de mais de uma ferramenta para remover o volume da cavidade, isto porque a fresa de maior diâmetro removerá um maior volume de material em menos tempo. Porém, essa fresa de maior diâmetro não consegue remover o material dos cantos porque o raio da fresa é maior do que o raio de concordância da cavidade. Nesse caso, uma segunda fresa se encarregará de remover este material (Figura 7.34).



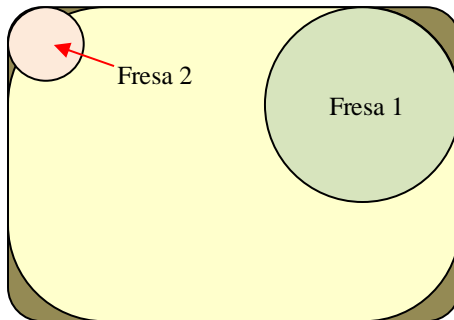


Figura 7.34. Áreas removidas por duas fresas de diferente diâmetro.  
Fonte: Criado pelo autor.

As trajetórias de movimentação da segunda fresa foram implementadas usando a modelagem descrita na seção 6.2.1.3.2 e na Figura 6.29. Modelagem das trajetórias em UML., onde vetores contendo elementos “*Path*” descrevem a trajetória que a ferramenta deverá percorrer. Portanto, esses vetores são mapeados e transformados em linhas de comando de código G. Se um elemento dentro do vetor de movimentação é do tipo “*LinearPath*”, então, provavelmente, o comando será G0 ou G1, se o atributo “*tipoDeMovimento*” é SLOW\_MOV, o comando gerado será G1, caso seja FAST\_MOV, o comando gerado será G0.

Se um elemento no vetor de movimentações é do tipo “*CircularPath*”, então se verifica pelo seu atributo “*sense*” se este atributo é CW, então o comando gerado será G2, e caso seja CCW o comando gerado será G3.

Os elementos do tipo “*Path*” têm como atributos básicos os pontos de início e de fim da movimentação. Estes pontos podem ser lidos e colocados junto à sua linha de comando correspondente. Objetos do tipo *CircularPath* têm adicionalmente o atributo que define a coordenada do centro.

### 7.3.3 Simulação da usinagem

Desenvolveu-se também um simulador. Este simulador usa o módulo de leitura (descrito no item 7.3.1) para extrair a partir do arquivo neutro informações tais como tamanho do bloco e diâmetro das ferramentas utilizadas para criar uma simulação do processo de

usinagem. Foram gerados vetores com os pontos (*Point3d*) de movimentação por onde a ferramenta deve passar.

O simulador lê estes pontos e os envia aos painéis desenhadores (de vista de topo e lateral) para criar nessa coordenada a representação visual da ferramenta. A Figura 7.35 apresenta a interface gráfica onde foi feita a simulação da movimentação para a face XY da peça da Figura 7.6.

A versão atual do programa consegue simular as movimentações e gerar o código G de *machining workingsteps* pertencentes a furos de base plana (acabamento e desbaste), furos passantes (acabamento e desbaste), furos cônicos (desbaste), ranhuras de perfil retangular (desbaste), degraus de perfil retangular (desbaste), cavidades de fundo plano (acabamento e desbaste) e cavidades passantes (acabamento e desbaste).

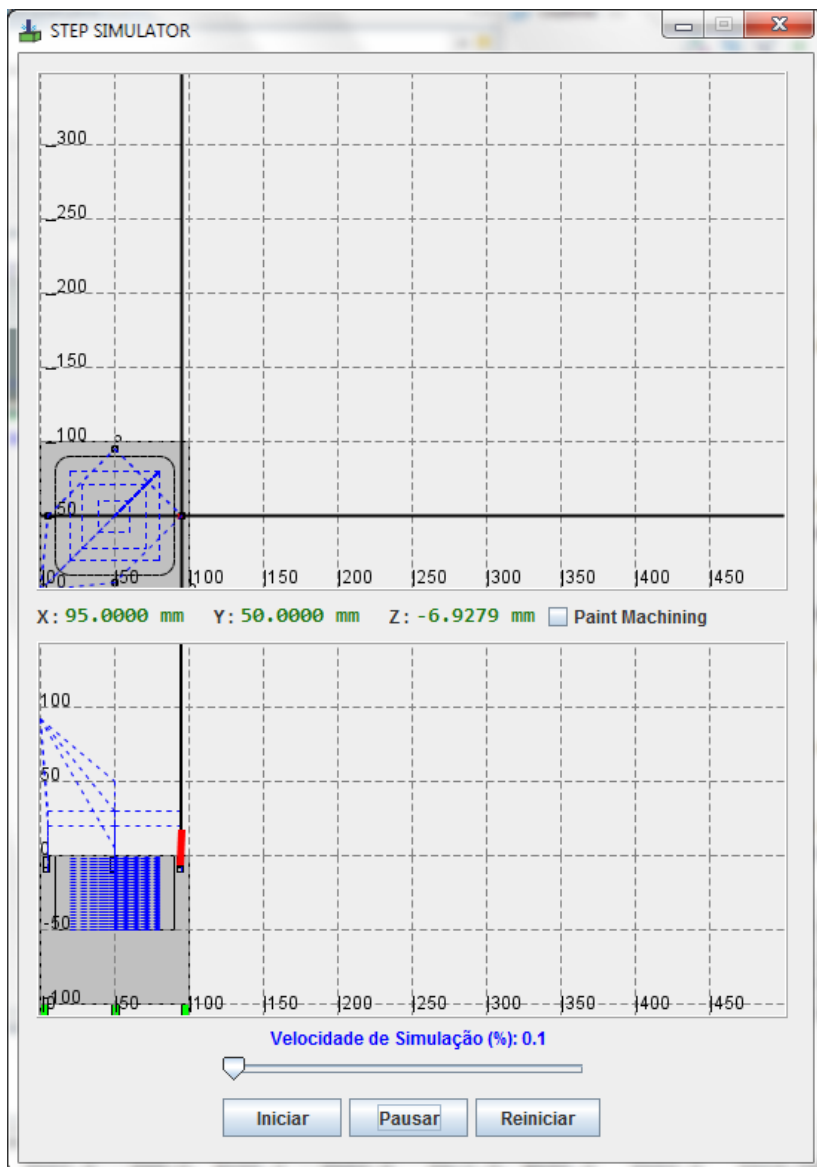


Figura 7.35. Simulador de trajetórias.  
Fonte: Criado pelo autor.

### **7.3.4 Disponibilização para usuários remotos**

Foi criada uma página de internet para que usuários cadastrados possam ter acesso ao sistema, onde o único requisito é ter um navegador com suporte a Java, e Java 3D instalado no computador (para visualização do sólido em 3D).

Um banco de dados foi implementado para salvar os dados dos usuários, os seus projetos realizados, grupos de ferramentas criadas e as ordens de fabricação que porventura sejam feitas. Esta página está disponível em [http://www.grima.ufsc.br/stepnc\\_project/](http://www.grima.ufsc.br/stepnc_project/).

Detalhes acerca da utilização do sistema podem ser vistos no apêndice C.

## 8 CONSIDERAÇÕES FINAIS

O presente capítulo tem por objetivo apresentar as conclusões referentes ao trabalho de pesquisa desenvolvido no âmbito desta tese.

### 8.1 CONCLUSÕES

- O protótipo CAD/CAPP/CAM proposto baseia-se em informações homogêneas e compatíveis. Isto permitiu a integração entre tais módulos e possibilitou o fluxo de informações bidirecional (um exemplo pode se ver no apêndice B).
- A hipótese feita no capítulo introdutório foi provada verdadeira, pois a implementação do protótipo usa o modelo de dados fornecido pelo padrão ISO 14649 e este foi essencial para conseguir fluxos de informações compatíveis.
- O modelo de dados STEP-NC (ISO 14649) que foi usado no presente trabalho de pesquisa fornece um modelo de dados orientado a objetos e facilitou a integração entre os módulos do sistema.
- A abordagem DSG (*Destructive Solid Geometry*), do projeto por *features* facilitou a modelagem dos dados de projeto, devido a que informações de alto nível são armazenadas na estrutura de dados do módulo CAD. Estas informações podem ser recuperadas e disponibilizadas para os outros módulos que compõem o sistema.
- O uso de modelos SDAI como ponte entre a linguagem EXPRESS e Java para leitura e escrita de dados em formato neutro mostrou-se importante, devido a que os modelos do produto, de processo e de recursos (ver seção 6.1.1) podem ser mapeados em um modelo SDAI equivalente com relativa facilidade. Desta maneira, separam-se tarefas: os módulos do sistema usam os modelos do produto, do processo e de recursos independentemente, e o modelo SDAI lida com as entidades definidas na norma STEP-NC e com os esquemas EXPRESS.
- Para a representação dos sólidos das *features* usando B-rep, foi necessária a implementação de classes e métodos de cálculo de

vértices, arestas e faces, e o sistema mostrou-se eficiente ao representar estes sólidos.

- A utilização da internet permitiu que dados acerca dos usuários e das ferramentas pudessem ser consultados remotamente a qualquer momento que o programa precisar, uma vez que o banco de dados encontra-se no servidor. Isto permite o gerenciamento dos dados mais seguro.
- A linguagem Java possibilitou ao programa poder ser inserido dentro de uma página PHP na internet, e desta maneira usuários em qualquer lugar do mundo podem ter acesso ao sistema desenvolvido.

## 8.2 CONTRIBUIÇÕES

Entre as principais contribuições do presente trabalho de tese podem-se citar:

- O modelo conceitual e a arquitetura do modelo do sistema computacional aderente ao padrão STEP-NC voltado para internet definidos no capítulo 6. Deve se notar que estes foram desenvolvidos a partir da questão de pesquisa em concordância com os objetivos geral e específicos estabelecidos no capítulo 1. O protótipo foi desenvolvido a partir das especificações do modelo.
- O sistema computacional é inédito no que concerne a sistemas que utilizam o padrão STEP-NC, permitindo sua execução através da internet, assim como gerando arquivos neutros em formato ISO 10303-21 e ISO 10303-28 (XML), tanto quanto lendo tais arquivos.
- A metodologia utilizada para a implementação do protótipo permite que futuras implementações sejam feitas sem muitas mudanças.
- A abordagem de modelagem por *features* contribui para a construção de futuras implementações adicionando-se outras *features* à biblioteca atual.
- O protótipo desenvolvido contribui com relação a sistemas remotos de manufatura voltados para *web*.
- O módulo CAPP verifica se com as ferramentas disponíveis é possível fabricar uma peça.

### 8.3 SUGESTÕES PARA FUTUROS TRABALHOS

Conforme mencionado acima, o presente trabalho de pesquisa contribuiu para as áreas de projeto, planejamento do processo e manufatura de peças prismáticas baseadas em *features* remotamente via internet. Entretanto, podem ser apontados futuros trabalhos de pesquisa que poderão ser desenvolvidos visando estender o sistema proposto em pontos que não foram abordados, os quais são os seguintes:

- Acrescentar à biblioteca de *features* outras *features*, especialmente módulos para a criação de padrões de *features* (por exemplo, vários furos iguais distribuídos ao longo de uma reta, ou de um círculo, ou ao longo de uma trajetória qualquer). Isto implica na extensão da estrutura interna de armazenamento dos dados, bem como na extensão das classes encarregadas do cálculo matemático para a geração dos sólidos que representarão tais *features*. Acrescentar *features* do tipo protuberâncias à biblioteca.
- Adicionar ao sistema CAPP um módulo de otimização usando, por exemplo, a matriz de Halevi (1999) ou grafos E/OU (DE MELLO e SANDERSON, 1990). Isto permitirá aos usuários do sistema terem planos de processos com alternativas, que contribuem para flexibilizar a programação da produção.
- As condições de usinagem para as diferentes operações foram tiradas dos catálogos dos fabricantes de ferramentas, e em futuros trabalhos pode-se usar algoritmos, modelos matemáticos e técnicas de inteligência artificial para calcular os parâmetros mais adequados para cada operação.
- No presente trabalho de tese se consideraram rugosidades das superfícies da *feature* e suas tolerâncias dimensionais. Poderão ser considerados futuramente outros tipos de tolerâncias.
- O sistema poderá ser estendido de maneira a fazer uso de agentes inteligentes para pesquisar recursos em outros centros de manufatura disponibilizados na internet (máquinas, ferramentas).

- No futuro o sistema desenvolvido poderá ser estendido de maneira que as peças consideradas possam ser fabricadas em máquinas de quatro ou cinco eixos.
- A determinação automática dos pontos de apoio poderá ser implementada em trabalhos futuros.
- Podem haver situações em que ocorrem interações entre *features*, e tais situações poderão ser consideradas em futuras implementações.
- O sistema utiliza a abordagem de projeto por *features* “*Destruction by Machining Features*” (seção 3.3.4), onde sempre se remove material a partir do bloco da peça bruta. Futuramente poderia se considerar também a “Síntese por *features* de projeto”, onde não é necessário iniciar com o bloco da peça bruta.
- O sistema desenvolvido no presente trabalho poderia ser acoplado a um sistema de controle CNC aderente ao padrão ISO 14649, para se ter um cenário de manufatura CNC onde se dispense o código G.



## 9 REFERÊNCIAS

AGUANNO KEVIN, PMP®, MAPM, **Managing Agile Projects**, first edition, Multi-media Publications Inc., Ontario, Canada, 2005.

ALLEN R. D., NEWMAN S. T., HARDING J. A., ROSSO JUNIOR. R.S.U., **The design of a STEP-NC compliant agent based CAD/CAM system**, Proceedings of the 13th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2003), Tampa, FL, USA, pp. 530–540, 2003.

ALLEN RD, HARDING JA, NEWMAN ST, **The application of STEP-NC using agent-based process planning**. International Journal of Production Research, v. 43, issue 4, p. 655–670, 2005.

ÁLVARES, ALBERTO JOSÉ, **Uma metodologia para integração CAD/CAPP/CAM voltada para manufatura remota de peças rotacionais baseada na internet**, 249f, Tese (Tese em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 2005.

ÁLVARES ALBERTO JOSÉ, FERREIRA JOÃO CARLOS ESPÍNDOLA, **Metodologia para implantação de Laboratórios Remotos Via Internet na área de Automação da Manufatura**, II COBEF, Uberlândia, MG, 05/2003.

AMAITIK SM, ENGIN KILIC S, **An intelligent process planning system for prismatic parts using STEP features**, The International Journal of Advanced Manufacturing Technology, v. 31, p. 978 – 993, 2007.

BELL GAVIN, PARISI ANTHONY, PESCE MARK., **The Virtual Reality Modeling Language - Version 1.0 Specification**. Documento disponível em <http://www.web3d.org/x3d/specifications/vrml/VRML1.0/index.html>, último acesso em 14/07/2011.

BLANCHARD BENJAMIN S, VERMA DINESH, PETERSON ELMER L, **MANTAINABILITY: A Key to Effective Serviceability**

**and Maintenance Management**, John Wiley & Sons, inc. NY, USA, 1995.

BLANCHARD EUGENE, **Introduction to Networking**, disponível em [http://www.rigacci.org/docs/biblio/online/intro\\_to\\_networking/book1.htm](http://www.rigacci.org/docs/biblio/online/intro_to_networking/book1.htm), último acesso em 20/07/2011.

BOUVIER DENNIS J., **Getting Started with the Java 3D API Chapter 1**, Sun Microsystems, disponível em [http://java.sun.com/developer/onlineTraining/java3d/j3d\\_tutorial\\_ch1.pdf](http://java.sun.com/developer/onlineTraining/java3d/j3d_tutorial_ch1.pdf), California, USA, 2000.

CERVO Amado Luiz; BERVIAN Pedro; SILVA Roberto da. **Metodologia científica**. 6 ed. São Paulo, Pearson Prentice Hall, 242p, 2007.

CHEN X, ZHANG C, LAN H, ZHAI P, WU H. **A framework for CNC turning system based on STEP-NC**. Proceedings of SPIE 6040, ICMIT 2005: Mechatronics, MEMS, and Smart Materials, 2005.

CHOI I, SUH SH, KIM K, SONG M, JANG M, LEE BE. **Development process and data management of TurnSTEP: a STEP-compliant CNC system for turning**. International Journal of Computer Integrated Manufacturing, v. 19, issue 6, p. 546–558, 2006.

DAUM, BERTHOLD; MERTER, UDO. **Arquitetura de sistemas com XML**. 1. ed. Rio de Janeiro: Campus, 441p., 2002.

DECEMBER J., **Java Makes Executable Content Possible**. In:Java Unleashed, p. 4-22 Ed. Sams.net. Indianápolis, EUA. 1996.

DE MELLO L. S. H. and SANDERSON A. C, **AND/OR Graph Representation of Assembly Plans**, IEEE Transactions on Robotics and Automation, v.6, n. 2, 1990.

DE OLIVEIRA L. G., **Construção de um sistema de Blackboard para gestão de documentos usando XML**, 2004, Trabalho de Conclusão de curso em Sistemas da Informação, PUCMG, Arcos, MG.

EVBUOMWAN N F O, SIVALOGANATHAN S, JEBB A, **A survey of design philosophies, models, methods and systems**, Journal of Engineering Manufacture, v. 210, 1996.

FEENEY A. B, KRAMER T, PROCTOR F, HARDWICK M, LOFFREDO D, **STEP-NC Implementation – ARM or AIM?**, ISO TC184/SC4 WG3 Document N1223, 2003.

FICHTNER D, NESTLER A, DANG TN, SCHULZE A, CARLSEN U, SCHREIBER S, **Use of agents and neural networks for acquisition and preparation of distributed NC information to support NC planning**. International Journal of Computer Integrated Manufacturing, v. 19, issue 6, p.581–592, 2006.

GARRIDO J., MARÍN R., IGNACIO J., SÁEZ J., **Extensión del estándar ISO 10303 AP-238 para la automatización del acceso de datos de procesos CNC**, Revista Iberoamericana de Automática e informática Industrial, Comité español de automática, Vol 6, p 105 – 116, 2009.

HALEVI G, **Restructuring the Manufacturing Process Applying the Matrix Method**, CRC Press, 1999.

HARDWICK M. **Digital manufacturing using STEP-NC**, Tech. Paper, Society of Manufacturing Engineers, MS02-242, 2002.

HARDWICK M, LOFFREDO D. **Lessons learned implementing STEP-NC AP-238**. International Journal of Computer Integrated Manufacturing, v.19 issue 6, p.523–532, 2006.

HEISIG PETER, CLARSON JOHN P, VAJNA SANDOR, **Modelling and Management of Engineering Process**, Springer, London, UK, 2010.

HENDERSON-SELLERS, B.; GIORGINI, P. **Agent-oriented Methodologies: An Introduction**, Hershey, PA: Idea Group Publishing, 2005.

HEUSINGER S, ROSSO Jr. RSU, KLEMM P, NEWMAN ST, RAHIMIFARD S. **Integrating the CAx process chain for STEP-compliant NC manufacturing of asymmetric parts**. International

Journal of Computer Integrated Manufacturing, v. 19, issue 6, p. 533 – 545, 2006.

HIGHSMITH JIM, **Agile Software Development Ecosystems**, Addison Wesley Inc., 448p, 2002.

HORSTMANN C. e CORNELL G., **Core Java – fundamentos**, Markon books, São Paulo, 2001.

HORSTMANN CAY S, CORNELL GARY, **Core Java - Fundamentals**, v.1, ed.8, Prentice Hall, NY, USA, 2008.

IBM (International Business Machines), **Introduction to Networking Technologies**, US, 1994.

ISO 6983-1, **Numerical control of machines – Program format and definition of address words – Part 1: Data format for positioning, line motion and contouring control systems**, 1982.

ISO 6983-1, **Automation systems and integration -- Numerical control of machines -- Program format and definitions of address words -- Part 1: Data format for positioning, line motion and contouring control systems**, 2009.

ISO 6983-2, **Automation systems and integration -- Numerical control of machines -- Program format and definition of address words -- Part 2: Coding of miscellaneous functions M (class 1 to 8)**, 1982

ISO 10303-11, **Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual**, 1994.

ISO 10303-21, **Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: “Clear text encoding of the exchange structure**. 1994.

**ISO 10303 – 22, Industrial automation systems and integration -- Product data representation and exchange -- Part 22: Implementation methods: Standard data access interface, 1998.**

**ISO 10303-23, Industrial automation systems and integration – Product data representation and exchange – Part 23: Implementation method: C++ language binding to the standard data access interface.**

**ISO10303-24, Industrial automation systems and integration – Product data representation and exchange – Part 24: Implementation method: C language binding to the standard data access interface, 2001.**

**ISO 10303 – 27, Industrial automation systems and integration -- Product data representation and exchange -- Part 27: Implementation methods: Java TM programming language binding to the standard data access interface with Internet/Intranet extensions, 2000.**

**ISO 14649-1, Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers -- Part 1 -- Overview and fundamental principles, 2002.**

**ISO 14649 – 10, Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers -- Part 10: General process data, 2002.**

**ISO 14649 – 11, Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers -- Part 11: Process data for milling, 2002.**

**ISO 14649 – 111, Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers -- Part 111: Tools for milling machines, 2002.**

**ISO/DIS 10303-238, Industrial automation systems and integration – Product data representation and exchange – Part 238: Application Protocols: Application interpreted model for computerized numerical controllers, 2003.**

ISO/IEC 9075, **Information technology – Database languages – SQL**, 1992.

ISO/IEC 19775-2, 2010, **Information technology -- Computer graphics and image processing -- Extensible 3D (X3D) -- Part 2: Scene access interface (SAI)**, 2010.

ISO/IEC 9899, **Programming languages – C**, 1990.

ISO/TS10303-27, **International Organization for Standardization – Technical Specification – ISO/TS 10303-27 -Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 27: Implementation Methods: Java™ Programming Language Binding to the Standard Data Access Interface with Internet/Intranet Extensions**. 2000.

ISO/TS10303-28, **International Organization for Standardization – Technical Specification – ISO/TS 10303-28 -Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 28: Implementation Methods: XML Implementation of EXPRESS Schemas and Data**. 2002.

JENKINS NICK, **A Software Testing Primer: An Introduction to Software Testing**, 43p, disponível em <http://www.nickjenkins.net>, San Francisco, California, USA, 2008.

KOC M, NI J, LEE J. **Introduction of e-manufacturing**, Proceeding of the International Conference on Frontiers on Design and Manufacturing, Dalian, China, July 2002.

LAN H, LIU R, ZHANG C, **A multi-agent-based intelligent STEP-NC controller for CNC machine tools**, International Journal of Production Research, v. 46, i. 14, p. 3887-3907, 2008.

LEE JAY, **E-manufacturing—fundamental, tools, and transformation**, Journal of Robotics and Computer Integrated Manufacturing v.19, p501–507, 2003.

LEE W, BANG YB. **Design and implementation of an ISO14649-compliant CNC milling machine**. International Journal of Production Research, v. 41, issue 13, p. 3007–3017, 2003.

LEE W, BANG YB, RYOU MS, KWON WH, JEE HS. **Development of a PC-based milling machine operated by STEP-NC in XML format**, International Journal of Computer Integrated Manufacturing, v. 19, issue 6, p. 593–602, 2006.

LIU R, ZHANG C, NEWMAN ST, **A framework and data processing for interfacing CNC with AP238**, International Journal of Computer Integrated Manufacturing, v. 19, issue 6, p. 516 – 522, 2006.

LOFFREDO, David. **Fundamentals of STEP Implementation**. STEP Tolls, Inc., New York, USA. Disponível em URL <http://www.steptools.com/library/fundimpl.pdf>, outubro 2000, último acesso em 12/05/2011.

MALECK LAYED L. ABDEL, WOLF CARL, GUYOT PATRICE D., **Telemanufacturing: A flexible manufacturing solution**, International Journal of Production Economics, v.56 – 57, p 1-12, 1998.

MAEDER Willy, NGUYEN Van Khai, JACQUES Richard, STARK John, **Standardisation of the Manufacturing Process : the IMS STEP-NC project**, ESPRIT Project, disponível em URL: [http://www.step-nc.org/data/IPLnetWorkshop2002\\_paper\\_27082002.pdf](http://www.step-nc.org/data/IPLnetWorkshop2002_paper_27082002.pdf), set 2002.

MOLINA ARTURO, RAMÍREZ SANTAELLA ARMANDO, **Achieving e-Manufacturing: multihead control and web technology for the implementation of a manufacturing execution system**, Journal of Intelligent Manufacturing, Volume 17, Number 6, p715-724, 2006.

MOULTIS, N PITTS e KIRK, C., **XML: black book**. 1. ed. São Paulo: MAKRON Books, 627p., 2000.

MOURA, ANTONIO ÁLVARO DE ASSIS, **Contribuições para o Desenvolvimento de um Modelador Baseado em Form Features com Interface STEP - ISO 10303**, 2003, 146f., dissertação (dissertação em engenharia da produção), Universidade Metodista de Piracicaba.

NASSEHI A, ALLEN RD, NEWMAN ST. **The application of multi-agent systems for STEP-NC computer aided process planning of**

**prismatic components.** International Journal of Machine Tools & Manufacture, v. 46, issue 2, p. 559–574, 2006a.

NASSEHI A, ALLEN RD, NEWMAN ST. **Application of mobile agents in interoperable STEP-NC compliant manufacturing,** International Journal of Production Research, v. 44 (18–19), p. 4159 – 4174, 2006b.

NASSEHI A, NEWMAN ST, ALLEN RD. **STEP-NC compliant process planning as an enabler for adaptive global manufacturing.** Robotics and Computer-Integrated Manufacturing, v. 22 (5–6), 456–467, 2006c.

NASSEHI AYDIN, **The Realisation of CAD/CAM/CNC Interoperability in Prismatic Part Manufacturing,** Doctoral Thesis, Department of Mechanical Engineering, University of Bath, UK., 2007.

NEWMAN STEPHEN.T., **Integrated CAD/CAM/CNC manufacture for the 21st century,** The 14th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2004), 12–14 July 2004, Ryerson University, Toronto, Canada, 2004.

NEWMAN ST, ALLEN RD, ROSSO Jr. RSU, **CAD/CAM solutions for STEP-compliant CNC manufacture.** International Journal of Computer Integrated Manufacturing, v.16, issue 7–8, p.590–597, 2003.

NEWMAN S. T., NASSEHI X. W., ROSSO Jr R. S. U, WANG L., YUSOF Y., ALI L., LIU R., ZHENG L. Y., KUMAR S., VICHARE P., and DHOKIA V., **Strategic advantages of interoperability for global manufacturing using CNC technology,** Robotics and Computer-Integrated Manufacturing, v.24, i.6, p. 699 – 708, 2008.

NIST, **STEP The Grand Experience,** Manufacturing Engineering Laboratory. National Institute of Standards and Technology. Gaithersburg (MD), USA, 1999.

OFFODILE FELIX O, MALEK LAYED L. ABDEL, **The virtual manufacturing paradigm: The impact of IT/IS outsourcing on manufacturing strategy,** International Journal of Production Economics, v. 75, p. 147 – 159, 2002.



OMG, OBJECT MANAGEMENT GROUP, **Introduction to OMG's Unified Modeling Language™ (UML®)**, disponível em [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm), ultimo acesso em 16/08/2011.

PACHECO, O. N; HARBS, E; ROSSO JUNIOR, R. S. U; LEAL, A. B; HOUNSELL, M. S; FERREIRA, J. C. E, **Application of the STEP-NC Standard in a Computer Numerical Controlled Machining Device**. In: 21st International Congress of Mechanical Engineering - COBEM'2011, 2011, Natal - RN. 21st COBEM. ABCM - Associação Brasileira de Engenharia e Ciências Mecânicas, 2011.

ROSSO JUNIOR ROBERTO S U, ALLEN R D, NEWMAN STEPHEN T, **Future issues for CAD/CAM and intelligent CNC manufacture**, Proceedings of the 19th International Manufacturing Conference, Queen's University Belfast – N. Ireland, 2002.

ROSSO Jr. RSU, NEWMAN ST, RAHIMIFARD S. **The adoption of STEP-NC for the manufacture of asymmetric rotational components**, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, v.218, p. 1639-1644, November 1, 2004.

ROSSO JUNIOR, ROBERTO SILVIO UBERTINO. **STEP Compliant CAD/CAPP/CAM System For Rotational Asymmetric Parts**. 2005. 175 f. Thesis (Doctoral) - Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, 2005.

SCHENCK, D. A. e WILSON, P. R. **Information Modeling: The EXPRESS Way**. New York: Oxford University Press, 388 p, 1994.

SCRA – South Carolina Research Authority, **STEP Application Handbook ISO 10303**, version 3. North Charleston, SC, USA, June, 2006.

SHAH, J. J., MÄNTYLÄ, M., **Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications**, John Wiley Sons Inc, New York, 1995.

SHIN SJ, SUH SH, STROUD I, **Reincarnation of G-code based part programs into STEP-NC for turning applications**, Computer-Aided Design, v. 39, p. 1–16, 2007.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. 3 ed, 2001.

SOUZA de, ADRIANO FAGALI; ULBRICH CRISTIANE B. L., **Engenharia integrada por computador e sistemas CAD/CAM/CNC, princípios e aplicações**, ed Artliber, São Paulo, Brasil, 2009.

SUH S, NOH S, CHO Y, **A PC-based retrofitting toward CAD/CAM/CNC integration**, Journal Computers and Industrial Engineering, v.28, issue 1, 133–146, 1995.

SUH S.H., CHEON S.U., **A Framework for an Intelligent CNC and Data Model**, The International Journal of Advanced Manufacturing Technology, v. 19, issue 10, p. 727 – 735, 2002.

SUH S. H., CHO J.H., and HONG H.D., **On the architecture of intelligent STEPcompliant CNC**, Journal of Computer Integrated Manufacturing v15, p. 168–177, 2002a.

SUH S. H, CHUNG D. H, LEE B. E, CHO J. H, CHEON S. U, HONG H. D, **Developing an integrated STEP-compliant CNC prototype**. Journal of Manufacturing Systems, v.21, n.5, p.350 – 62, 2002b.

SUH S H, LEE B E, CHUNG D H, CHEON U S, **Architecture and implementation of a shop-floor programming system for STEP-compliant CNC**, Computer-Aided Design v.35, p.1069–1083, 2003.

SUH SH, CHUNG DH, LEE BE, SHIN S, CHOI I, KIM KM, **STEP-compliant CNC system for turning: data model, architecture, and implementation**. CAD Computer-Aided Design, v. 38, issue 6, p. 677–688, 2006.

TIWARI, M. K; JHA, KUMAR SUJIT, ANAND, Raj Bardhan. **Operation allocation and part type selection in E-manufacturing: An auction based heuristic supported by agent technology**, Robotics and Computer-Integrated Manufacturing, USA, vol 26, p. 312 – 324, 2010.

TORNINCASA STEFANO, CHIRONE EMILIO, **The Leonardo Webd Project: An Example of the Web3d technology applications for distance training and learning**, XIV Congreso Internacional de Ingeniería Gráfica, Santander, España, 2002.

VENKATESH S, ODENDAHI D, XU X, MICHALOSKI J, PROCTOR F, KRAMER T, **Validating portability of STEP-NC tool center programming**, In: Proceedings of the ASME international design engineering technical conferences and computers and information in engineering conference—DETC2005, p. 285–290, 2005.

VICHARE, P., NASSEHI, A., KUMAR, S., NEWMAN, S., ZHENG, L., and DHOKIA, V., **Towards a STEP-NC compliant model for representation of machine tools**, Proceedings of the 4th International Conference on Digital Enterprise Technology, 2007.

W3C, **Extensible Markup Language (XML)**, disponível em <http://www.w3.org/XML/>, último acesso em 26/05/2011.

WANG, H.P. e LI, J.K., **Computer-Aided Process Planning**, Advances in Industrial Engineering, Vol. 13, Elsevier, 1991.

WANG H., XU X.W., **A STEP-compliant 'Adaptor' for linking CAPP with CNC**, 34th International MATADOR Conference, 7th–9th July 2004, UMIST, Manchester, UK, 2004.

WECK, **STEP-NC—A new interface closing the gap between planning and shopfloor**, STEP-NC Workshop, WZL RWTH Aachen, <http://www.step-nc.org/>, Aachen, Germany, 2003.

WILSON, P. R. **A short history of CAD data transfer standards**. Journal IEEE Computer Graphics. & Applications. v.6. issue 6, p. 64-67, 1987.

XU X, and MAO J., “**A STEP-Compliant Collaborative Product Development System**”, Proceedings of the 33rd International Conference on Computers and Industrial Engineering, March 25-27, Ramada Plaza, Oriental Hotel, Jeju, Korea. CIE598, 2004.

XU XW, WANG J, **Development of a G-Code free, STEP-compliant CNC lathe**, International Mechanical Engineering Congress and Exposition (IMECE2004), p. 75 – 82, California, USA, 2004.

XU XW, **Realization of STEP-NC enabled machining**, Robotics and Computer-Integrated Manufacturing, v. 22, issue 2, p. 144–153, 2006.

XU X. W., NEWMAN S. T., **Making CNC machine tools more open, interoperable and intelligent – a review of the Technologies, Computers in Industry**, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, vol 57, págs 141 – 152, 2006.

XU XUN W., WANG LIHUI, RONG YIMING, **STEP-NC and Function Blocks for Interoperable Manufacturing**, IEEE, Transactions on Automation Science and Engineering, v.3, n.3, 2006.

YUSOF Y, **Exploring the Application of New Data Standard for CNC Machining**, wcecs, pp.201-210, Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science, 2008.

YUSOF Y, ROSSO Jr RSU, CASE K, NEWMAN ST. **The design of a STEP-NC compliant CAD/CAP/CAM system for the manufacture of rotational parts on a CNC turning centre, in ‘Innovations in manufacturing’**, Proceedings of the 23rd international manufacturing conference, Northern Ireland: Ulster, p. 19 – 28, 2006.

YUSRI YUSOF, NURUL ZAKIAH ZAMRI TAN and NOORDIANA KASSIM. **ISO14649 Code Generator for Intelligent Manufacture for STEP-NC Compliant Machining**, Asia Pacific Industrial Engineering & Management Systems Conference, Kitakyushu, Japan, p. 1937 – 1945. 1 nov. 2009.

ZAIDAT A., BOUCHER X., VINCENT, L., **A framework for organization network engineering and Integration**, Robotics and Computer-Integrated Manufacturing. Vol. 21, p. 259-271, 2005.

ZEID, IBRAHIM. CAD/CAM Theory and practice. McGraw-Hill: New York. 1991.

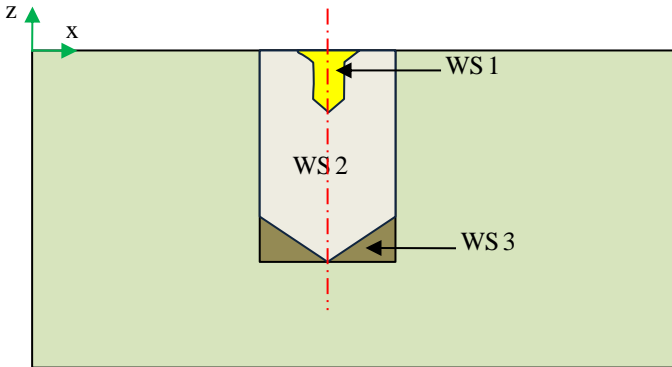
ZHANG X., LIU R., NASSEHI A., NEWMAN S. T., **A STEP-compliant process planning system for CNC turning operations**, Robotics and Computer – Integrated Manufacturing. vol 27 i2. p. 349 – 356. April 2011.

ZHAO FIONA., XU XUN, XIE SHANE., **STEP-NC enabled on-line inspection in support of closed-loop machining**, Robotics and Computer-Integrated Manufacturing, v. 24, i. 2, p. 200-216, 2008.

## APÊNDICE A – Mapeadoras de *workingsteps*

Este apêndice tem como finalidade mostrar quais regras foram seguidas para o mapeamento das diferentes *features* para *machining workingsteps*.

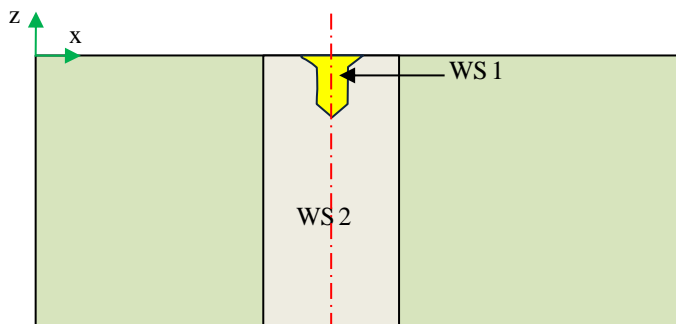
### 1. Furo com base plana (Desbaste):



<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Furação de centro	Broca de centro	nulo
WS 2	Furação	Broca helicoidal	WS 1
WS 3	Fresamento	Fresa de topo	WS 2

Figura A.1. Mapeamento de um furo base plana em *machining workingsteps*.  
Fonte: Criado pelo autor.

## 2. Furo passante (Desbaste)

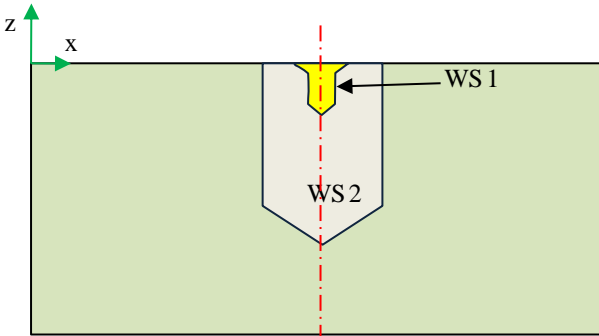


<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Furação de centro	Broca de centro	nulo
WS 2	Furação	Broca helicoidal	WS 1

Figura A.2. Mapeamento de um furo passante em *machining workingsteps*.  
 Fonte: Criado pelo autor.

Adicionalmente, se a broca helicoidal não consegue remover todo o diâmetro do furo, pode ser acrescentada uma outra operação de fresamento de contorno.

## 3. Furo com base cônica (Desbaste)



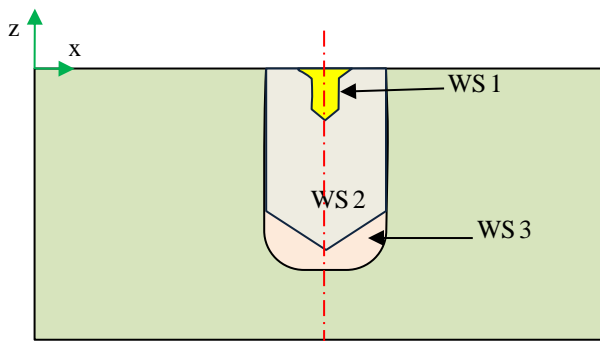
<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Furação de centro	Broca de centro	nulo
WS 2	Furação	Broca helicoidal	WS 1

Figura A.3. Mapeamento de um furo base cônica em *machining workingsteps*.

Fonte: Criado pelo autor.



## 4. Furo com base arredondada (Desbaste)

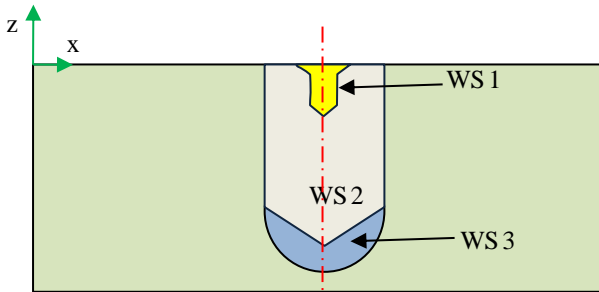


<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Furação de centro	Broca de centro	nulo
WS 2	Furação	Broca helicoidal	WS 1
WS 3	Fresamento	Fresa esférica	WS 2

Figura A.4. Mapeamento de um furo base arredondada em *machining workingsteps*.

Fonte: Criado pelo autor.

## 5. Furo com base esférica (Desbaste)

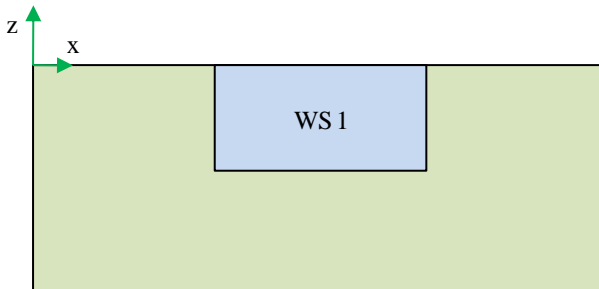


<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Furação de centro	Broca de centro	nulo
WS 2	Furação	Broca helicoidal	WS 1
WS 3	Fresamento	Fresa esférica	WS 2

Figura A.5. Mapeamento de um furo base esférica em *machining workingsteps*.

Fonte: Criado pelo autor.

## 6. Ranhura simples (Desbaste)

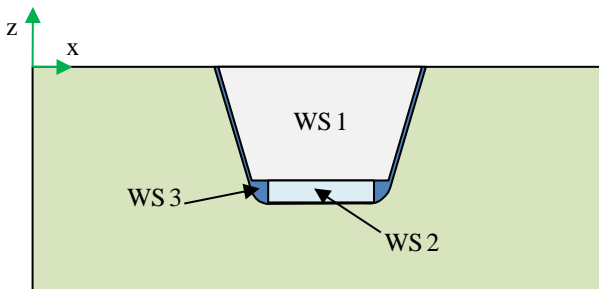


<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Fresamento desbaste	Fresa de topo	nulo

Figura A.6. Mapeamento de uma ranhura com perfil retangular em *machining workingsteps*.

Fonte: Criado pelo autor.

### 7. Ranhura perfil quadrado U (Desbaste)

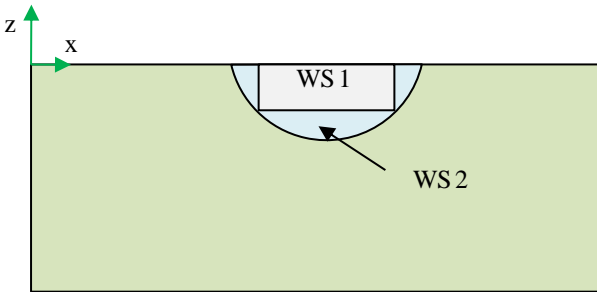


<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Fresamento desbaste	Fresa de topo com diâmetro maior	Nulo
WS 2	Fresamento desbaste	Fresa de topo com diâmetro menor	WS1
WS 3	Fresamento	Fresa esférica	WS 2

Figura A.7. Mapeamento de uma ranhura perfil quadrado U em *machining workingsteps*.

Fonte: Criado pelo autor.

## 8. Ranhura perfil parcial circular (Desbaste)

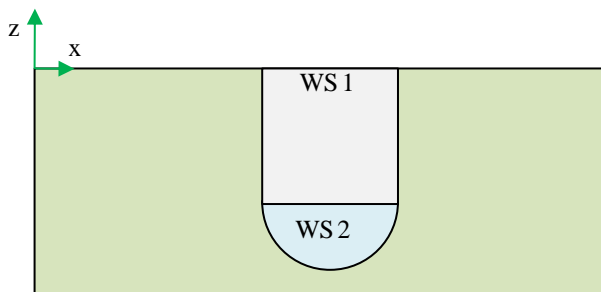


<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Fresamento desbaste	Fresa de topo	Nulo
WS 2	Fresamento	Fresa esférica	WS1

Figura A.8. Mapeamento de uma ranhura com perfil circular parcial em *machining workingsteps*.

Fonte: Criado pelo autor.

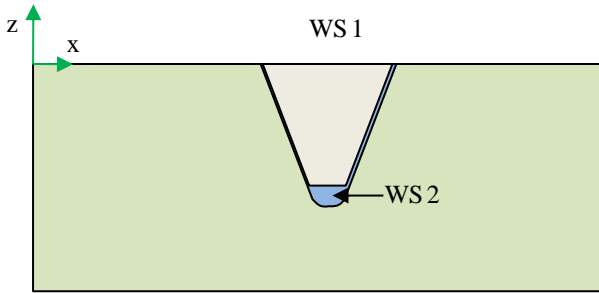
## 9. Ranhura perfil Rounded U (Desbaste)



<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Fresamento desbaste	Fresa de topo	Nulo
WS 2	Fresamento	Fresa esférica	WS1

Figura A.9. Mapeamento de uma ranhura perfil U em *machining workingsteps*.  
Fonte: Criado pelo autor.

## 10. Ranhura perfil Vee (Desbaste)



<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Fresamento desbaste	Fresa de topo	Nulo
WS 2	Fresamento	Fresa esférica	WS1

Figura A.10. Mapeamento de uma ranhura perfil Vee em *machining workingsteps*.

Fonte: Criado pelo autor.

## 11. Ranhura perfil genérico (Desbaste)

Descrito na seção 7.2.2.

## 12. Degrau (Desbaste)

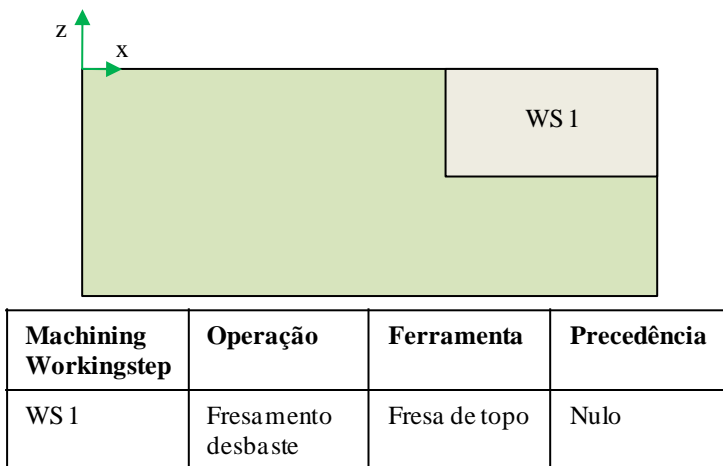


Figura A.11. Mapeamento de um degrau em *machining workingsteps*.  
Fonte: Criado pelo autor.

### 13. Cavidade com fundo plano e passante (Desbaste)

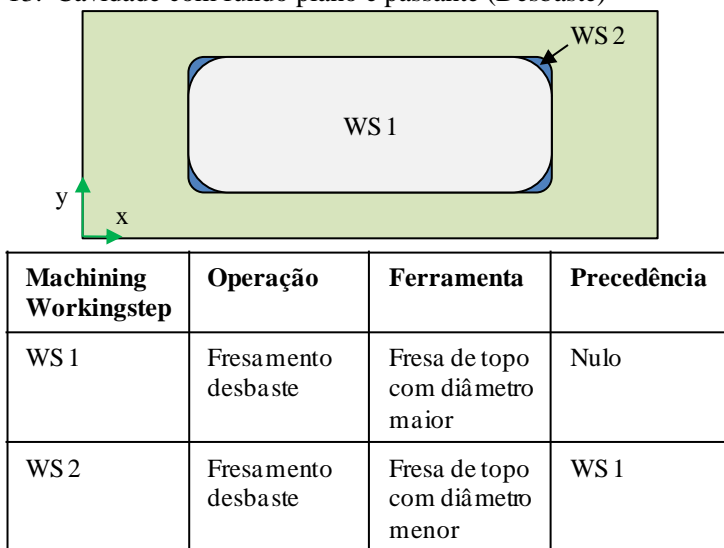
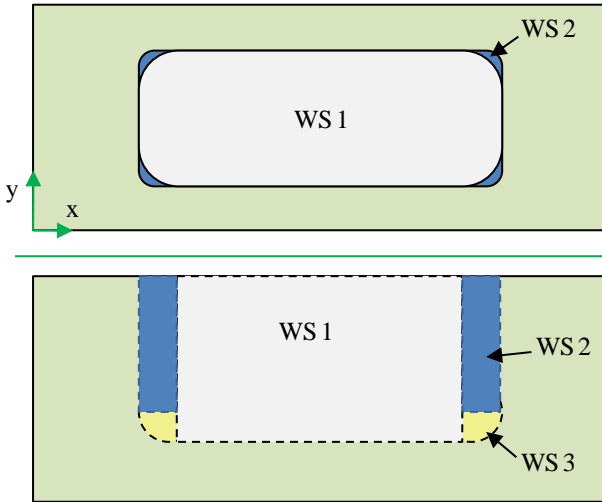


Figura A.12. Mapeamento de uma cavidade retangular com fundo plano em *machining workingsteps*.  
Fonte: Criado pelo autor.

## 14. Cavidade com fundo arredondado (Desbaste)



<b>Machining Workingstep</b>	<b>Operação</b>	<b>Ferramenta</b>	<b>Precedência</b>
WS 1	Fresamento desbaste	Fresa de topo com diâmetro maior	Nulo
WS 2	Fresamento desbaste	Fresa de topo com diâmetro menor	WS 1
WS 3	Fresamento	Fresa esférica	WS 2

Figura A.13. Mapeamento de uma cavidade retangular com fundo arredondado em *machining workingsteps*.

Fonte: Criado pelo autor.



## APÊNDICE B – Fluxo de informações bidirecional

Neste apêndice é apresentado o caso de se usinar a face XY de uma peça denominada “Boxy” (Figura B.1) com diferentes conjuntos de ferramentas. Não é raro que mudanças aconteçam no chão de fábrica, por exemplo, parâmetros de corte diferentes dos calculados pelo programa talvez sejam mais eficientes, ou novas ferramentas que entram no almoxarifado da empresa. Logo, se a empresa usa o arquivo neutro no formato STEP parte 21, a tarefa de modificação deste arquivo é muito mais fácil do que tentar reutilizar os programas NC feitos para outras condições. Usando-se o arquivo neutro nesse formato resulta em um fluxo do chão de fábrica de volta para os níveis superiores.

Supõe-se neste caso que a empresa possua o grupo de ferramentas “A”, e por alguma razão ela trocou a terceira ferramenta por duas outras, formando o conjunto de ferramentas “B” (tabela B.1).

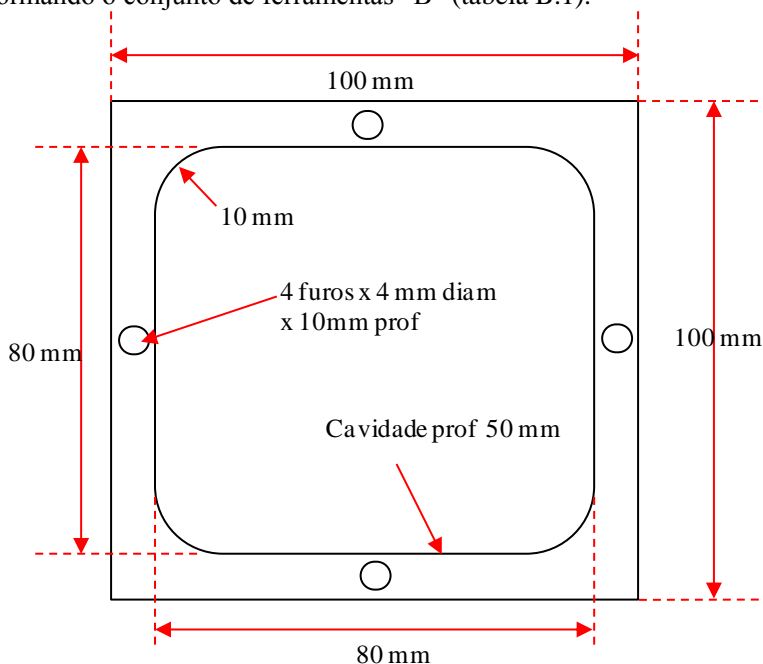


Figura B.1 Face XY da peça “Boxy”

**Tabela B.1. Conjuntos diferentes de ferramentas para a usinagem da mesma peça.**



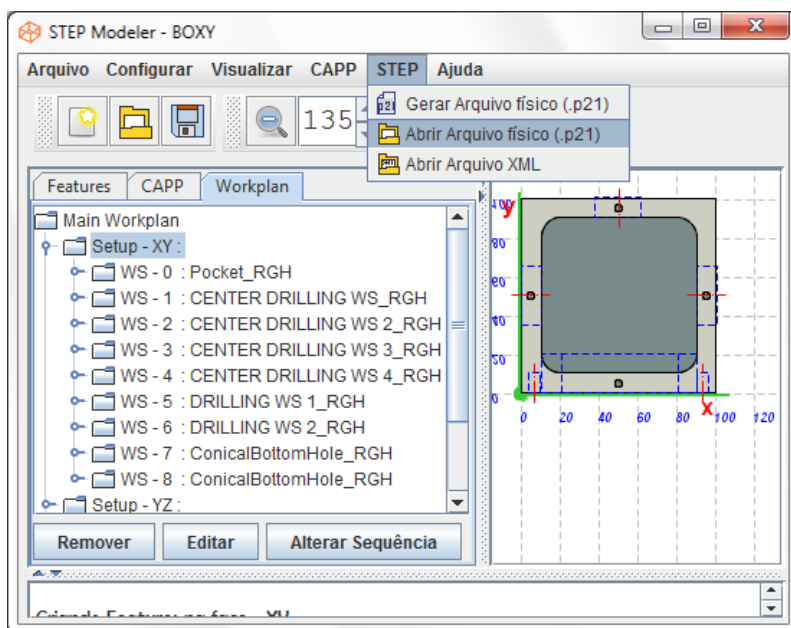


Figura B.3. Abrir arquivo físico.

Fonte: Criado pelo autor.

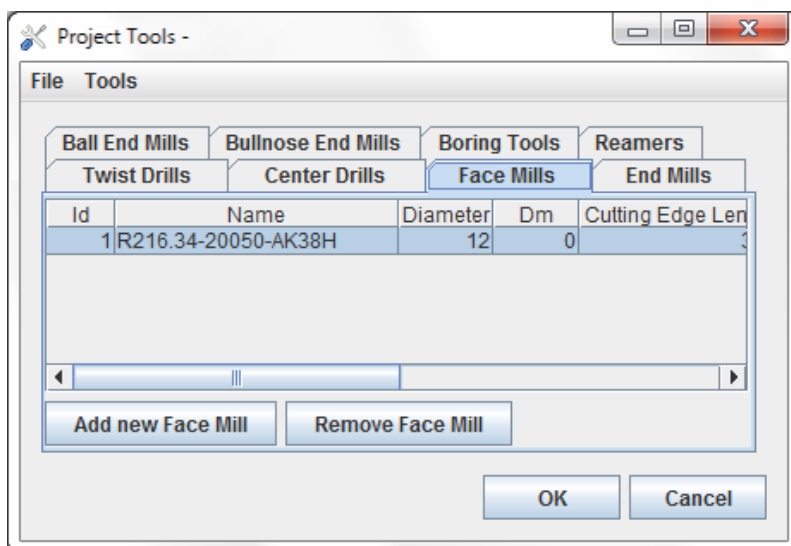


Figura B.4. Fresas disponíveis no grupo de ferramentas “A” (Fresa de 12 mm) para o fresamento da cavidade.

Fonte: Criado pelo autor.

A criação de um novo arquivo “.21” que contemple a utilização do novo conjunto de ferramentas “B” implica em atualizar a tabela de ferramentas disponíveis com as novas ferramentas (Fresa de 20 mm e Fresa de 25 mm).

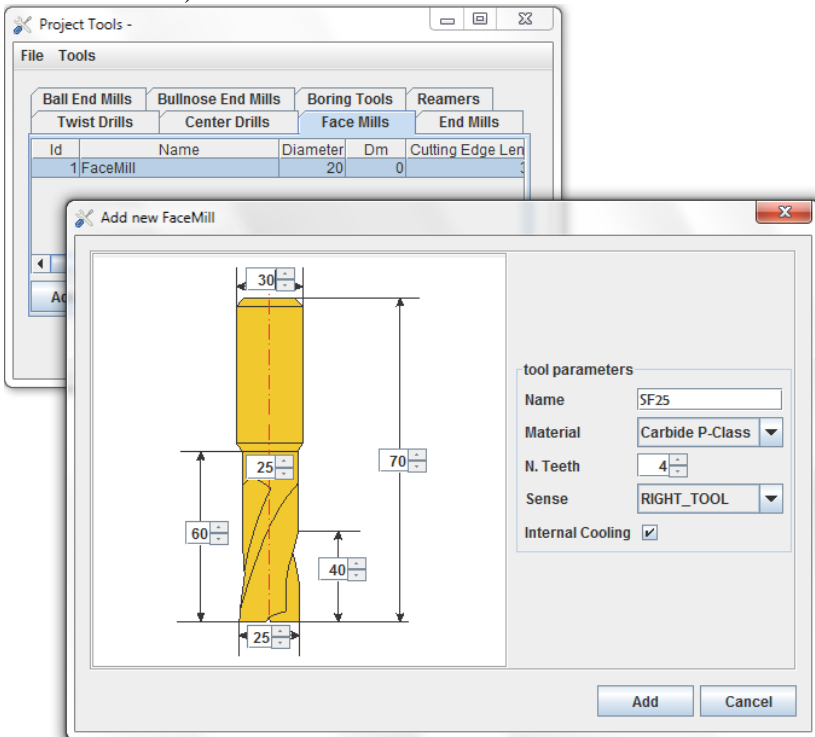


Figura B.5. Adição das novas fresas à tabela de ferramentas do projeto.

Fonte: Criado pelo autor.

Agora simplesmente se gera o novo plano do processo por meio do menu “CAPP” da tela principal submenu “gerar plano do processo”. Logo, pode-se gerar um novo arquivo físico (menu “STEP”, submenu “gerar arquivo físico (.p21)”).

```

#212=DRILLING($,$,'Drilling',5.0,#220,#213,#218,#219,$,10.727940
468532404,$,$,$,
#221);
#213=MILLING_CUTTING_TOOL('R840-0400-x0-AyA',#214,(#216),$,$,$);
#214=TWIST_DRILL(#215,2,.LEFT,$,24.0);
#215=TOOL_DIMENSION(4.0,0.3490658503988659,0.0,17.0,0.0,0.0,0.0);
#216=CUTTING_COMPONENT(66.0,#217,$,$,$);
#217=MATERIAL('ISO - P',$,$);
#218=MILLING_TECHNOLOGY(0.175,-TCP.,110.0,8753.0,$.F.,.F.,.F.,$);
#219=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,$.T.,$,$,$);
#220=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#221=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#222=MACHINING_WORKINGSTEP('D
WS4_RGH',#21,#165,#212,$);
#223=WORKPLAN('workplan setup
XY',(#67,#82,#106,#130,#154,#178,#189,#200,#211,#222),
$,#16,$);
#224=CENTER_DRILLING($,$,'Center
Drilling' 5.0 #225 #226 #231 #232 $ 9.5 $ $ $ #233);

```

Figura B.6. Arquivo físico gerado para o grupo de ferramentas “B” (10 *Machining workingsteps* no *workplan*).

Fonte: Criado pelo autor.

Este novo arquivo físico (figura B.6) tem um *machining workingstep* a mais pois subdivide-se a remoção do material da cavidade retangular (antes era só com fresa de 12 mm) em dois *machining workingsteps*, devido à existência de uma ferramenta que consegue remover maior volume de material em menor tempo (fresa de 25 mm de diâmetro).

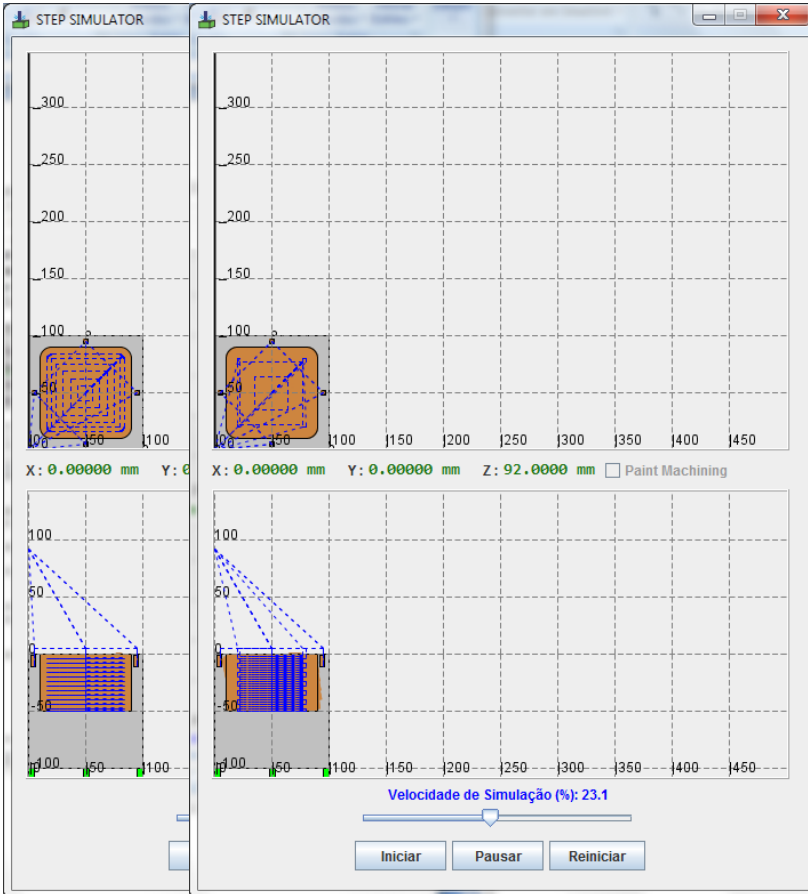
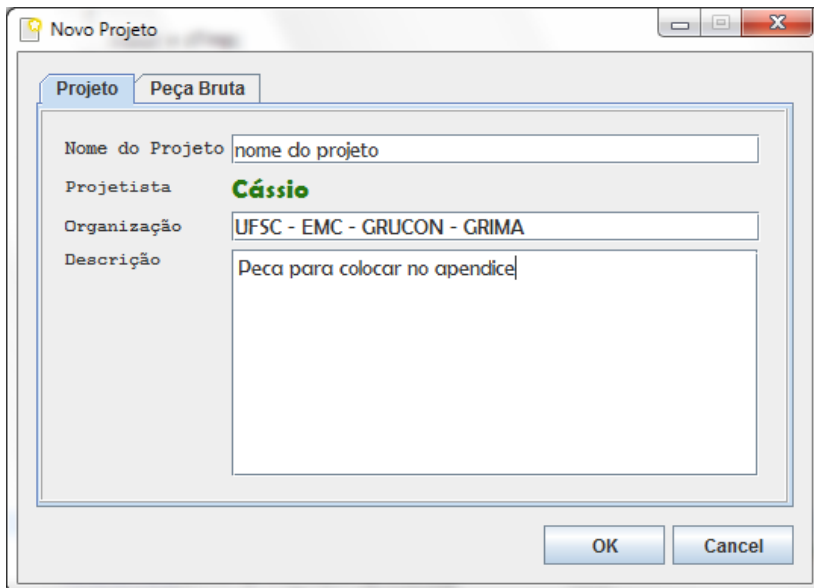


Figura B.7. Diferentes movimentações: na esquerda, movimentação com grupo de ferramentas “A”. Na direita, movimentação com grupo de ferramentas “B”.  
Fonte: Criado pelo autor.

## APÊNDICE C – Utilização do sistema

Neste apêndice se apresenta um exemplo de como criar peças, determinar as ferramentas, criar o *workplan*, editar os *machining workingsteps*, gerar o arquivo físico e exportá-lo como XML.

### 1. Criação da peça de trabalho:



The image shows a software dialog box titled "Novo Projeto". It has two tabs: "Projeto" (selected) and "Peça Bruta". The "Projeto" tab contains the following fields:

- Nome do Projeto: nome do projeto
- Projetista: Cássio
- Organização: UFSC - EMC - GRUCON - GRIMA
- Descrição: Peca para colocar no apêndice

At the bottom right of the dialog, there are two buttons: "OK" and "Cancel".

Figura C.1. Os dados colocados nesta janela aparecerão no cabeçalho do arquivo físico, na seção “HEADER”.

Fonte: Criado pelo autor.

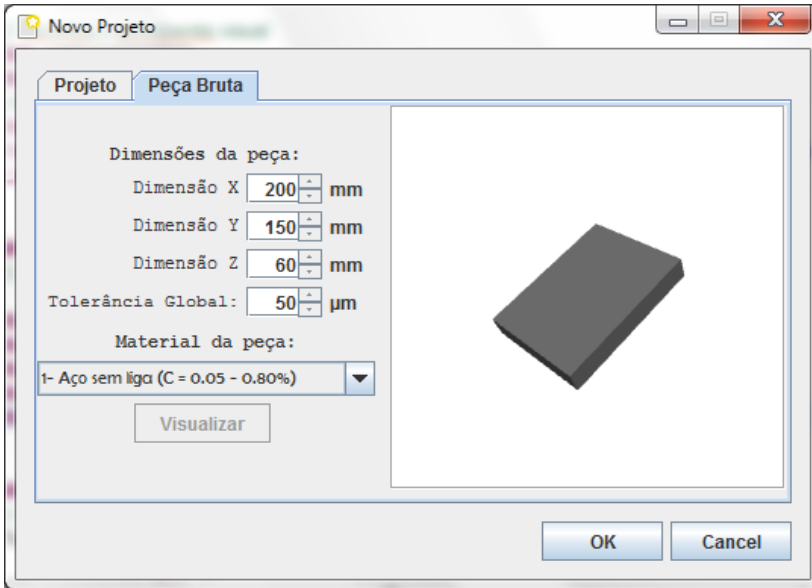


Figura C.2. Dimensões e material da peça bruta.

Fonte: Criado pelo autor.

## 2. Seleção da face na que será inserida a *feature*.

No menu “configurar”, submenu “plano de trabalho”, pode-se escolher qualquer uma das seis faces do bloco.



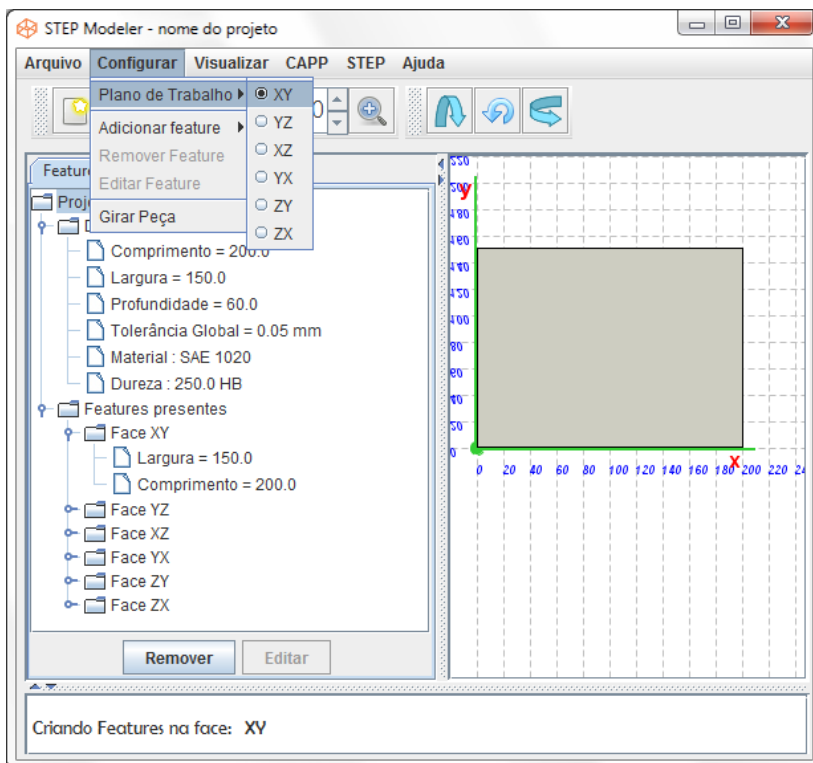


Figura C.3. Configuração da face de trabalho.

Fonte: Criado pelo autor.

### 3. Inserção de *features* na peça de trabalho.

Pode-se escolher dentre várias *features* disponíveis na biblioteca (Menu “configurar”, submenu “adicionar *feature*”)

#### (a) Inserção de um perfil de Bèzier em toda a face.

Marcando o *checkbox* “Em toda a peça” (ver figura C.4), a janela distribui uniformemente os pontos de controle ao longo da face inteira. O botão “visualizar perfil” permite ver como o perfil ficaria ao longo da peça.

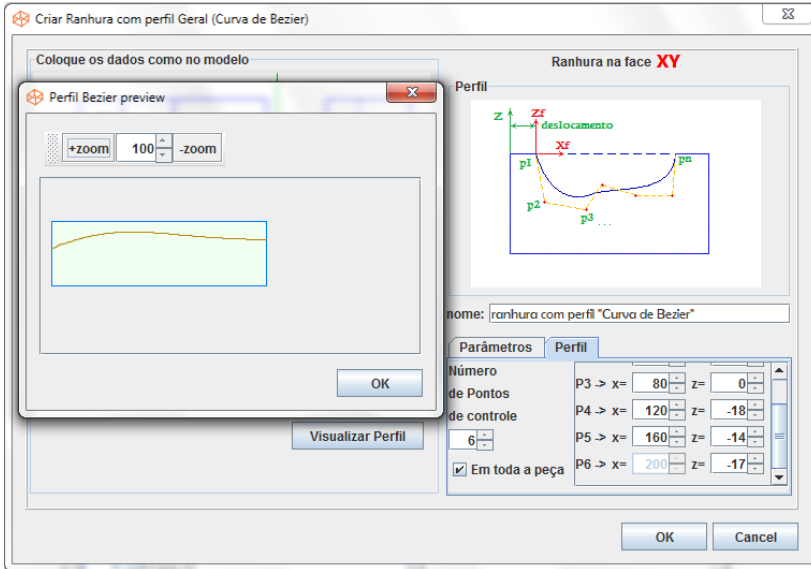


Figura C.4. Interface de criação da feature perfil Bèzier.  
Fonte: Criado pelo autor.

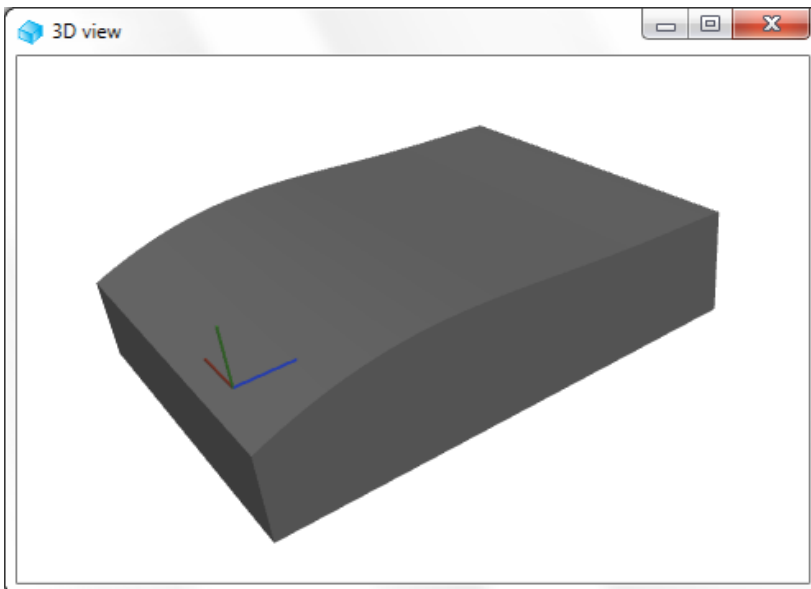


Figura C.5. Perfil de Bèzier inserido na face XY.  
Fonte: Criado pelo autor.

(b) Inserção de uma cavidade com fundo arredondado.

Deve-se notar que a tolerância dos raios e a textura destas superfícies são de  $\pm 10 \mu\text{m}$  e  $10 \mu\text{m}$  respectivamente.

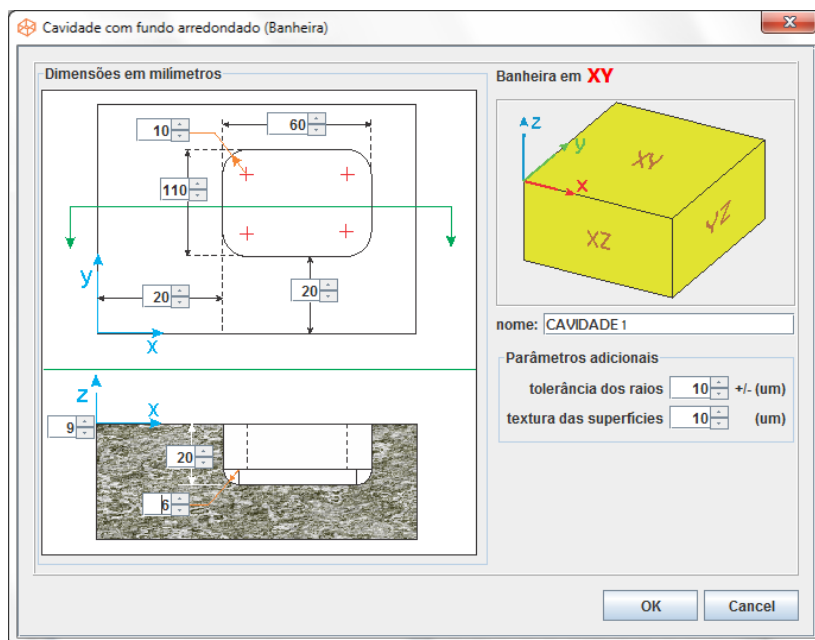


Figura C.6. Interface de criação de uma cavidade com fundo arredondado  
Fonte: Criado pelo autor.

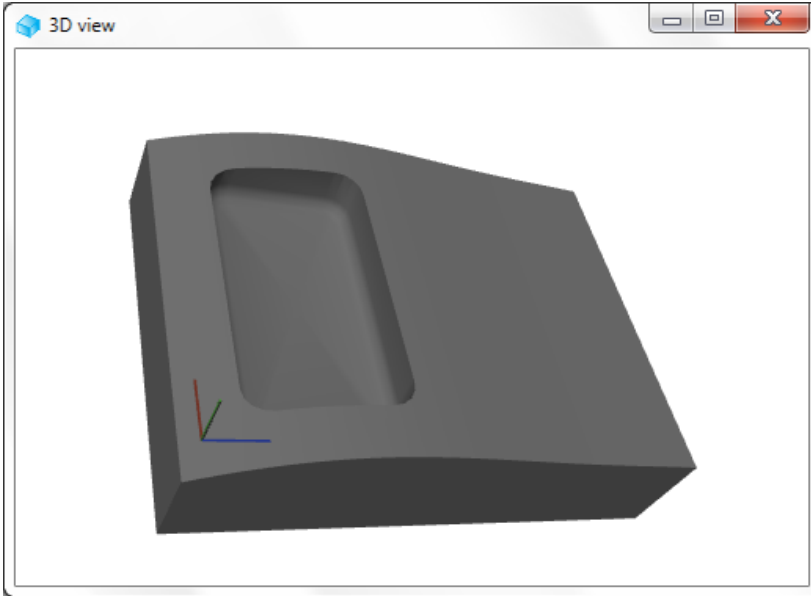


Figura C.7. Modelo gerado em 3D.

Fonte: Criado pelo autor.

(c) Inserção de uma ranhura perfil quadrado U.

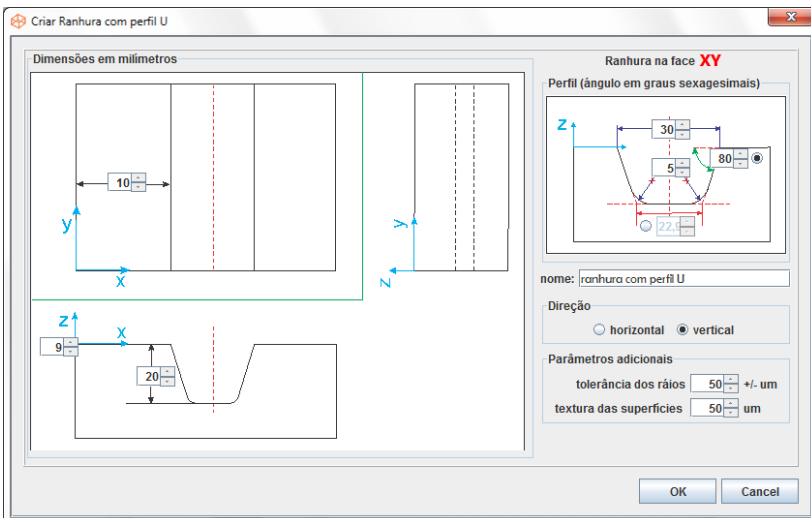


Figura C.8. Janela de entrada dos dados da ranhura perfil quadrado U.

Fonte: Criado pelo autor.

O perfil da ranhura pode ser definido especificando-se a largura superior e a largura inferior ou o ângulo de inclinação da parede do perfil (figura C.8).

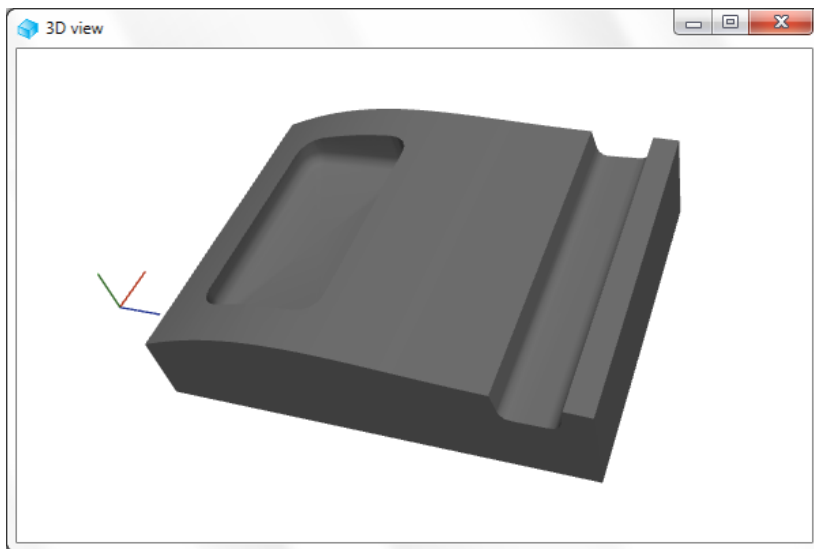


Figura C.9. Sólido resultante da inserção da ranhura perfil quadrado U.  
Fonte: Criado pelo autor.

#### (d) Inserção de furos passantes

Deve-se marcar a opção “passante” na janela, e adicionalmente pode-se especificar a tolerância do diâmetro e a rugosidade da superfície.

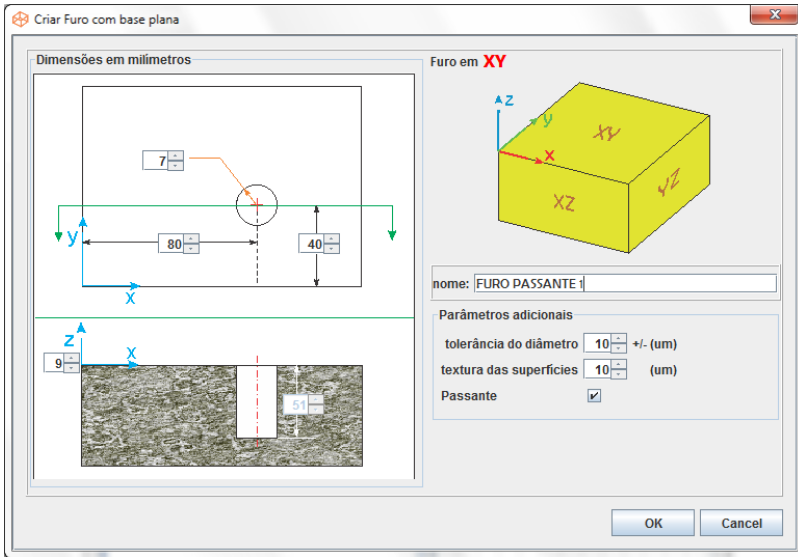


Figura C.10. Janela de criação do furo passante.

Fonte: Criado pelo autor.

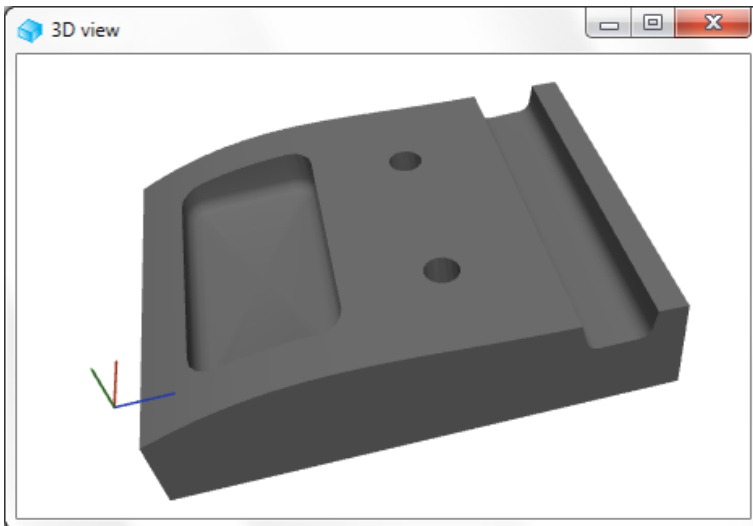


Figura C.11. Sólido resultante da inserção dos furos.

Fonte: Criado pelo autor.

#### 4. Criação de *features* em outras faces.

Selecione a face XZ como no item 2 deste apêndice, para posteriormente serem introduzidas as *features*.

(a) Inserção de furos com base esférica

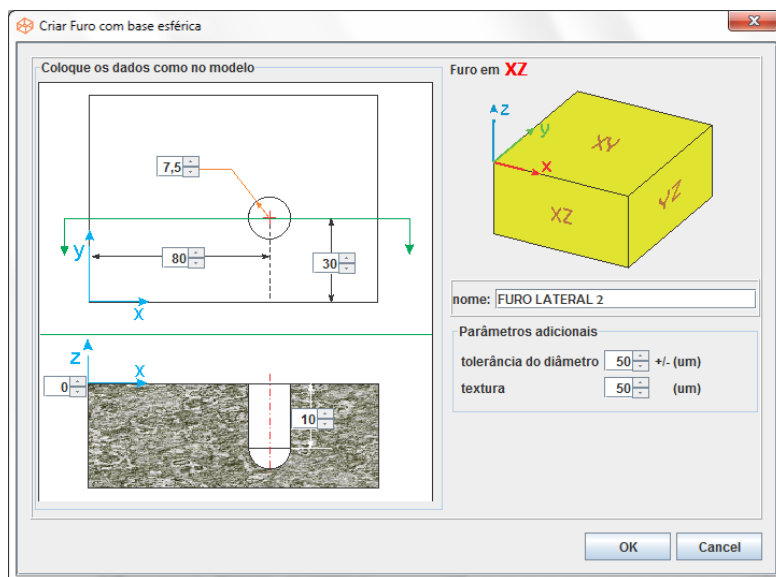


Figura C.12. Janela de criação do furo esférico.  
Fonte: Criado pelo autor.

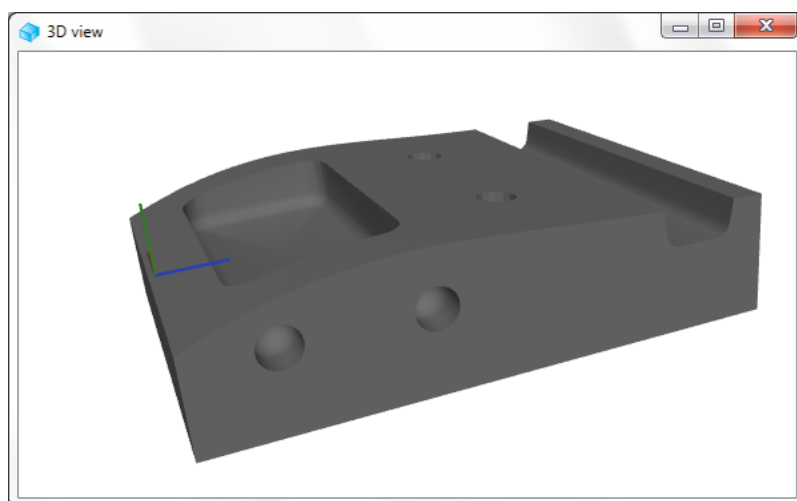


Figura C.13. Sólido resultante.

Fonte: Criado pelo autor.

### 5. Definição de um conjunto de ferramentas de usinagem

No menu CAPP, submenu “Gerenciar ferramentas”, pode-se definir o conjunto das ferramentas de usinagem com as quais o sistema gerará o plano do processo.

Por “default” aparece uma tabela vazia( figura C.14), pois ainda não se associou nenhum conjunto de ferramentas.

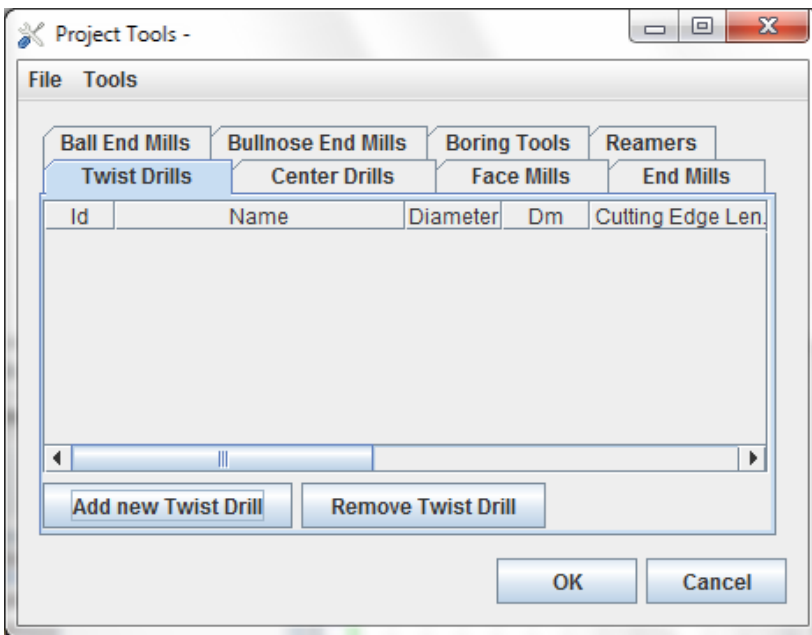


Figura C.14. Janela que mostra as ferramentas associadas ao projeto.

Fonte: Criado pelo autor.

Algumas das possibilidades de serem definidas estas ferramentas são listadas abaixo:

- (a) Criando-se manualmente botão “add new...”
- (b) Selecionando-se ferramentas de um catálogo de ferramentas armazenadas no servidor
- (c) Recuperando-se as ferramentas usadas em um projeto anterior.



(d) Adicionando-as automaticamente a partir do catálogo de ferramentas no servidor.

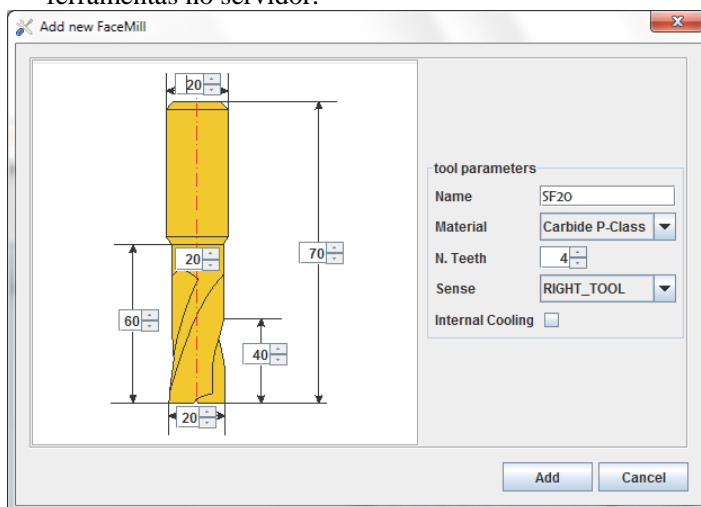


Figura C.15. Janela que adiciona uma nova fresa de topo.  
Fonte: Criado pelo autor.

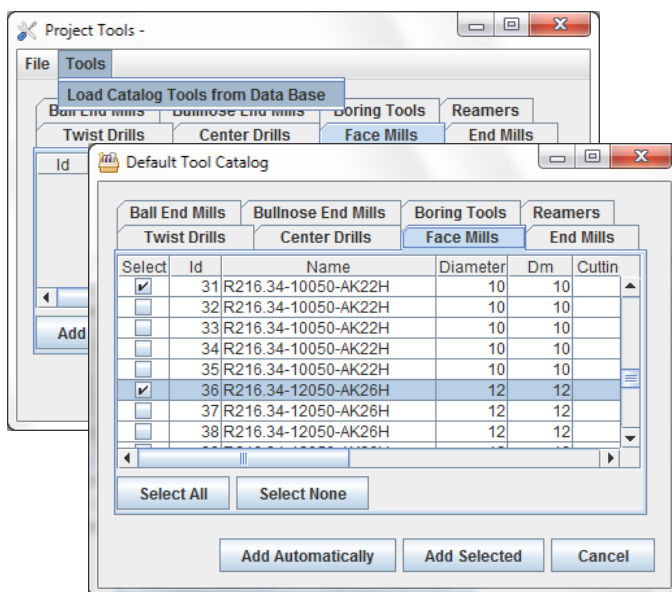


Figura C.16. Ferramentas selecionadas a partir do catálogo no servidor.  
Fonte: Criado pelo autor.

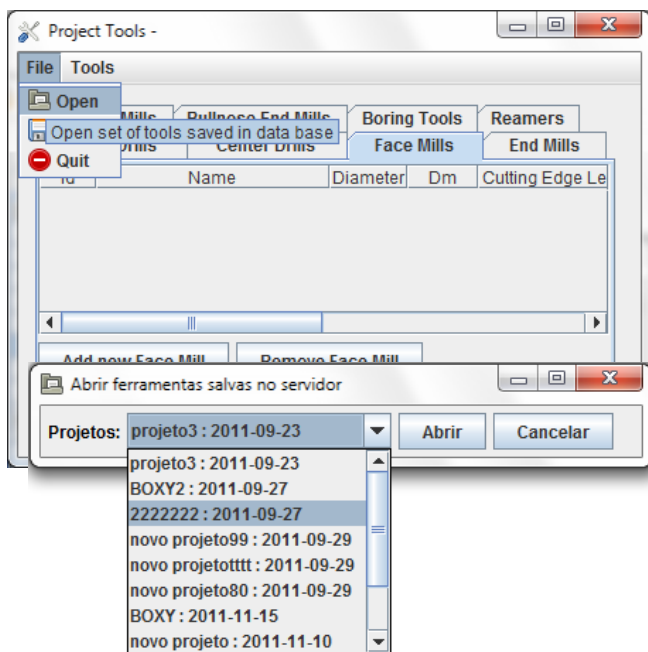


Figura C.17. Janela que permite recuperar as ferramentas de um projeto anterior.

Fonte: Criado pelo autor.

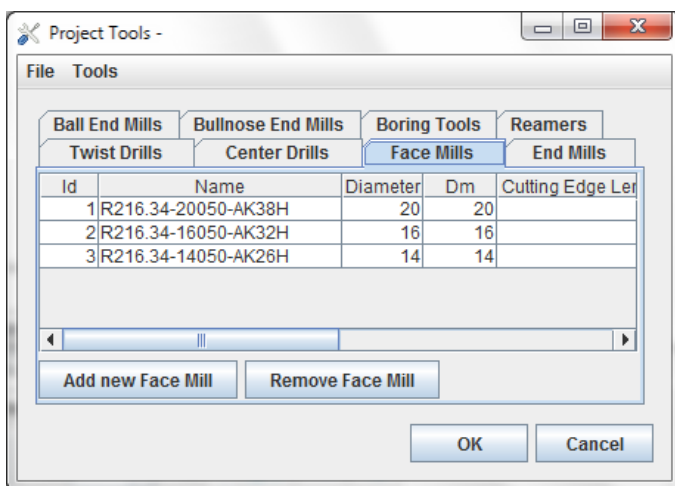


Figura C.18. Janela com as ferramentas (fresas) do projeto atual.

Fonte: Criado pelo autor.

## 6. Criação do *workplan*

A criação de um *workplan* pode ser realizada clicando-se no menu “CAPP”, submenu “Gerar plano do processo”. Como resultado tem-se os *machining workingsteps*, que removerão os volumes necessários para se obter a peça desejada (figura C.19).

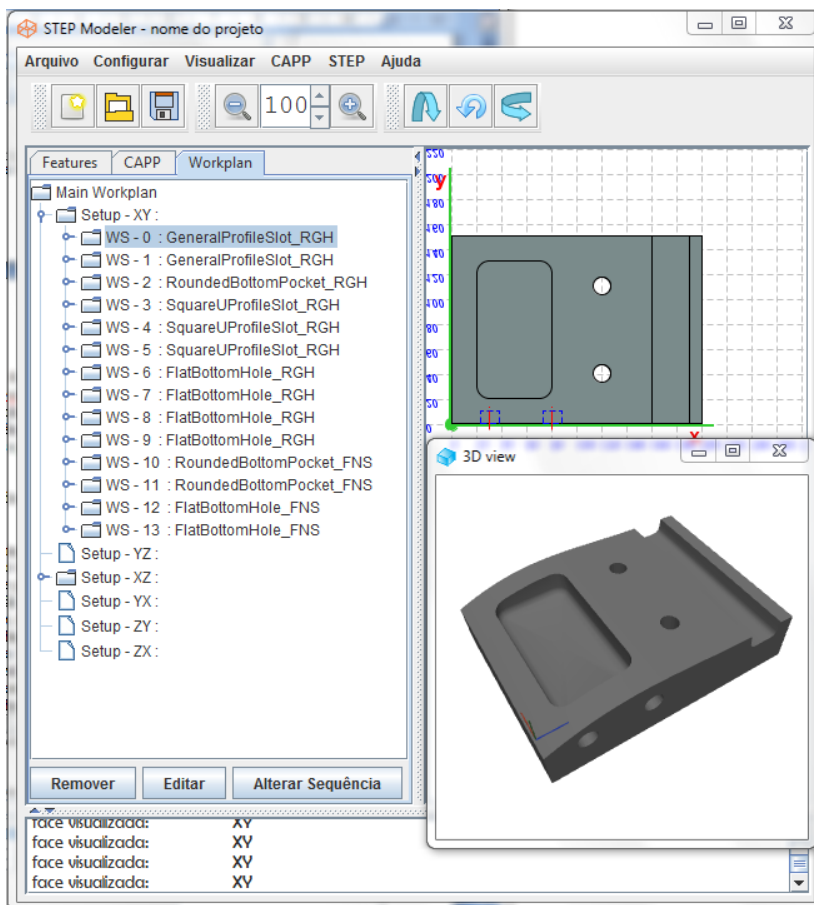


Figura C.19. Estrutura de árvore com os *machining workingsteps* na ordem de execução.

Fonte: Criado pelo autor.

## 7. Edição dos *machining workingsteps*

No campo da esquerda da interface principal, há o botão “Editar”. Ao selecionar-se um *machining workingstep* na estrutura de árvore, e clicando-se o botão “editar”, uma interface aparece (figura C.20), por meio da qual pode-se ajustar os parâmetros principais que compõem o *machining workingstep* (operação, ferramenta e parâmetros de corte).

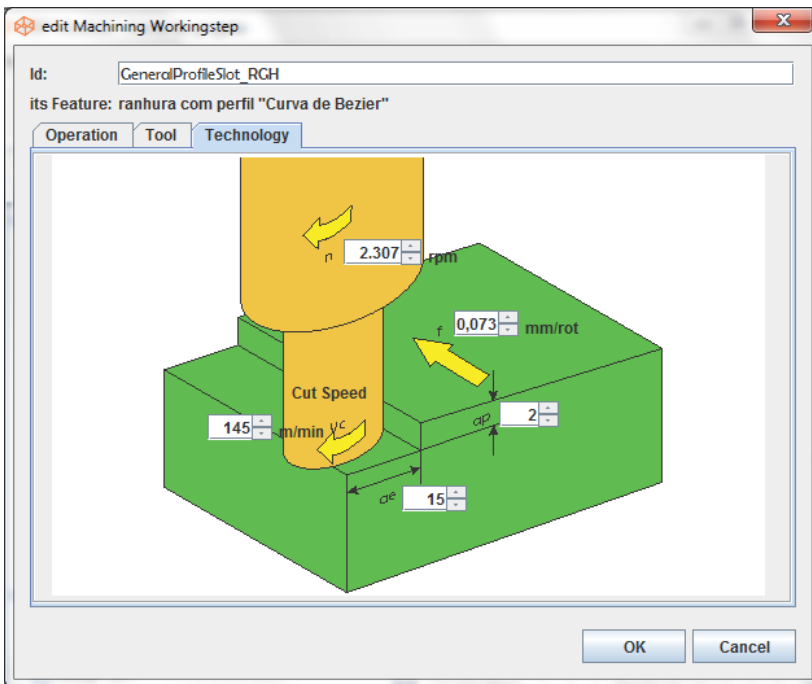


Figura C.20. Interface gráfica onde se podem mudar os parâmetros de usinagem do machining workingstep.

Fonte: Criado pelo autor.

## 8. Determinação dos pontos de apoio

Ao clicar-se no menu “CAPP”, submenu “Definir pontos de apoio”, a interface gráfica aparece (figura C.21). Na versão atual, apenas a parte manual está funcionando.

Para serem posicionados os pontos de apoio, deve-se selecionar a opção “manual” e a quantidade de pontos desejada. Uma tabela aparecerá, e as coordenadas poderão ser inseridas. Os pontos de apoio aparecem em linhas descontínuas e em fundo azul celeste.

Ao pressionar-se o botão “confirmar”, os pontos são armazenados e associados à peça (eles aparecem no arquivo step.21).

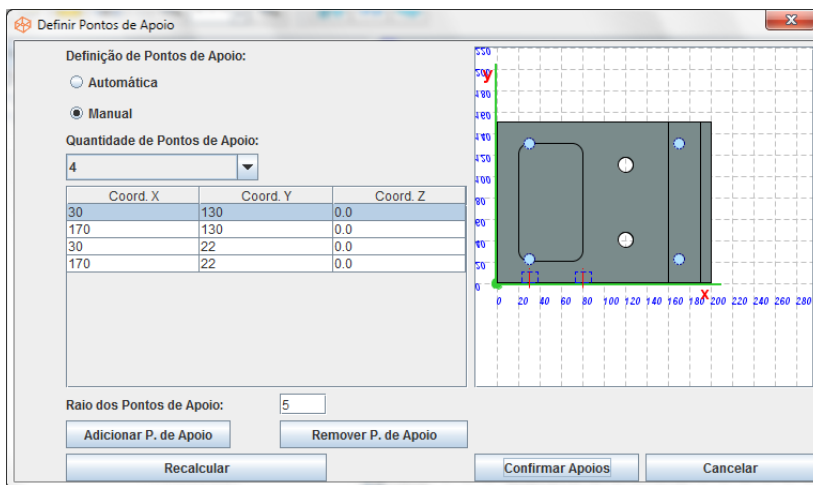
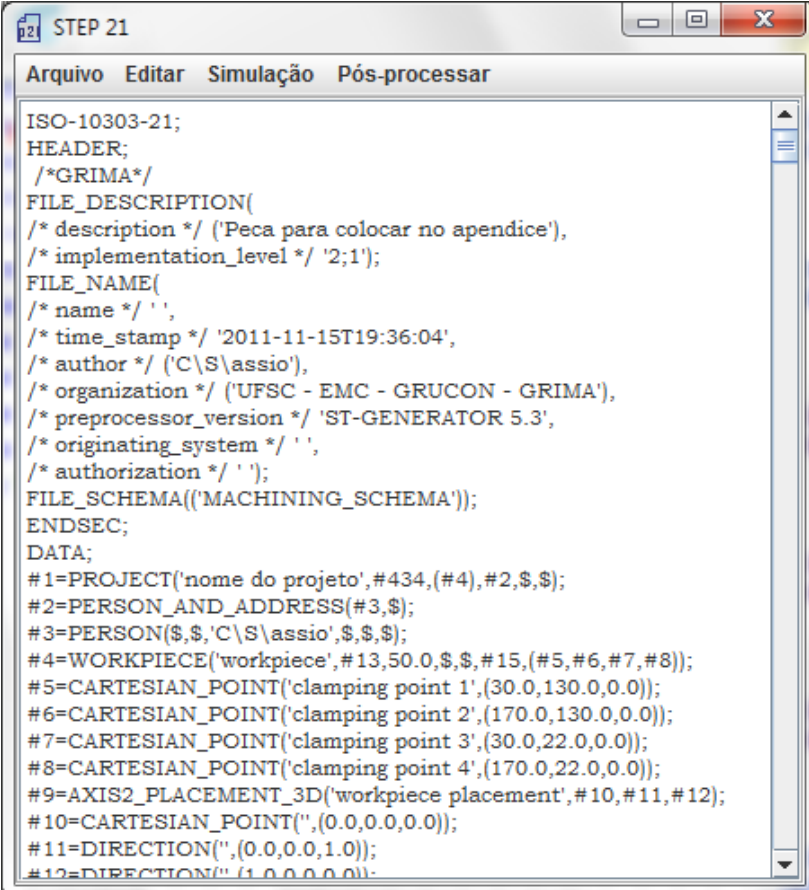


Figura C.21. Interface gráfica onde se podem inserir os pontos de apoio.  
Fonte: Criado pelo autor.

## 9. Geração do arquivo neutro STEP-NC.

Clicando-se no menu “STEP”, submenu “Gerar Arquivo Físico”, o arquivo STEP-NC (parte 21 da norma STEP) é gerado (figura C.22).



```

STEP 21
Arquivo  Editar  Simulação  Pós-processar

ISO-10303-21;
HEADER;
/*GRIMA*/
FILE_DESCRIPTION(
/* description */ ('Peca para colocar no apendice'),
/* implementation_level */ '2;1');
FILE_NAME(
/* name */ ' ',
/* time_stamp */ '2011-11-15T19:36:04',
/* author */ ('C\S\assio'),
/* organization */ ('UFSC - EMC - GRUCON - GRIMA'),
/* preprocessor_version */ 'ST-GENERATOR 5.3',
/* originating_system */ ' ',
/* authorization */ ' ');
FILE_SCHEMA(('MACHINING_SCHEMA'));
ENDSEC;
DATA;
#1=PROJECT('nome do projeto',#434,(#4),#2,$,$);
#2=PERSON_AND_ADDRESS(#3,$);
#3=PERSON($,$,'C\S\assio',$,$,$);
#4=WORKPIECE('workpiece',#13,50.0,$,$,#15,(#5,#6,#7,#8));
#5=CARTESIAN_POINT('clamping point 1',(30.0,130.0,0.0));
#6=CARTESIAN_POINT('clamping point 2',(170.0,130.0,0.0));
#7=CARTESIAN_POINT('clamping point 3',(30.0,22.0,0.0));
#8=CARTESIAN_POINT('clamping point 4',(170.0,22.0,0.0));
#9=AXIS2_PLACEMENT_3D('workpiece placement',#10,#11,#12);
#10=CARTESIAN_POINT("",(0.0,0.0,0.0));
#11=DIRECTION("",(0.0,0.0,1.0));
#12=DIRECTION("",(1.0,0.0,0.0));

```

Figura C.22. Arquivo físico gerado.

Fonte: Criado pelo autor.

## 10. Exportar como arquivo XML.

Isto é efetuado clicando-se no menu “arquivo” da janela que mostra o arquivo físico, e selecionando-se o submenu “Exportar como XML”. A figura C.23 mostra um trecho deste arquivo.

```

<?xml version="1.0" encoding="UTF-8"?>
- <iso_10303_28 version="PDTS" representation_category="LB">
  - <iso_10303_28_header>
    <document_name>Peca para colocar no apendice</document_name>
    <time_stamp>2011-11-16T14:46:03</time_stamp>
    <author>Cássio</author>
    <originating_organization>UFSC - EMC - GRUCON - GRIMA</originating_organization>
    <authorization></authorization>
    <originating_system></originating_system>
    <preprocessor_version>ST-GENERATOR 5.3</preprocessor_version>
  </iso_10303_28_header>
- <express_data id="m0" name="STEP_NC">
  - <schema_instance schema_name="machining_schema">
    - <entity_instance id="e5" express_entity_name="axis2_placement_3d">
      - <attribute_instance express_attribute_name="ref_direction">
        <entity_instance_ref refid="e8"/>
      </attribute_instance>
      - <attribute_instance express_attribute_name="axis">
        <entity_instance_ref refid="e7"/>
      </attribute_instance>
      - <partial_entity_instance express_entity_name="placement">
        - <inherited_attribute_instance express_attribute_name="location">
          <entity_instance_ref refid="e6"/>
        </inherited_attribute_instance>
      </partial_entity_instance>
      - <partial_entity_instance express_entity_name="representation_item">
        - <inherited_attribute_instance express_attribute_name="name">
          - <type_literal express_type_name="label">
            <string_literal>workpiece placement</string_literal>
          </type_literal>
        </inherited_attribute_instance>
      </partial_entity_instance>
    </entity_instance>
  </schema_instance>
</express_data>

```

Figura C.23. Trecho do arquivo XML.

Fonte: Criado pelo autor.

## 11. Simulação e geração do código G.

Na versão atual, a simulação e a geração de código G está disponível apenas para as seguintes *features*: Furos com base plana (desbaste e acabamento), Furos passantes (desbaste e acabamento), Furos com base cônica (desbaste e acabamento), Ranhuras com perfil retangular (desbaste), Degraus (desbaste), Cavidades com fundo plano (desbaste e acabamento), Cavidades passantes (desbaste e acabamento).