



**DESENVOLVIMENTO DE UMA FRESADORA CNC ADERENTE À NORMA  
STEP-NC BASEADO NO CONTROLADOR DE MÁQUINA AVANÇADO  
(EMC2)**

Miguel Eduardo Gutierrez Paredes

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

**FACULDADE DE TECNOLOGIA  
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**DESENVOLVIMENTO DE UMA FRESADORA CNC ADERENTE À  
NORMA STEP-NC BASEADO NO CONTROLADOR DE MÁQUINA  
AVANÇADO (EMC2)**

**MIGUEL EDUARDO GUTIERREZ PAREDES**

**ORIENTADOR: ALBERTO JOSÉ ALVARES**

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS**

**PUBLICAÇÃO: ENM.DM – 61A/13  
BRASÍLIA/DF: SETEMBRO – 2013**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**DESENVOLVIMENTO DE UMA FRESADORA CNC ADERENTE À  
NORMA STEP-NC BASEADO NO CONTROLADOR DE MÁQUINA  
AVANÇADO (EMC2)**

**MIGUEL EDUARDO GUTIERREZ PAREDES**

**DISSERTAÇÃO DE MESTRADO SUBMETIDA AO  
DEPARTAMENTO DE ENGENHARIA MECÂNICA DA FACULDADE  
DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO  
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO  
GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS**

APROVADA POR:

---

Prof. Alberto José Álvares, Dr. Eng. (ENM-UnB)  
(Orientador)

---

Prof. Antônio Piratelli Filho, Dr. Eng. (ENM-UnB)  
(Examinador Interno)

---

Prof. João Carlos Espindola Ferreira, Dr. Eng. (ENM-UFSC)  
(Examinador Externo)

BRASÍLIA/DF, 03 DE SETEMBRO DE 2013

## FICHA CATALOGRÁFICA

PAREDES, MIGUEL EDUARDO GUTIERREZ

Desenvolvimento de uma Fresadora CNC Aderente à Norma STEP-NC Baseado no Controlador de Máquina Avançado (Emc2). [Distrito Federal] 2013.

xxi, 227p., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2013).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

- |                           |                    |
|---------------------------|--------------------|
| 1. Engenharia de Produção | 2. CNC             |
| 3. STEP-NC                | 4. Linux e EMC2    |
| I. ENM/FT/UnB             | II. Título (série) |

## REFERÊNCIA BIBLIOGRÁFICA

GUTIERREZ P., M. E. (2013). Desenvolvimento de uma Fresadora CNC Aderente à Norma STEP-NC Baseado no Controlador de Máquina Avançado (Emc2). Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-61A/13, Departamento de Engenharia Mecânica,

Universidade de Brasília, Brasília, DF, 227p.

## CESSÃO DE DIREITOS

AUTOR: Miguel Eduardo Gutierrez Paredes.

TÍTULO: Desenvolvimento de uma Fresadora CNC Aderente à Norma STEP-NC Baseado no Controlador de Máquina Avançado (Emc2).

GRAU: Mestre

ANO: 2013

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Miguel Eduardo Gutierrez Paredes.  
ADE Águas Claras, Arniquieras.  
Conjunto 2, Condomínio Jerusalém  
SHVG CH 53/2, Lt 18/19  
71998-180 Brasília – DF – Brasil.

## AGRADECIMENTOS

Agradeço...

*Em primeiro lugar a Deus por me dar vida, sabedoria e força para terminar todas as metas que me propus.*

*Aos meus Pais, RIGOBERTO e LUZ AMPARO, que são o mais importante da minha vida e que construíram em mim a pessoa que sou hoje.*

*À minha família, irmãos, tios e aos avós porque me apoiam e cada dia contribuem com sua alegria em minha vida para seguir adiante com meus estudos.*

*A meu tio e engenheiro WILLINTON por seu apoio incondicional, por me deixar no belo caminho da engenharia e me colaborar em cada etapa dos meus estudos.*

*À minha amiga e meu amor THALLITA RAQUEL por sua paciência, apoio e força, me ajudando a terminar os estudos de mestrado.*

*A todas as pessoas, familiares e amigos que contribuíram com este grande logro.*

*MIGUEL EDUARDO GUTIERREZ PAREDES*

## RESUMO

Este trabalho apresenta o desenvolvimento e a validação de uma arquitetura de um controlador numérico computadorizado (CNC) aderente à norma STEP-NC por meio de um adaptador de STEP-NC a Código G. O CNC está associado ao processo de fresamento. O controlador STEP-NC desenvolvido é baseado no controlador de código aberto EMC (*Enhanced Machine Controller*) aderente à norma RS-274, que originalmente trabalha com programa NC (*Numerical Control*) usando códigos G e M. Para validação da arquitetura do controlador foi desenvolvida uma máquina-ferramenta controlada numericamente com topologia tipo Router (fresadora com estrutura em portal) com três graus de liberdade tendo como controlador o EMC em plataforma x86 (computador pessoal) usando o sistema operacional Ubuntu/Linux. O controlador EMC foi integrado com o modelo de dados STEP-NC Parte 21 utilizando um adaptador do arquivo físico p21 a código G, permitindo que o EMC receba como entrada um programa de comando numérico em formato STEP-NC Part-21. O formato se baseia no conceito de *workingsteps*, *features* de usinagem e operações de usinagem, entre outras entidades descritas pela norma; ou seja, o programa STEP-NC foca-se em uma descrição de alto nível baseada em *features* de usinagem e em um modelo de dados mais sofisticado, mas não na movimentação dos eixos da máquina. O comando numérico com código STEP-NC é gerado pelo sistema integrado de projeto, planejamento de processo e manufatura auxiliado por computador (CAD/CAPP/CAM) denominado *STEP Modeler* ([http://www.grima.ufsc.br/stepnc\\_project/](http://www.grima.ufsc.br/stepnc_project/)), desenvolvido pelo GRIMA/UFSC, que permite a modelagem de uma peça prismática via Web baseada em *features* de usinagem, tendo um modelo de dados aderente ao projeto STEP, viabilizando a integração do ciclo de vida de uma peça, do projeto à fabricação, via Web. O controlador EMC com o adaptador STEP-NC utiliza o código gerado pelo *STEP Modeler* e o executa de forma transparente para o operador da Router CNC, especialmente desenvolvida para validação da arquitetura idealizada. Este desenvolvimento permitiu a validação do conceito de integração do ciclo de vida do projeto à fabricação baseado no modelo de dados STEP-NC, utilizando os conceitos de *features* de projeto e usinagem. A arquitetura é apresentada usando modelos IDEF0 (*Integration Definition for Function Modeling*) e UML (*Unified Modeling Language*), mostrando a especificação do sistema integrado CAD/CAPP/CAM, em especial a etapa associada ao CAM Execução, ou seja, o controlador/adaptador STEP-NC desenvolvido baseado no EMC. O projeto detalhado da Router é apresentado de forma resumida. Os resultados obtidos são mostrados através de estudo de caso com a fabricação de peças exemplos, contemplando sua medição e análise. O ambiente STEP-NC foi concebido para um contexto Web sendo também usado em disciplinas de Automação da Manufatura de graduação e pós-graduação da UnB e da UFSC, visando à apresentação dos conceitos e modelos de dados baseados em STEP, que será o futuro da Automação da Manufatura, estando a cada ano mais madura.

## ABSTRACT

This document describes the development and validation of the architecture of a numerical computer controller that is adherent to the standard STEP-NC by means of an adapter of STEP-NC to G-Code. The CNC is associated with the milling process. The STEP-NC controller development is based on the open source *driver* EMC (Enhanced Machine Controller) adherent to the standard RS-274 that originally works with the NC program using G/M codes. In order to validate the controller architecture, the NC machine-tool topology type Router (portal structure) was developed with three degrees of freedom to have the EMC controller in x86 platform (personal computer) using the operating system Ubuntu/Linux. The EMC controller was integrated using an adapter to the Data Model STEP-NC Part- 11, allowing the EMC to receive as an input a numerical control program in a format STEP-NC Part-21 also called STEP. The format is based on the concept of workingsteps, machining features and machining operations, among others that are described by the standard, i.e., the STEP program focuses on a high-level description based on machining features and a more sophisticated data model but not the movement of the machine axes. This numerical control program with STEP-NC code is generated by the integrated design, process planning and computer-aided manufacturing (CAD / CAPP / CAM) called *STEP Modeler* ([http://www.grima.ufsc.br/stepnc\\_project/](http://www.grima.ufsc.br/stepnc_project/)), developed by GRIMA/UFSC, that allows the modeling of a prismatic part web-based machining features, having a data model adhering to the STEP project, enabling the integration of the life cycle of a piece, from design to manufacturing, via Web. The EMC controller with STEP-NC adapter uses codes generated by STEP Modeler and runs transparently to the operator of specially developed CNC Router to validate the architecture conceived. This development allows the validation of the concept of integration lifecycle from design to manufacturing based on the data model of STEP-NC, using the concepts of design and machining features. The architecture is presented to use models IDEF0 (Integration Definition for Function Modeling) and UML (Unified Modeling Language), showing the specification of integrated system CAD / CAPP / CAM, particularly the stage associated with CAM Execution, i.e., the controller/adapter STEP-NC developed based on EMC. The detailed project of the Router CNC is presented in a summarized way. The results are shown through study of cases, process planning and manufacturing parts examples, with measurement and analysis. The environment STEP-NC has been designed for Web context being also used in the disciplines of the Manufacturing Automation of graduate and postgraduate of UNB and UFSC, aiming at presentation of concepts and data models based on STEP, which will be the future of Automation Manufacturing, maturing through time and technological improvements.

# SUMÁRIO

|  |       |
|--|-------|
| RESUMO .....   | v     |
| ABSTRACT.....  | vi    |
| LISTA DE TABELAS .....   | xii   |
| LISTA DE FIGURAS .....   | xiii  |
| LISTA DE SÍMBOLOS E ABREVIATURAS.....                            | xviii |
| 1. INTRODUÇÃO .....  | 1     |
| 1.1. CONTEXTUALIZAÇÃO .....                                      | 1     |
| 1.2. PROBLEMA DE PESQUISA.....                                   | 4     |
| 1.3. OBJETIVOS .....   | 5     |
| 1.3.1. Objetivo Geral.....                                       | 5     |
| 1.3.2. Objetivos específicos.....                                | 5     |
| 1.4. DELIMITAÇÃO DO TRABALHO.....                                | 5     |
| 1.5. APRESENTAÇÃO DO DOCUMENTO .....                             | 5     |
| 2. REVISÃO DA LITERATURA SOBRE STEP-NC.....                      | 7     |
| 2.1. SISTEMAS ASSISTIDOS POR COMPUTADOR .....                    | 7     |
| 2.1.1. Sistema CAD.....  | 7     |
| 2.1.2. Sistema CAPP .....  | 7     |
| 2.1.3. Sistema CAM .....   | 8     |
| 2.2. INTEGRAÇÃO DE SISTEMAS CAD/CAPP/CAM/CNC .....               | 8     |
| 2.2.1. Sistema integrado CAD/CAPP/CAM/CNC aderente STEP-NC ..... | 9     |
| 2.3. A ORGANIZAÇÃO INTERNACIONAL DE NORMALIZAÇÃO (ISO) .....     | 11    |
| 2.4. O ISO6983 - CÓDIGO G/M.....                                 | 11    |
| 2.4.1. Partes do ISO6983.....                                    | 11    |
| 2.4.2. Programação de máquinas NC com o ISO6983 .....            | 12    |
| 2.4.3. O intérprete NIST RS274NGC.....                           | 13    |
| 2.5. O ISO10303 - STEP .....                                     | 13    |
| 2.5.1. A relação entre STEP e STEP-NC .....                      | 14    |
| 2.5.2. Protocolo de Aplicação .....                              | 15    |
| 2.5.3. O Esquema EXPRESS .....                                   | 15    |
| 2.6. O ISO14649 - STEP-NC .....                                  | 17    |
| 2.6.1. O Sistema STEP-NC .....                                   | 17    |
| 2.6.2. STEP-NC e o Novo Modelo de dados para CNC.....            | 18    |

|        |   |    |
|--------|---|----|
| 2.6.3. | ISO 10303 Parte 21 .....  | 20 |
| 2.6.4. | Partes do ISO 14649 geral e para fresamento .....   | 20 |
| 2.7.   | PROJETOS RELACIONADOS À NORMA STEP-NC.....  | 32 |
| 2.7.1. | Projeto STEP-NC <i>MACHINE</i> .....  | 33 |
| 2.7.2. | Projeto Internacional IMS STEP-NC.....  | 33 |
| 2.7.3. | Projeto ISO14649 <i>ToolKit</i> .....   | 33 |
| 2.7.4. | Projeto STEP <i>Modeler</i> UFSC/GRIMA .....  | 34 |
| 2.7.5. | Projeto Europeu <i>ESPRIT</i> STEP-NC.....  | 35 |
| 2.7.6. | Projeto Suíte de Manufatura STEP (SMS).....   | 35 |
| 2.7.7. | Projeto para Rápida Aquisição de Peças Manufaturadas (Project RAMP) .....   | 35 |
| 2.7.8. | Projeto de Manufatura inteligente para Usinagem e Inspeção compatível com STEP-NC<br>36                           |    |
| 2.8.   | STEP-NC E <i>FUNCTION BLOCKS</i> .....  | 36 |
| 2.8.1. | Blocos de função para processo de programação de acesso a dados.....  | 37 |
| 2.8.2. | O Protótipo STEPNCMillUoA.....  | 37 |
| 2.8.3. | CNC aderente as normas STEP-NC e FB .....   | 38 |
| 2.9.   | CONVERSORES RELACIONADOS COM O PADRÃO STEP-NC E CÓDIGO G....  | 38 |
| 2.9.1. | Patente 1 No US 8,041,445 B2: Método de transformação de Código G num programa de peça em STEP-NC [58].....       | 39 |
| 2.9.2. | Patente 2 No US 6,795,749 B2: Método para geração automática de um programa de peça para uso no STEP-NC [59]..... | 40 |
| 2.10.  | TIPOS DE CONTROLADORES ADERENTES A STEP-NC .....  | 41 |
| 2.11.  | O PROTOCOLO DE APLICAÇÃO 238 OU AP238 .....   | 42 |
| 3.     | MODELAGEM DO CNC ADERENTE A STEP-NC.....  | 44 |
| 3.1.   | ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA.....  | 44 |
| 3.2.   | MODELAGEM FUNCIONAL DO CNC ADERENTE À NORMA STEP-NC .....   | 44 |
| 3.2.1. | Extrair informações básicas.....  | 50 |
| 3.2.2. | Extração de informações avançadas.....  | 52 |
| 3.2.3. | Geração de Código G/M .....   | 55 |
| 3.2.4. | Configurar EMC aderindo o <i>MTConnect</i> .....  | 60 |
| 4.     | IMPLEMENTAÇÃO DO PROTÓTIPO DE ADAPTADOR DE STEP-NC A CÓDIGO G.  | 62 |
| 4.1.   | IMPLEMENTAÇÃO DAS CLASSES PARA A EXTRAÇÃO DE INFORMAÇÕES .  | 63 |
| 4.2.   | CARACTERÍSTICAS IMPORTANTES REFERENTES À PROGRAMAÇÃO NC .....   | 67 |
| 4.2.1. | Descrição das funções preparatórias e miscelâneas .....   | 69 |
| 4.2.2. | Segmentação geométrica de <i>features</i> complexas .....   | 74 |
| 4.2.3. | Geração de trajetórias de ferramenta ( <i>Toolpath</i> ).....   | 77 |

|        |   |     |
|--------|---|-----|
| 4.2.4. | Trajetória de ferramenta no plano XY .....  | 80  |
| 4.2.5. | Equações para cálculo de trajetórias no plano XZ e/ou YZ .....  | 82  |
| 4.3.   | GERAÇÃO DE CÓDIGO G & M .....   | 92  |
| 4.4.   | GERAÇÃO DO ARQUIVO .TBL.....  | 99  |
| 4.5.   | CÓDIGO EM JAVA DO ALGORITMO DO PROTÓTIPO DE ADAPTADOR DE STEP-NC A CÓDIGO G .....   | 99  |
| 5.     | IMPLEMENTAÇÃO DO CONTROLADOR LINUXCNC (ou EMC) ADERENTE À NORMA STEP-NC.....  | 100 |
| 5.1.   | METODOLOGIA PARA ADIÇÃO DO ADAPTADOR DE STEP-NC PARA CÓDIGO G AO CONTROLADOR LINUXCNC .....                                     | 100 |
| 5.2.   | METODOLOGIA PARA CRIAR UMA GUI PARA VISUALIZAR O ARQUIVO FÍSICO P21 JUNTO COM A GUI AXIS DO LINUXCNC .....                      | 101 |
| 5.2.1. | A GUI <i>AXIS</i> do LinuxCNC.....  | 101 |
| 5.2.2. | Implementação de uma GUI para visualizar o arquivo físico p21 usando Glade .....  | 103 |
| 5.2.3. | Implementação das funções da GUI que visualiza o arquivo físico p21 .....   | 104 |
| 5.3.   | METODOLOGIA PARA AGREGAR A GUI DE VISUALIZAÇÃO NA GUI AXIS DO LINUXCNC.....   | 104 |
| 6.     | ESTUDO DE CASOS.....  | 107 |
| 6.1.   | ESTUDO DE CASO – CNC ADERENTE A STEP-NC.....  | 107 |
| 6.1.1. | Esboço da peça teste do estudo de caso .....  | 107 |
| 6.1.2. | Gerar um arquivo físico p21 no STEP <i>Modeler</i> de acordo com o esboço da peça teste   | 108 |
| 6.1.3. | Carregar um arquivo físico p21 no LinuxCNC .....  | 111 |
| 6.1.4. | Simulação do código G gerado pelo adaptador de STEP-NC a código G em Vericut  | 112 |
| 6.1.5. | Usinagem da peça de trabalho.....   | 113 |
| 6.1.6. | Comparação entre as medidas teóricas (esboço peça teste) com as práticas (peça teste usinada)                                   | 115 |
| 6.1.7. | Comparação entre código G feito manualmente e o código G gerado pelo adaptador  | 117 |
| 7.     | CONCLUSÕES E TRABALHOS FUTUROS .....  | 119 |
| 7.1.   | CONCLUSÕES DO TRABALHO .....  | 119 |
| 7.2.   | SUGESTÕES PARA TRABALHOS FUTUROS .....  | 121 |
|        | REFERENCIAS BIBLIOGRÁFICAS .....  | 123 |
|        | APÊNDICE A. METODOLOGIA DE TRABALHO E CONCEITOS SOBRE TRADUTOR DE LINGUAGEM PARA A MODELAGEM DO CNC ADERENTE A NORMA STEP-NC... | 132 |
| A.1.   | METODOLOGIA DE TRABALHO .....   | 132 |
| A.1.1. | Metodologias ágeis para desenvolvimento de <i>software</i> .....  | 132 |

|             |   |     |
|-------------|---|-----|
| A.2.        | TRADUTOR DE LINGUAGEM.....  | 133 |
| A.2.1.      | O Compilador.....   | 133 |
| A.2.2.      | O Intérprete .....  | 133 |
| A.2.3.      | O Conversor Fonte - Fonte.....  | 134 |
| APÊNDICE B. | O CONTROLADOR EMC SUAS CARACTERÍSTICAS E FUNÇÕES.....   | 135 |
| B.1.        | O CONTROLADOR LINUXCNC .....  | 135 |
| B.2.        | PROJETO LINUXCNC.ORG.....   | 135 |
| B.3.        | INSTALAÇÃO DO LINUXCNC .....  | 136 |
| B.4.        | CONFIGURAÇÃO DO CNC TIPO ROUTER COM O EMC2.....   | 137 |
| B.4.1.      | O arquivo .INI .....  | 143 |
| B.4.2.      | O arquivo .TBL .....  | 143 |
| APÊNDICE C. | PROJETO DE DESENHO E FABRICAÇÃO DE UMA ROUTER CNC DE TRÊS EIXOS PARA PEÇAS PRISMÁTICAS .....  | 146 |
| C.1.        | COMPONENTES DE UMA ROUTER CNC.....  | 146 |
| C.2.        | PROJETO CAD DE DESENHO E SIMULAÇÃO DA ROUTER CNC .....  | 150 |
| C.3.        | USINAGEM E MONTAGEM DAS PEÇAS DO ROUTER CNC.....  | 161 |
| C.4.        | DISPOSITIVO DE TROCA DE FERRAMENTA .....  | 164 |
| C.5.        | MOTORES DE PASSO .....  | 166 |
| C.6.        | SISTEMA ELETRÔNICO DE CONTROLE DO ROUTER CNC .....  | 168 |
| C.7.        | PARÂMETROS DE CORTE DO ROUTER CNC.....  | 171 |
| APÊNDICE D. | CRIAÇÃO DE PROGRAMAS STEP-NC <i>PART 21</i> USANDO O STEP <i>MODELER</i> 175  |     |
| D.1.        | Exemplo a) Programa de peça em STEP-NC: Peça com uma <i>feature</i> (“simples”) tipo furo com base cônica gerado pelo STEP <i>Modeler</i> ..... | 175 |
| D.2.        | Exemplo b) Programa de peça em STEP-NC: Peça com uma <i>feature</i> tipo Ranhura com perfil em U gerada pelo STEP <i>Modeler</i> .....          | 176 |
| APÊNDICE E. | IMPLEMENTAÇÃO DO ALGORITMO DA GUI DE VISUALIZAÇÃO USANDO A LINGUAGEM PYTHON .....   | 178 |
| APÊNDICE F. | ESTUDO DE CAPABILIDADE DA ROUTER CNC .....  | 179 |
| F.1.        | PROCEDIMENTOS PARA OBTENÇÃO DE AMOSTRAS E APRESENTAÇÃO DE RESULTADOS .....  | 182 |
| F.1.1.      | Procedimento para obtenção de amostras para identificar o valor do erro de retilinearidade, paralelismo e perpendicularidade.....               | 182 |
| F.1.2.      | Resultados dos erros de forma macrogeométricos da peça teste relacionados com retilinearidade, paralelismo e perpendicularidade.....            | 183 |
| F.1.3.      | Procedimento de toma de amostras para identificar o valor do erro de circularidade.....   | 186 |
| F.1.4.      | Resultado do erro de forma macrogeométrico da peça teste relacionado com circularidade.....   | 186 |

|   |  |     |
|---|--|-----|
| F.1.5.  | Procedimento de toma de amostras para identificar o valor do erro de posicionamento dos eixos X, Y e Z ..... | 187 |
| F.1.6.  | Resultado do erro de posicionamento da peça teste relacionado com os eixos X, Y e Z da Router CNC.....       | 188 |
| APÊNDICE G. ARQUIVO DE CONFIGURAÇÃO USADO PELO ADAPTADOR DE STEP-NC A CÓDIGO G. 189 |  |     |
| ANEXO 1.  | LISTA DE CÓDIGOS G/M PARA FRESAMENTO.....  | 191 |
| ANEXO 2.  | CARACTERÍSTICAS GERAIS DO MOTOR DE PASSO KTC-HT23-400, KALATEC® 193  |     |
| ANEXO 3.  | CARATERÍSTICAS GERAIS DA LIGA DE ALUMÍNIO ALZINTOK 75-T651 194   |     |
| ANEXO 4.  | SMC-B-PRO GUIA DE UTILIZAÇÃO.....  | 195 |
| ANEXO 5.  | BPC-PRO GUIA DE UTILIZAÇÃO.....  | 199 |
| ANEXO 6.  | CÓDIGO G CRIADO PELO ADAPTADOR A PARTIR DE UM PROGRAMA DE PEÇA EM STEP-NC .....                              | 203 |
| ANEXO 7.  | EQUAÇÕES USADAS NO MODELAMENTO MATEMÁTICO DA GEOMETRIA DE FEATURES.....                                      | 204 |
| ANEXO 8.  | MODELO E CERTIFICADO DE CALIBRAÇÃO CMM MITUTOYO.....   | 206 |

## LISTA DE TABELAS

|   |     |
|---|-----|
| Tabela 2.1: <i>Features</i> de usinagem, condições e operações usadas no software <i>STEP Modeler</i> ..... | 34  |
| Tabela 4.1: Conceitos relacionados com a programação NC .....   | 67  |
| Tabela 4.2: Códigos G – Funções preparatórias .....   | 68  |
| Tabela 4.3: Códigos G – Funções miscelâneas e outros códigos auxiliares .....                               | 68  |
| Tabela 4.4: <i>Toolpaths</i> 2½D dispostos na norma ISO14649-11 .....                                       | 80  |
| Tabela 4.5: Equações para descrever a geometria de uma <i>feature</i> simples .....                         | 83  |
| Tabela 4.6: Seções da ranhura com perfil quadrado U, Figura 4.34 .....                                      | 84  |
| Tabela 4.7: Características da <i>feature</i> de exemplo. ....  | 84  |
| Tabela 4.8: Conversão de coordenadas.....   | 90  |
| Tabela 4.9: Cálculo do comprimento e profundidade da semicircunferência da <i>feature</i> .....             | 91  |
| Tabela 4.10: Informações para criar um exemplo de programa P21 .....  | 92  |
| Tabela 4.11: Ferramentas do programa P21 - segundo exemplo .....  | 95  |
| Tabela 4.12: <i>Features</i> e a suas ferramentas - segundo exemplo.....                                    | 95  |
| Tabela 4.13: <i>Features</i> e a suas ferramentas rearranjadas – segundo exemplo.....                       | 96  |
| Tabela 4.14: <i>Features</i> funcionais no adaptador .....  | 97  |
| Tabela 6.1: Itens e suas características para o projeto da peça .....                                       | 107 |
| Tabela 6.2: Medidas encontradas na peça teste usinada .....   | 116 |
| Tabela 6.3: Medidas encontradas na peça teste usinada .....   | 116 |
| Tabela 6.4: Comparação entre medidas teóricas e práticas da peça teste .....                                | 117 |
| Tabela 6.5: Medidas encontradas nas peças simuladas usinadas .....  | 118 |
| Tabela B.1: Hardware do PC usado na Router CNC .....  | 140 |
| Tabela C.1: Fusos de esferas recirculantes .....  | 147 |
| Tabela C.2: Características do Motor de passo KTC-HT23-400.....   | 147 |
| Tabela C.3: Resumo das características de ferramentas elétricas. ....                                       | 148 |
| Tabela C.4: Matérias para a estrutura da Router.....  | 149 |
| Tabela C.5: Componentes para o Projeto.....   | 150 |
| Tabela C.6: Componentes mecânicos para o Módulo do eixo Z .....   | 152 |
| Tabela C.7: Componentes eletromecânicos para o Módulo do eixo Z .....                                       | 152 |
| Tabela C.8: Componentes mecânicos para o Módulo do eixo X.....  | 154 |
| Tabela C.9: Componentes eletromecânicos para o Módulo do eixo X.....  | 154 |
| Tabela C.10: Componentes mecânicos para o Módulo do eixo Y .....  | 155 |
| Tabela C.11: Componentes eletromecânicos para o Módulo do eixo Y.....                                       | 156 |
| Tabela C.12: Perfis Industriais [97]. ....  | 157 |
| Tabela C.13: Lista de novos materiais .....   | 161 |
| Tabela C.14: Velocidades de avanço .....  | 173 |
| Tabela C.15: Aceleração de avanço.....  | 173 |
| Tabela D.1: Informações da peça bruta .....   | 175 |
| Tabela D.2: Informações da <i>feature</i> .....   | 175 |
| Tabela D.3: Informações das ferramentas.....  | 175 |
| Tabela F.1: Parâmetros de corte da Router CNC encontrados experimentalmente .....                           | 181 |
| Tabela F.2: Resultados do ensaio de retilineidade .....   | 183 |
| Tabela F.3: Resultados do ensaio de paralelismo (180°).....   | 184 |
| Tabela F.4: Resultados do ensaio de perpendicularismo (90°).....  | 184 |
| Tabela F.5: Resultados do ensaio de circularidade .....   | 186 |
| Tabela F.6: Resultados do ensaio posicionamento eixos X, Y e Z.....   | 188 |

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1.1: Estrutura geral da organização de dados na ISO 14649 (Adaptado de [2]).....  | 2  |
| Figura 1.2: Saída de dados para fabricação de uma peça em STEP-NC (Adaptado de [2]) .....  | 3  |
| Figura 1.3: Tentativa do sistema finalizado .....  | 4  |
| Figura 2.1: Modelos de integração CAD/CAPP/CAM/CNC (Adaptado de [17]) .....  | 10 |
| Figura 2.2: Sistema Integrado CAD/CAPP/CAM/CNC usando STEP e STEP-NC (Adaptado de [20]).....   | 11 |
| Figura 2.3: Linha de programa típica usando a norma ISO 6983 aplicando como referência o intérprete RS274NGC (Adaptado de [11])..... | 13 |
| Figura 2.4: A relação entre a ISO 14649 e ISO 10303 (Adaptado de [31]) .....   | 15 |
| Figura 2.5: Exemplo simples do esquema EXPRESS textual [33] .....  | 16 |
| Figura 2.6: Exemplo simples do esquema EXPRESS-G [33].....   | 17 |
| Figura 2.7: Comparação entre Código G/M e STEP-NC (Adaptado de [31]). .....  | 19 |
| Figura 2.8: Descrição geral do modelo de dados do ISO14649 (Adaptado de [31]). .....   | 20 |
| Figura 2.9: Estrutura do modelo de dados STEP-NC para fresamento (Adaptado de [27]) .....  | 21 |
| Figura 2.10: Estrutura do modelo de dados STEP-NC para fresado no diagrama EXPRESS-G (Adaptado de [7]). .....                        | 21 |
| Figura 2.11: Estrutura de dados do ISO 14649 (Adaptado de [38]).....   | 22 |
| Figura 2.12: Diagrama <i>EXPRESS-G</i> da entidade <i>project</i> (Adaptado de [36]). .....  | 23 |
| Figura 2.13: Diagrama <i>EXPRESS-G</i> , <i>Executables</i> (Adaptado de [36]). .....  | 24 |
| Figura 2.14: Diagrama <i>EXPRESS-G</i> do <i>workplan</i> (Adaptado de [36]).....  | 24 |
| Figura 2.15: Diagrama <i>EXPRESS-G</i> do <i>workingstep</i> (Adaptado de [36]). .....   | 25 |
| Figura 2.16: Diagrama <i>EXPRESS-G</i> do <i>machining_workingstep</i> (Adaptado de [36]). .....                                     | 25 |
| Figura 2.17: Diagrama <i>EXPRESS-G</i> do <i>manufacturing_feature</i> [11].....   | 26 |
| Figura 2.18: Diagrama <i>EXPRESS-G</i> do <i>operation</i> (Adaptado de [36]). .....   | 26 |
| Figura 2.19: Diagrama <i>EXPRESS-G</i> do <i>machining_operation</i> (Adaptado de [36]). .....                                       | 27 |
| Figura 2.20: Diagrama <i>EXPRESS-G</i> do <i>milling_machining_operation</i> (Adaptado de [39]). .....                               | 28 |
| Figura 2.21: Diagrama <i>EXPRESS-G</i> do <i>milling_tipe_operation</i> (Adaptado de [39]). .....                                    | 28 |
| Figura 2.22: Diagrama <i>EXPRESS-G</i> do <i>drilling_tipe_operation</i> (Adaptado de [39]). .....                                   | 29 |
| Figura 2.23: Diagrama <i>EXPRESS-G</i> do <i>milling technology</i> [39].....  | 29 |
| Figura 2.24: Diagrama <i>EXPRESS-G</i> do <i>milling machine functions</i> (Adaptado de [39]). .....                                 | 30 |
| Figura 2.25: Diagrama <i>EXPRESS-G</i> do <i>cutting tool for milling</i> (Adaptado de [11]). .....                                  | 31 |
| Figura 2.26: Diferentes tipos de ferramentas e suas Dimensões (Adaptado de [40]). .....  | 31 |
| Figura 2.27: Diagrama <i>EXPRESS-G</i> do <i>Milling tool body</i> (Adaptado de [40]). .....   | 32 |
| Figura 2.28: Diagrama <i>EXPRESS-G</i> do <i>Tool dimension</i> (Adaptado de [40]). .....  | 32 |
| Figura 2.29: Os principais contribuintes para a abordagem do STEP-NC [53] .....  | 36 |
| Figura 2.30: Passos para transformar Código G para STEP-NC (Adaptado de [58]).....   | 40 |
| Figura 2.31: Diagrama IDEFØ da função principal da patente 2 (Adaptado de [59]) .....  | 41 |
| Figura 2.32: Tipos de controladores STEP-NC (Adaptado de [60]) .....   | 42 |
| Figura 2.33: Extensão de <i>nc_functions</i> [62].....   | 43 |
| Figura 3.1: Diagrama geral IDEFØ do Sistema integrado CAD/CAPP/CAM/CNC aderente STEP-NC. ....  | 46 |
| Figura 3.2: Diagrama IDEFØ do sistema integrado, dentro do nível A0.....   | 47 |
| Figura 3.3: Diagrama IDEFØ do CNC aderente STEP-NC, dentro do nível A2.....  | 48 |
| Figura 3.4: Diagrama IDEFØ do Adaptador STEP-NC a Código G/M, dentro do nível A21 .....  | 49 |

|   |    |
|---|----|
| Figura 3.5: Diagrama IDEFØ - Extrair informações básicas, dentro do nível A211 .....                                  | 51 |
| Figura 3.6: Diagrama UML, classes da extração de informações básicas .....  | 52 |
| Figura 3.7: Diagrama IDEFØ - Extrair informações Avançadas, dentro do nível A212 .....                                | 53 |
| Figura 3.8: Diagrama UML, classes da extração de informações Avançadas .....  | 54 |
| Figura 3.9: Exemplo de criação de uma <i>feature</i> no software <i>STEP Modeler</i> .....                            | 55 |
| Figura 3.10: Informações que deverá conter o <i>ArrayList</i> - InformaçõesAvançadas .....                            | 55 |
| Figura 3.11: Diagrama IDEFØ – Gerar código G/M, dentro do nível A213 .....  | 57 |
| Figura 3.12: Diagrama IDEFØ – Gerar programa NC, dentro do nível A2136 .....  | 58 |
| Figura 3.13: Diagrama UML, classes da geração do Código G/M .....   | 59 |
| Figura 3.14: Diagrama UML, classes da geração do arquivo “ <i>tool.tbl</i> ” .....                                    | 60 |
| Figura 3.15: Diagrama IDEFØ – Configurar o EMC aderindo o <i>MTConnect</i> , dentro do nível A22 .....                | 61 |
| Figura 4.1: Diagrama de fluxo da arquitetura do protótipo do adaptador, parte 1 .....                                 | 62 |
| Figura 4.2: Diagrama de fluxo da arquitetura do protótipo do adaptador, parte 2 .....                                 | 63 |
| Figura 4.3: Programa de peça STEP-NC .....  | 64 |
| Figura 4.4: Estrutura de dados STEP-NC para CAD/CAM [2]. .....  | 65 |
| Figura 4.5: Endereços e linhas do programa. ....  | 65 |
| Figura 4.6: Informações básicas: dados da peça bruta .....  | 66 |
| Figura 4.7: Informações básicas: plano de segurança .....   | 66 |
| Figura 4.8: Mapeamento de Informações Avançadas .....   | 66 |
| Figura 4.9: Informações Avançadas (Adaptado de [69]) .....  | 68 |
| Figura 4.10: Interpolação linear (Adaptado de [71]).....  | 69 |
| Figura 4.11: Interpolação circular (Adaptado de [72]).....  | 70 |
| Figura 4.12: Diferenças entre os formatos de interpolação circular (Adaptado de [71]) .....                           | 71 |
| Figura 4.13: Compensação de raio de corte (Adaptado de [72]).....   | 71 |
| Figura 4.14: Sistemas de coordenadas .....  | 72 |
| Figura 4.15: Coordenadas de distância G90 e G91 (Adaptador de [73]) .....   | 73 |
| Figura 4.16: Janela: Troca manual de ferramenta [74] .....  | 74 |
| Figura 4.17: <i>Workingsteps</i> de um furo com base plana (Adaptado de [11]).....                                    | 74 |
| Figura 4.18: <i>Workingsteps</i> de um furo com base arredondada (Adaptado de [11]).....                              | 75 |
| Figura 4.19: <i>Workingsteps</i> de um furo com base esférica (Adaptado de [11]) .....                                | 75 |
| Figura 4.20: <i>Workingsteps</i> de uma ranhura perfil quadrado U (Adaptado de [11]).....                             | 75 |
| Figura 4.21: <i>Workingsteps</i> de uma ranhura perfil parcial circular (Adaptado de [11]).....                       | 76 |
| Figura 4.22: <i>Workingsteps</i> de uma cavidade com fundo plano (Adaptado de [11]).....                              | 76 |
| Figura 4.23: <i>Workingsteps</i> de uma cavidade com fundo arredondado (Adaptado de [11]) .....                       | 76 |
| Figura 4.24: Planos de trabalho.....  | 77 |
| Figura 4.25: Plano de trabalho XZ – A etapa de desbaste gera degraus .....  | 78 |
| Figura 4.26: <i>Toolpath</i> - Ziguezague .....   | 78 |
| Figura 4.27: <i>Toolpath</i> - Espiral retangular com quina circular (Adaptado de [76]).....                          | 79 |
| Figura 4.28: <i>Toolpath</i> - Isoparamétrico [77]. .....   | 79 |
| Figura 4.29: <i>Toolpath</i> - Preenchimento usando curvas <i>Hilbert's</i> [75] .....                                | 79 |
| Figura 4.30: <i>Toolpath</i> - <i>Contour parallel</i> [76].....  | 80 |
| Figura 4.31: <i>Toolpath</i> para furos (adaptado de [76]). .....   | 81 |
| Figura 4.32: <i>Toolpath</i> para Ranhuras e Degraus (adaptado <i>A, B</i> de [80] e <i>C</i> de [39]).....           | 81 |
| Figura 4.33: Vistas dos planos de trabalho.....   | 82 |
| Figura 4.34: Decomposição e análises matemático/geométrico da <i>feature</i> tipo ranhura com perfil quadrado U ..... | 83 |

|  |     |
|--|-----|
| Figura 4.35: Características da <i>feature</i> de exemplo ( <i>feature</i> tipo ranhura com perfil quadrado U) . | 84  |
| Figura 4.36: Divisão da <i>feature</i> em partes mais simples.....   | 85  |
| Figura 4.37: Triângulo retângulo parte da <i>feature</i> de exemplo.....   | 86  |
| Figura 4.38: Triângulo retângulo mais retângulo, parte da <i>feature</i> exemplo.....                            | 87  |
| Figura 4.39: Análise geométrica do ponto de intercepção.....   | 88  |
| Figura 4.40: ponto de intercepção e a suas medidas.....  | 89  |
| Figura 4.41: Porção de semicircunferência e ângulos em sequência.....  | 89  |
| Figura 4.42: Comprimento em sequência.....   | 90  |
| Figura 4.43: Comprimento em sequência (Adaptado de [81]).....  | 90  |
| Figura 4.44: Comprimentos e profundidades.....   | 91  |
| Figura 4.45: Arquivo físico p21 gerado pelo STEP Modeler.....  | 93  |
| Figura 4.46: Informações extraídas do arquivo físico p21.....  | 94  |
| Figura 4.47: Ferramentas do arquivo físico p21.....  | 95  |
| Figura 4.48: Simulação de trajetórias de ferramenta e geometria final (usinagem).....                            | 97  |
| Figura 4.49: Código G gerado a partir do Programa de peça em STEP-NC de exemplo.....                             | 98  |
| Figura 5.1: Ferramentas e suas características.....  | 101 |
| Figura 5.2: GUI AXIS do LinuxCNC.....  | 102 |
| Figura 5.3: Widgets do PyVCP.....  | 102 |
| Figura 5.4: GUI para visualização do arquivo físico p21.....   | 104 |
| Figura 5.5: Diagrama de fluxo do algoritmo da GUI de visualização.....   | 105 |
| Figura 5.6: Fragmento original e modificado da GUI AXIS.....   | 105 |
| Figura 5.7: Menu “View” original e modificado da GUI AXIS.....   | 106 |
| Figura 6.1: Esboço da peça teste do estudo de caso.....  | 108 |
| Figura 6.2: Projeto CAD/CAPP/CAM da peça teste no <i>software</i> STEP Modeler.....                              | 108 |
| Figura 6.3: Modelo 3D da peça projetada gerado pelo STEP Modeler.....  | 109 |
| Figura 6.4: Diferentes vistas 2D da peça modelada no STEP Modeler.....   | 109 |
| Figura 6.5: <i>Features</i> , CAPP e <i>workplan</i> do projeto.....   | 109 |
| Figura 6.6: Ferramentas utilizadas no projeto.....   | 110 |
| Figura 6.7: Fragmento do arquivo físico p21 gerado pelo STEP Modeler para a peça teste.....                      | 110 |
| Figura 6.8: Abrindo o arquivo físico p21.....  | 111 |
| Figura 6.9: Trajetórias geradas pelo adaptador visualizadas na GUI AXIS.....                                     | 111 |
| Figura 6.10: Editor de ferramentas com as ferramentas usadas no Código G.....                                    | 112 |
| Figura 6.11: GUI AXIS e a GUI de visualização.....   | 112 |
| Figura 6.12: Resultado da simulação da Router CNC com a peça teste usinada.....                                  | 113 |
| Figura 6.13: Resultado da simulação da peça teste usinada.....   | 113 |
| Figura 6.14: Fixação da peça teste na mesa de trabalho.....  | 114 |
| Figura 6.15: Resultado da usinagem da peça do segundo estudo de caso.....  | 114 |
| Figura 6.16: Identificação das <i>features</i> da peça usinada.....  | 114 |
| Figura 6.17: Divisão das <i>features</i> da peça teste – pontos para medição.....                                | 115 |
| Figura 6.18: Simulação de usinagem CNC no Vericut.....   | 118 |
| Figura 6.19: Exemplo de montagem com duas peças teste.....   | 118 |
| Figura A.1: Desenho preliminar de um tradutor (Adaptado de [89]).....  | 133 |
| Figura B.1: Tela de início do programa LinuxCNC <i>Stepconf Wizard</i> .....                                     | 136 |
| Figura B.2: Tela de escolha configuração do EMC2.....  | 137 |
| Figura B.3: Tela de Informações básicas da máquina.....  | 138 |
| Figura B.4: Sinais do <i>Driver Timing Settings</i> [92].....  | 138 |
| Figura B.5: Janela do Teste de latência [72].....  | 139 |

|   |     |
|---|-----|
| Figura B.6: Opções de configuração avançada do LinuxCNC [72].....   | 140 |
| Figura B.7: Configuração dos pines da porta paralela .....  | 141 |
| Figura B.8: Configuração Eixo X .....   | 141 |
| Figura B.9: Configuração Eixo Y .....   | 142 |
| Figura B.10: Configuração Eixo Z.....   | 142 |
| Figura B.11: Edição de um arquivo “.tbl” com o Editor de Ferramentas .....  | 144 |
| Figura B.12: Ferramentas e suas características no Editor de Ferramenta.....                                      | 145 |
| Figura B.13: arquivo TBL no Editor de texto .....   | 145 |
| Figura C.1: Mesa de trabalho para a Router.....   | 146 |
| Figura C.2: Acople de motor de passo com fuso de esferas (Adaptado de [93]) .....                                 | 147 |
| Figura C.3:Tupia Makita 3709 [94] .....   | 148 |
| Figura C.4:Troca de ferramenta: (a) Manual [95], (b) Automática [93] .....  | 149 |
| Figura C.5: Chapas de alumínio, vários calibres.....  | 150 |
| Figura C.6: Módulos: eixo Y (a e b), eixo X (c) e eixo Z (d) (Adaptado de [96]) .....                             | 151 |
| Figura C.7: Protótipo da Router CNC .....   | 151 |
| Figura C.8: Desenho CAD - Peças do Cabeçote.....  | 153 |
| Figura C.9: Desenho CAD - Peças do Grupo de movimentação do eixo Z.....   | 153 |
| Figura C.10: Desenho CAD - Módulo do eixo Z completo .....  | 153 |
| Figura C.11: Desenho CAD - Peças do módulo do eixo X.....   | 155 |
| Figura C.12: Desenho CAD – Peças da base portal do modulo do eixo Y .....   | 156 |
| Figura C.13: Desenho CAD - Peças do Portal do modulo do eixo Y .....  | 156 |
| Figura C.14: Desenho CAD - Modulo do eixo Y (portal) completo.....  | 157 |
| Figura C.15: Desenho CAD Suporte .....  | 158 |
| Figura C.16: Desenho CAD Router CNC completo .....  | 158 |
| Figura C.17: Desenho CAD Aparelho para troca de ferramenta.....   | 159 |
| Figura C.18: Desenho CAD Router CNC múltiplas vistas. ....  | 160 |
| Figura C.19: Impressão 3D do Router CNC, escala 1:10 .....  | 161 |
| Figura C.20: Montagem do Cabeçote (peças e tupia) .....   | 162 |
| Figura C.21: Montagem grupo de movimentação eixo Z (peças, motor e acople) .....                                  | 162 |
| Figura C.22: Módulo do Eixo Z.....  | 162 |
| Figura C.23: Módulo do Eixo X.....  | 163 |
| Figura C.24: Portal com o módulo do eixo X e Z.....   | 163 |
| Figura C.25: Base do Portal .....   | 163 |
| Figura C.26: Montagem do módulo do eixo Y com o módulo do eixo X e Z .....  | 164 |
| Figura C.27: Mesa de trabalho e motor do módulo do eixo Y .....   | 164 |
| Figura C.28: Montagem completa da Router CNC, mesa de trabalho e suporte .....                                    | 165 |
| Figura C.29: Montagem do dispositivo de troca de ferramenta .....   | 165 |
| Figura C.30: Dispositivo de troca de ferramenta: Superfície de contato com ferramenta de corte do Router CNC..... | 166 |
| Figura C.31: Desenho CAD da integração entre motor, fuso e carga (Adaptado de [92]).....                          | 166 |
| Figura C.32: Perfil trapezoidal (Adaptado de [98]) .....  | 168 |
| Figura C.33: <i>Driver</i> SMC-B-PRO [100].....   | 169 |
| Figura C.34: Esquema de conexão <i>Driver</i> /Motor/Fonte de alimentação (Adaptado de [100]) ....                | 169 |
| Figura C.35: Interface opto isolada BPC-PRO [100].....  | 170 |
| Figura C.36: Identificação dos pontos de conexão do BPC-PRO [100] .....   | 170 |
| Figura C.37: Sistema de controle do Router CNC (Adaptado de [84]).....  | 171 |
| Figura C.38: Montagem do sistema de controle do Router CNC.....   | 171 |

|  |     |
|--|-----|
| Figura C.39: Referencias da Ferramenta de corte.....   | 172 |
| Figura C.40: Profundidade de Corte (Adaptado de [106]).....  | 174 |
| Figura D.1: Peça bruta, <i>feature</i> e ferramentas.....  | 176 |
| Figura D.2: Características da Peça bruta.....   | 176 |
| Figura D.3: Características da <i>feature</i> .....  | 176 |
| Figura D.4: Vista 3D da <i>feature</i> .....   | 177 |
| Figura D.5: Características das ferramentas.....   | 177 |
| Figura F.1: Esboço de referência da peça teste.....  | 179 |
| Figura F.2: Código G da peça teste.....  | 179 |
| Figura F.3: Resultado da simulação de usinagem da peça teste no <i>software Vericut</i> .....                                      | 180 |
| Figura F.4: Trajetórias da ferramenta de corte (linhas pretas) mostradas pelo LinuxCNC de acordo com o código G da peça teste..... | 180 |
| Figura F.5: Resultado da peça teste usinada na Router CNC.....   | 180 |
| Figura F.6: Novo esboço de referência da peça teste.....   | 181 |
| Figura F.7: Pontos de referência para medição da peça teste.....   | 182 |
| Figura F.8: CMM QM-353 e sistema de palpação.....  | 183 |
| Figura F.9: Procedimento para medição em um ponto.....   | 183 |
| Figura F.10: Apresentação gráfica dos resultados do ensaio de retilineidade.....   | 184 |
| Figura F.11: Apresentação gráfica dos resultados do ensaio de paralelismo.....   | 185 |
| Figura F.12: Apresentação gráfica dos resultados do ensaio de perpendicularismo.....   | 185 |
| Figura F.13: Fragmento da peça teste – Centro da peça.....   | 186 |
| Figura F.14: Apresentação gráfica dos resultados do ensaio de circularidade.....   | 187 |
| Figura F.15: Fragmento da peça teste – Centro da peça teste.....   | 187 |
| Figura F.16: Fragmento da peça teste – planos de referencia.....   | 188 |
| Figura G.1: Exemplo do arquivo de configuração usado pelo adaptador.....   | 190 |
| Figura An7.1: Triangulo retângulo.....   | 204 |

## LISTA DE SÍMBOLOS E ABREVIATURAS

|                  |  |
|------------------|--|
| <b>AIM</b>       | <i>Application Interpreted Model</i> - Modelo de Aplicação Interpretada                            |
| <b>AMST</b>      | <i>Center of Mechanical &amp; Manufacturing Engineering</i>  |
| <b>AP</b>        | Protocolo de aplicação   |
| <b>ARM</b>       | <i>Application Reference Model</i> - Modelo de Referência de Aplicação                             |
| <b>CAD</b>       | <i>Computer Aided Design</i> - Projeto Auxiliado por Computador                                    |
| <b>CAM</b>       | <i>Computer Aided Manufacturing</i> - Manufatura Auxiliada por Computador                          |
| <b>CAPP</b>      | <i>Computer Aided Process Planning</i> - Planejamento de Processo Auxiliado por Computador         |
| <b>CAx</b>       | <i>Computer-Aided x (sistem)</i> – (sistema) x Auxiliado por Computador                            |
| <b>CCW</b>       | Sentido anti-horário   |
| <b>CMM</b>       | <i>Coordinate Measuring Machine</i> - Máquina de Medição por Coordenadas                           |
| <b>CMM</b>       | <i>Coordinate Measuring Machine</i> – Máquina de medição por coordenadas                           |
| <b>CMS</b>       | Fabricante de maquinas-ferramenta CNC  |
| <b>CNC</b>       | <i>Computer Numeric Control</i> - Controle Numérico Computadorizado                                |
| <b>CW</b>        | Sentido horário  |
| <b>EDM</b>       | <i>Engineering Data Management</i> - Engenharia de Gestão de Dados                                 |
| <b>EMC</b>       | <i>Enhanced Machine Controller</i> - Controlador de Máquinas Avançado                              |
| <b>FLOSS</b>     | <i>Free/Open-Source Software</i> - Software de Código-Aberto/Livre                                 |
| <b>FreeBSD</b>   | Sistema operacional baseado em UNIX  |
| <b>Glade</b>     | <i>Glade Interface Designer</i> – Desenhador de interfaces Glade                                   |
| <b>Glxgears</b>  | Teste gráfico de referencia em Linux Ubuntu  |
| <b>GNU</b>       | <i>General Public License</i> - Licença Pública Geral  |
| <b>GTK/GNOME</b> | <i>GIMP Toolkit /GNU Network Object Model Environment</i>  |
| <b>GUI</b>       | <i>Graphical User Interface</i> - Interface Gráfica do Usuário                                     |
| <b>HAL</b>       | <i>Hardware Abstraction layer</i> - Camada de abstração de hardware                                |
| <b>ICS</b>       | <i>Information Content And Semantics</i> - Conteúdo da informação semântica                        |
| <b>IDE</b>       | <i>Integrated Development Environment</i> - Ambiente integrado de desenvolvimento                  |
| <b>IGES</b>      | <i>Initial Graphics Exchange Specification</i>   |
| <b>IMS</b>       | <i>Intelligent Manufacturing Systems</i> – Sistema de manufatura inteligente                       |
| <b>INI</b>       | Extensão padrão para arquivos de configuração  |
| <b>ISO</b>       | <i>International Organization for Standardization</i> - Organização internacional de normalização  |
| <b>LinuxCNC</b>  | Controlador de Máquinas Avançado   |
| <b>LTP</b>       | <i>Line Print Terminal</i> - Interface de porta paralela de um computador                          |
| <b>MIT</b>       | <i>Massachusetts Institute of Technology</i>   |
| <b>NC</b>        | <i>Numerical Control</i> – Controle numérico   |
| <b>NGC</b>       | <i>Next Generation Controller</i> – Controlador da próxima geração                                 |
| <b>NIST</b>      | <i>National Institute of Standards and Technology</i> - Instituto Nacional de Padrões e Tecnologia |
| <b>NML</b>       | <i>motion commands</i> – Comandos de movimento   |
| <b>OpenGL</b>    | <i>Open Graphics Library</i> – Livraria gráfica aberta   |
| <b>OS</b>        | <i>Operating System</i> – Sistema Operacional  |
| <b>OSAI</b>      | Empresa especializada em controle numérico e sistemas de automação                                 |
| <b>PC</b>        | <i>Personal computer</i> – Computador pessoal  |

|                |  |
|----------------|--|
| <b>PLC</b>     | <i>Programmable Logic Controller</i> - Controlador Lógico Programável  |
| <b>PyVCP</b>   | <i>Python Virtual Control Panel</i> – Panel de Controle Virtual  |
| <b>RT</b>      | <i>Real time</i> – Tempo real  |
| <b>SC</b>      | <i>Sub-committees</i> - Sub-comitês  |
| <b>SCADA</b>   | <i>Supervisory Control And Data Acquisition</i> - Sistemas de Supervisão e Aquisição de Dados                    |
| <b>SET</b>     | <i>Standard Exchange and Transfer</i>  |
| <b>SFP</b>     | <i>Shop Floor Programming</i> - Programação de chão de fábrica   |
| <b>SMS</b>     | <i>STEP Manufacturing Suite</i> - Suíte de Manufatura STEP   |
| <b>STEP</b>    | <i>Standard for the Exchange of Product model data</i> - Padrão para o intercâmbio de dados do modelo de produto |
| <b>STEP-NC</b> | Linguagem de controle de máquinas-ferramenta baseado em STEP   |
| <b>TACOM</b>   | <i>Tank-Automotive and Armaments Command</i> – Tanque-Automotivo e Comando de Armamento                          |
| <b>TBL</b>     | <i>Tool_Table</i> - Extensão do arquivo tool do EMC  |
| <b>TC</b>      | <i>Technical Committee</i> - Comitê Técnico  |
| <b>Tcl/Tk</b>  | <i>Tool Command Language / Tool Kit</i>  |
| <b>VDA</b>     | Association of the Automotive Industry   |
| <b>XML</b>     | <i>eXtensible Markup Language</i> – Linguagem de marcação extensível   |

# 1. INTRODUÇÃO

## 1.1. CONTEXTUALIZAÇÃO

Este trabalho de mestrado visa a concepção de um adaptador de programa de peça em STEP-NC a programa de peça em Código G para o controlador LinuxCNC (conhecido também como EMC2 - *Enhanced Machine Controller* versão 2) de forma a torná-lo aderente à norma ISO14649 para permitir que carregue o arquivo físico P21 (programa de peça em STEP-NC) de forma original. Também é desenvolvida uma Router CNC para validar o controlador LinuxCNC aderente á norma STEP-NC que será voltado para aplicação em fresadoras ou Routers CNCs de três eixos. É usada como plataforma o Linux Ubuntu (<http://www.linuxcnc.org>) e *opensource* (GNU/GPL license).

A manufatura rápida, eficiente e integrada tem sido um dos objetivos de pesquisadores desde o aparecimento da máquina a CNC (*Computer Numerical Control*) durante a década de 1980 [1] - [2]. No início dos sistemas CAD (*Computer Aided Design*) e CAM (*Computer Aided Manufacturing*), o problema de portabilidade dos modelos de um sistema para outro foi uma das questões chave que dificultaram e dificultam a disseminação destas ferramentas. Muitas soluções foram propostas na direção da padronização e do uso de normas para troca de dados tais como os padrões SET, VDA e IGES os quais obtiveram sucesso parcial [1]. Porém não eram totalmente adequados às necessidades da indústria de CAD, CAPP (*Computer Aided Process Planning*) e CAM.

No início da década de 1990, a comunidade internacional desenvolveu o conjunto inicial da norma ISO 10303 [2] - [3] conhecida como STEP (*STandard for the Exchange of Product model data*), que tem seus fundamentos nas normas mencionadas anteriormente [2].

A norma corrente, RS 274 [4] para programas NC (*Numerical Control*) utiliza padrões que não mudaram de maneira significativa desde a década de 1950 quando a primeira máquina NC foi desenvolvida no M.I.T. (*Massachusetts Institute of Technology*). Do mesmo modo que as primeiras máquinas NC, as atuais máquinas que usam CNC, continuam usando o mesmo padrão de programação, normalizado pela ISO 6893 [5], conhecidos como códigos G/M que se baseiam na descrição dos movimentos da ferramenta.

Atualmente a troca de programas NC entre máquinas CNC com controladores diferentes é virtualmente impossível a menos que tenham controladores do mesmo fabricante e uma configuração semelhante [2]. Em caso de correções ou modificações nestes programas NC, feitas pelo operador da máquina, não há um padrão que permita atualizar automaticamente as bases de dados de processo e projeto do produto. Estes problemas tem dificultado a integração da manufatura. Como consequência da necessidade de novos controladores associados ao padrão STEP e para resolver estas questões, na segunda metade da década de 1990, uma nova interface de dados chamada ISO 14649 [6] passa a ser desenvolvida usando como base a norma ISO 10303 (STEP), e sob direção do Comitê Técnico TC184 da ISO (*International Organization for Standardization*) dentro dos Sub-comitês SC1 e SC4.

Neste projeto a ISO 14649 [7] também é chamada com o termo STEP-NC. Porém é necessário notar que STEP-NC é o título de um programa da Organização IMS (*Intelligent Manufacturing Systems*) que combina os programas chamados Super-Model nos EUA e STEP-NC na Comunidade Europeia, Suíça e Coréia do Sul.

A ISO 14649 [7] define um modelo de dados orientado a objetos para CNC, com uma estrutura detalhada de interface de dados que incorpora a programação baseada em *features* [8], onde há uma gama de informações tais como a *feature* a ser usinada, tipo de ferramentas a usar, as operações a realizar, e o plano de trabalho [2] - [9]. Tudo isto em um arquivo físico no formato STEP [10]. Para cada operação a realizar sobre uma ou mais *features* é definida uma instrução chamada *workingstep*. Estes *workingsteps* fornecem as bases para o funcionamento dos planos de trabalho definidos em instruções chamadas *workplan* que definem a fabricação do componente. A Figura 1.1 descreve a estrutura geral da organização de dados da norma ISO14649.

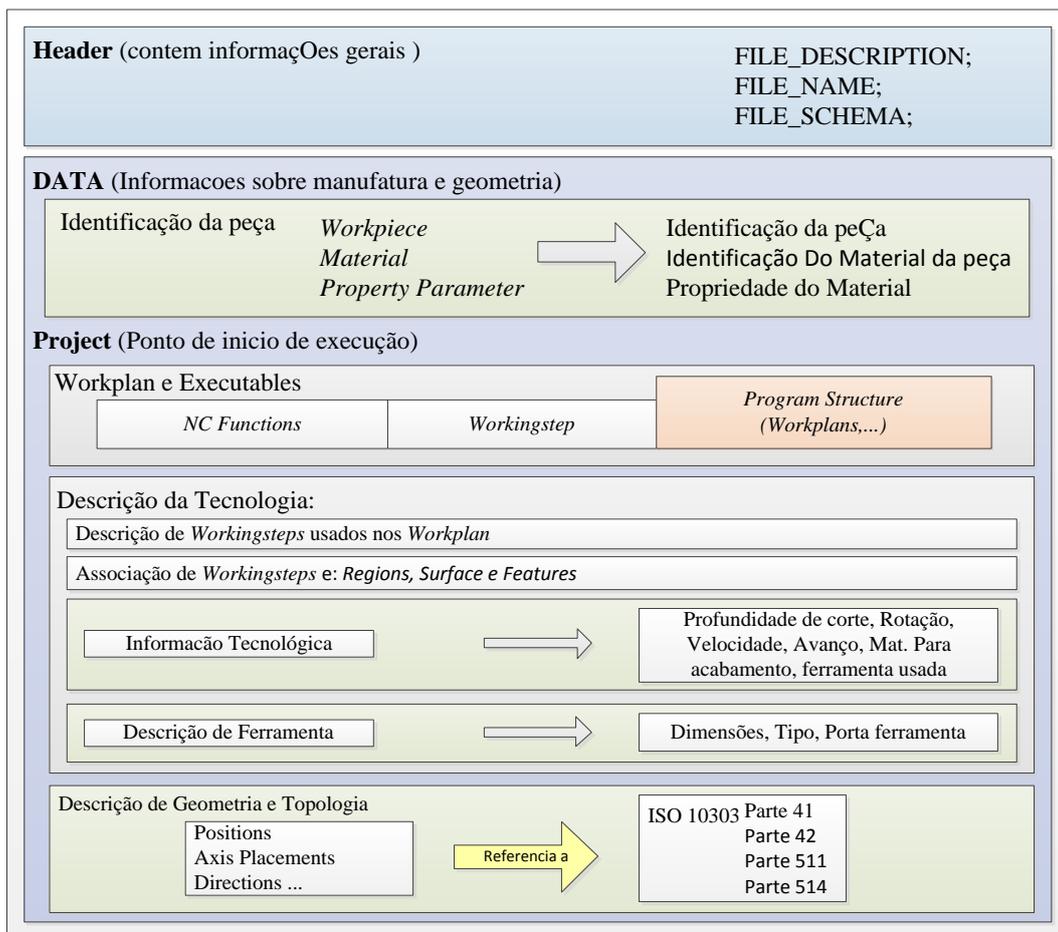


Figura 1.1: Estrutura geral da organização de dados na ISO 14649 (Adaptado de [2])

Pesquisadores da Universidade Federal de Santa Catarina, estão trabalhando em um sistema computacional CAD/CAPP/CAM aderente a STEP-NC, voltado à manufatura remota de peças mecânicas de característica prismática utilizando a Internet como meio de comunicação. Este sistema é chamado “STEP Modeler” [11].

No STEP Modeler, foram disponibilizadas vários tipos de *features* (de acordo com a norma ISO14649) para o usuário remoto. Um dos módulos do sistema verifica se a peça criada pelo usuário é válida do ponto de vista de fabricabilidade com as ferramentas disponíveis em uma máquina de comando numérico de três eixos, e no caso de não-fabricabilidade o sistema informa ao usuário o que a causou. Por fim, o sistema gera automaticamente o arquivo físico no formato P21 [10], 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 P21 como entrada. Segundo

WANG [12], as extensões do arquivo físico “.STEP” e “.STP” indicam que o arquivo contém dados em conformidade com os protocolos de aplicação (AP) do STEP.

O arquivo físico P21 (também chamado STEP ou ainda STP) gerado pelo software *STEP Modeler*, poderá ser utilizado em máquinas CNC reais compatíveis com o formato STEP parte 21. A Figura 1.2 ilustra um fragmento de um arquivo físico p21. Este arquivo é usado para fabricação de uma peça prismática. Pode-se observar, no programa de peça em STEP-NC, o *workplan* (e.g. #20 na Figura) constituído por *workingsteps* para faceamento, furação e usinagem de uma cavidade.

```
HEADER: FILE DESCRIPTION ((Test Program for SLOT,POCKET,HOLE3,1);
FILE NAME ('tape.STP',$(ISO14649),('),R.D.Allen', 'Loughborough University', 'UK');
FILE_SCHEMA (('MACHINING_SCHEMA', 'MILLING_SCHEMA'));
ENDSEC;
#1=WORKPIECE('PART2',#2,0.1,$,$,($200#201#202#203#204));
#2=MATERIAL('FC200', 'CAST_IRON',(#0));
...
#9-CLOSED_POCKET('pocket#1,(#40,#41.),#74,#0,#0,#0,#0,#0,#0,#17,#160)
#10=ROUND_HOLE('hole',#1,(#42,#43),#76,#146,#0,#0,#0.0,#0.0,#147,0.0,#18,#0,#0)
...
#19=PROJECT("EXECUTE EXAMPLE2"),#20,(#1,)
#20=WORKPLAN(#21,#22 #28#24,#28#26,#27,#28,#28#30,9 #71);
#21=MACHINING_WORKINGSTEP('ROUGH_POCKET1',#86,#89,#40);
#22=MACHINING_WORKINGSTEP('FINISH_POCKET1',#86,#9,#41);
#23=MACHINING-WORKINGSTEP('DRILL_HOLE1',#87,#10,#42);
#24=MACHINING-WORKINGSTEP('REAM_HOLE1',#87,#10,#43);
#25=MACHINING-WORKINGSTEP('ROUGR_SLOT1',#88,#11,#44);
#26=MACHINING-WORKINGSTEP('FINISH_SLOT1',#88,#11,#45);
...
#41=BOTTOM_AND_SIDE_FINISH_MILLING(#0,#0,'FINISH_POCKET1',#51,15.0,#0,#97,#61,#70,#0,
#0,2.0,10.0);
#42=DRILLING(#0,#0,'DRILL_HOLE1',#52,10.0,#0,#98,#62,#70,0.0,0.0)
#43=REAMING(06,#0,'REAM_HOLE1',#53,10.0,#0,#99,#63,870,0.0,0.0,.,F.,0.0,5)
...
#50=CONTOUR_BIDIRECTIONAL(0.0,.,F.,0,80,0);
#51=CONTOUR_PARALLEL(4.0,5.0,.,T.,CW,0.0,CONVENTIONALL);
#52=DRILLING_TYPE_STRATEGY(0.0,0.0,0.0,0.0,0.0,2.0,2.0,200.0);
#53=DRILLING_TYPE_STRATEGY(75.0,50.0,2.0,50.0,8.0,10.0,0.0,200.0);
...
#60=MILLING_TECHNOLOGY(0.0,TCP,#0.2,0.0,0.3,.,F.,.,F.);
...
#71=SETUP(SETUP1,#72,#85,(#83)); #72=AXIS2_PLACEMENT_3D('SETUP1',#164,#168#166);
...
#83=WORKPIECE_POSmONS(#1,#84,#0);
...
#96=MILLING_CUTTING_TOOL('TAPERED_ENDMILL_20MM',#126,(#106),(50.0),80.0,#91,0.0);
...
#125=MATERIAL('M3DU46','HSS/C0',(#0)); #126=TAPERED_ENDMILL(#136,4,null,.,F.,#0,#0);
...
#136=TOOL_DIMENSION(20.0,0.0,0.0,0.0,1.5,0.0,0.0);
...
#159=TOLERANCED_LENGTH_MEASURE(0.0,null,0.0,0.0);
#160=POLYLINE('pocket1',(#205,#206,#207,#208,#209,#210,#211));
...
#200=CARTESIAN_POINT('CLAMPING_POSITION1',(0.0,20.0,25.0));
...
#217=CARTESIAN_POINT('SLOT3_2',(10.0,20.0,0.0));
...
ENDSEC;
```

Figura 1.2: Saída de dados para fabricação de uma peça em STEP-NC (Adaptado de [2])

Para utilizar um programa de peça em STEP-NC é necessário desenvolver uma máquina CNC compatível com o formato STEP parte 21 usando, neste caso, o EMC como controlador. Este controlador é de código aberto e permite a integração de novas funcionalidades. Através da concepção de um adaptador é possível ler o arquivo físico P21 ou programa de peça em STEP-NC como entrada de dados em lugar do padrão RS-274.

Também existe a possibilidade de incluir a geração de código G para outras máquinas CNC industriais, baseadas só em código G, pois o adaptador desenvolvido é independente do CNC gerando código G a partir de um programa de peça em STEP-NC.

Na Figura 1.3, pode-se observar as atividades essenciais apoiadas ao ciclo de vida da concepção e fabricação de uma peça baseada em STEP-NC. O sistema *STEP Modeler* é usado para gerar o arquivo físico P21, enquanto o controlador EMC2 aderente á norma STEP-NC controla a fresadora CNC de acordo com um arquivo físico p21. Neste texto e usado o termo Router CNC para se referir à fresadora CNC.

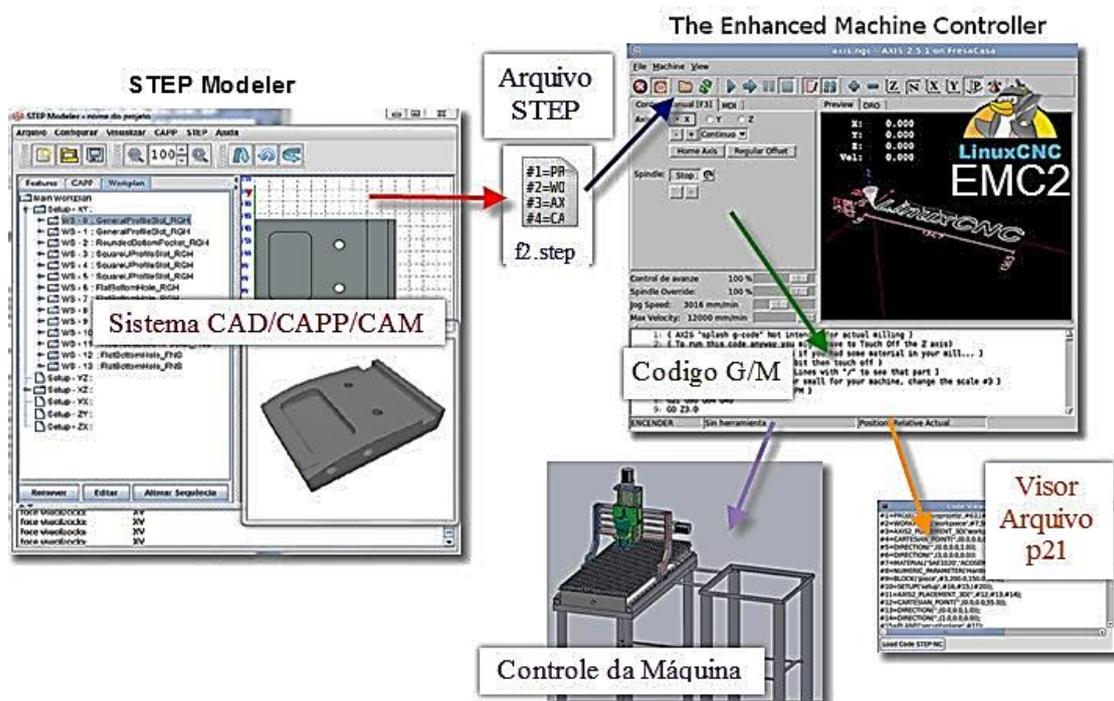


Figura 1.3: Tentativa do sistema finalizado

## 1.2. PROBLEMA DE PESQUISA

O contexto apresentado no item anterior mostra alguns motivos que levaram ao desenvolvimento do presente trabalho de mestrado. Logo, o problema consiste na falta de controladores NC baseados no padrão STEP-NC que permitam a integração da manufatura.

O padrão STEP-NC fornece um modelo de dados que permite a integração da manufatura entre sistemas CAD/CAPP/CAM e CNC. Este padrão deverá ser incluído nos futuros controladores NC. Portanto, comprovar que é viável a integração desta norma no CNC, propor-se o desenvolvimento de uma máquina-ferramenta tipo Router (fresadora com topologia tipo portal) controlada por meio do controlador EMC, adicionando a funcionalidade de carregar programas de peça em STEP-NC, e usando com estratégia o desenvolvimento de um adaptador de STEP-NC a RS-274, já que a proposta

é usar um interprete de código G baseado no RS-274, e para tal foi escolhido o controlador EMC (LinuxCNC) que usa este interpretador baseado em código G/M (RS-274).

### **1.3. OBJETIVOS**

#### **1.3.1. Objetivo Geral**

O objetivo geral é conceber um adaptador STEP-NC/RS274 para o controlador de máquina avançado EMC (LinuxCNC) de forma a torna-lo aderente à norma ISO 14649 para permitir carregar o arquivo físico P21 (programa de peça em STEP-NC) de forma original validando-o por meio de um estudo de caso em um Router CNC desenvolvido para esta finalidade.

#### **1.3.2. Objetivos específicos**

- ✓ Desenvolver uma arquitetura de integração de STEP-NC/RS274;
- ✓ Desenvolver um adaptador do arquivo físico p21 (programa de peça em STEP-NC) para o código de peça em Código G;
- ✓ Utilizar programas de peça em STEP-NC gerados por um software CAD/CAPP/CAM, como entrada no adaptador de STEP-NC a Código G/M;
- ✓ Adicionar o adaptador de STEP-NC a Código G/M como uma nova funcionalidade do controlador EMC;
- ✓ Visualizar o código de peça em STEP-NC original na GUI do controlador EMC;
- ✓ Projetar e fabricar um Router CNC a partir de referências de máquinas do mesmo tipo;
- ✓ Validar o controlador EMC2 aderente a STEP-NC por meio de um estudo de caso usando o Router CNC;

### **1.4. DELIMITAÇÃO DO TRABALHO**

O presente trabalho de mestrado é focado em aderir a norma ISO14649 no controlador de máquina avançado EMC, por meio de um adaptador que converte o arquivo físico p21 em código G. Um Router CNC é desenvolvido para validar a aderência à norma, por meio de um estudo de caso.

Como entrada, o EMC receberá de forma nativa programas de peça em STEP-NC provenientes do software *STEP Modeler* da UFSC/GRIMA, convertendo-o a código G de forma transparente para o operário do Router CNC. Portanto, a conversão de STEP-NC a Código G depende inteiramente das informações contidas nesse programa.

O *STEP Modeler* fornece várias *features* das contidas na norma ISO14649 parte 10. Consequentemente, o adaptador só tratará as *features* do programa da UFSC/GRIMA.

Já o desenvolvimento da Router é dividido em duas partes, desenho 3D de todos os componentes mecânicos da Router e fabricação/montagem dos componentes mecânicos e eletrônicos. Essas partes serão explicadas brevemente pois não são o foco do presente trabalho, mas é o meio pelo qual é validada a aderência do controlador LinuxCNC a STEP-NC.

### **1.5. APRESENTAÇÃO DO DOCUMENTO**

No capítulo 2 é apresentada a revisão bibliográfica em torno do problema da integração da manufatura e a norma ISO14649. Primeiramente são explicados individualmente os sistemas CAD,

CAPP, CAM e CNC, assim como a integração entre eles. Na sequência, são definidos os padrões e projetos que estão envolvidos com o padrão STEP-NC. Por último, se apresentam alguns trabalhos relacionados com a conversão de STEP-NC a Código G.

O capítulo 3 refere-se ao modelamento de um CNC aderente a STEP-NC, onde são explicadas em sequência as diferentes características e componentes que deve possuir o CNC com relação a carregar programas de peça em STEP-NC. Também são mostradas as funcionalidades do CNC aderente a STEP-NC, assim como, o modelamento de um adaptador de STEP-NC a Código G, que permitirá converter um programa de peça em STEP-NC a um programa de peça em Código G.

No capítulo 4, é mostrada a implementação do CNC aderente a STEP-NC. Esta implementação se divide em duas partes, um protótipo de adaptador de STEP-NC a Código G e o CNC com o adaptador incluso, o que permite garantir aderência à norma ISO14649. Ao final, o capítulo trata somente a implementação do protótipo do adaptador em quanto ao algoritmo que descreve suas funcionalidades. A segunda parte, é questão do capítulo seguinte.

O capítulo 5 ilustra a implementação do controlador LinuxCNC aderente a STEP-NC, por meio do adaptador desenvolvido no capítulo 4. São expostas as características do controlador, como ele é instalado e configurado. Apresenta-se também a metodologia para adição de novas funcionalidades relacionadas com o arquivo físico p21, carregá-lo e mostrá-lo junto à GUI do EMC.

Finalmente, no capítulo 6, têm-se os principais resultados obtidos da implementação do CNC aderente à norma STEP-NC por meio de um estudo de caso. Para encerrar, no capítulo 7 são apresentadas as conclusões sobre este trabalho e sugestões para trabalhos futuros.

Já no apêndice A mostra-se uma metodologia de trabalho e conceitos sobre tradutor de linguagem para a modelagem do CNC aderente a norma STEP-NC. No apêndice B é apresentada uma revisão de literatura sobre o controlador LinuxCNC suas características e funções. Seguidamente no apêndice C é exibido o projeto e fabricação da Router CNC de forma resumida. No apêndice D são apresentados vários exemplos sobre a criação de programas STEP-NC parte 21 usando o STEP Modeler. No apêndice E é mostrado o algoritmo da GUI de visualização do arquivo físico p21 usando a linguagem Python. Este programa em Python é executado junto com a GUI do controlador LinuxCNC.

O apêndice F ilustra o estudo de capacidade da Router desenvolvida neste trabalho por meio de um estudo de caso de uma peça teste usinada. Por último está o apêndice G que apresenta o formato do arquivo de configuração usado pelo adaptador de STEP-NC a código G. Neste arquivo estão os dados referentes aos parâmetros de corte da Router, entre outros dados usados pelo adaptador. Nos anexos se encontram informações relacionadas com os códigos G usados no adaptador, assim como características de alguns componentes usados no desenho e montagem da Router CNC.

## **2. REVISÃO DA LITERATURA SOBRE STEP-NC**

O intercâmbio de informações entre CAD/CAPP/CAM e CNC tem sido, na últimas décadas, um dos maiores problemas na indústria dado a perda de informações no ciclo de desenvolvimento de um produto. Atualmente existe um padrão que permite esse intercâmbio, padrão este conhecido como STEP-NC (ISO14649).

Este capítulo apresenta a revisão bibliográfica feita da teoria ao redor da norma ISO 14649. São apresentados conceitos básicos que serão utilizados no resto do desenvolvimento do trabalho. Também são mostrados alguns trabalhos relacionados com STEP-NC, destacando-se a suas contribuições no presente projeto.

### **2.1. SISTEMAS ASSISTIDOS POR COMPUTADOR**

Os sistemas assistidos por computador aproveitam a tecnologia da computação para facilitar muitos dos trabalhos realizados na área da engenharia, e.g. desenho CAD, manufatura CNC, etc. A seguir é apresentada uma resenha básica de alguns dos sistemas assistidos por computador utilizados na área da manufatura e de fabricação.

#### **2.1.1. Sistema CAD**

O Projeto Assistido por Computador (*Computer-Aided Design* ou CAD) é um sistema eletrônico que permite desenhar novas peças e produtos ou modificar outros já existentes, substituindo o desenho tradicional na mão [13]. O sistema CAD ajuda a poupar tempo e agilizar as tarefas de edição de projetos existentes em lugar de concebê-los do início.

Atualmente 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, desenho de placas eletrônicas.
- Arquitetura: projetos arquitetônicos de qualquer tipo.
- Engenharia civil: edificações, projeto de tubulações, projetos elétricos, etc.

Segundo TICONA [11], apesar da sigla "CAD" incluir o termo "design", observa-se que são poucos os casos em que o computador efetivamente projeta alguma coisa, servindo mais como uma ferramenta de auxílio à confecção de desenhos de engenharia. Sua maior contribuição ocorre na modelagem dos produtos e componentes e no detalhamento de seus desenhos.

#### **2.1.2. Sistema CAPP**

O Planejamento do Processo Assistido por Computador (*Computer Aided Process Planning* ou CAPP) se ocupa da seleção de métodos de produção, ou seja: ferramentas, suportes, maquinaria, sequência de operações e montagem [14], sendo uma das tarefas mais difíceis e especializadas na área de produção segundo TICONA [11].

Segundo GROOVER [15], o CAPP pode significar: “a atividade responsável pela determinação dos processos e sua sequência para a transformação da matéria prima na peça acabada”.

No geral, o CAPP demanda grande consumo de tempo ao considerar uma operação (e.g. um produto terminado) como um sistema integrado, pois é uma tarefa muito complexa, isto é, identificar todas as informações, operações e passos individuais que intervêm na fabricação de uma ou várias peças coordenadas entre si, sendo realizada em forma eficiente e confiável. É por isto que o sistema CAPP é um parte importante entre CAD e CAM.

### **2.1.3. Sistema CAM**

A Manufatura Assistida por Computador (*Computer Aided Manufacturing* ou CAM) está enfocada para a automatização de um processo de manufatura, de manipulação de materiais mediante técnicas integradas de controle automático (SCADA, PLC, etc.) e de controle numérico computadorizado (CNC). Essa integração fornece ao processo de manufatura uma grande flexibilidade e melhora notavelmente a reprodutibilidade das peças fabricadas [16].

Segundo GROOVER [15], o CAM se define como o uso eficaz da tecnologia informática no planejamento de fabricação e controle, estando mais estreitamente relacionada com as funções de engenharia de fabricação, tais como processos de planejamento e de controle numérico (NC).

Assim, o propósito principal do sistema CAM é criar um processo de produção mais rápido, componentes e ferramentas com mais precisão e uso de menor quantidade de matéria-prima e energia.

## **2.2. INTEGRAÇÃO DE SISTEMAS CAD/CAPP/CAM/CNC**

Segundo XU [17], as tecnologias concernentes a CAD, planejamento do processo, fabricação e controle numérico, amadureceram a um ponto que as soluções de software comercializadas e sistemas industriais podem ser adquiridas facilmente. Estas soluções ou sistemas estão conectados mas não necessariamente integrados de uma forma perfeita.

O termo “*Island of Automation*” (“ilhas de automação”) tem sido utilizado para descrever estes grupos desconectados de sistemas de integração. Estas “ilhas de automação” prejudicam o usuário final.

As empresas de engenharia estão executando seus trabalhos cada vez mais de uma forma globalizada. Essas ilhas de automação precisam estar conectadas para atender e servir melhor ao ambiente colaborativo e distribuído.

É evidente que as empresas estão se esforçando com essa estratégia de integração para incluir todos os níveis como CAD, CAPP, CAM e CNC. Em alguns casos, onde a integração não existe entre estes níveis, as tecnologias de produtos pode vir a uma parada súbita por causa dessa barreira.

Segundo XU [18], no esforço para alcançar a integração dos sistemas CAD/CAPP/CAM/CNC, apareceram dois tipos de modelos, sendo eles: modelo centralizado e modelo colaborativo.

Em um modelo centralizado, as atividades de fabricação ocorrem dentro de um único fabricante ou algum fabricante que possui infraestrutura de informação similar. Neste modelo, os formatos de dados de propriedade são comumente usados.

Em um modelo colaborativo, uma categoria intermediária é adicionada usando por exemplo um formato de troca de dados neutro. Como tal, com esses modelos de colaboração, torna-se mais fácil o ambiente de produção. A Figura 2.1 apresenta os fluxos de dados nestes dois modelos.

No modelo centralizado, os sistemas CAD e CAM usa o mesmo formato de dados proprietário. Ao longo dos anos, os fornecedores de sistemas CAD/CAM tiveram êxito no desenvolvimento de diferentes formatos de dados proprietários para apoiar seus sistemas durante os processos de manufatura e projeto. As vantagens desse modelo são óbvias. Sistemas CAD e CAM são unificadas pelo mesmo formato de dados para que a incompatibilidades de dados entre eles sejam eliminadas. Além da geometria pura, os sistemas integrados CAD/CAM podem servir para outras atividades que vão desde a concepção à programação NC.

Existe, porém, um problema com o modelo centralizado. Seu modelo de dados é proprietário e não permite a manufatura colaborativa devido à incompatibilidade de dados entre os vários fabricantes.

O modelo colaborativo é destinado a resolver este problema por meio de da utilização de formatos de dados neutros, tais como IGES [19].

O Formato de dados neutros fornece uma camada intermediária para conectar sistemas CAD e CAM. Com a ajuda de formatos de dados neutros, o modelo colaborativo cria um ambiente de manufatura colaborativa e faz com que a troca de dados seja possível para grandes projetos em nível internacional.

Assim como XU [17] apresenta estes dois métodos, existem vários sistemas integrados CAD/CAPP/CAM/CNC aderentes ou relacionados a STEP-NC.

### **2.2.1. Sistema integrado CAD/CAPP/CAM/CNC aderente STEP-NC**

Os modelos centralizado e colaborativo, apresentados anteriormente, possuem vários problemas no intercâmbio de dados. Segundo XU [17], no modelo colaborativo há perda de dados quando ocorre a transmissão desde o desenho (CAD) à manufatura (CAM).

Em um modelo centralizado (proprietário), é ignorada a informação adicional do desenho e da manufatura (e.g. informações de *feature*) o que impossibilita usá-lo com outros formatos de dados como STEP-NC.

Por outro lado, o IGES foi designado somente para o intercâmbio de informação geométrica. Às vezes, durante a transferência de dados, algumas informações são perdidas, o que faz necessário a reconstrução geométrica (CAD) ou reparação do formato de dados (CAM). Além disso o IGES não é uma norma internacional.

Em contrapartida, a comunidade internacional da indústria da manufatura apoiada pela Organização Internacional de Normalização, fez um grande esforço para introduzir o padrão STEP no modelo colaborativo do sistema integrado CAD/CAPP/CAM/CNC. O padrão STEP está sendo usado recentemente na indústria para melhorar o intercâmbio de dados entre estes sistemas. O padrão é informalmente conhecido como STEP-Compatível com Controle Numérico (*STEP-Compliant Numerical Control*) ou STEP-NC de forma mais breve.

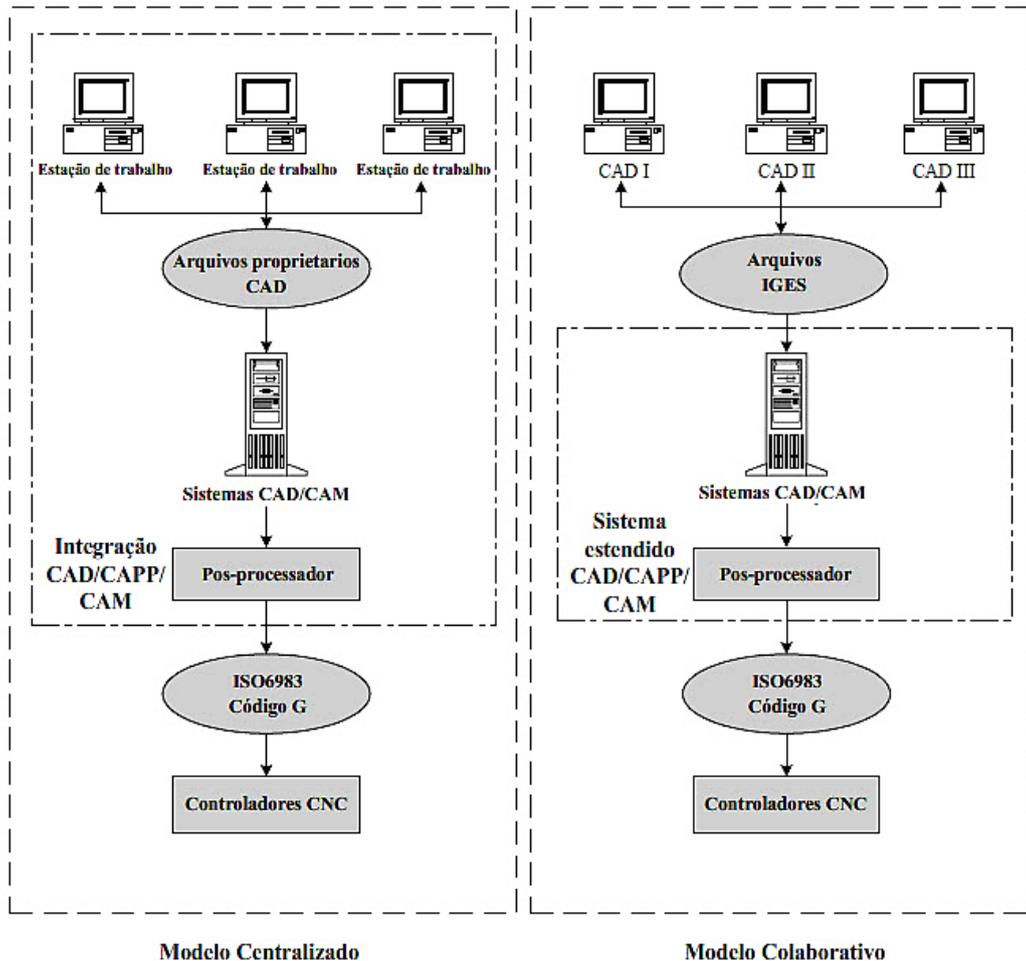


Figura 2.1: Modelos de integração CAD/CAPP/CAM/CNC (Adaptado de [17])

O STEP-NC é identificado pelo “ISO14649: Modelo de Dados para Controladores Numéricos Computadorizados [7]”. Esse padrão define o Modelo de Referência de Aplicação STEP-NC. Com STEP, começa-se a estender o modelo de informações de manufatura, emergindo como um novo paradigma de integração de CAD/CAPP/CAM/CNC [20]. Na Figura 2.2, apresenta-se esta integração.

Segundo XU [17], a chave desse paradigma é que não é necessário a conversão de dados, preservando assim os dados de todo o projeto e cadeia de produção. Portanto, XU [17] propõe o uso das normas ISO10303 e ISO14649 para apoiar o intercâmbio de dados entre sistemas CAD, CAPP, CAM e CNC e assim facilitar o fluxo de dados entre eles.

XU [21] e NEWMAN et al. [22] expõem que embora tenha havido avanços significativos que melhoram o software e arquitetura de máquinas ferramenta CNC, ainda há a busca por uma linguagem comum para CAD, CAPP, CAM e CNC, que integre e traduza o conhecimento de cada etapa na manufatura. Segundo XU et al. [21] e NEWMAN et al. [22], a linguagem que está em desenvolvimento para fornecer padrões consistentes orientados para manufatura automática e de qualidade usando CNC é o *STEP-Compliant NC* ou como foi exposto anteriormente STEP-NC.

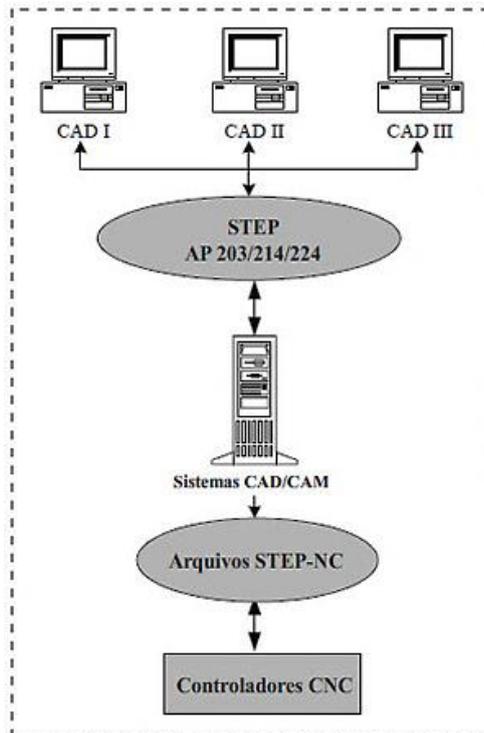


Figura 2.2: Sistema Integrado CAD/CAPP/CAM/CNC usando STEP e STEP-NC (Adaptado de [20])

### 2.3. A ORGANIZAÇÃO INTERNACIONAL DE NORMALIZAÇÃO (ISO)

A ISO (*International Organization for Standardization*) é o maior desenvolvedor do mundo de normas internacionais voluntárias. Normas Internacionais que proporcionam o estado de arte para especificações produtos, serviços e boas práticas, ajudando a tornar a indústria mais eficiente e eficaz. Desenvolvido através de um consenso global, a ISO ajuda a quebrar as barreiras do comércio internacional [23].

A revisão das normas ISO (Padrões) que se expõem a seguir estão relacionados tanto com o antigo quanto o atual modelo de dados para CNC, a norma ISO14649.

Primeiramente, revisa-se a ISO6983 para verificar as suas características e entender as suas falências, segue-se com a norma ISO10303, pois é nela que está baseada a norma ISO14649.

### 2.4. O ISO6983 - CÓDIGO G/M

O padrão ISO6983 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 [11]. Neste texto usa-se código G ou código G/M para referir-se a este padrão.

#### 2.4.1. Partes do ISO6983

O padrão ISO6983 não é muito flexível e, portanto, não garante a interoperabilidade entre máquinas NC, a não ser que essas tenham o mesmo intérprete NC. Além disso, o padrão ISO 6983 encontra-se dividido principalmente em duas partes, mostradas a seguir:

### **i. Parte 1 – Modelo de dados para posicionamento e movimentação**

Esta parte do ISO foi publicada em 1982 [24] e sua última versão em 2009 [5]. 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 o intercâmbio de programas entre máquinas controladas numericamente do mesmo tipo por classificação, processo, função, tamanho e precisão [11].

### **ii. Parte 2 – Funções preparatórias e Miscelâneas – G/M**

Especifica a codificação e a elaboração das funções preparatórias (G) e miscelâneas (M) (classes 1 a 8) [11]. Esta parte é apenas um rascunho [25].

#### **2.4.2. Programação de máquinas NC com o ISO6983**

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” [26].

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, que 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 [27].

Assim como existe o ciclo fixo G83 para furação, também existem outros ciclos específicos implementados por cada fabricante de máquinas CNC.

Outra característica da norma ISO6983 é que ela permite a implementação de subprogramas e macros. Subprogramas são conjuntos de linhas de código que se usam várias vezes em um ou mais programas principais. A implementação destes subprogramas é específico de cada fabricante e tem diferenças entre controladores [27].

Estas variações de programação dos ciclos fixos e dos subprogramas, que segundo MAEDER et al. [26], são mais de 500 “dialetos”, resulta em problemas de interoperabilidade entre máquinas CNC. Portanto, é necessário uma abordagem padronizada que permita a interoperabilidade na manufatura CNC, e é por esta razão que os padrões relacionados à fabricação evoluíram [27] - [28].

Existe um padrão chamado RS274 que é baseado no padrão ISO6983. Esse padrão foi uma revisão da parte 1 do padrão ISO6983 aprovada em 1980. Assim como padrão RS274 é baseado no ISO6983 há outras modificações ou revisões que tomam nomes diferentes, e.g. BLC, RS274NGC, DIN, entre outros.

Como já foi mencionado, o padrão RS274 (a versão mais atual é chamada RS274D) possui algumas modificações com relação com o ISO6983, no entanto neste texto é usado o termo código G ou G/M como referência tanto do padrão ISO6983 quanto o padrão RS274 e o RS274NGC.

### 2.4.3. O intérprete NIST RS274NGC

Este intérprete lê o código de controle numérico NC (ISO 6983 ou Código G/M) e produz as chamadas para um conjunto de funções canônicas para realizar a usinagem. O intérprete é um software escrito em C++. A saída do intérprete pode ser usada para movimentar centros de usinagem de três a seis eixos. A entrada para o intérprete RS274 é código no dialeto definido pelo controlador de Próxima Geração (NGC ou RS274NGC), que é um código G/M com algumas modificações [4].

O intérprete pode ser compilado como um programa de computador autônomo ou pode ser integrado com o controlador da máquina avançada EMC2. Para a entrada do intérprete é possível inserir um arquivo físico com código G/M ou um usuário digitando inteiramente o código G/M. Os comandos de saída podem ser impressos para uso futuro ou podem ser executados diretamente em um centro de usinagem. A Figura 2.3 mostra um fragmento de uma linha de programação NC.

Existe ainda um relatório que inclui uma descrição completa da linguagem de entrada RS274/NGC e as funções canônicas de usinagem chamados pelo intérprete [4].

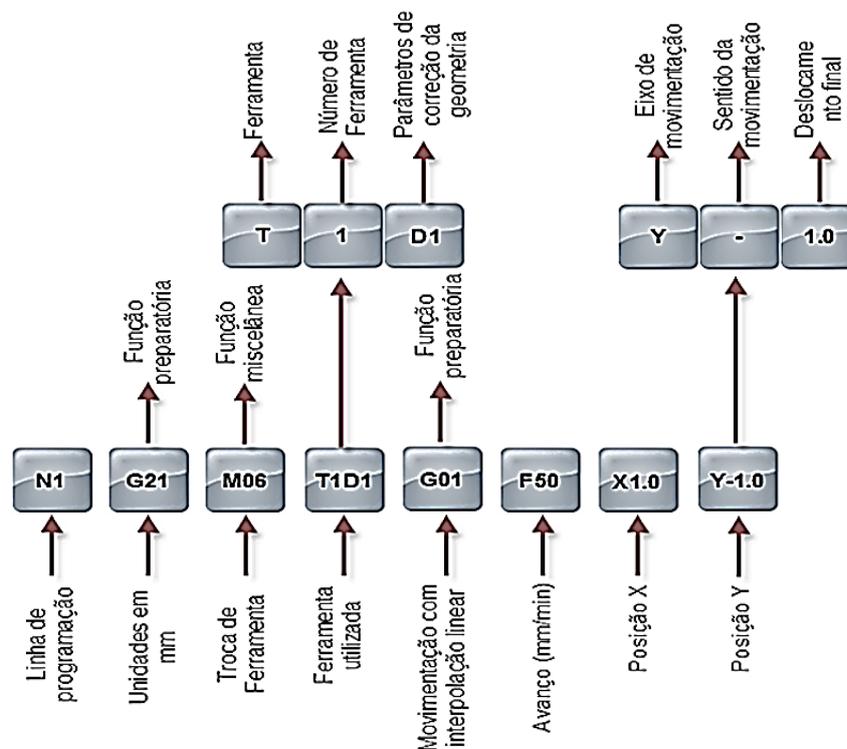


Figura 2.3: Linha de programa típica usando a norma ISO 6983 aplicando como referência o intérprete RS274NGC (Adaptado de [11])

### 2.5. O ISO10303 - STEP

O título oficial do ISO10303 é “Automatização de sistemas industriais e integração – representação e intercâmbio de dados entre produtos”, mas é também conhecido como STEP (acrônimo de *Standard for the Exchange of Product model data*) ou Padrão para o intercâmbio de dados do modelo de produto.

O STEP é um padrão internacional para a representação e intercâmbio de informação de produtos industriais, sendo dividido em várias partes. Essas partes cobrem diferentes temas da norma ISO10303. O objetivo é facilitar um mecanismo que seja capaz de descrever a informação de um

produto através do ciclo de vida do mesmo, independentemente de qualquer sistema (CAD, CAPP, CAM) em particular [29].

A referência inicial do STEP foi ter a coleção de todos os dados do modelo do produto. A ideia 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) [30].

Além disso, O STEP não apenas serve para um intercâmbio neutral de arquivos, como também é uma base para desenvolver e compartilhar bases de dados de produtos e arquivos.

Normalmente STEP pode ser usado para intercambiar dados entre sistemas CAD, CAM, CAE, PDM/EDM e outros sistemas CAx. O STEP suporta modelos de desenho mecânico, elétrico, análises e manufatura, adicionando informações específicas de várias indústrias como automotriz, aeroespacial, construção de edifícios, barcos, plantas de processo, etc [29].

Valendo-se da flexibilidade do ISO 10303 para suportar vários modelos de dados, foi criada uma extensão para a representação de um novo modelo de dados para CNC conhecido como STEP-NC.

### 2.5.1. A relação entre STEP e STEP-NC

O padrão STEP-NC adota a definição de STEP modificando-o em conformidade com o CNC, na Figura 2.4 mostra-se a relação entre STEP e STEP-NC.

A norma ISO14649 usa dois Protocolos de Aplicação ou AP (*Application Protocol*) da norma ISO 10303, a saber:

- O AP 203 com o qual se determinam as informações de referência para superfícies *free-form* e desenho 3D [29].
- O AP 224 para as características de usinagem 2.5D [29].

Ao mesmo tempo existe um esquema identificado como EXPRESS, que é usado para o modelo de dados do STEP-NC. Este esquema é feito com referência na norma ISO 10303 Parte 11 conhecida como *EXPRESS* e a implementação do método para o formato do arquivo físico com o ISO10303 parte 21 e ISO10303 parte 28. A parte 21 (regra de codificação de texto claro) é para o formato de arquivo STEP ou STP e a parte 28 para o formato do arquivo em XML.

Neste projeto, trabalhar-se-á com o esquema *EXPRESS* para modelo de dados e somente com a codificação de texto da Parte 21 do STEP. Já o STEP-NC está sendo desenvolvido pelo ISO TC 184 (*Technical Committee in International Standardization Organization*) CS1 e SC4 (Sub-Comissões de 1 e 4). Estritamente falando, existem duas versões do modelo de dados STEP-NC. O primeiro é o Modelo de Referência de Aplicação (ARM), a saber, a norma ISO14649 que descreve os processos de usinagem em termos de domínio da terminologia. O segundo é o Modelo de Aplicação Interpretada (AIM), a saber, a norma ISO10303 AP 238 [29] que mapeia o ARM para os recursos existentes integrados do STEP (Padrões de intercâmbio de dados do modelo do produto como o ISO 10303 formalizado).

Segundo SUK et al. [31], recentemente, o padrão STEP-NC foi harmonizado com o conjunto de normas do STEP como a norma ISO10303 AP238, que oferece a possibilidade de integração de aplicação durante o desenho e a manufatura.

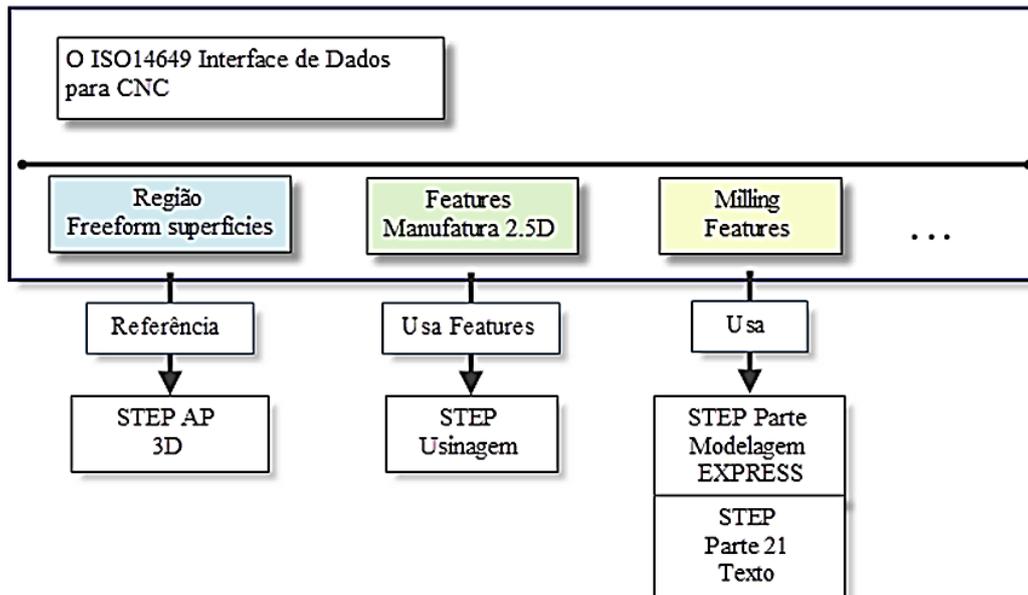


Figura 2.4: A relação entre a ISO 14649 e ISO 10303 (Adaptado de [31])

Este esforço para uma melhor integração é para possibilitar o intercâmbio de informações baseadas na manufatura, nomeada como a suíte de STEP-Manufacturing recomendado pela ISO TC184 SC4 e SC1. Essa suíte mostra as funções e as informações associadas aos padrões em termos de Protocolos de Aplicação do STEP [31], mencionados anteriormente.

Neste texto, usa-se a referência à norma ISO 14649 como o modelo de dados de STEP-NC.

### 2.5.2. Protocolo de Aplicação

Um protocolo de aplicação ou *Application Protocol* (AP) é uma parte do STEP destinada a ser implementado no uso industrial levando um objetivo específico, por exemplo: o AP 21 do ISO 10303 é usado para a codificação do arquivo físico do STEP-NC.

Segundo SCRA [29], o STEP usa protocolos de aplicação (APs) para especificar a representação de informações sobre o produto para uma ou mais aplicações.

Espera-se que várias centenas de APs possam ser desenvolvidos para suportar muitas aplicações industriais que o padrão STEP espera cobrir.

### 2.5.3. O Esquema EXPRESS

O Esquema *EXPRESS* é uma linguagem de modelagem de dados padrão para dados de produto definido na norma ISO10303 parte 11, conhecido também como o manual de referência da linguagem *EXPRESS*.

Existem dois métodos de linguagem ou esquema *EXPRESS*:

- Método *EXPRESS* textual
- Método *EXPRESS* Gráfico ou *EXPRESS-G*

O *EXPRESS* textual refere-se a uma linguagem similar à linguagem de programação orientada a objetos, por exemplo C. Isto é, *EXPRESS* suporta a descrição de estrutura de dados de um objeto, mas os objetos não têm nenhum método ou função executável. No *EXPRESS*, a definição de um tipo de objeto é chamada de “entidade” ao invés de classes (termo usado em C). Uma propriedade de uma entidade é chamada de atributo da entidade (por exemplo um atributo de um quadrado é a área). O *EXPRESS* também suporta a relação pai/filho entre entidades. O pai é chamado supertipo e o filho de subtipo. Uma entidade filho (subtipo) herda todos os atributos de seus pais(supertipo) [32]. Na Figura 2.5 pode-se observar um exemplo simples do esquema *EXPRESS* textual:

```

SCHEMA Family;

ENTITY Person
  ABSTRACT SUPERTYPE OF (ONEOF (Male, Female));
  name: STRING;
  mother: OPTIONAL Female;
  father: OPTIONAL Male;
END_ENTITY;

ENTITY Female
  SUBTYPE OF (Person);
END_ENTITY;

ENTITY Male
  SUBTYPE of (Person);
END_ENTITY;

END_SCHEMA;

```

Figura 2.5: Exemplo simples do esquema *EXPRESS* textual [33]

O *EXPRESS-G* descreve um método gráfico por meio de diagramas. Os diagramas estão descritos por:

- Entidades em caixas sólidas.
- Tipos de dados simples em caixas sólidas com uma linha dupla na extremidade direita.
- Tipos de dados definidos em caixas com bordas pontilhadas.
- Enumeração de tipos de dados em caixas com bordas pontilhadas e uma linha dupla na extremidade direita.
- Subtipos como uma linha grossa para conectar uma entidade supertipo a uma entidade subtipo com um círculo no final subtipo.
- Os atributos obrigatórios como uma linha fina para a conexão de uma entidade a um atributo da entidade, com um círculo no final atributo e o nome do atributo (adicionando descrições) no texto ao lado da linha.
- Atributos opcionais como uma fina linha tracejada para a conexão de uma entidade a um atributo da entidade, com um círculo no final atributo e o nome do atributo (adicionando descrições) no texto ao lado da linha

Na Figura 2.6, é mostrado o esquema *EXPRESS-G* com relação ao exemplo *EXPRESS* textual.

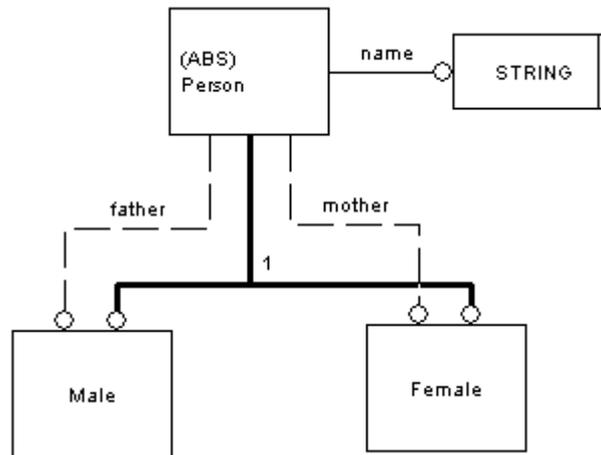


Figura 2.6: Exemplo simples do esquema EXPRESS-G [33]

## 2.6. O ISO14649 - STEP-NC

O STEP-NC é uma extensão do STEP, logo, este padrão aproveita a característica do formato neutro de dados do STEP para a informação digital de um produto, então esses dados podem ser compartilhados e intercambiados entre várias plataformas, sendo a única forma de compatibilidade.

A norma ISO14649 padroniza a maneira de como a informação sobre a usinagem CNC pode ser adicionada às peças representadas no modelo de produto do STEP [34]. Assim, esse padrão foi desenvolvido com a finalidade de fornecer um modelo de dados para um novo conjunto de controladores CNC inteligentes e, conseqüentemente, apresentar um homólogo melhorado do antigo Código G/M como foi mostrado na sessão 2.4.

Como já foi mencionado, o padrão STEP-NC tem um Modelo de Referência de Aplicação (ARM) identificado com o número ISO 14649-1 [7], o mesmo está direcionado especificamente à programação NC, buscando satisfazer o objetivo de um controlador CNC em um nível mais elevado, tornando a geração do programa mais rápida e eficiente. Em conjunto com o ARM está o Modelo Interpretado da Aplicação (AIM) do STEP-NC, ISO 10303 parte 238 [35].

O ARM também fornece um modelo de dados orientado a objetos voltado para CNCs com uma interface de dados estruturada e detalhada que incorpora a programação baseada em *features*<sup>1</sup> a serem usinadas, os tipos de ferramentas usadas, as operações a serem realizadas (*workingsteps*) e o plano de processo (*workplan*) [11].

Por último está o AIM do STEP-NC, cuja principal importância está em melhorar a troca de dados entre sistemas CAD/CAM e CNC [21].

### 2.6.1. O Sistema STEP-NC

O padrão STEP-NC foi abordado por muitas filiações, instituições e empresas desde vários pontos de vista. Assim sendo, existem várias terminologias relacionadas com o sistema STEP-NC. A seguir são apresentadas diversas definições dos autores que trabalham com ele.

<sup>1</sup> Uma *feature* descreve a área de remoção de material em uma peça. Também, descreve o resultado desejado de um processo de usinagem.

- **STEP-NC:** há dois sentidos, o curto e o amplo. Em sentido curto, STEP-NC é o idioma da nova interface entre CAM e CNC. Em um sentido amplo, STEP-NC inclui não apenas o idioma da nova interface, mas também serve para implementar tecnologias de CAD/CAM e software CNC ou produtos baseados na nova interface.
- **ISO 14649:** ISO 14649 é um padrão internacional de especificação que define o modelo de dados para STEP-NC. Ele especifica o conteúdo da informação e semântica (ICS) para diversos processos de fabricação CNC e recursos, incluindo ferramentas de corte e máquinas-ferramentas.
- **STEP-NC modelo de dados:** STEP-NC modelo de dados significa o conteúdo da ISO 14649. Em outras palavras, é o mesmo que o sentido curto do STEP-NC.
- **STEP- Compilador CNC:** se relaciona a uma espécie de novo controlador CNC com a implementação do STEP-NC, ou seja, tendo STEP-NC como entrada e assim controlar o movimento da máquina-ferramenta. Dependendo de como é a interface do STEP-NC e de como é usado, o compilador é classificado em três tipos (ver sessão 2.10).
  - Tipo A (Convencional)
  - Tipo B (Básico)
  - Tipo C (Inteligente)
- **STEP-CNC:** é uma abreviação de STEP-compilador CNC.
- **Tecnologia STEP-NC:** descreve várias tecnologias necessárias para a implementação de software e produtos baseado na interface STEP-NC [31].

Estritamente neste projeto, usa-se a definição de STEP-NC no sentido curto, voltado para a aplicação em fresamento em 3 eixos (XYZ).

### 2.6.2. STEP-NC e o Novo Modelo de dados para CNC

A norma ISO14649 descreve um novo modelo de transferência de dados entre CAD/CAM e os sistemas de máquinas CNC, que substitui a norma ISO6983 [36]. Porém, para entender o modelo de dados do STEP-NC, deve-se entender primeiramente as limitações que a norma ISO6983 possui. O código G contém apenas (como mostrado na Figura 2.7) o movimento dos eixos, velocidade, avanço, troca de ferramenta, posição da ferramenta e *On/Off* do fluido refrigerante.

Essas informações são muito básicas e dificultam a compreensão do fluxo operacional, as condições de usinagem e a especificação de ferramentas quando se lê o programa NC de uma peça. É por isto que é virtualmente impossível para o controlador CNC executar um controle inteligente, autônomo, lidando com erros e casos de emergência [31].

Para enfrentar essas limitações da norma ISO6983, o padrão STEP-NC contém informações funcionais necessárias (como se mostra na Figura 2.7) como os dados geométricos, dados das *features* de usinagem e dados do processo de usinagem.

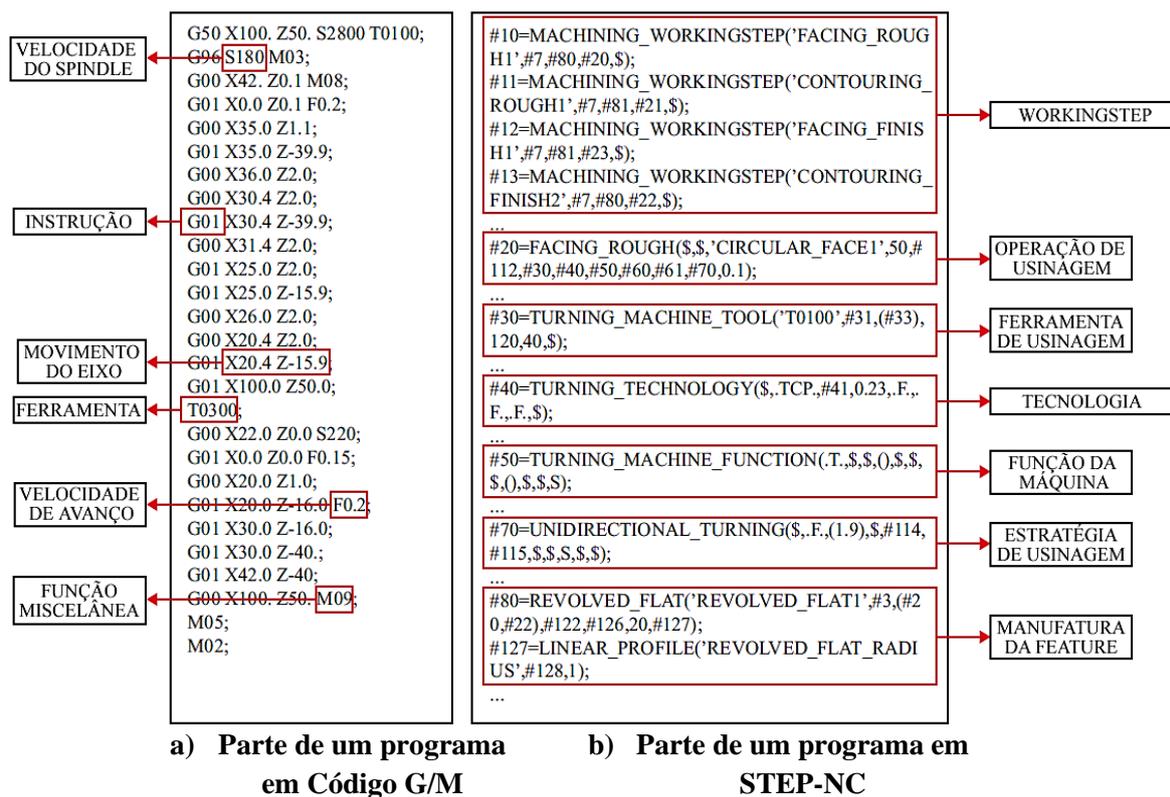


Figura 2.7: Comparação entre Código G/M e STEP-NC (Adaptado de [31]).

Os dados geométricos normalmente são originados a partir de sistemas CAD e são descritos no protocolo de aplicação 203 da norma ISO10303 [29]. Esta norma inclui toda a informação necessária para definir a geometria de uma peça acabada.

A norma ISO14649 usa o protocolo de aplicação 224 da norma ISO10303 como base para definição de *features* [37]. Os dados de features de usinagem e os dados do processo de usinagem se originam normalmente em sistemas CAM. Esses dados definem os parâmetros tecnológicos a serem empregados durante o processo de corte, que compreendem velocidade de corte, velocidade de rotação da ferramenta e descrições do ferramental requerido para executar cada uma das operações de usinagem [11].

O padrão ISO14649 se harmoniza com o padrão ISO10303 AP 224 [7]. A Figura 2.8 apresenta um esquema com a descrição geral do modelo de dados do ISO14649.

É importante ressaltar que o modelo de dados do STEP-NC usa a parte 21 da norma ISO10303 como método de implementação do arquivo físico.

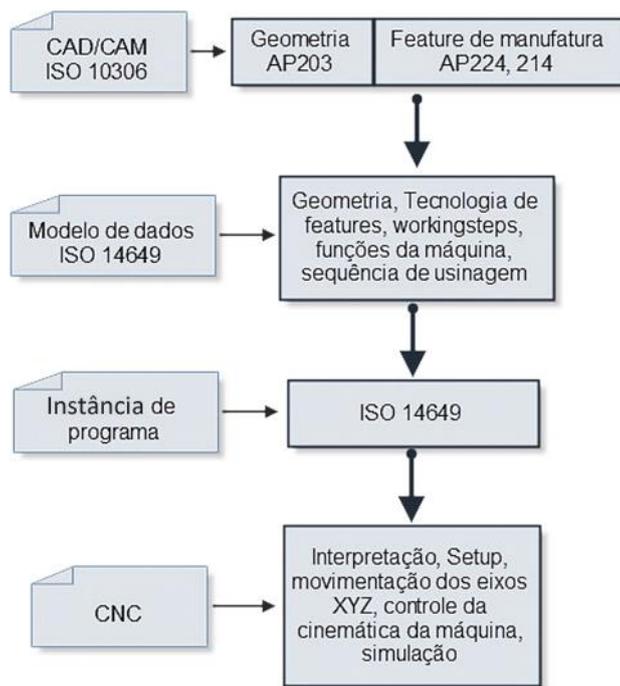


Figura 2.8: Descrição geral do modelo de dados do ISO14649 (Adaptado de [31]).

### 2.6.3. ISO 10303 Parte 21

A parte 21 do ISO 10303 especifica o método de implementação de arquivo físico, o qual define as regras básicas para armazenar dados usando o esquema *EXPRESS* em um arquivo físico. O arquivo da parte 21 só define as relações entre entidades que são definidas por um esquema *EXPRESS* externo [12].

A parte 21 ou também ISO 10303-21 define o mecanismo de codificação na forma de representar os dados de acordo com um esquema *EXPRESS* dado, mas não é o próprio esquema *EXPRESS*.

O arquivo STEP-NC também é chamado de arquivo p21, arquivo STEP ou ainda arquivo STP. As extensões do arquivo físico “.STP” e “.STEP” indicam que o arquivo contém dados em conformidade com os protocolos de aplicação (AP) do STEP, enquanto a extensão “.P21” deve ser utilizada para todos os outros fins.

A seguir são descritas as partes do ISO 14649 para fresado que incluem o esquema *EXPRESS*, utilizado para especificar o modelo de informação NC.

### 2.6.4. Partes do ISO 14649 geral e para fresamento

Neste projeto faz-se referência somente à tecnologia de usinagem para fresamento, portanto, as seguintes partes do ISO 14649 descrevem apenas os esquemas *EXPRESS-G* para texto geral (partes 1 e 10) e *EXPRESS-G* para este tipo de tecnologia (partes 11 e 111). Na Figura 2.9, é exposta uma estrutura onde se observa a relação entre as partes que compõem o modelo de dados da norma ISO14649 e, imediatamente após, é exibido o diagrama *EXPRESS-G* do mesmo na Figura 2.10.

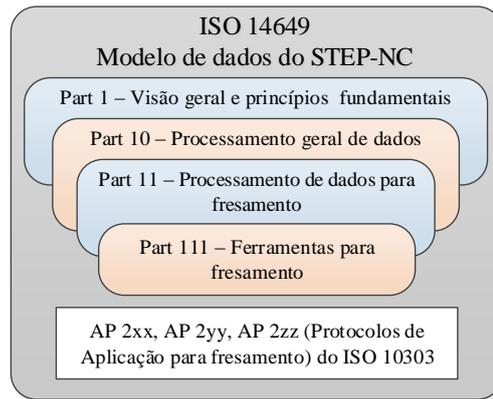


Figura 2.9: Estrutura do modelo de dados STEP-NC para fresamento (Adaptado de [27])

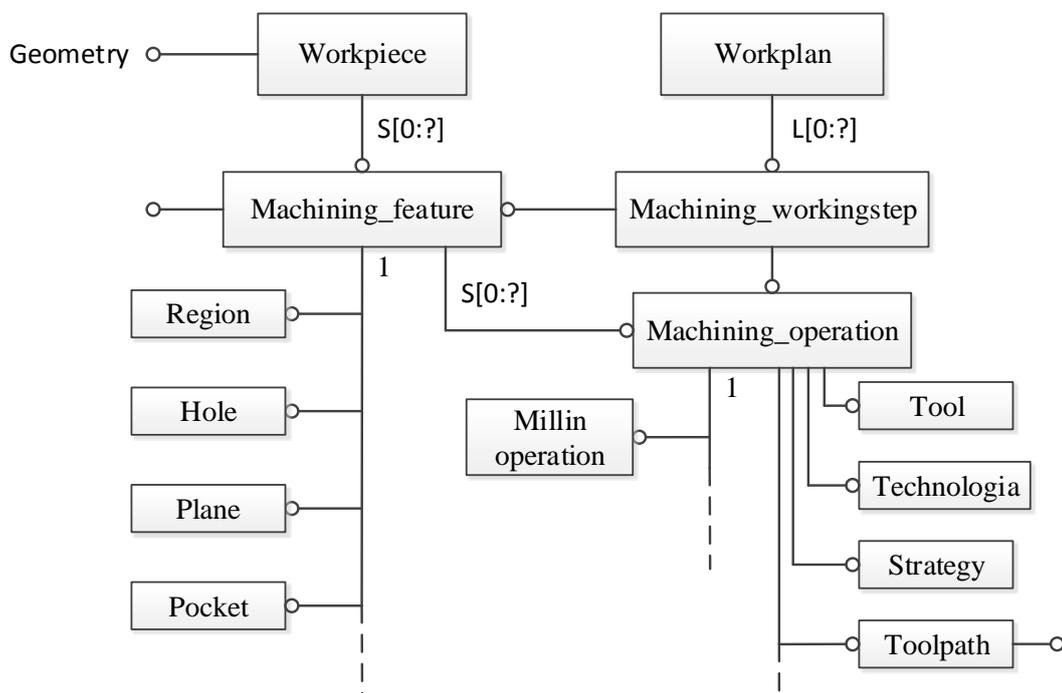


Figura 2.10: Estrutura do modelo de dados STEP-NC para fresado no diagrama EXPRESS-G (Adaptado de [7]).

Depois de identificar as partes do modelo de dados do STEP-NC para fresamento, como visto na Figura 2.10, segue-se com a descrição geral das mesmas.

#### i. Parte 1 – Visão geral e princípios fundamentais

Esta parte da norma ISO14649 fornece uma introdução e organização geral de um modelo de dados para Controladores Numéricos Computadorizados com base nos conceitos de dados do produto.

Um programa de uma peça é descrito em um formato de arquivo físico de acordo com a ISO 10303-21. A primeira seção do programa é parte da seção do cabeçalho, marcado pela palavra-chave "HEADER". Neste cabeçalho, algumas informações gerais e comentários sobre o programa da peça são colocados, como nome, autor, data, organização, etc.

A segunda e principal seção do arquivo do programa é a seção marcada pela palavra-chave "DATA". Essa contém todas as informações sobre geometria, funções e tarefas de fabricação. O conteúdo da seção de dados está dividido em três partes significativas:

- o *workplan* e *executables* (Plano de trabalho e executáveis) com sua descrição da tecnologia.
- *features* de usinagem.
- descrição de geometria.

As entidades do projeto servem como uma referência explícita para o ponto de partida das tarefas de fabricação. A Figura 2.11 mostra a relação entre estas partes significativas de um conjunto de dados da norma ISO14649.

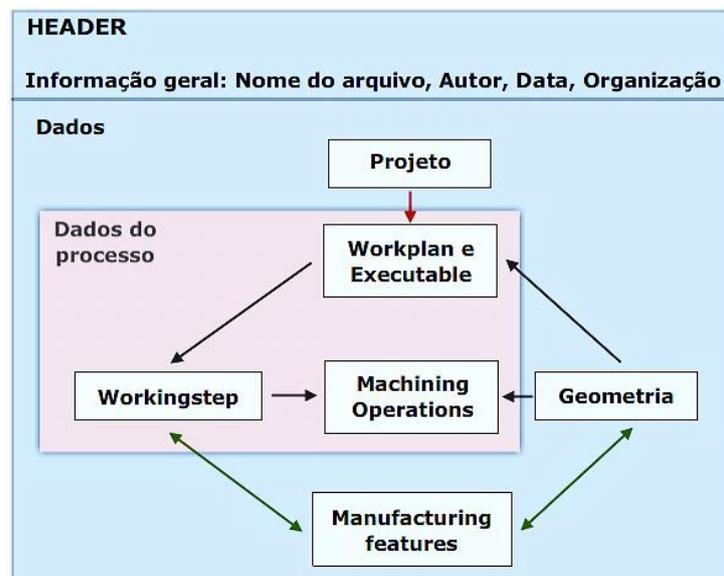


Figura 2.11: Estrutura de dados do ISO 14649 (Adaptado de [38]).

A estrutura e a finalidade que definem as características e os dados do processo estão descritos mais detalhadamente na parte 10.

## ii. Parte 10 – Processamento geral de dados

A norma Internacional ISO14649-10 foi preparada pelo Comitê Técnico ISO / TC 184, TC Sistemas de Automação Industrial e Integração, Subcomitê SC 1, SC Dispositivo de Controle Físico. Esta parte da norma ISO14649 especifica os dados do processo, que geralmente são necessários para a programação-NC dentro de todas as tecnologias de usinagem. Na programação-NC, o programa inclui informações geométricas e tecnológicas. Isso pode ser descrito utilizando esta parte (ISO14649-10) da norma, além de conter informações de tecnologia específica, por exemplo a norma ISO14649-11 para fresamento ou a norma ISO14649-12 para torneamento. Esta parte da norma ISO14649 fornece as estruturas de controle para a sequência de execução do programa-NC, principalmente a sequência de etapas de trabalho e funções associadas à máquina. Existe também o "*machining\_schema*" definido nesta parte da ISO e contém a definição de tipos de dados que são geralmente relevantes para diferentes tecnologias (por exemplo, fresamento, torneamento, retificação, entre outras). Além disso, a norma ISO14649-10 inclui a definição da peça (*workpiece*) assim como um catálogo contendo características que podem ser referenciadas por várias tecnologias, executáveis gerais e bases para uma definição de operação.

Como já foi mencionado, as tecnologias específicas estão definidas em partes separadas na ISO14649. A norma ISO14649-10 não pode ficar sozinha. Um programa-NC precisa da norma ISO14649-10 e deve ter pelo menos uma parte de tecnologia específica (por exemplo, ISO14649-11 para a fresamento, ISO14649-12 para torneamento, etc.).

A descrição dos dados de processo é feita utilizando a linguagem ou esquema EXPRESS conforme é definido na norma ISO10303-11. A codificação dos dados é feita usando-se a norma ISO10303-21 [36]. Em seguida, são descritas as partes que contêm as entidades (para ter uma concordância com os gráficos do digrama *EXPRESS-G*, as entidades conservam o seu nome em inglês) mais relevantes definidas na norma ISO14649, que são:

- **Project**

Quando se usa o modelo de dados da norma ISO14649 para escrever um programa de uma peça, esse programa deve (depois do *Header*, ver Figura 2.12) incluir uma entidade de nível superior (*top-level*) chamada “*project*”. Essa entidade indica outras duas importantes entidades. A primeira é o plano de trabalho ou “*workplan*” e a segunda é a peça de trabalho ou “*workpiece*”. A seguir, explica-se de forma geral essas entidades. A Figura 2.12 expõe a entidade *project* e seus atributos por meio do diagrama *EXPRESS-G*.

- **Entidades Executáveis**

As entidades executáveis são a base de todos os objetos executáveis. Há três tipos de objetos executáveis: o *workplan*, a *NC\_function* e, por último, o *workingstep*. Eles iniciam as ações em uma máquina e devem ser organizados segundo o modelo de dados. Na Figura 2.13 pode-se notar a ordem.

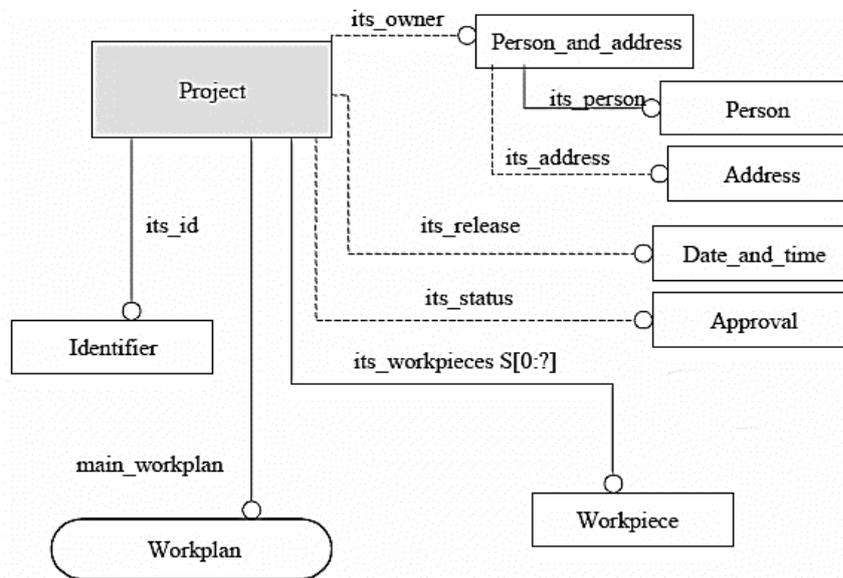


Figura 2.12: Diagrama *EXPRESS-G* da entidade *project* (Adaptado de [36]).

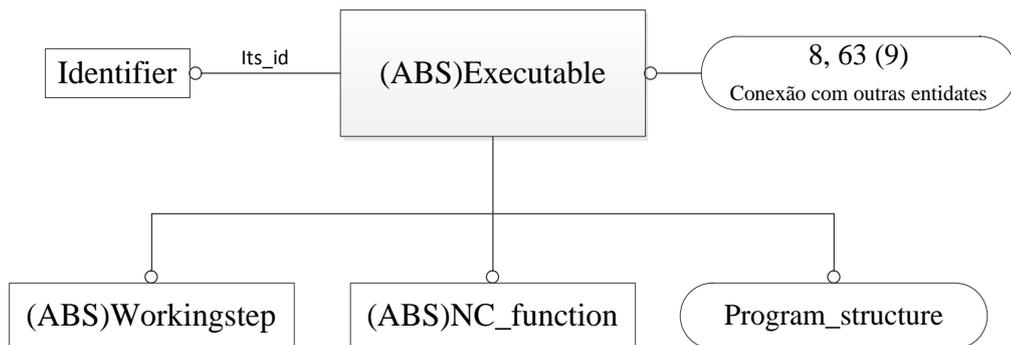


Figura 2.13: Diagrama EXPRESS-G, Executables (Adaptado de [36]).

#### ❖ Workplan

Na sequência lógica de um programa de uma peça, a entidade seguinte é o *workplan* (Figura 2.14). O *workplan* faz parte das entidades executáveis e pode conter outros *workplans*, um ou vários *workingsteps* e *NC\_functions*, todos dispostos em uma ordem linear.

A entidade que contém o *workplan* é a entidade *oplevel project*. A seguir mostra-se o diagrama EXPRESS-G do *workplan*.

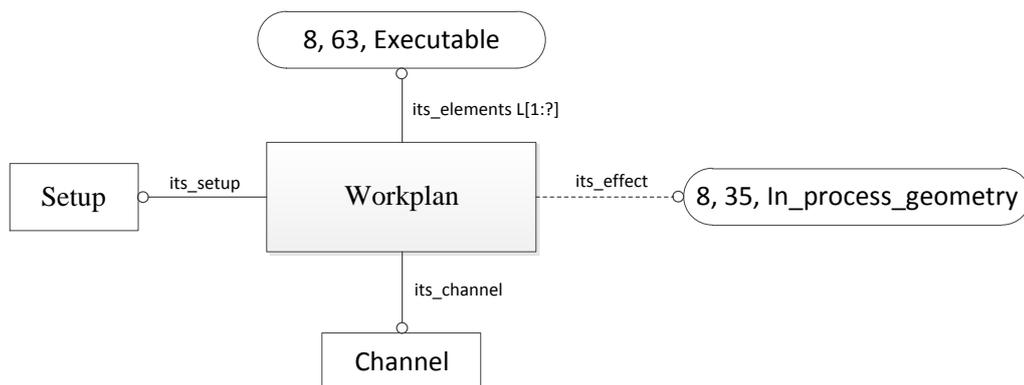


Figura 2.14: Diagrama EXPRESS-G do *workplan* (Adaptado de [36]).

Seguindo com os objetos executáveis, apresentam-se as entidades mostradas na Figura 2.13.

#### ❖ Workingstep

O *workingstep* é o coração de um programa NC. Nele estão contidas múltiplas ações, entre elas uma muito importante: o tipo de tecnologia (fresamento, torneamento) que descreve uma única operação de usinagem utilizando uma só ferramenta. Na Figura 2.15, pode-se observar esta entidade e suas ações.

Da entidade *workingstep*, a ação (que também é uma entidade) mais importante é a *machining\_workingstep*.

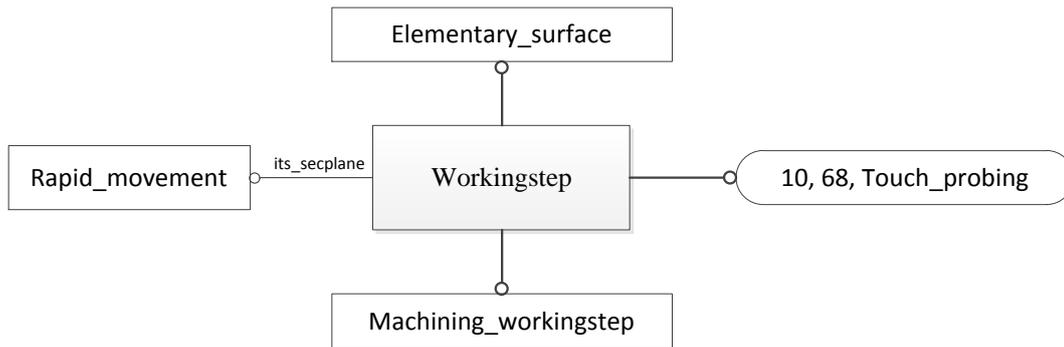


Figura 2.15: Diagrama EXPRESS-G do *workingstep* (Adaptado de [36]).

#### ❖ *Machining\_workingstep*

Esta entidade representa o processo de usinagem de uma determinada área em uma peça de trabalho. A *machining\_workingstep* não é independente de uma *feature* (por exemplo uma ranhura), mas especifica a associação entre a *feature* e a operação para usiná-la. Portanto, com esta entidade entende-se a diferença entre *feature* e operação. A Figura 2.16 apresenta a descrição básica da entidade *machining\_workingstep* no diagrama EXPRESS-G.

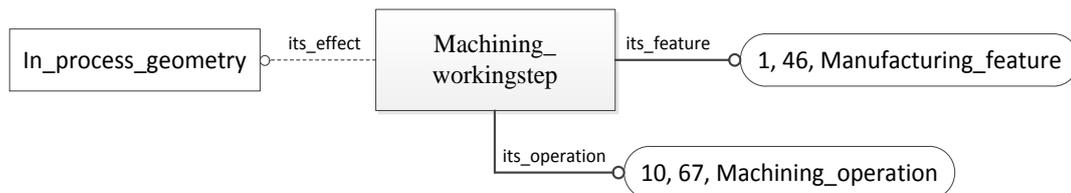


Figura 2.16: Diagrama EXPRESS-G do *machining\_workingstep* (Adaptado de [36]).

#### ❖ *Manufacturing feature*

Esta entidade é o supertipo (*oplevel*) de todas as *features* de manufatura. Quando se trata de manufatura 2½D, as *features* são do tipo *two5D\_manufacturing\_feature* e podem ser furos, cavidades, degraus, ranhuras, etc. Se for considerada a usinagem de forma livre, usam-se regiões (entidade tipo “*region*”). Nesta entidade, são descritas por exemplo as propriedades geométricas da peça, mas não contém nenhuma instrução de como usinar uma peça. Essas instruções de usinagem estão contidas nas operações.

#### ❖ *Operation*

A entidade operação (*operation*) é uma entidade tipo *oplevel* e especifica os dados genéricos exigidos do conteúdo de um *workingstep*. Na norma ISO14649-10 [36], são descritos três tipos de operações: a) operação de usinagem (*machining\_operation*); b) movimentação rápida (*rapid\_movement*), e c) prova de toque (*touch\_probing*), como mostrado na Figura 2.18. 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* [11].

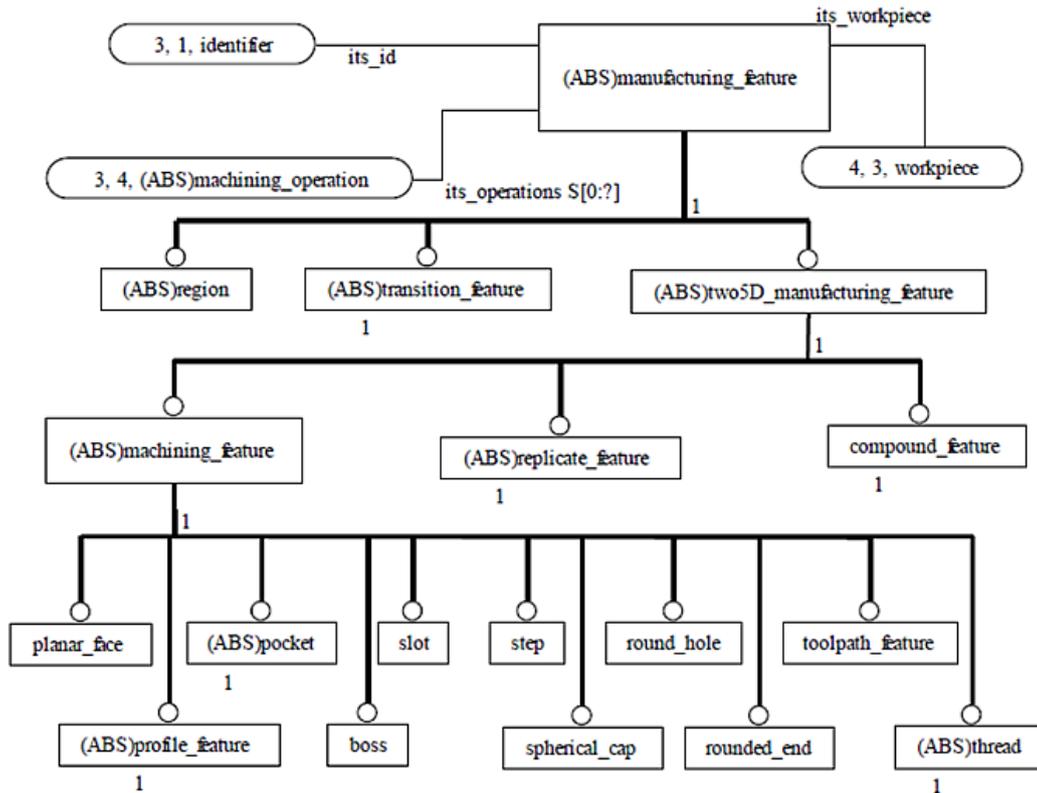


Figura 2.17: Diagrama EXPRESS-G do *manufacturing\_feature* [11].

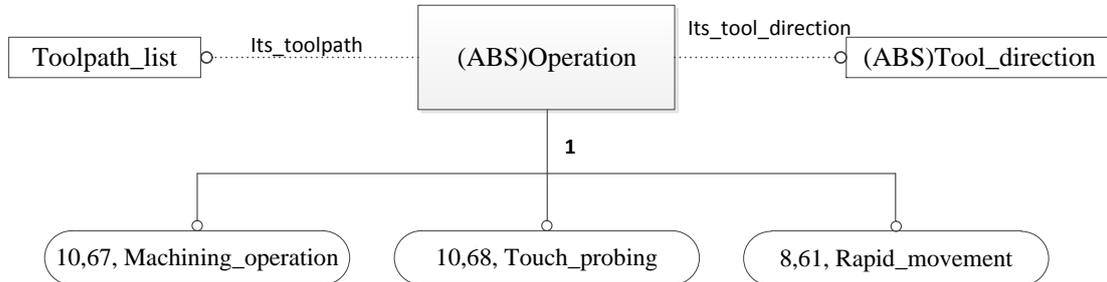


Figura 2.18: Diagrama EXPRESS-G do *operation* (Adaptado de [36]).

Uma operação pode estar associada a múltiplas ou a nenhuma *features*. Assim, uma mesma operação pode ser executada em diferentes lugares no código da peça. A finalidade disso é a reutilização dos dados de uma operação em diferentes *features*.

Todas as operações descritas na norma têm como opção a especificação das trajetórias da ferramenta de forma explícita, seja em dois ou três eixos. Para operações de usinagem, os controladores numéricos deveriam especificar os ciclos para gerar as suas próprias trajetórias.

Entretanto, para controladores numéricos antigos, ou que não são capazes de gerar as suas próprias trajetórias, ou que por razões tecnológicas requerem a execução de trajetórias exatamente determinadas, pode se usar o atributo *its\_toolpath*.

Neste texto só será descrita a entidade *machining\_operation* ou Operação de usinagem.

### ❖ *Machining\_operation*

A entidade *machining\_operation* define o método de usinagem para uma área específica da peça de trabalho. Nela fica explícita a ferramenta a ser utilizada e um conjunto de parâmetros tecnológicos para usinar a *feature* à que foi vinculada. Na Figura 2.19, apresenta-se esta entidade e seus atributos.

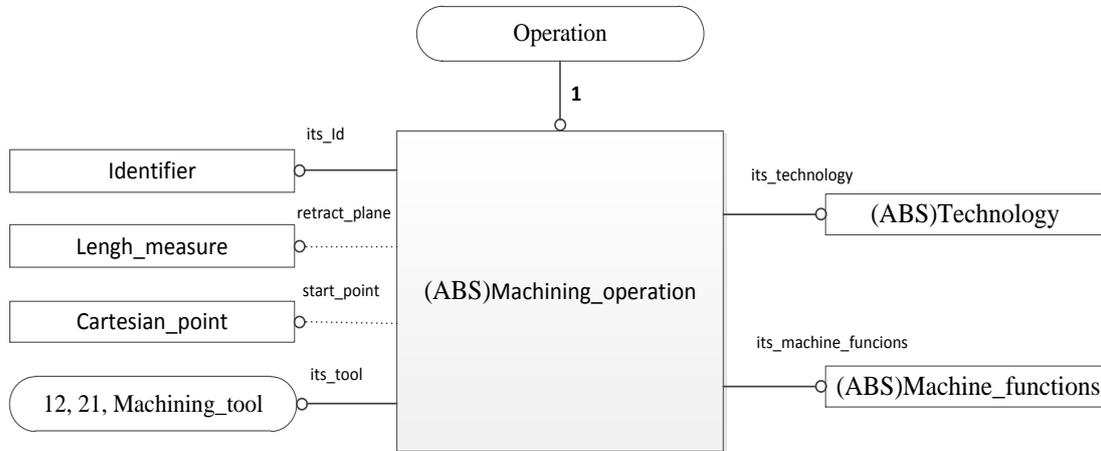


Figura 2.19: Diagrama EXPRESS-G do *machining\_operation* (Adaptado de [36]).

Até aqui foram descritas várias das entidades importantes da norma ISO 14649 parte 10. Por isso as operações de usinagem específicas para processos de fresamento são definidas na parte 11 da norma ISO14649.

### iii. Part 11 – Processamento de dados para fresamento

A parte 11 da norma ISO14649 especifica os elementos de dados específicos de tecnologia necessários para os processos de fresamento. Juntamente com os dados gerais da parte 10, descreve a interface entre um controlador numérico computadorizado e seu sistema de programação (sistema CAM ou sistema de programação do chão de fábrica) para fresamento [39]. Esta parte inclui apenas processos para fresamento e furação.

Também define o “*milling\_schema*”, o mesmo que define os tipos de dados que representam os processos de fresamento e furação [11]. Em seguida, apresentar-se-á algumas das entidades mais ressaltantes definidas nesta parte (para ter uma concordância com os gráficos no digrama EXPRESS-G, as entidades conservam o seu nome em inglês).

### ❖ *Milling machining operation*

Esta é a classe base para todas as operações descritas na norma ISO14649-11, sendo subtipo da entidade *machining\_operation* definida na parte 10. Divide-se em duas operações: fresamento (*milling*) e furação (*drilling*). A Figura 2.20 apresenta um diagrama EXPRESS-G desta entidade.

### ❖ *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 desses processos. Em geral, há dois tipos de operações de usinagem: desbaste e acabamento. As operações de desbaste removem mediante camadas todo o

material da peça dependendo do tipo de *feature* a ser usinada, deixando por último uma camada (*allowance*) para fazer o acabamento. A operação de acabamento remove essa camada (*allowance*) até a superfície final da *feature* em uma ou várias passadas. A Figura 2.21 apresenta esta entidade com seus atributos e subtipos (operações de forma livre e operações em 2½D).

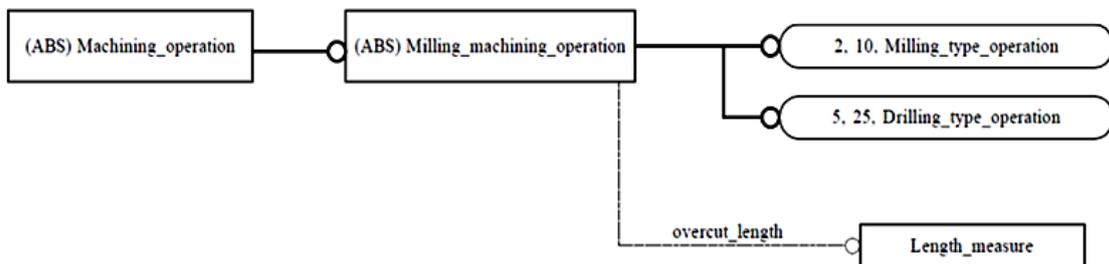


Figura 2.20: Diagrama EXPRESS-G do *milling\_machining\_operation* (Adaptado de [39]).

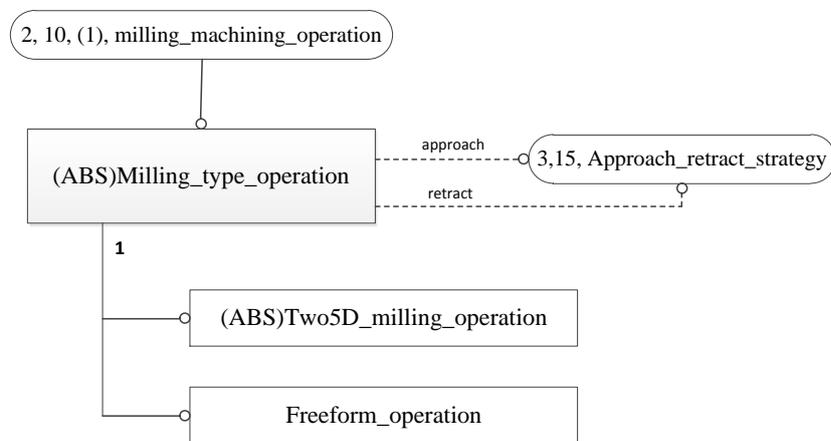


Figura 2.21: Diagrama EXPRESS-G do *milling\_type\_operation* (Adaptado de [39]).

### ❖ *Drilling type operation*

Esta é a classe base para todas as operações pertinentes a furação, alargamento e *sinking*. Esta entidade 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 especificar o diâmetro prévio (atributo *previus\_diameter*) [11]. A Figura 2.22 apresenta um diagrama desta entidade, onde também pode ser visto que ela possui cinco subtipos.

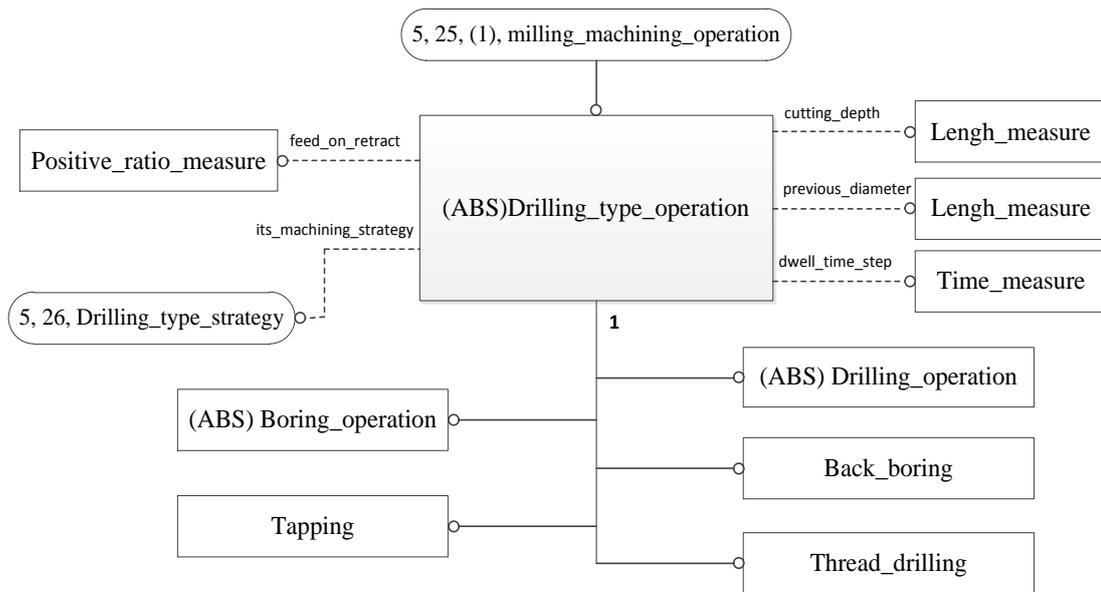


Figura 2.22: Diagrama EXPRESS-G do *drilling\_tipe\_operation* (Adaptado de [39]).

#### ❖ *Milling technology*

Esta entidade define os parâmetros tecnológicos das operações de fresamento e de furação como a velocidade de corte, a velocidade de rotação, o avanço, etc. A Figura 2.23 apresenta esta entidade e seus atributos em um diagrama EXPRESS-G.

#### ❖ *Milling machine functions*

A entidade *milling\_machine\_funcions* descreve o estado de várias funções da máquina como a refrigeração, a remoção de cavacos, etc. Estas funções são aplicadas durante o tempo de uma operação. Esta entidade é subtipo da entidade *technology* definida no ISO4649-10 [36]. A Figura 2.24 apresenta um diagrama desta entidade e seus atributos.

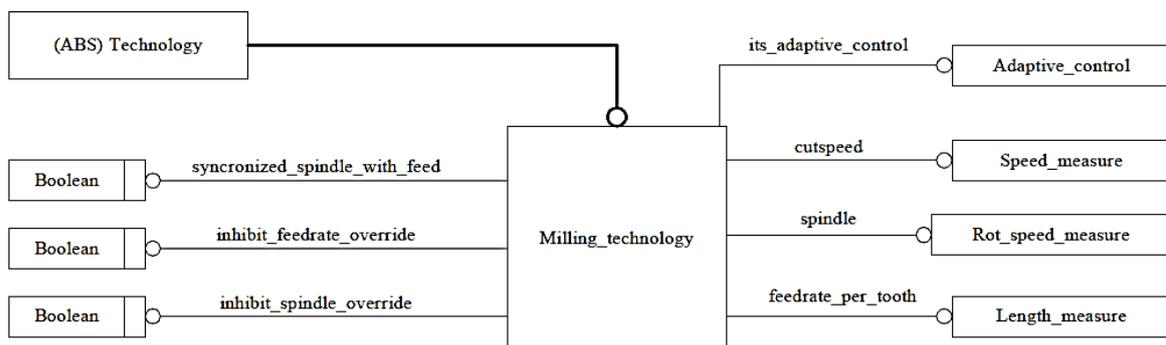


Figura 2.23: Diagrama EXPRESS-G do *milling technology* [39].

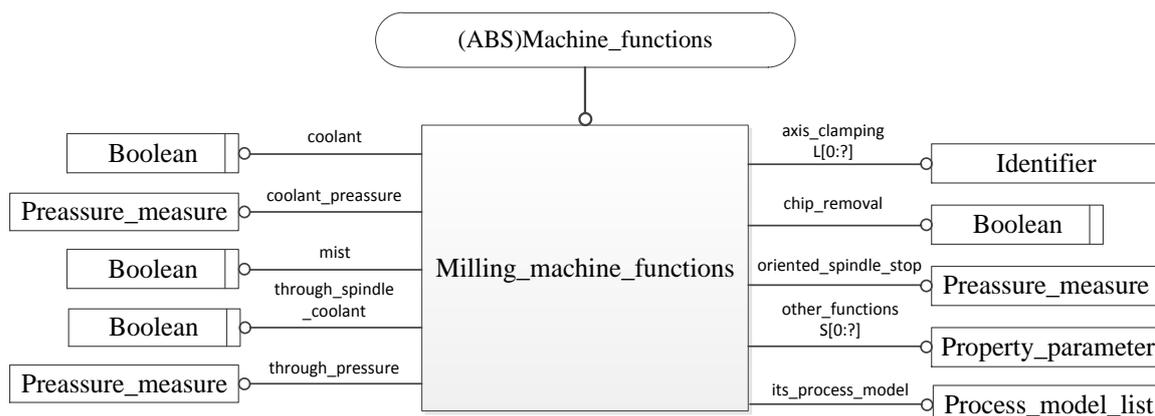


Figura 2.24: Diagrama EXPRESS-G do *milling machine functions* (Adaptado de [39]).

A seguir são descritas algumas das ferramentas usadas em usinagem tanto para fresamento como para furação, que está incluído na norma ISO14649 parte 111.

#### iv. Parte 111 – Ferramentas para fresamento

Esta parte especifica os elementos de dados necessários para representar as ferramentas para fresamento, trabalhando em conjunto com a parte 11 do mesmo ISO14649. Estes elementos de dados podem ser utilizados como critério para selecionar uma ou várias ferramentas para fresamento ou para furação [40].

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 [40].

Em relação ao que foi dito e como foi descrito na norma ISO6983 (secção 2.3.1), a ferramenta é definida apenas com um número (por exemplo, T1) e nenhuma informação adicional relacionada ao tipo ou à geometria da ferramenta é dada [11].

Os dados das ferramentas que estão dentro da entidade *milling\_tool\_schema* são: tipo, geometria e a vida esperada da ferramenta. A seguir são apresentadas as principais entidades definidas neste esquema.

##### ❖ Ferramenta de corte para fresamento (*Cutting tools for milling*)

Esta entidade descreve a informação referente à tecnologia necessária para descrever ferramentas de corte para operações de fresamento, incluindo tipos de ferramentas como alargadores, brocas, fresas, etc.

É subtipo da entidade genérica *cutting\_tool* definida na parte 10 da norma ISO 14649 [40]. A Figura 2.25 apresenta o diagrama desta entidade e seus atributos.

Como pode ser observado na Figura 2.25, a entidade *Milling cutting tool* herda os atributos da sua classe pai, a entidade *cutting tool*.

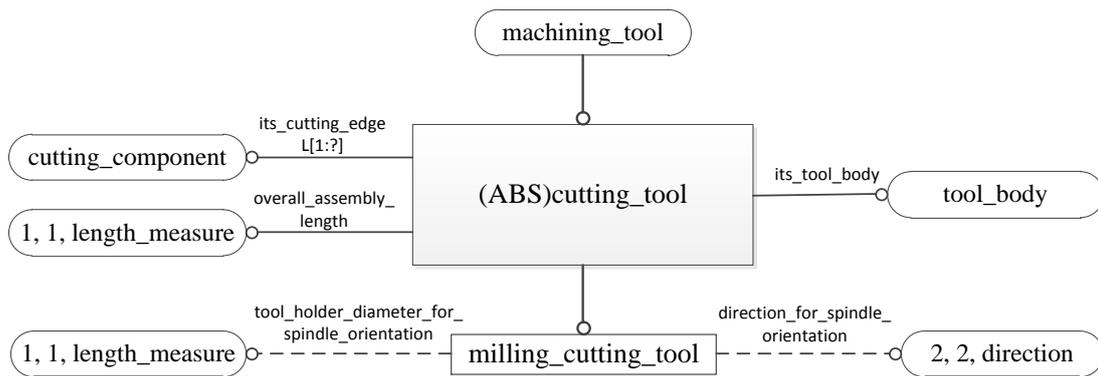


Figura 2.25: Diagrama EXPRESS-G do cutting tool for milling (Adaptado de [11]).

### ❖ Milling tool body

Esta é a classe base para todos os tipos de corpos de ferramenta usados em processos de fresamento sendo subtipo da entidade *tool body* definida na parte 10. A Figura 2.27 mostra um diagrama desta entidade com seus atributos e seus subtipos mais a *tool body*.

### ❖ Tool dimension

Esta entidade descreve as dimensões do corpo da ferramenta. A Figura 2.26 descreve uma ferramenta genérica, uma fresa esférica e uma de perfil arredondado, e a Figura 2.28 apresenta um diagrama em EXPRESS-G desta entidade e seus atributos.

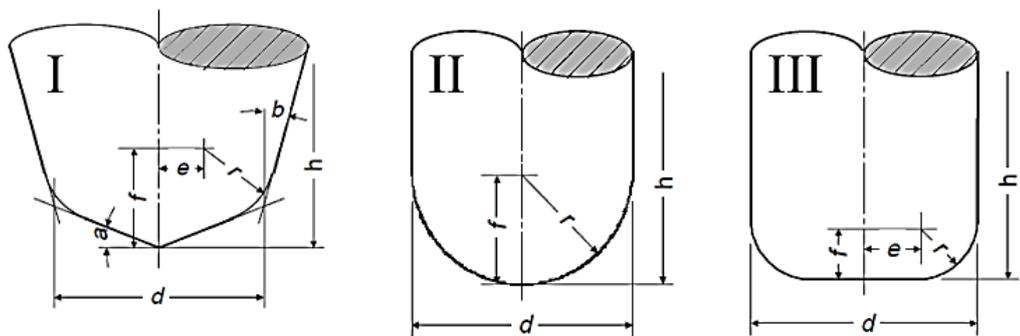


Figura 2.26: Diferentes tipos de ferramentas e suas Dimensões (Adaptado de [40]).

Da figura 2.26 se tem as seguintes descrições:

- I. Ferramenta genérica
- II. Fresa esférica
- III. Fresa de perfil arredondado

Onde:

- |  |                                      |
|--|--------------------------------------|
| $a$ : ângulo da ponta da ferramenta          | $f$ : centro da ponta vertical       |
| $b$ : ângulo de circunferência da ferramenta | $h$ : comprimento da aresta de corte |
| $d$ : diâmetro                               | $r$ : raio de quina                  |
| $e$ : centro da borda horizontal             |                                      |

Todas as dimensões aparecem como atributos da entidade *tool dimension*, como observado na Figura 2.28.

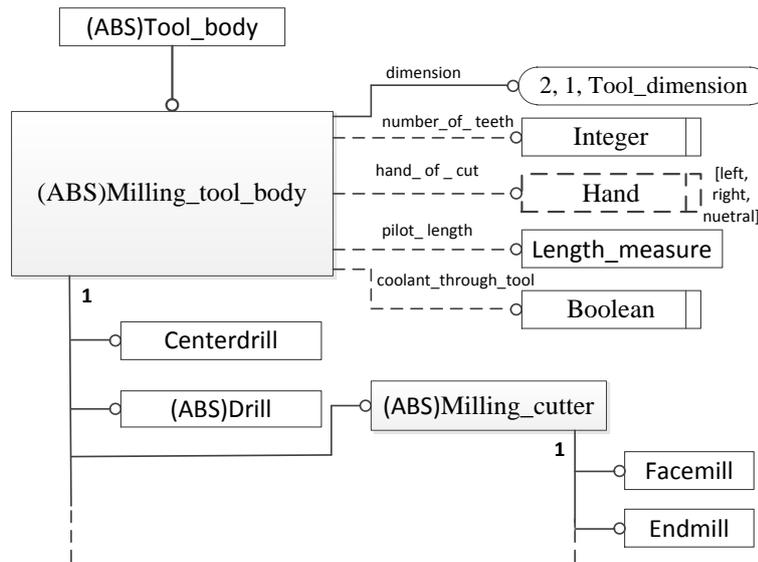


Figura 2.27: Diagrama EXPRESS-G do *Milling tool body* (Adaptado de [40]).

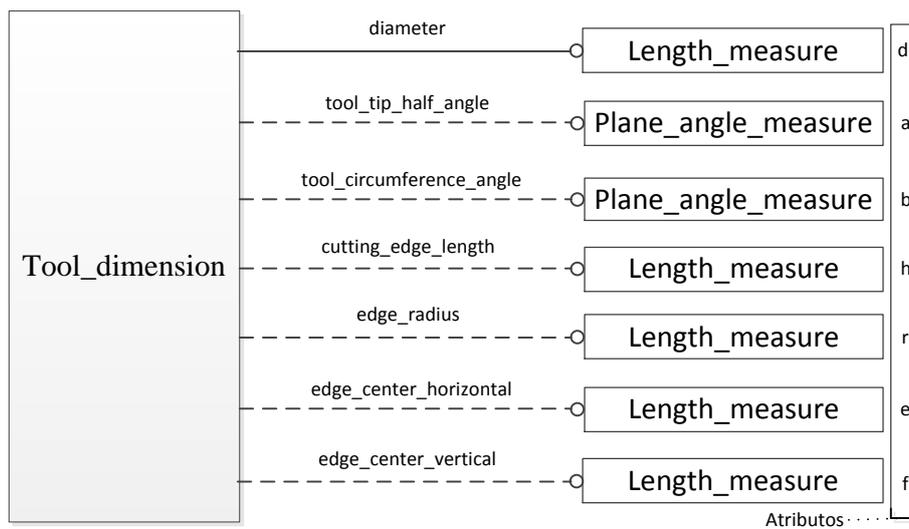


Figura 2.28: Diagrama EXPRESS-G do *Tool dimension* (Adaptado de [40]).

Depois de revisar as normas relacionadas com o STEP-NC, estudar-se-á sobre alguns dos projetos mais importantes que existem até o momento no contexto da norma ISO14649.

## 2.7. PROJETOS RELACIONADOS À NORMA STEP-NC

Existem vários projetos que constroem, desenvolvem e trabalham com a norma ISO14649. Alguns deles continuam atualmente ativos em conjunto com o desenvolvimento de outras normas. A seguir são apresentados alguns desses projetos.

### **2.7.1. Projeto STEP-NC MACHINE**

Este projeto é implementado a partir do antigo projeto *SuperModel* da empresa de *software* STEP Tools [41]. O resultado final é o software STEP MACHINE que tem sido alvo de provas e melhoras contínuas desde sua concepção até hoje. Deve-se frisar que esse software, desde o ano 2000, tem sido utilizado como banco de ensaio STEP-NC em uma série contínua de demonstrações de tecnologia para verificar o padrão ISO10303, as implementações da unidade de comandos de máquina STEP-NC e dar à indústria da manufatura experiência com a norma STEP-NC [42]. Atualmente a versão disponível do STEP-NC MACHINE é a 9.52.

Em junho de 2003 a STEP Tools disponibilizou um *plugin* para o software MASTERCAM. Esse *plugin* é baseado no software STEP-NC MACHINE [43]. Esse software possui várias versões. Ressalta-se que na segunda versão foram construídas as primeiras interfaces de controle baseado em STEP-NC para máquinas CNC *Fanuc* e *Siemens* que originalmente usavam código G. Os resultados obtidos foram mostrados no primeiro fórum de teste industrial em 2005. Desde 2010, esse software está na oitava versão (STEP-NC MACHINE v8) [42].

### **2.7.2. Projeto Internacional IMS STEP-NC.**

O projeto Sistema de Manufatura Inteligente (*Intelligent Manufacturing Systems*) para STEP-NC foi iniciado no verão de 2001, sendo coordenadas as diferentes atividades internacionais entre EU, Suíça, USA e Coreia [44].

O projeto IMS STEP-NC começou como uma extensão do projeto *ESPRIT* STEP-NC tendo como objetivo principal a ampliação das tecnologias de STEP-NC, a interoperabilidade entre o chão de fábrica e o desenho, o fortalecimento de funções e ferramentas e a difusão do novo modelo de dados no mundo todo.

As novas tecnologias desenvolvidas são entendidas como novas cadeias de processos entre sistemas CAD e máquinas de medição. Por outro lado, foi ampliado o modelo de dados existente para fresamento [45]. O projeto atualmente continua em construção e desenvolvimento.

### **2.7.3. Projeto ISO14649 Toolkit**

Este projeto foi iniciado em dezembro de 2008 e concluído em fevereiro de 2009 por Mark Pictor e Thomas Kramer, ambos programadores de software de manufatura aberta [46]. Eles criaram um conjunto de ferramentas (*toolkit*) para STEP-NC baseado nos códigos: *sebnf2pars* [47]. O *sebnf2pars* é uma ferramenta para a construção de analisadores de arquivos de STEP Parte 21 e do código do NIST [48] da norma ISO14649 partes 10, 11 e 111.

Entre as ferramentas inclusas no software do *toolkit*, existe uma que permite analisar código no formato STEP Parte 21 ou “.p21” para saber se a sintaxe está correta. Também há uma ferramenta para interpretar e imprimir código NC para usinagem de 3 eixos.

O *toolkit* está programado em linguagem C++ e admite a modificação do seu código fonte para acoplá-lo às necessidades de qualquer outro projeto com relação a controladores NC, como por exemplo o EMC2.

Na revisão deste projeto são identificadas algumas ferramentas que marcam a base para o desenvolvimento da arquitetura do adaptador de STEP-NC a código G/M do presente projeto. É possível usar como exemplo a estrutura do programa (*toolkit*) e a forma de identificação dos dados inclusos no arquivo físico p21. Isto com o fim de fazer o modelado funcional do adaptador.

#### 2.7.4. Projeto STEP Modeler UFSC/GRIMA

Este projeto de doutorado foi desenvolvido Na Universidade Federal de Santa Catarina (UFSC) em conjunto com Grupo de Integração da manufatura (GRIMA) com o nome: “UM SISTEMA PARA O PROJETO E FABRICAÇÃO DE PEÇAS MECÂNICAS A DISTÂNCIA VIA INTERNET ADERENTE À NORMA ISO 14649 (STEP-NC)” [11], o resultado final foi o software chamado “STEP Modeler”.

O STEP Modeler aproveita a tecnologia da internet introduzindo a *e-manufatura*. A *e-manufatura* consiste em um fluxo de dados compatíveis e sem brechas na cadeia CAD-CAPP-CAM-CNC [11]. Este software usa o padrão ISO14649 voltado para fresamento (partes 10, 11 e 111). Apesar do padrão ISO14649 conter muitas informações para usinagem de peças prismáticas e outras, só foram disponibilizadas algumas das *features* descritas na norma ISO14649 parte 11.

Com o STEP Modeler é possível visualizar uma peça prismática junto com *features* adicionadas, isso com o fim de ter uma ideia da peça acabada além de se verificar se a peça criada é válida desde o ponto de vista de fabricabilidade com as ferramentas disponíveis. Por último o sistema gera automaticamente o arquivo físico no formato p21.

Concluído parcialmente em dezembro de 2011, atualmente apresentam-se modificações no seu código, melhorando-se constantemente com relação à geração do código no formato físico STEP parte 21 ou “.p21” usando a norma ISO14649 de 2002.

É importante ressaltar o funcionamento deste software, pois será usado neste projeto para gerar o arquivo físico de uma peça CAD projetada, exportando-o no formato de saída “STP” ou “p21” (padrão ISO 10303 parte 21). O arquivo de saída será utilizado como entrada no EMC (LinuxCNC) que irá controlar a Router CNC. Desta forma, verifica-se a operabilidade e a funcionalidade do CNC na aderência à norma ISO14649 mediante estudo de casos, além de validar o trabalho da UFSC associado ao gerador de código STEP-NC. Adverte-se também que o STEP Modeler só trabalha com as *features* de usinagem dispostas na Tabela 2.1 onde são mostradas as características e operações delas.

Tabela 2.1: *Features* de usinagem, condições e operações usadas no software STEP Modeler.

| <i>Feature</i> de usinagem                    | Condições                      | Operações   |
|---|--------------------------------|---|
| <i>Pocket - Bottom_and_side_rough_milling</i> | <i>Pocket_button_condition</i> | Ferramenta,<br>Velocidade de<br>avanço, parâmetros<br>tecnológicos da<br>máquina. |
| <i>Slot - Bottom_and_side_rough_milling</i>   | <i>Open_slot_end_type</i>      |   |
| <i>Step - Bottom_and_side_rough_milling</i>   | <i>Open_boundary</i>           |   |
| <i>Round Hole – Drilling, Center Drilling</i> | <i>Hole_button_condition</i>   |   |

### **2.7.5. Projeto Europeu *ESPRIT* STEP-NC**

Este projeto começou em janeiro de 1999 e terminou em dezembro de 2001. Foi desenvolvido por vinte parceiros industriais e acadêmicos com experiência nas áreas de CAD/CAM (*Dassault, Open Mind*), controle (*Siemens, Osai*), máquinas-ferramentas (CMS) e usuários finais (*Daimler Chrysler, Volvo*) na Europa formando o consórcio europeu STEP-NC [49].

O protótipo para a fresamento foi demonstrado pelo consórcio em outubro de 2001 em Aachen, Alemanha. Este protótipo foi baseado em sistemas comerciais da *Siemens, Open Mind* e Sistemas *Dassault*. Os módulos CAM *Open Mind (HyperFact)* e *Dassault (Sistema Catia V5)* geram os programas STEP-NC, incluindo um alto nível de dados de geometria CAD, dados e informações de operação de sequenciamento. Este arquivo de programa foi, então, passado para o chão de fábrica, onde um controlador Siemens 840D *Sinumerik*, usando *ShopMill*, foi adaptado para processá-los.

O trabalho de pesquisa sobre STEP-NC EDM é baseado em uma máquina-ferramenta *Agie Charmilles* em *CADCAMation* que formalizou um cenário orientado STEP-NC para a linha EDM. Osai e CMS coletivamente criaram uma saída CAM para corte de contorno usando as informações STEP-NC em uma máquina-ferramenta controlada por *Osai* para usinagem de madeira.

Em 2002, o projeto Europeu STEP-NC mudou-se para sua segunda fase. Além de reforçar ainda mais os sistemas previstos e desenvolvidos na primeira fase, a segunda fase concentrou-se nos mecanismos de realimentação da STEP-NC [50]. O objetivo da segunda fase foi avaliar se a usinagem é adequada e devidamente executada no chão de fábrica. Para fazer isso, os dados de controle foram recolhidos e foi definido um modelo de dados para manipulá-los dentro de um programa STEP-NC. Também está sendo investigado, paralelamente, um cenário para a verificação do modelo de dados e a avaliação de seus benefícios.

### **2.7.6. Projeto Suíte de Manufatura STEP (SMS)**

A SCRA (Carolina do Sul *Research Authority*), sob o patrocínio do Centro Nacional Automotivo do Exército dos EUA *Tank-Automotive and Armaments Command (TACOM)*, coordenou o projeto SMS que define a Suíte STEP como um conjunto de Protocolos de Aplicação e uma arquitetura de implementação, isso habilitou a produção de STEP em duas aplicações: as *comerciais* e de defesa [51].

Estes conceitos foram obtidos a partir da entrada de muitos países e empresas que participam nos grupos de trabalho do TC184 da ISO SC4, como parte da comunidade STEP. Esses países incluem Canadá, Alemanha, Suécia, Japão, Reino Unido e EUA. As empresas são lideradas por Boeing, General Dynamics Land Systems, Ltd. LSC, SCRA e STEP Tools Inc [49].

### **2.7.7. Projeto para Rápida Aquisição de Peças Manufaturadas (Project RAMP)**

O Projeto RAMP existe desde 1986, abordando padrões de aplicações orientadas para a fabricação de peças mecânicas e elétricas e a sua montagem. A maior conquista deste projeto foi o desenvolvimento e a implementação da etapa padrão no AP224 como parte do Programa RAMP. Este programa foi inicialmente financiado pelo Comando Naval de Sistemas de Abastecimento (NSNC) e mais tarde pela Agência de Defesa logística (DLA). TACOM NAC é o patrocinador atual da tecnologia, tendo um contrato com *DLA's Strategic Technologies Sourcing*.

O Programa RAMP do Reino Unido é uma implementação da tecnologia RAMP no UK, financiado pelo Ministério da Defesa Britânico. Este programa está em vigor desde 1998 e em uso na indústria [52].

### 2.7.8. Projeto de Manufatura inteligente para Usinagem e Inspeção compatível com STEP-NC

Este projeto nascido no Reino Unido, é financiado pelo *Engineering & Physical Sciences Research Council* (EPSRC) e começou em maio de 2004, sendo liderado pelo Centro de AMST da Universidade de *Loughborough* [49]. Este trabalho tem como objetivo ampliar a investigação da UE com o CNC *Siemens 840D* no processo de medição na máquina CNC e também explorar a aplicação e integração da etapa de padrões compatíveis com NC entre a máquina CNC e uma Máquina de Medição por Coordenadas (CMM).

Um tema importante ainda subjacente da pesquisa é a aplicação da inteligência artificial ao processo CAPP/CAM/CNC com a aplicação da tecnologia de agentes através do uso de extração de dados. O projeto é apoiado por uma série de parceiros industriais, especificamente: Renishaw, Siemens, Grupo LSC, Delcam e Rolls Royce.

Alguns dos projetos anteriores podem ser localizados na Figura 2.29, que mostra o mapa mundial dos principais contribuintes do STEP-NC.

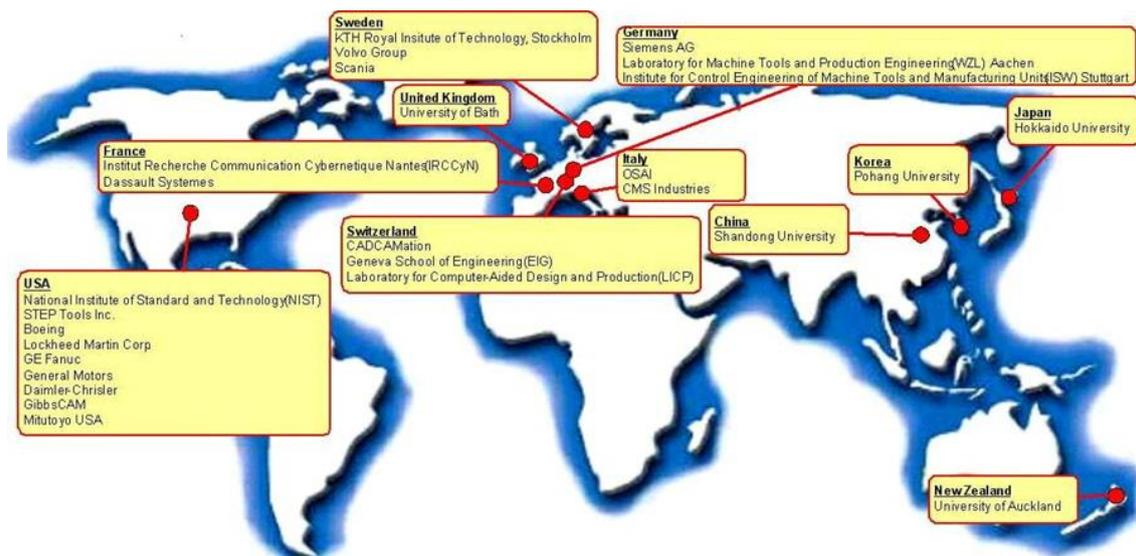


Figura 2.29: Os principais contribuintes para a abordagem do STEP-NC [53]

## 2.8. STEP-NC E FUNCTION BLOCKS

Os padrões STEP-NC (ISO14649) e *Function Block* (IEC61499) são usados em conjunto para permitir a integração da manufatura. Por isso, a junção desses dois padrões tem sido objeto de estudo de pesquisadores. Neste texto são apresentadas apenas três abordagens relacionadas ao padrão ISO14649 e à norma IEC61499. Essas três abordagens são escolhidas por sua relação com o tema tratado neste trabalho de mestrado.

### 2.8.1. Blocos de função para processo de programação de acesso a dados

Este trabalho foi desenvolvido por [54] e trata sobre o uso de blocos de função (*Function Blocks* ou FB) para implementar a lógica de acesso a dados em programas controladores. A implementação deste trabalho resulta em um protótipo para ambientes CAD/CAM/CNC para automatizar os processos de fabricação de acesso a dados e de gravação. Isso visa apoiar os serviços de fiscalização à atividade de produção, dados de fabricação, rastreabilidade e gravação de informações relevantes dos processos de fabricação para futura revisão em caso de falha, etc.

Este projeto foi aplicado no campo das máquinas CNC onde um novo conjunto de *nc-function* de rastreabilidade (*traceability*) foram discutidas para serem incluídas em futuras edições do padrão STEP-NC, a fim de automatizar as atividades de acesso a dados e monitoramento.

Com monitoramento e rastreabilidade em *nc\_functions* inseridas em um programa de peça CAM (ou um programa de peça em STEP-NC), as ações de rastreabilidade e de monitoramento podem ser programadas e executadas automaticamente pelo controlador NC. É esperado que, como resultado, exista um vínculo entre um registro entre dados guardados e as *features* de usinagem correspondentes ao programa de peça em STEP-NC (AP238).

Este trabalho permite ver a necessidade da interoperabilidade entre vários sistemas CAx e CNC, assim como uma ideia de monitoramento e estudo de falhas do CNC, o que poderá ser incluído no padrão STEP-NC. Desse modo, neste projeto é usada a ideia de monitoramento do CNC apenas na modelagem funcional do CNC aderente a STEP-NC, mesmo sem ser implementada. Ou seja, houve apenas uma idealização.

### 2.8.2. O Protótipo STEPNCMillUoA

O *STEPNCMillUoA* é o protótipo de um novo sistema CNC que utiliza o modelo dados descrito no padrão STEP-NC e a norma IEC61499 (blocos de função ou FB). O STEP-NC fornece um modelo de dados de alto nível e permite a usinagem baseado em *features* enquanto que a arquitetura de blocos de função é usada para simplificar o desenho de um CNC [55].

Segundo MUNHATA et al. [55], recentemente, há surgido uma série de pesquisadores que consideram o uso combinado destes padrões. A pesquisa teve início na década de 1990, quando os pesquisadores estavam tentando aumentar a adaptabilidade, flexibilidade e interoperabilidade para operações de usinagem dos controladores NC.

Segundo MINHATA et al. [56], é possível usar a norma IEC61499 como arquitetura de referência, usando-a eficazmente para o desenvolvimento de uma nova geração de controlador CNC com características de interoperabilidade, portabilidade, configuração e distribuição.

O sistema do *STEPNCMillUoA* está composto basicamente de quatro partes. A primeira é a geração de um arquivo físico p21 contendo informações como ferramentas, *features* de usinagem, estratégias e parâmetros tecnológicos. A segunda é a criação de um tradutor de STEP-NC a FB que toma as informações contidas no programa de peça em STEP-NC e as transforma em blocos de funções. Isso é realizado usando uma arquitetura de capas responsáveis pelo processamento de dados, armazenamento e execução do tradutor. A terceira parte é a criação de um protótipo de CNC com aderência a FB chamado CNC-FB. A última parte é um controlador STEP-NC encarregado de gerar

trajetórias de ferramenta (*toolpath*) baseado nas FB criadas na segunda parte. Por fim, este protótipo foi testado por meio de simulação e usinagem real.

Os resultados obtidos permitiram inferir que a arquitetura em camadas de FB prevê flexibilidade na adoção do controlador para novas máquinas e equipamentos com relação à plataforma hardware/software. Neste projeto também apresentou-se uma metodologia para facilitar a transição de um sistema centralizado para um sistema distribuído. O controlador STEP-NC fazendo uso da IEC61499 tira o máximo proveito para a integração da manufatura de acordo aos sistemas CAD/CAM e CNC.

A arquitetura, a estrutura e a metodologia usadas no sistema supracitado são tomadas como referência para o desenvolvimento deste trabalho de mestrado em quanto a modelagem funcional do CNC aderente a norma STEP-NC.

### **2.8.3. CNC aderente as normas STEP-NC e FB**

Este projeto é centrado no uso das normas ISO14649 e IEC61499 de maneira integrada de modo a obter um CNC flexível, adaptável, escalável e interoperável. O modelo de dados fornecido pelo padrão STEP-NC é usado para a transferência/compartilhamento de dados entre sistemas CAD/CAM e CNC. A norma IEC61499 apresenta uma arquitetura padronizada para controle de processo de automação baseada em blocos de funções. Essas normas foram usadas para a concepção de um controlador NC para uma máquina-ferramenta protótipo servoacionada, mas o alvo deste trabalho foi o desenvolvimento do software responsável pelo controle do servoacionamento. Para validação do funcionamento, foi realizada a usinagem de uma peça teste com várias *features* [57].

O sistema usa como entrada um arquivo STEP-NC em formato ISO10303-21 e por meio de um conversor é gerado como código alvo, arquivos aderentes à norma IEC61499 no formato XML, contendo as informações STEP-NC e toda a estratégia de controle da máquina-ferramenta representada em redes de FBs. Para validação da proposta de integração das normas, foi desenvolvido um controlador numérico para uma fresadora de 3 eixos para testes dos programas construídos.

Este projeto possui limitações tanto hardware quanto software, no entanto foi usinada uma peça teste com algumas das *features* disponíveis na norma STEP-NC. O trabalho mostrou um sistema CNC capaz de sintetizar a estrutura de alto nível da STEP-NC em instruções de baixo nível dentro do controlador para o acionamento dos servomotores, o que permite ao CNC utilizar arquivos STEPNC sem a necessidade de linguagens intermediárias [57].

## **2.9. CONVERSORES RELACIONADOS COM O PADRÃO STEP-NC E CÓDIGO G**

Desde que o STEP-NC começou a ser desenvolvido a fim de substituir o antigo Código G, várias instituições e pessoas levaram a iniciativa para desenvolver um conversor entre esses formatos.

Entre esses conversores, encontra-se o projeto anteriormente mencionado na sessão 2.7.3, sendo diretamente um conversor do tipo intérprete.

Também existem algumas patentes que descrevem tanto os métodos relacionados à conversão de um formato a outro quanto o planejamento de processos. A seguir, são apresentadas essas patentes.

### 2.9.1. Patente 1 No US 8,041,445 B2: Método de transformação de Código G num programa de peça em STEP-NC [58]

Esta patente refere-se a um método de transformação de *G-CODE* (código G) em um programa de peça em STEP-NC. Especificamente um método para permitir que um programa de peça em Código G possa ser usado num controlador STEP-NC sem gerar erros na criação automática de um programa de peça em STEP-NC.

Este método é composto das informações de operação de usinagem, informação das *features* de usinagem, informação da estratégia de usinagem e outros similares. Por último, é feito um processo de análise do código G e da informação da ferramenta.

No desenvolvimento da invenção desta patente, apresenta-se o típico problema do sistema do código G que tem dificultado o intercâmbio, acarretando a perda de informações com outros sistemas como CAD, CAPP e CAM. Assim a patente apresenta uma solução usando o Código G para gerar um programa de peça em STEP-NC, fazendo que este último seja mais aceitado na indústria e se expanda junto aos futuros controladores STEP-NC. Para atingir o objetivo deste método, é necessário seguir estes passos:

- Receber um programa da peça em Código G, informação da ferramenta e informação de um controlador numérico (NC) capaz de analisar o correspondente Código G.
- Analisar o programa da peça em Código G de acordo com o tipo do NC e criar blocos de informação do código.
- Separar os blocos de informação do Código G, agrupando os blocos em recursos para criar uma base para saber se há um ou vários *workingsteps*.
- Criar a informação da ferramenta, informação das funções da máquina e informação da tecnologia de usinagem para cada um dos *workingsteps*.
- Criar a informação interna dos *machining\_workingstep* e *machining\_operations*, detalhando o desbaste e o acabamento da *feature*.
- Criar as informações das estratégias de usinagem e de ferramenta.

Depois de seguir todos esses passos, é possível organizar os *workingsteps* numa mesma sequência de operações de usinagem executadas da mesma forma que no programa da peça em código G. Por último, é criado o programa da peça em STEP-NC para um controlador STEP-NC.

Na Figura 2.30 é possível observar a sequência de passos para transformar o Código G para STEP-NC.

Enfim, a presente patente mostra que não é suficiente ter só o programa da peça em código G, mas é preciso ter outras informações importantes como as da ferramenta para se conseguir um programa de peça em STEP-NC completo de outra forma seria difícilimo.

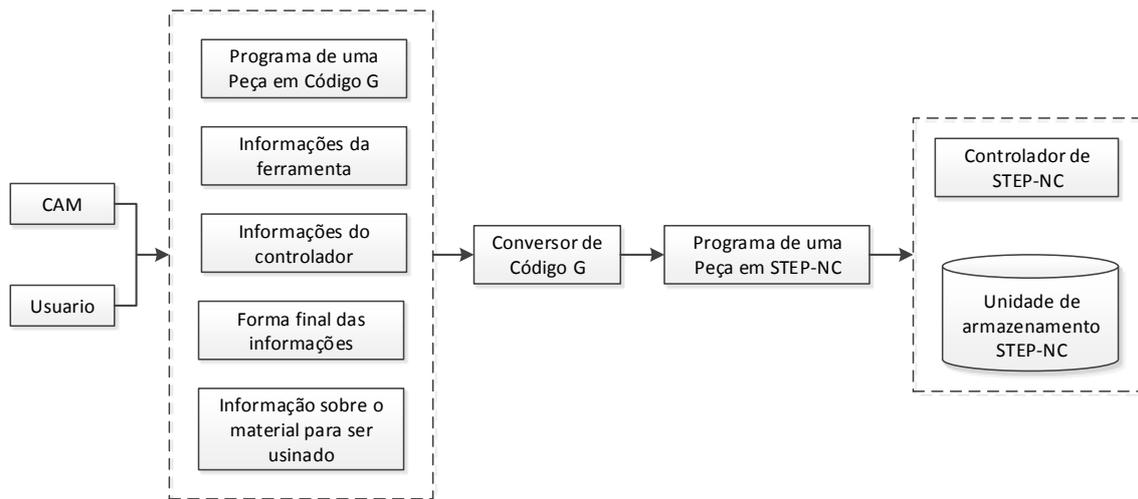


Figura 2.30: Passos para transformar Código G para STEP-NC (Adaptado de [58])

### 2.9.2. Patente 2 No US 6,795,749 B2: Método para geração automática de um programa de peça para uso no STEP-NC [59]

Esta patente refere-se a um método para gerar automaticamente um programa de peça (STEP-NC) para uso num controlador STEP-NC em um sistema SFP (programação de chão de fábrica) com base no modelo de dados fornecido pela norma ISO14649.

O desenvolvimento da invenção que se apresenta nesta patente descreve um procedimento de várias fases. Tais fases são:

- Primeiro, é feita a interpretação de um arquivo físico STEP ou de um programa de peça em ISO14649. Isto gera uma base de dados com as *features* reconhecidas. Em seguida, é estabelecido um plano de processos de acordo a dita norma.
- Segundo, o plano de processos é editado e um programa de peça em STEP-NC é gerado.
- Terceiro, de acordo com as informações das *features* de usinagem é feita a trajetória da ferramenta para usiná-las cada qual independentemente. Depois a trajetória é verificada num equipamento CNC.

Por fim, ao incluir as funções de um sistema CAD/CAPP/CAM, a presente invenção faz que seja possível gerar rapidamente um programa de peça em STEP-NC para utilizá-lo num controlador CNC.

Na Figura 2.31, apresenta-se um diagrama IDEF0 representando a função principal do método descrito anteriormente.

É importante ressaltar o método usado nesta patente, pois serve como ponto de referência para desenvolver o software que permitirá aderir a norma ISO14649 no EMC2.

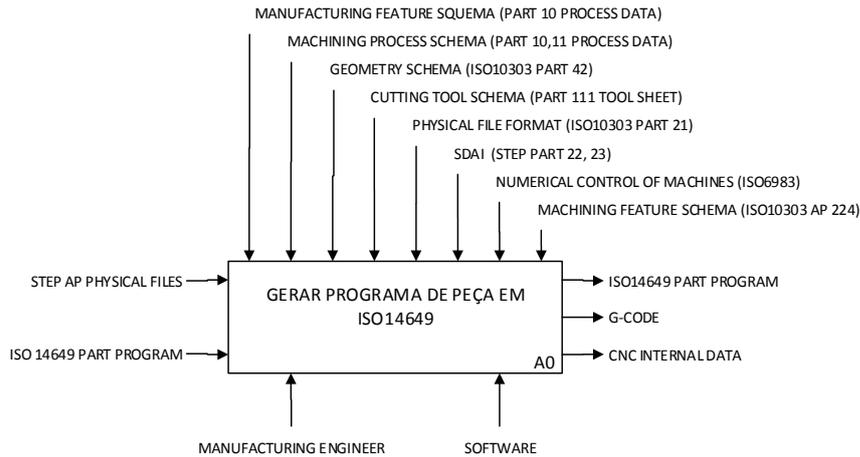


Figura 2.31: Diagrama IDEFØ da função principal da patente 2 (Adaptado de [59])

## 2.10. TIPOS DE CONTROLADORES ADERENTES A STEP-NC

Segundo SUH et al. [31], os controladores aderentes são classificados geralmente em três tipos. Na Figura 2.32, amostra-se a classificação.

- O tipo A incorpora a STEP-NC para o controle convencional via pós-processador. Nesse caso o CNC pode ser usado sem modificações, isto não pode ser considerado como um CNC preparado para STEP-NC, mas deve ao menos ser capaz de ler programas de peça em STEP-NC
- O tipo B é o “novo tipo de controle” que tem um interpretador STEP-NC como parte do CNC, o *workingstep* programado é executado pelo kernel do CNC com capacidade de geração de trajetória de ferramenta. Esse é o tipo básico onde o movimento é executado fielmente. É baseado na estratégia e sequência de usinagem como especificado pelo STEP-NC. Ou seja o controle não é inteligente mas só gera trajetória de ferramenta.
- O tipo C é muito mais funcional que os tipos A e B, e é o “novo controle inteligente” em que o CNC pode executar de maneira autônoma e inteligente as tarefas de usinagem baseado na informação de um programas de peça em STEP-NC. Um exemplo desse tipo de controlador é o STEP-NC *MACHINE* da empresa *STEP Tools* (ver sessão 2.7.1).

Segundo o exposto no texto anterior, no presente trabalho o tipo de controlador será do tipo A, pois o controlador EMC não faz o processamento direto de um programa de peça em STEP-NC, mas sim poderá ler esse programa mediante um adaptador que irá convertê-lo em Código G/M o qual é o código nativo deste controlador.

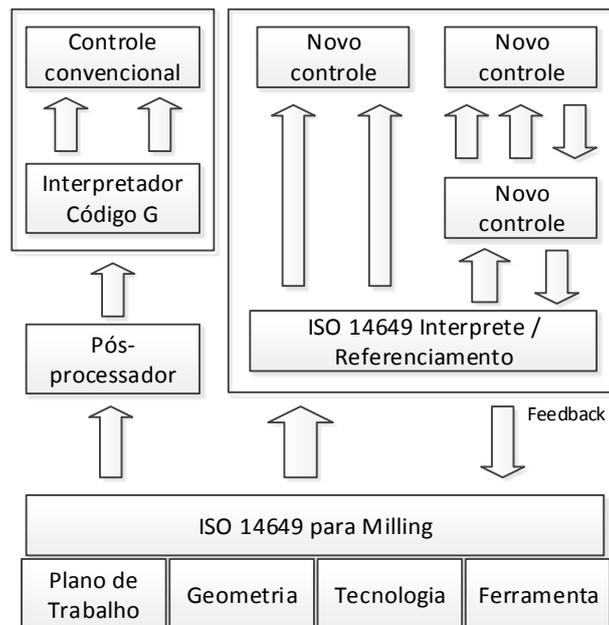


Figura 2.32: Tipos de controladores STEP-NC (Adaptado de [60])

Depois de expor as anteriores referências, é importante ressaltar que existe um Protocolo de Aplicação que está relacionado com o modelo de dados de STEP-NC descrito na sessão 2.5.2, este AP faz parte do padrão STEP e apresenta o STEP-NC como um padrão.

## 2.11. O PROTOCOLO DE APLICAÇÃO 238 OU AP238

O título oficial do ISO10303-238 é “Automatização de sistemas industriais e integração – representação e intercâmbio de dados de produto – Parte 238: protocolo de aplicação: Modelo interpretado de aplicação para controladores numéricos computadorizados” [61], mas é também conhecido simplesmente como AP238. Esse AP é o resultado da ampliação do padrão ISO10303 combinado com a norma STEP-NC. Atualmente o termo STEP-NC representa tanto o ISO14649 quanto o ISO10303-238.

O AP238 permite que um programador descreva não apenas "como fazer" uma peça, mas também "o que fazer", incorporando o conhecimento do produto baseado em *features* diretamente para os programas do controlador CNC, como parte dos requisitos de informação disponíveis para a programação NC [62].

No AP-238, a informação requerida para controlar uma máquina é acoplada à informação gerada pelo sistema CAD/CAM para criar uma verdadeira independência do Controlador CNC. Ele toma o modelo de dados e o leva dentro do conjunto de normas do STEP. Com isso, os CNCs podem ser totalmente integrados com os sistemas CAD, CAM, CAPP e outros CA-aplicações. Um exemplo de uso do AP238 é o software STEP-NC *MACHINE*.

AP-238 define uma nova interface para sistemas existentes. Este AP não necessita novas capacidades para ser adicionado a sistemas existentes. Em vez disso, define um novo padrão para a informação existente que já está em sistemas CAM para que as informações sejam passadas para o sistema CNC, tornando-os mais fáceis de programar e operar. Há vários exemplos de software CAM

comercial como CATIA, GibbsCAM, UGS NX e Mastercam, que demonstram essa funcionalidade. Eles usam o padrão (ISO10303-238) por meio de plug-ins (tradutores de exportação) [63].

Além disso, o AP238 propõe a implementação de uma nova extensão de *nc\_functions* para rastreabilidade e monitoramento de dados de acordo com o programa STEP-NC. Na Figura 2.33, essa ideia é apresentada [62].

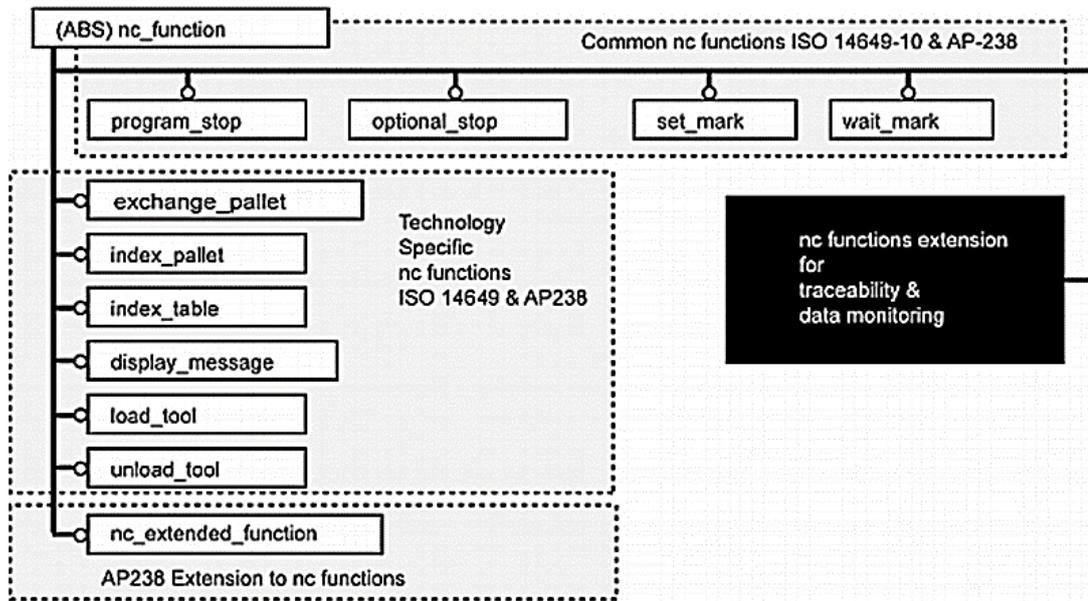


Figura 2.33: Extensão de *nc\_functions* [62]

A primeira edição deste protocolo de aplicação foi publicada em 2007 e, atualmente, o fórum de STEP-NC está preparando a segunda edição do AP-238, adicionando novas capacidades [41].

### 3. MODELAGEM DO CNC ADERENTE A STEP-NC

Neste capítulo é apresentada a metodologia para modelar as funcionalidades do CNC aderente a STEP-NC. Essa modelagem está dividida em duas partes: modelagem funcional de um protótipo de adaptador de programa de peça em STEP-NC a Código G/M e a modelagem que descreve a inclusão deste (o adaptador) no Controlador NC, neste caso o LinuxCNC (EMC) junto ao padrão *MTConnect*. Primeiramente, são usados alguns conceitos sobre a metodologia de trabalho, além de informações necessárias para a modelagem funcional do protótipo do adaptador. Esses conceitos estão descritos no apêndice 1. Em seguida é feita a modelagem funcional do CNC aderente a STEP-NC.

#### 3.1. ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA

A qualidade com que seja realizada a especificação dos requisitos vai influenciar no processo todo de desenvolvimento do software, impactando as fases restantes de desenvolvimento dele mesmo. Em seguida são especificados os requisitos funcionais mais importantes para a modelagem da arquitetura do adaptador de STEP-NC a Código G/M.

- Requisitos funcionais
  - Extrair as informações de usinagem do programa de peça em STEP-NC de acordo com seu modelo de dados.
  - Converter um programa de peça em STEP-NC a Código G/M de acordo com as informações de usinagem.
  - Construir o arquivo físico p21 (ou STP) original para visualizá-lo na GUI AXIS do controlador EMC.
  - Gerar um arquivo (.tbl: *tool table*) com as ferramentas usadas na usinagem para o EMC.
  - A conversão fonte-fonte deve ser automática e transparente para o usuário final.

Tomando como referência os requisitos gerais supracitados é possível iniciar a modelagem das funções do adaptador, levando em conta também conceitos como: Conversão fonte-fonte, identificação de *tokens*, modularidade, recursividade entre outros antes referidos.

#### 3.2. MODELAGEM FUNCIONAL DO CNC ADERENTE À NORMA STEP-NC

Na Figura 3.1 é apresentado um modelo funcional geral de um sistema integrado CAD/CAPP/CAM/CNC aderente ao padrão STEP-NC usando o diagrama IDEF0. Este sistema permite visualizar o ciclo de vida, o projeto, o planejamento, o CAPP com geração programa STEP-NC (por parte do *STEP Modeler*), e o CAM execução por parte do Controlador NC (EMC) aderente STEP-NC, graças ao adaptador de STEP-NC a Código G.

A modelagem funcional ilustrada na Figura 3.2 vai além do implementado neste projeto. Para isto são adicionados dois componentes que apenas serão modelados em conjunto com o adaptador para criar o modelo integrado CAD/CAPP/CAM/CNC aderente ao padrão STEP-NC. O primeiro dos componentes adicionais é o *STEP Modeler* e o segundo é o padrão *MTConnect*. Esses componentes não serão implementados neste trabalho.

O software *STEP Modeler* descreve um sistema integrado CAD/CAPP/CAM [11]. Deste modelo é isolada uma das suas funções: a geração do arquivo físico p21 ou STP. Esse arquivo será usado como entrada no controlador CNC aderente a STEP-NC e conseguir a usinagem da peça dependendo dos *workingsteps*. A modelagem funcional do *STEP Modeler* não é implementada neste trabalho.

Já o *MTCConnect* é um protocolo aberto e um padrão baseado em XML para integração de dados, que pode atuar como um facilitador para os padrões de nível superior [64]. Também é um conjunto de normas destinadas a facilitar a comunicação entre controladores e dispositivos no chão de fábrica com aplicativos de software, através da publicação de dados por meio de redes, utilizando o Protocolo de Internet (IP) [65]. Sua arquitetura pode ser facilmente implementada e adaptada às máquinas existentes no chão de fábrica, portanto, proporciona funções de flexibilidade e portabilidade para vários tipos de ambientes de usinagem.

O *MTCConnect* tem quatro componentes: dispositivo, adaptador, Agente e Cliente; eles agem coletivamente como a coluna vertebral do padrão de comunicação. O dispositivo são todos os componentes, tais como controlador, sensores e máquinas-ferramenta, que são responsáveis pelo fornecimento dos dados monitorados. Estes dados são adquiridos pelo sistema de aquisição de dados e recolhidos por um adaptador. O adaptador é responsável por comunicar e transmitir isso para o agente em um formato padrão. O processo de aquisição de dados funciona com a Interface de programação de aplicação do dispositivo (API), com o qual o adaptador se comunica. O agente, então, aceita os pedidos de dados de um aplicativo cliente, que retorna os dados em formato XML. O cliente pode então extrair os dados do documento e apresentá-los para o usuário. Estes dados podem ser avaliados posteriormente pelo operador, tendo em conta a condição atual da máquina-ferramenta [66]. Um exemplo do uso do padrão *MTCConnect* é monitoramento remoto de uma máquina-ferramenta mediante um programa (web browser) e a comunicação com seu Controlador NC. A modelagem funcional do *MTCConnect* e o CNC não é implementada neste trabalho.

Na Figura 3.2 pode-se observar os níveis A1 e A2. No nível A1 está modelada a primeira parte do sistema integrado sendo referência do sistema *STEP Modeler* (Modelo geral, não é implementado). O nível A2 mostra a modelagem do CNC aderente a STEP-NC equipado com *MTCConnect*, contendo em seus sub-níveis a implementação do adaptador de STEP-NC a Código G/M (nível A21). O modelado do sistema de monitoramento via web é feito por meio do padrão *MTCConnect* (nível A221) sendo integrado ao controlador EMC como se observa na Figura 3.16.

É necessário explicar que da modelagem proposta na Figura 3.2, só é feita a especificação do modelo funcional do CNC aderente ao padrão STEP-NC. Portanto, é realizada unicamente neste projeto a implementação do protótipo do adaptador de STEP-NC a Código G/M (ver Figura 3.4) adicionando-o no controlador NC.

Para realizar a abordagem funcional e modelar as atividades do protótipo de adaptador de STEP-NC a Código G/M é imprescindível fazer uma análise das informações de usinagem da norma STEP-NC, conforme descrito nas sessões de 2.6 a 2.10, ou seja, fazer uma modelagem baseada em entidades, *features*, operações e ferramentas.

Como já foi afirmado, o sistema *STEP Modeler* da UFSC/GRIMA é usado para gerar o arquivo fonte STP (Figura 3.2), com isto se consegue validar não só o protótipo do adaptador (e consequentemente o CNC aderente a STEP-NC), mas também o sistema da UFSC. Como resultado, o adaptador deve permitir que o controlador LinuxCNC (ou EMC2) execute o programa de peça em STEP-NC de maneira automática e transparente para o operador da máquina-ferramenta. A seguir, modela-se o adaptador utilizando os diagramas IDEFØ e UML para descrever seu funcionamento.

Na Figura 3.3, mostra-se a arquitetura do adaptador em seu nível mais alto (A21) usando o diagrama IDEFØ. Posteriormente, na Figura 3.5, essa atividade (A21) é separada em cada uma de suas sub-funções, assim: extração de informações básicas (A211), extração de informações avançadas

(A212), gerar código G/M (A213). A seguir serão descritas detalhadamente cada uma delas. Também serão mencionados e mostrados o nível A22 com seus sub níveis A221, A222, A223.

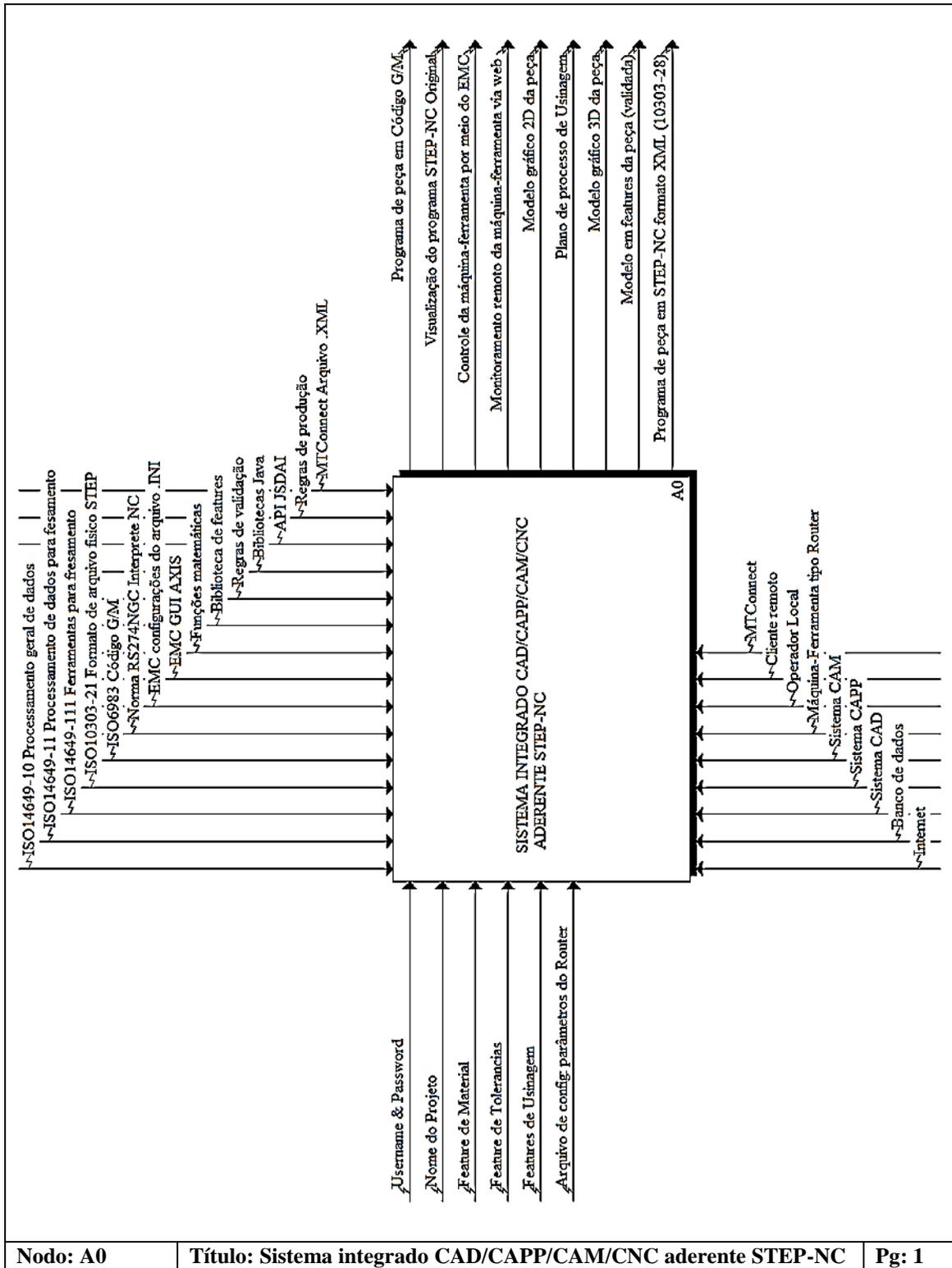


Figura 3.1: Diagrama geral IDEF0 do Sistema integrado CAD/CAPP/CAM/CNC aderente STEP-NC.

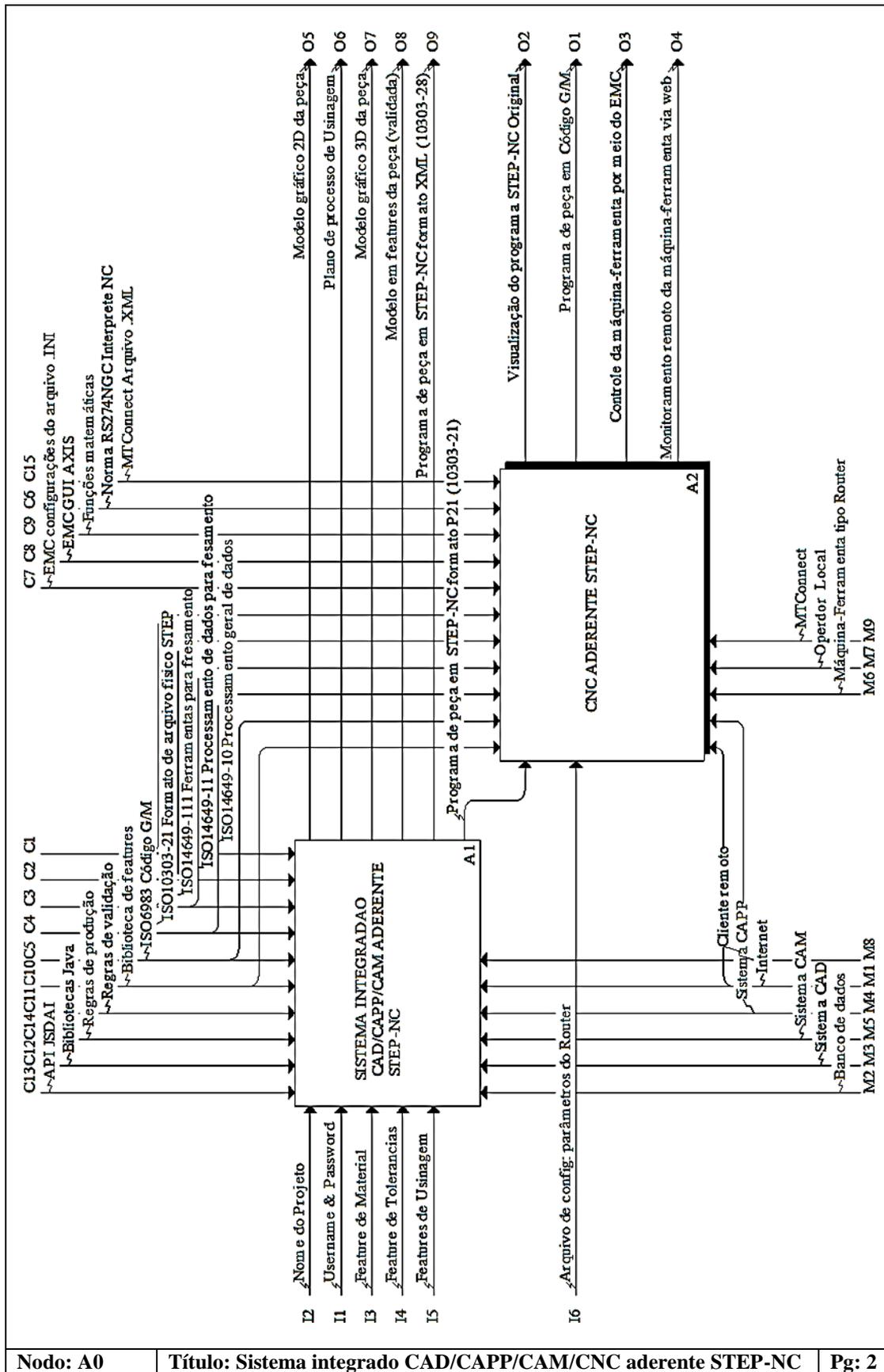


Figura 3.2: Diagrama IDEF0 do sistema integrado, dentro do nível A0

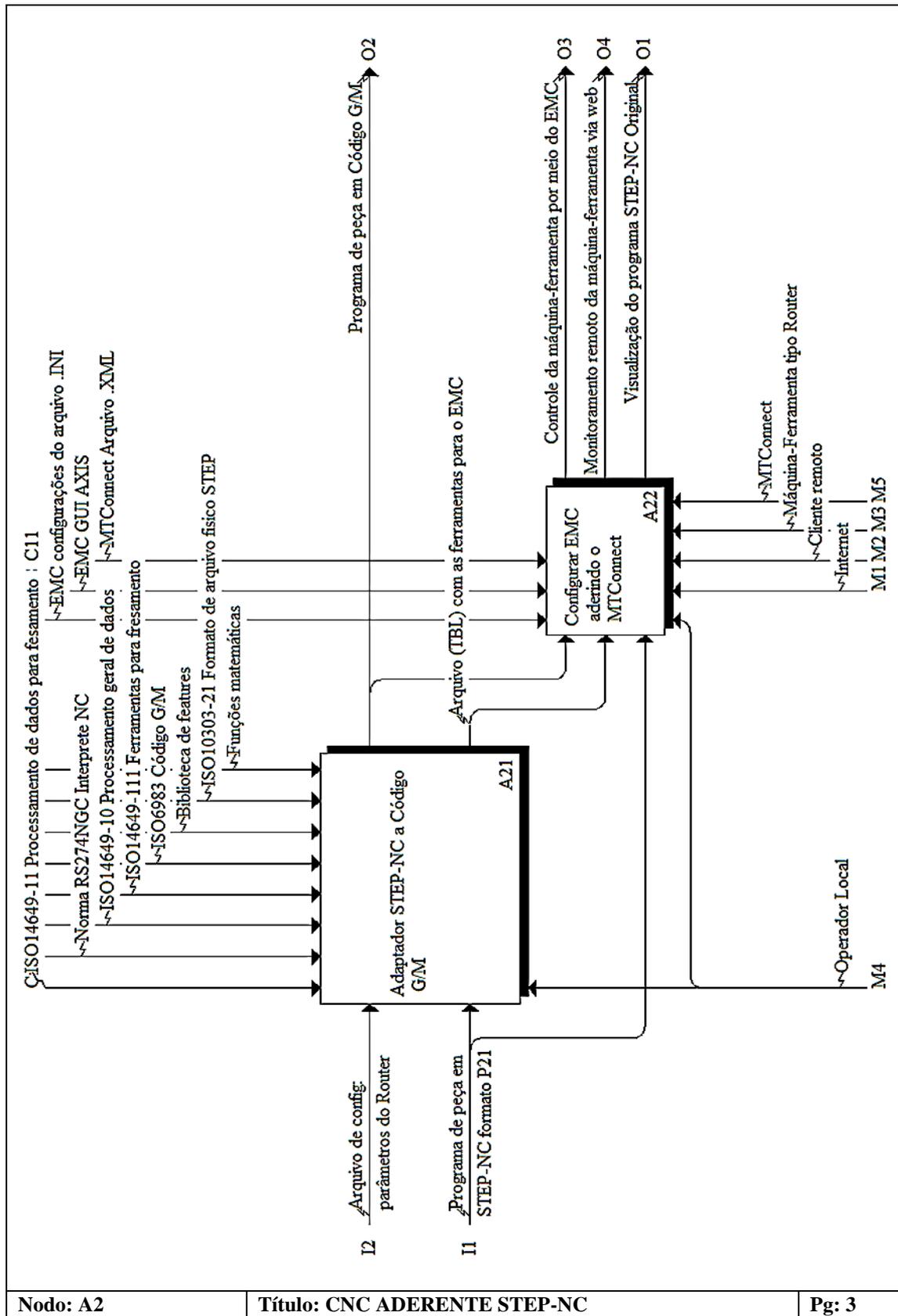


Figura 3.3: Diagrama IDEF0 do CNC aderente STEP-NC, dentro do nível A2

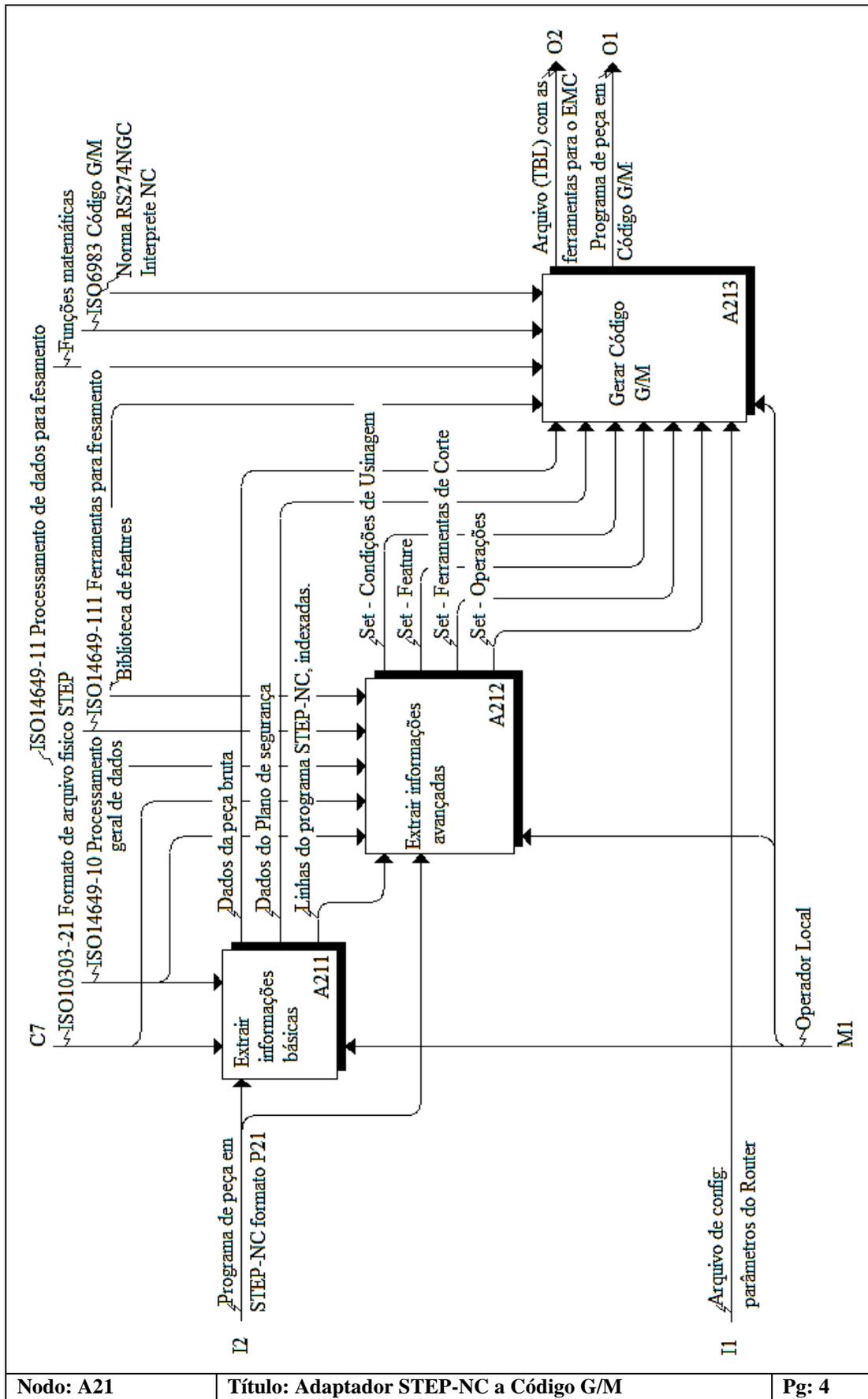


Figura 3.4: Diagrama IDEF0 do Adaptador STEP-NC a Código G/M, dentro do nível A21

### 3.2.1. Extrair informações básicas

Esta atividade (nível A211) tem a função de extrair principalmente duas informações básicas do programa de peça STEP-NC, dados da peça de trabalho (peça bruta) e dados do plano de segurança, no entanto, possui outras funções.

Os dados da peça de trabalho relaciona-se com a extração das características geométricas dimensionais da peça bruta, item essencial para algumas das *features* e em consequência a geração do código G. Os dados do plano de segurança relaciona-se com a posição fixa num ponto coordenado XYZ, importante por ser a referência para a mudança de uma *feature* para outra, assim como para mudar de um plano coordenado (ver sessão 4.2), isto com o fim de colocar um ponto coordenado específico para a troca de ferramenta manual. Também, esta atividade tem como função passar o programa de peça em STEP-NC de entrada para uma GUI onde posteriormente irá ser visualizado junto com a GUI do controlador NC.

Da Figura 3.5 pode-se observar que esta atividade possui as seguintes funções:

- **A2111 - Tirar *tokens* irrelevantes:** esta função é simples, mas é muito importante. Sem ela a extração de informação se vê comprometida. Para extrair dados do programa de peça em STEP-NC, é necessário identificar os *tokens* que nele se encontram, mas este formato do arquivo físico p21 contém espaços em branco, delimitadores e comentários que prejudicam a legibilidade do programa e, portanto, são completamente irrelevantes, é por isto que é imprescindível que exista esta função.
- **A2112 - Separar linha a linha atribuindo um endereço:** esta função é a base para adquirir todas as informações, pois faz uma referência das linhas do código do arquivo físico, ou seja, cada linha tem um endereço (e.g. #10, #240) com o qual é identificada, como pode se observar nas Figura 1.2 e Figura 2.7b. Desta forma pode-se indexar cada linha e extrair as informações contidas de cada entidade.
- **A2113 - Extrair informações da geometria da peça bruta:** esta função procura o endereço da linha que contém a entidade *workpiece* e em consequência o endereço da linha que identifica a entidade *block* que contém a informação do comprimento, largura e espessura da peça de trabalho.
- **A2114 - Extrair posição XYZ do Plano de Segurança:** esta função procura o endereço da linha que contém a entidade *workplan*, e a partir desta procura-se a entidade *setup* e, posteriormente, a entidade *plane* que contém a informação da posição XYZ do plano de segurança.

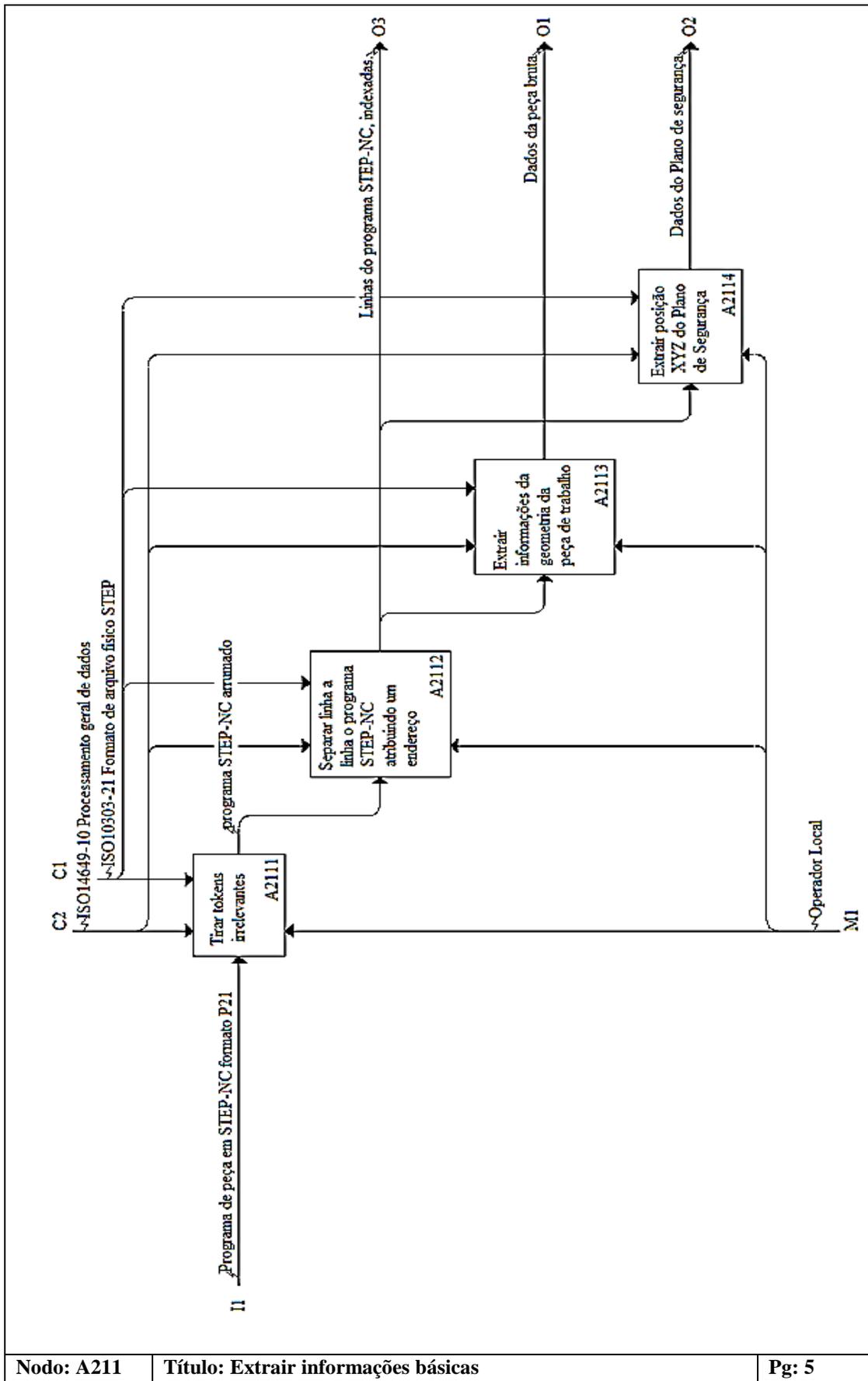


Figura 3.5: Diagrama IDEF0 - Extrair informações básicas, dentro do nível A211

Fazendo uso da linguagem UML e especificamente utilizando diagrama de classes, serão apresentados os atributos principais e a relação das classes envolvidas na extração das informações básicas do programa da peça em STEP-NC. A Figura 3.6 ilustra o diagrama UML da atividade A2111. É importante salientar que as relações entre as classes que compõem o adaptador são prioritariamente do tipo dependente (a alteração de um objeto independente afeta diretamente o objeto dependente). É criada também uma classe principal chamada “Integrador”. Essa classe comanda a sequência de execução do algoritmo do adaptador.

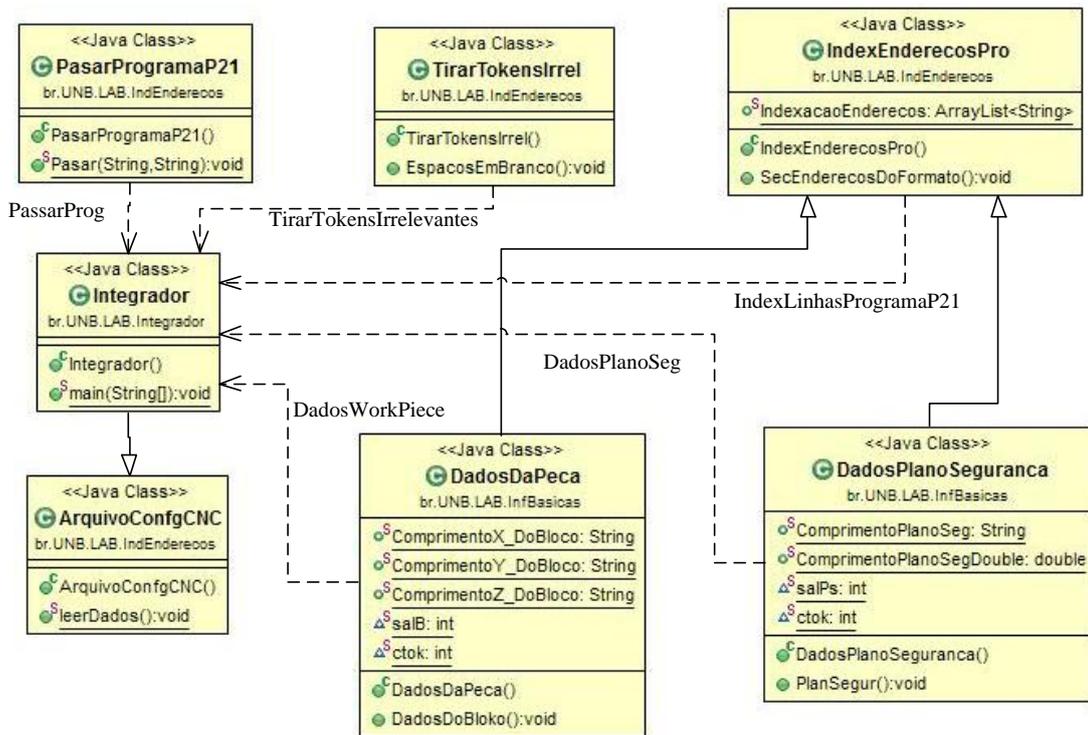


Figura 3.6: Diagrama UML, classes da extração de informações básicas

### 3.2.2. Extração de informações avançadas

Esta atividade (nível A212 na Figura 3.7) identifica a entidade *Machining\_workingstep* mapeando as *features*, condições de usinagem, operações e ferramentas (contidas no programa de peça em STEP-NC) com o propósito de obter suas características e suas informações. O mapeamento dessas informações é feito de acordo com o tipo de *feature* de usinagem segundo a norma ISO14649.

Na Figura 3.7 poder se observar que esta atividade possui as seguintes funções:

- **A2121 - Extrair a(s) *machining\_workingstep*:** esta função identifica e extrai todas as *machining\_workingstep* do programa da peça em STEP-NC. Em seguida são listadas para posteriormente extrair os elementos do processo de usinagem, como as *features* de usinagem, as operações de usinagem entre outros.
- **A2122 - Extrair informações da *machining\_workingstep* para o processo de usinagem:** as dados que estão contidos na entidade *machining\_workingstep* são o ponto de partida para a geração do código G. Por meio desta função são obtidos um conjunto de *features*, condições de usinagem, operações e ferramentas que indicam o processo de usinagem descrito pelo programa da peça em STEP-NC.

Usando UML e diagrama de classes serão apresentados a seguir os atributos principais e a relação das classes envolvidas na extração das informações avançadas do programa de peça em STEP-NC. O diagrama UML da atividade A2112 (Figura 3.8) segue a ordem de execução dos dados de processo que são necessários para a programação NC, segundo a linguagem ou esquema EXPRESS, conforme as entidades definidas nas partes 10, 11 e 111 da norma ISO14649.

É importante notar que todas as informações obtidas de uma *machining\_workingstep* serão armazenadas em um *ArrayList*<sup>2</sup> tipo *String* chamado “*InformacoesAvançadas*”, deste serão extraídos os dados que utilizará o gerador de código G.

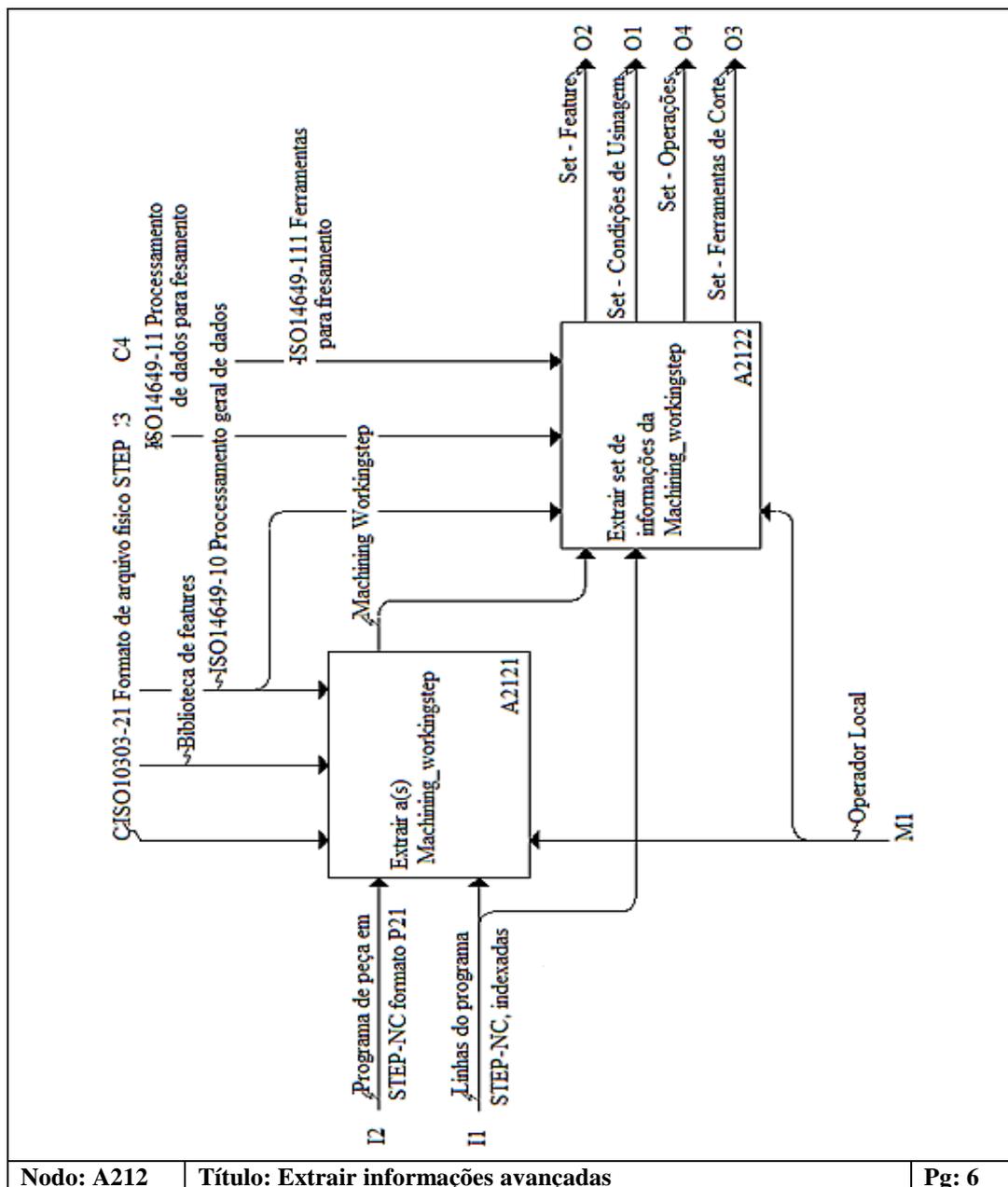


Figura 3.7: Diagrama IDEF0 - Extrair informações Avançadas, dentro do nível A212

<sup>2</sup> Lista que permite conter e ordenar objetos. Seu tamanho é dinâmico, ou seja, esta lista vai crescer à medida que mais elementos são inseridos nela [116].

Já fazendo uso do software *STEP Modeler*, pode-se criar, e.g. um programa de peça em STEP-NC de uma peça bruta com uma *feature* tipo furo com base plana (ver Figura 3.9), isto com o fim de visar a modelagem funcional e a organização dos dados no *ArrayList* das informações avançadas.

A organização dos dados no *ArrayList* (InformaçõesAvançadas) será feita da seguinte forma: Tipo de *feature* + Número de *feature*, Número de ferramenta + dados da ferramenta, Posição XYZ, Profundidade Z, Dados da *feature*, Tipo de operação. A Figura 3.10 exibe os dados esperados no *ArrayList* - InformaçõesAvançadas.

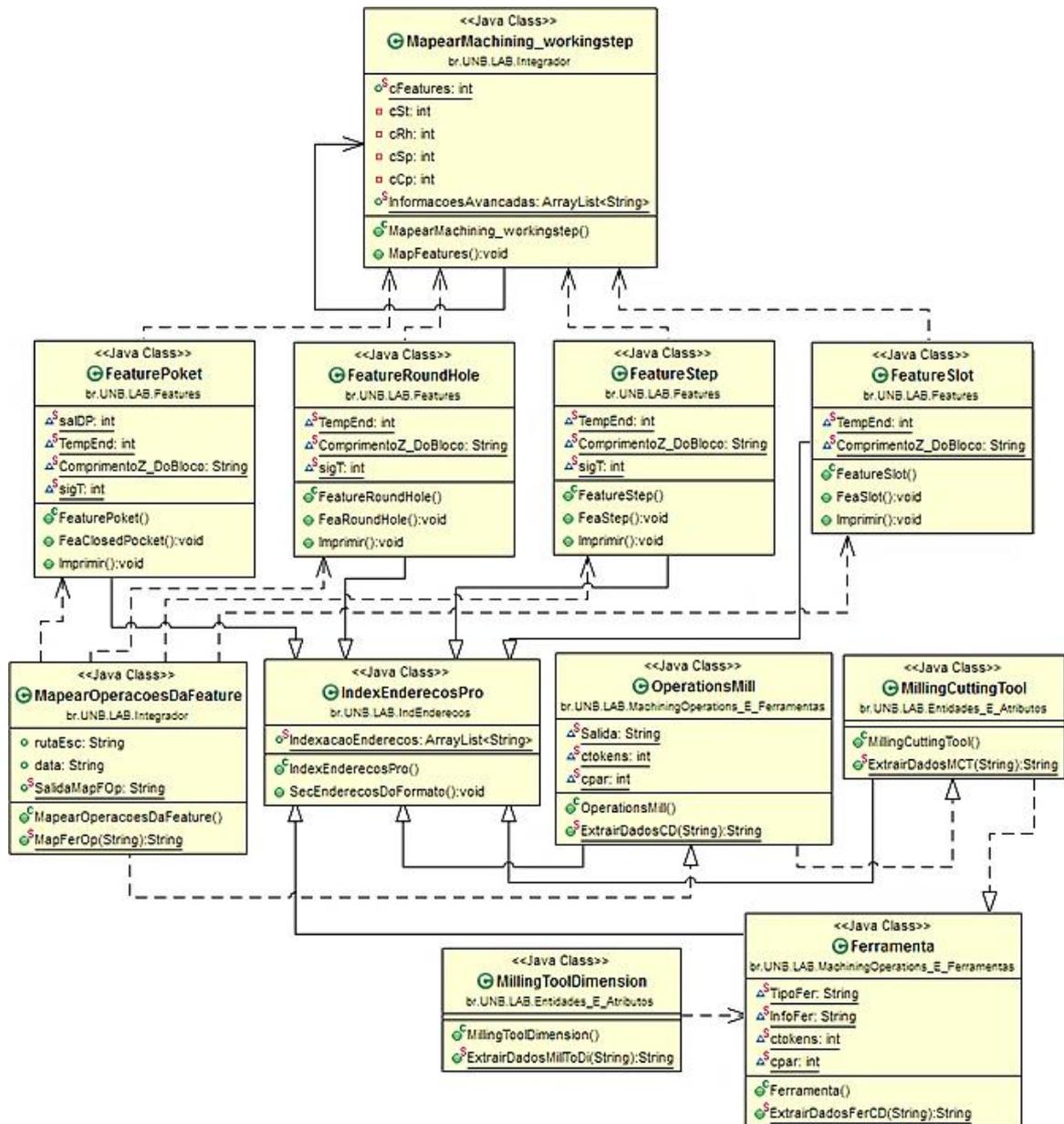


Figura 3.8: Diagrama UML, classes da extração de informações Avançadas

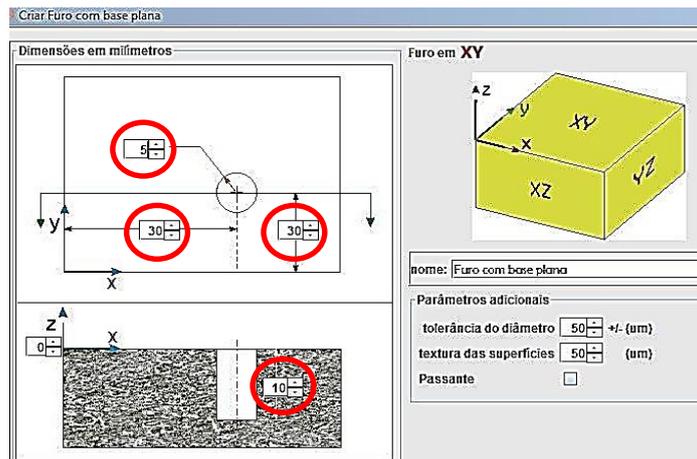


Figura 3.9: Exemplo de criação de uma *feature* no software STEP Modeler

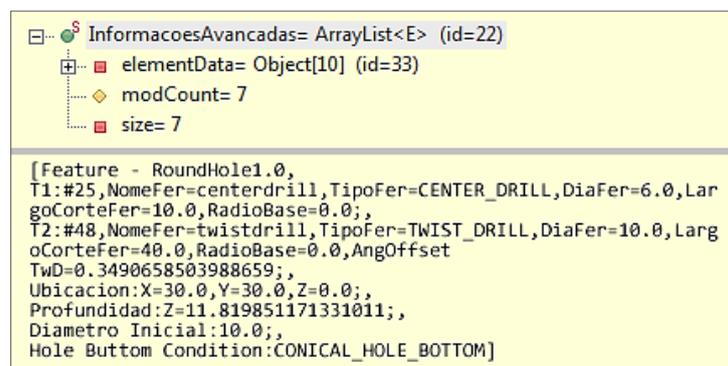


Figura 3.10: Informações que deverá conter o *ArrayList* - *InformaçõesAvançadas*

Obtendo-se os dados essenciais tanto básicos como avançados, prossegue-se com a atividade encarregada da geração do código G/M.

### 3.2.3. Geração de Código G/M

A geração do código G/M parte das informações básicas e avançadas. Com isto, adota-se como base o formato das instruções do programa para máquinas de controle numérico contidas no ISO6983, levando em conta as modificações do código G/M aceitas pelo intérprete RS274NGC (intérprete de códigos G do controlador EMC usado neste projeto).

A atividade que gera o código G/M está definida no nível A213 do modelo IDFEØ como mostrado na Figura 3.4. Sua funcionalidade reside na individualização de cada *feature* e a sua relação de informações.

Cada *feature* é extraída de forma individual do conjunto de *features*, assim como as suas condições de usinagem, operação e ferramentas para usiná-la. Depois de obter os dados necessários pode-se gerar o código de programa NC contendo as trajetórias da ferramenta (*toolpath*), velocidade de avanço (*feedrate*), troca de ferramenta manual, plano de segurança, entre outras [4], [5], [25]. Também, esta atividade tem a função de criar uma lista contendo todas as ferramentas usadas no programa NC, tanto para uma como várias *features* (apêndice B.4.2, o arquivo tbl). Na Figura 3.11 apresenta-se o diagrama IDEFØ da atividade: “Gerar Código G/M” da arquitetura do adaptador.

Como foi referido anteriormente, as atividades A2131, A2132, A2133 e A2134 individualizam os dados referentes a uma só *feature* com suas informações.

Depois de gerar o código G/M de uma *feature*, prossegue com a seguinte (se houver) até terminar todo o *set* de *features*.

A atividade A2135 é a encarregada de gerar um arquivo “*tool.tbl*” que contém a tabela com os dados das ferramentas de corte usadas para a usinagem de todo o *set* de *features* contidas no programa NC [67]. O arquivo *tool.tbl* é usado no EMC para a execução correta do programa NC, da compensação de ferramenta e da troca de ferramenta. A exposição deste arquivo é feita no apêndice B.4.2.

Por último, encontra-se a atividade A2136. Essa é o coração do adaptador e depende completamente das atividades anteriores. Nessa se sintetizam todos dados das informações básicas e avançadas, além das informações (parâmetros de configuração como *feedrate* máxima, profundidade máxima de corte, etc.) da máquina-ferramenta contidas em um arquivo externo. Com esses dados é possível gerar o programa NC levando em conta a sintaxe descrita no padrão ISO6983 e o intérprete RS247 para programas NC. Este programa de peça em Código G/M ou NC contém a trajetória da ferramenta (*toolpath*), velocidade de avanço (*feedrate*), indexação das ferramentas para a troca manual de ferramenta, a posição em um ponto coordenado XYZ do plano de segurança entre outras (ver Figura 3.12).

Em seguida se mostra os diagramas de classe UML para descrever a atividade do nível A213. A Figura 3.13 apresenta a relação das classes envolvidas na geração do código G/M, depois na Figura 3.14, mostra-se a relação das classes envolvidas na geração do arquivo *tool.tbl* tendo como exemplo uma *feature* tipo *Step* (ver sessão 2.7.4), já que trabalha-se da mesma forma para as outras *features*.

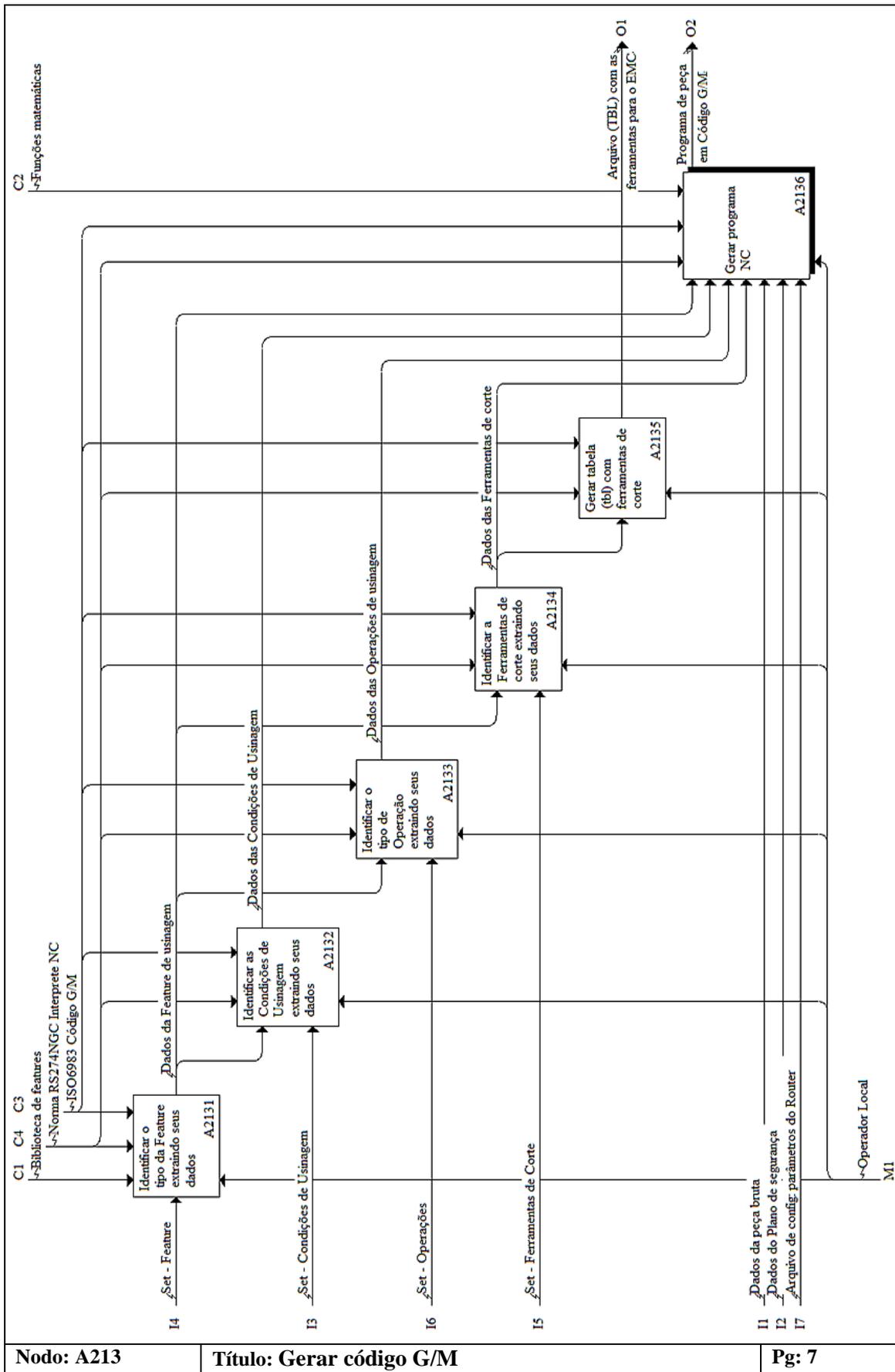


Figura 3.11: Diagrama IDEF0 – Gerar código G/M, dentro do nível A213

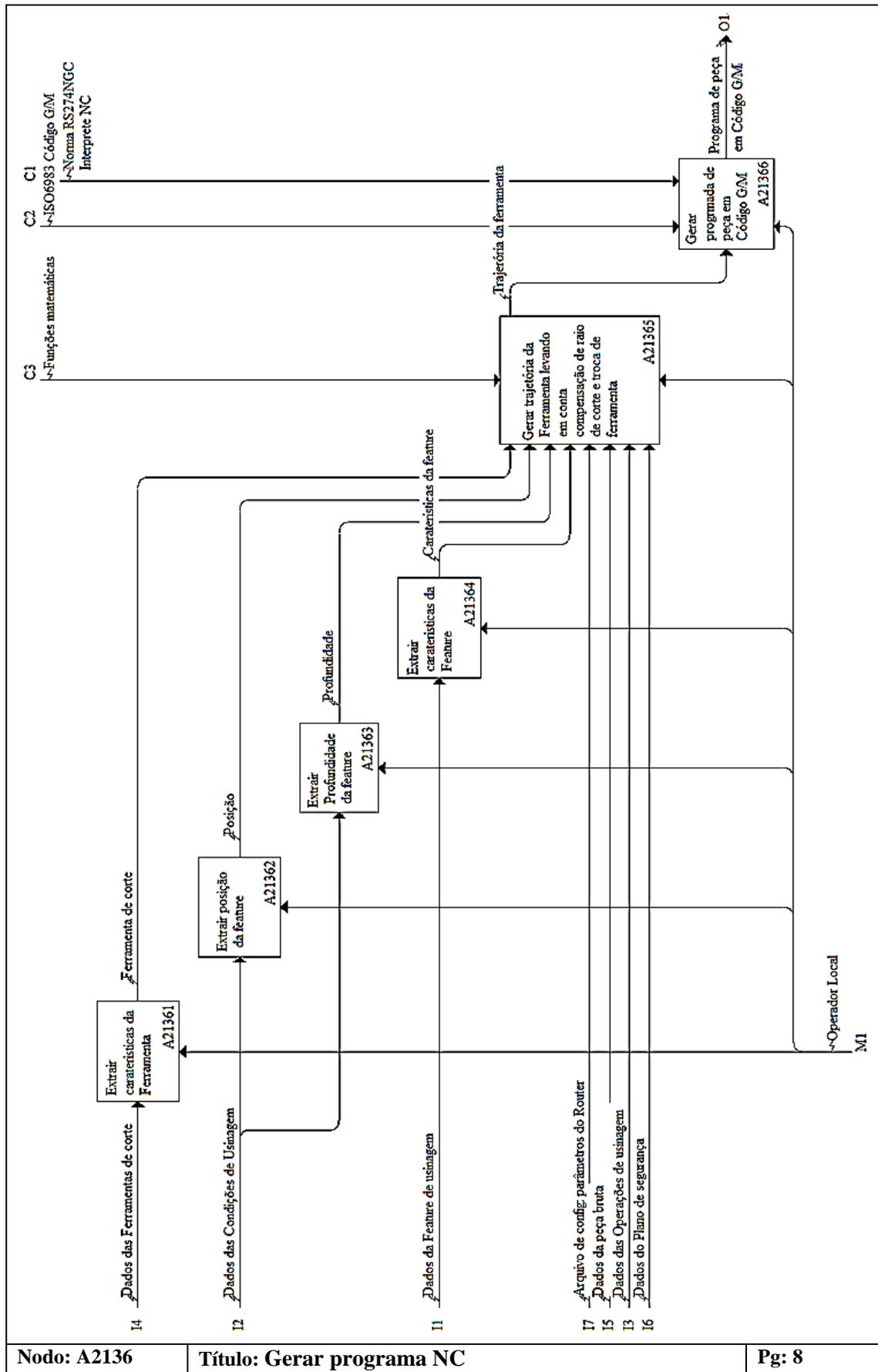


Figura 3.12: Diagrama IDEF0 – Gerar programa NC, dentro do nível A2136



Como foi exposto no início deste capítulo, este projeto está limitado somente ao CNC aderente a STEP-NC e consequentemente só será implementado o adaptador de STEP-NC a Código G/M, os demais componentes mostrados no modelo da Figura 3.3 apenas são modelados funcionalmente, sem implementação.

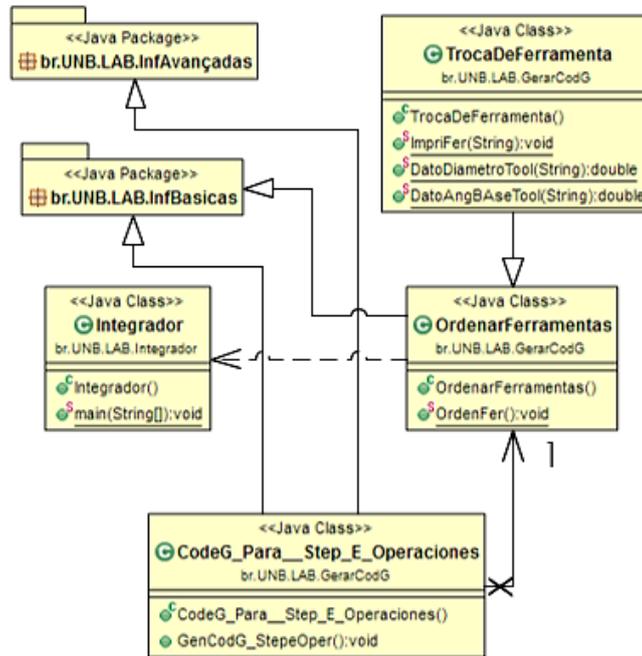


Figura 3.14: Diagrama UML, classes da geração do arquivo “*tool.tbl*”

### 3.2.4. Configurar EMC aderindo o *MTCConnect*

A segunda parte da modelagem funcional do CNC aderente a STEP-NC é a adesão do adaptador de STEP-NC a Código G/M ao CNC junto com o padrão *MTCConnect* para o monitoramento remoto da máquina-ferramenta. Essa última parte (uso do padrão *MTCConnect* para o monitoramento remoto) não é implementada neste trabalho.

Na modelagem funcional do CNC aderente a STEP-NC a modelagem do adaptador é externo ao CNC, mas na implementação ele fica configurado dentro do CNC, permitindo que o usuário da máquina CNC possa abrir programas de peça em STEP-NC em formato p21 ou STP convertendo-o a código G de forma automática e transparente. O nível “A22: Configurar EMC aderindo o *MTCConnect*” do diagrama IDEFØ da Figura 3.3 internamente tem três funções (Figura 3.15). A primeira é encarregada da configuração do *MTCConnect*, a segunda permite adicionar as saídas do adaptador junto com o monitoramento remoto pelo *MTCConnect* na interface do controlador EMC (ver sessão 5.6.1) e a terceira é o controle da máquina mediante o software HAL (*Hardware Abstraction layer*) (ver sessão B.2). A Figura 3.15 mostra as atividades do nível A22.

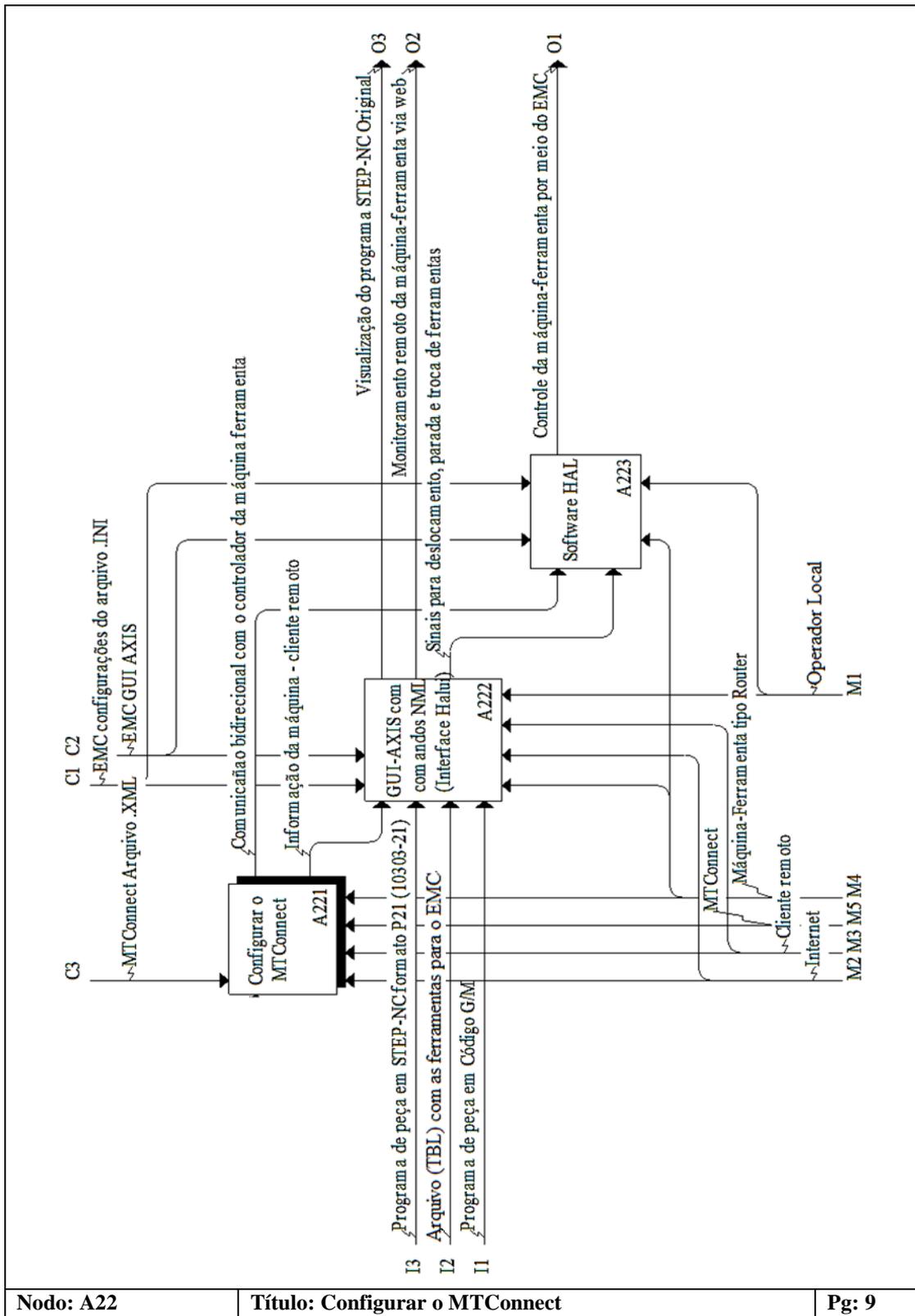


Figura 3.15: Diagrama IDEF0 – Configurar o EMC aderindo o *MTConnect*, dentro do nível A22

## 4. IMPLEMENTAÇÃO DO PROTÓTIPO DE ADAPTADOR DE STEP-NC A CÓDIGO G

A implementação do CNC aderente a STEP-NC consiste em duas etapas principais. A primeira etapa é a implementação de um adaptador de STEP-NC a Código G/M e a segunda é a adição desse adaptador ao controlador LinuxCNC. Este capítulo apresentará a implementação do protótipo de adaptador proposto e modelado no capítulo anterior. A segunda etapa é implementada no capítulo seguinte. A seguir é apresentada a implementação do adaptador de STEP-NC a Código G/M baseado nas sessões 2.6.4, 2.7.3, 2.7.4, 2.8.2 e 2.8.3. É proposto então um diagrama de fluxo que representa a arquitetura do protótipo do adaptador. Ele está dividido em duas partes: Figura 4.1, extração de informações do arquivo físico STEP e Figura 4.2, geração do código G a partir das informações extraídas.

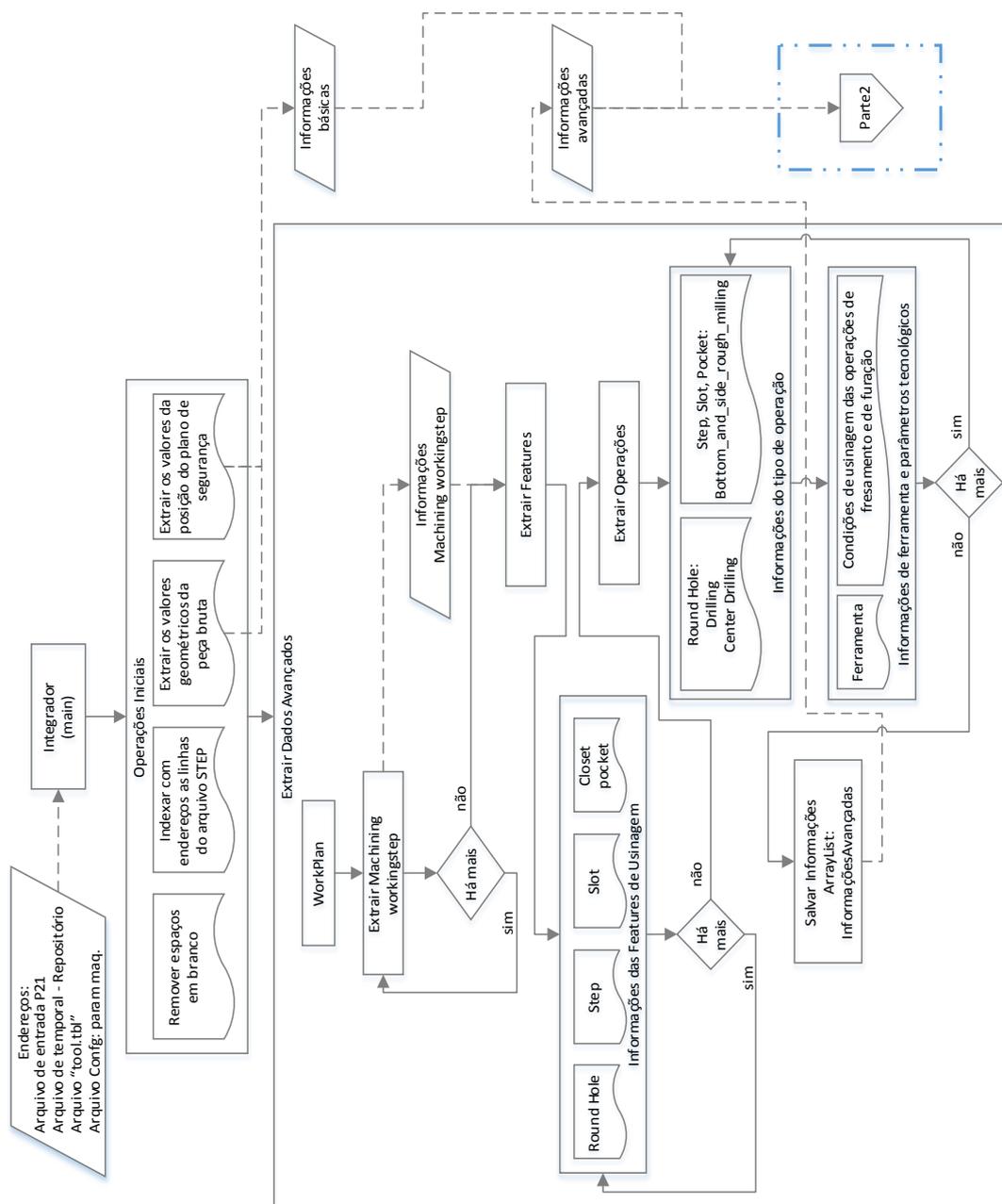


Figura 4.1: Diagrama de fluxo da arquitetura do protótipo do adaptador, parte 1

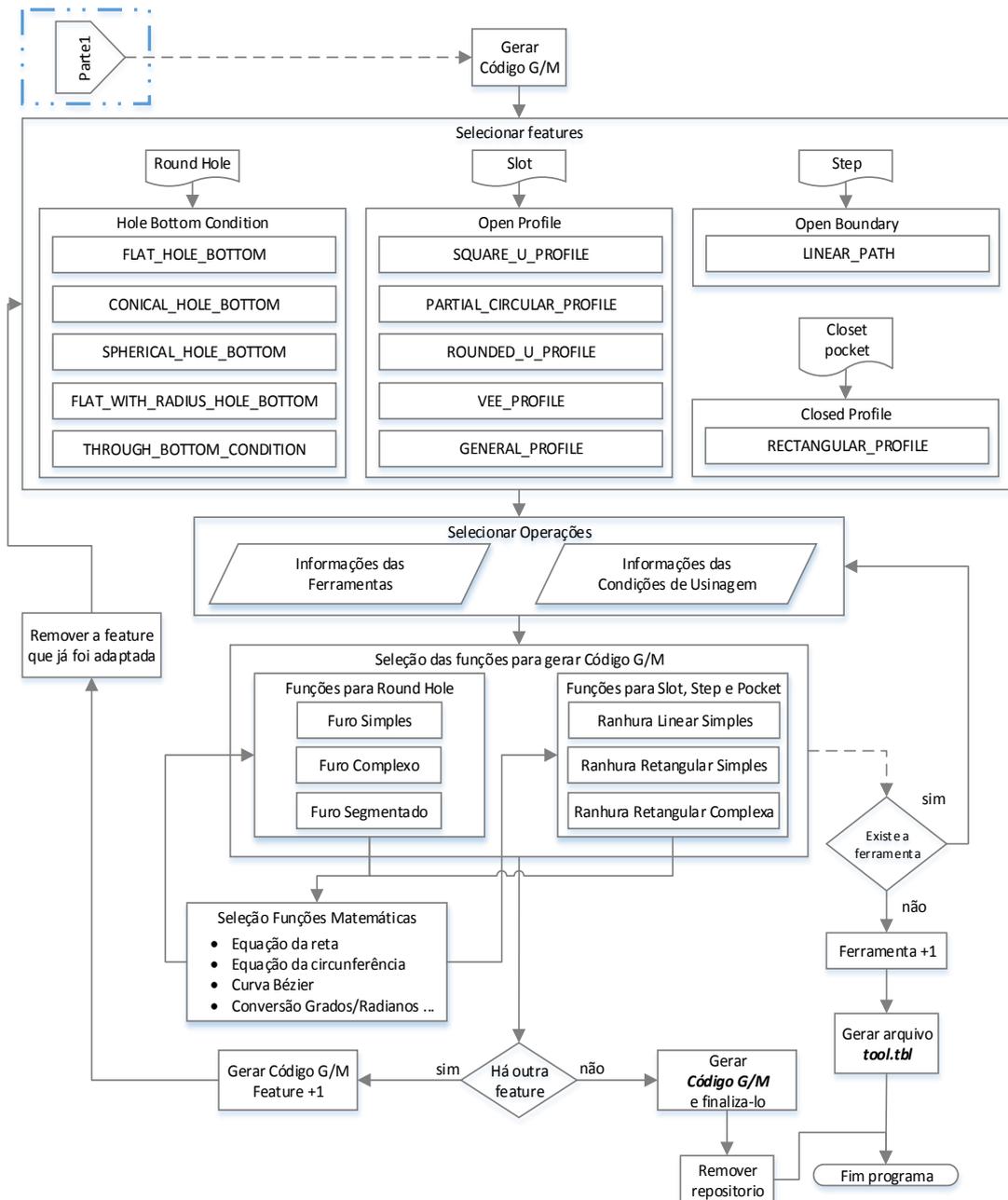


Figura 4.2: Diagrama de fluxo da arquitetura do protótipo do adaptador, parte 2

De acordo com o diagrama de fluxo (parte 1 e 2) e os diagramas IDEFØ (modelagem funcional referente ao adaptador de STEP-NC a Código G), serão implementadas várias classes<sup>3</sup> para a programação do protótipo do adaptador.

#### 4.1. IMPLEMENTAÇÃO DAS CLASSES PARA A EXTRAÇÃO DE INFORMAÇÕES

Para entender melhor a extração das informações do programa de peça em STEP-NC em formato P21 ou STP (ou programa de peça em STEP-NC), usa-se como estudo de caso um programa de peça em STEP-NC de uma peça com uma *feature* (“simples”) tipo furo com base cônica (ver apêndice D

<sup>3</sup> Uma classe é uma abstração que define as características e os comportamentos de um objeto [116].

exemplo a). O arquivo físico P21 da peça projetada gerado pelo *STEP Modeler* é mostrado na Figura 4.3.

O fragmento do programa físico p21 mostrado na Figura 4.3, é usado como estudo de caso. Com isso como exemplo, inicia-se análise da informação contida nele e, portanto, a implementação do protótipo das classes do adaptador de acordo com a modelagem funcional.

Antes de começar com o desenvolvimento do *software* do adaptador, é importante lembrar que a estrutura de dados STEP-NC usa um modelo orientado a objetos para descrever as entidades tais como *workplan*, *workingstep*, assim como para identificar *features*, operações, ferramentas e todas as entidades geométricas. Desta maneira, cada entidade pertence a uma classe de objetos e deve ser utilizada como tal. Isso é feito apontando-se para as partes de cada objeto ao invés de um único registro físico/linha (ver Figura 4.4). As partes da informação são também indexadas em um índice principal de entidades que gerencia a localização de cada informação [27]. As informações são recuperadas e, montadas juntas para posterior uso na geração do código G.

ROSSO et al [2]. e NEWMAN et al. [22] implementaram uma estrutura de dados de acordo com um sistema CAD/CAM com núcleo e estrutura de dados aderentes a STEP-NC. A Figura 4.4 descreve a estrutura de dados implementada.

As informações são armazenadas em estruturas para cada classe de entidade e o programa é guiado pelos planos de trabalho descritos pela entidade *workplan* [2]. Neste projeto utiliza-se o conceito anterior apresentado por ROSSO et al. [2] e NEWMAN et al. [22].

```
ISO-10303-21;
HEADER;
...
ENDSEC;
DATA;
#1=PROJECT('novo projeto',#60,(#2),$,,$,$);
#2=WORKPIECE('workpiece',#7,50.0,$,$,#9,());
#3=AXIS2_PLACEMENT_3D('workpiece placement',#4,#5,#6);
#4=CARTESIAN_POINT('',(0.0,0.0,0.0));
#5=DIRECTION('',(0.0,0.0,1.0));
#6=DIRECTION('',(1.0,0.0,0.0));
#7=MATERIAL('SAE 1020','ACO SEM LIGA',(#8));
#8=NUMERIC_PARAMETER('Hardness',250.0,'HB');
...
#13=DIRECTION('',(0.0,0.0,1.0));
#14=DIRECTION('',(1.0,0.0,0.0));
#15=PLANE('security plane',#11);
#16=AXIS2_PLACEMENT_3D('origin',#17,#18,#19);
#17=CARTESIAN_POINT('',(0.0,0.0,0.0));
...
#21=AXIS2_PLACEMENT_3D('',#22,#23,#24);
#22=CARTESIAN_POINT('',(0.0,0.0,0.0));
#23=DIRECTION('',(0.0,0.0,1.0));
#24=DIRECTION('',(1.0,0.0,0.0));
...
#33=MILLING_MACHINE_FUNCTIONS(.T.,$,,$,.F.,$,(),.T.,$,,$,());
#34=ROUND_HOLE('Furo com base conica',#2,(#25,#48),#35,#43,#44,$,#46);
#35=AXIS2_PLACEMENT_3D('Furo com base conica placement',#36,#37,#38);
...
#51=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#52=CUTTING_COMPONENT(70.0,#53,$,$,$);
#53=MATERIAL('P','CARBIDE',());
...
#58=MACHINING_WORKINGSTEP('ConicalBottomHole_RGH',#15,#34,#48,$);
#59=WORKPLAN('workplan setup plano: XY',(#47,#58),$,#10,$);
#60=WORKPLAN('Main workplan',(#59),$,,$,$);
ENDSEC;
END-ISO-10303-21;
```

Figura 4.3: Programa de peça STEP-NC

O *software* STEP Modeler utiliza a estrutura de dados exposta na Figura 4.4. Portanto, neste trabalho, é usada essa estrutura de dados como referência para identificar e indexar as linhas do programa de peça em STEP-NC e, por fim, extrair as informações contidas em cada entidade.

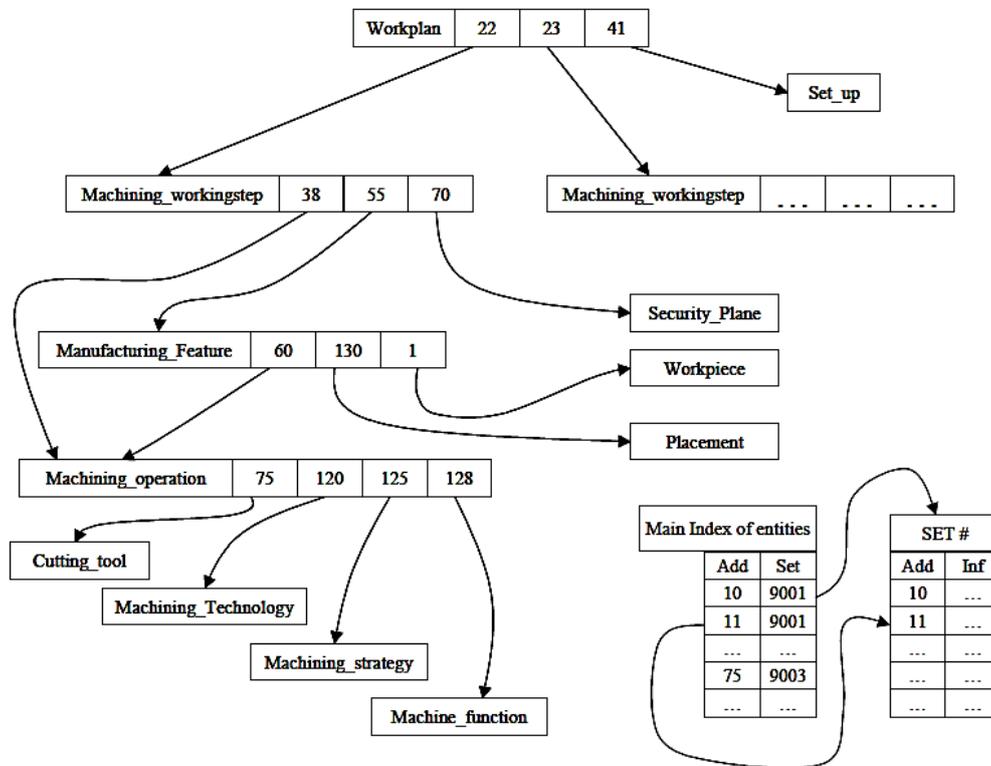


Figura 4.4: Estrutura de dados STEP-NC para CAD/CAM [2].

O desenvolvimento do *software* começa com a criação de uma classe principal chamada “Integrador”. Ela está encarregada de comandar a sequência de execução das classes que compõem a arquitetura do protótipo do adaptador. Em suma, a classe principal “integrador” chama em sequência as seguintes classes:

➤ **IndexEndereçosProg:** esta classe organiza e indexa as linhas do programa STEP-NC, fornecendo uma lista com o endereço inicial de cada linha. Na Figura 4.5 é apresentado graficamente o endereço (em sequência) no qual é assignada uma linha do programa STEP-NC.

| Endereço | Linhas do programa                                     |
|----------|--|
| 1        | #1=PROJECT('novo projeto',#60,(#2),\$,,\$);            |
| 2        | #2=WORKPIECE('workpiece',#7,50.0,\$,\$,#9,());         |
| 3        | #3=AXIS2_PLACEMENT_3D('workpiece placement',#4,#5,#6); |
| 4        | #4=CARTESIAN_POINT('',(0.0,0.0,0.0));                  |
| 5        | ....   |

Figura 4.5: Endereços e linhas do programa.

Com a lista contendo os endereços, pode-se achar as informações em cada uma das entidades, como *workplan*, *machining\_workingstep*, *block* entre outras. Entenda-se que o nome de cada entidade está relacionado com um *token* de identificação (e.g. *token*: *WORKPLAN*, *PROJECT*, etc.) e cada endereço fornece uma linha do programa.

➤ **DadosDaPeça:** esta classe identifica a rota onde se encontra a entidade *block* usando o index das linhas do programa. Essa entidade contém as informações geométricas dimensionais da peça de trabalho. A classe “*DadosDaPeça*” extrai essas informações e as armazena. Na Figura 4.6 é mostrado o mapeamento da entidade *block* de acordo com o programa do estudo de caso (Figura 4.3).

```
#1=PROJECT('novo projeto',#60,(#2),$,,$,$);
#2=WORKPIECE('workpiece',#7,50.0,$,$,#9,());
#9=BLOCK('piece',#3,200.0,150.0,30.0);
```

Figura 4.6: Informações básicas: dados da peça bruta

Na figura anterior, pode-se observar a caixa vermelha referindo-se às dimensões (eixo X: comprimento, eixo Y: altura e eixo Z: largura) da peça de trabalho segundo a entidade *block* descrita na *part 10* da norma ISO14649 (ver sessão 2.6.4).

➤ **PlanoSegurança:** esta classe mapeia a entidade *machining\_workingstep* que faz referência a um plano coordenado onde está posicionado o plano de segurança, i.e. um ponto cartesiano XYZ. Na Figura 4.7, é mostrada a sequência do mapeamento das entidades: #59=*Machining\_workingstep* / #15=*Plane* / #11=*Axis2\_Placement\_3D* / #12=*Cartesian\_Point* de acordo com o programa do caso de estudo da Figura 4.3. Na Figura 4.7, pode-se observar na caixa a informação da posição XYZ do plano de segurança.

```
#1=PROJECT('novo projeto',#60,(#2),$,,$,$);
#60=WORKPLAN('Main workplan',(#59),$,,$,$);
#59=WORKPLAN('workplan setup plano: XY',(#47,#58),$,#10,$);
#47=MACHINING_WORKINGSTEP('ConicalBottomHole_RGH',#15,#34,#48,$);
#15=PLANE('security plane',#11);
#11=AXIS2_PLACEMENT_3D('',#12,#13,#14);
#12=CARTESIAN_POINT('',(0.0,0.0,55.0));
```

Figura 4.7: Informações básicas: plano de segurança

➤ **MapearMachining\_workingstep:** esta classe extrai e salva os dados de uma ou mais *features* de usinagem incluídas no programa STP. As *features* estão acompanhadas com os parâmetros tecnológicos de usinagem, assim como as ferramentas utilizadas para usinar cada *feature*. A informações das *features* e todas a suas características são chamadas de “*informações avançadas*”. Na Figura 4.8 é apresentado o mapeamento de uma *machining\_workingstep*, uma *feature*, uma Ferramenta e a suas Condições de Usinagem.

```
#47=MACHINING_WORKINGSTEP('ConicalBottomHole_RGH',#15,#34,#25,$);
#34=ROUND_HOLE('Furo com base conica',#2,(#25,#48),#35,#43,#44,$,#46);
#25=CENTER_DRILLING($,$,'CenterDrilling',5.0,#26,#27,#32,#33,$,10.0,$,$,$,$);
#27=MILLING_CUTTING_TOOL('center drill',#28,(#30),$,,$,$);
#28=CENTER_DRILL(#29,2,.RIGHT,$,25.0);
#29=MILLING_TOOL_DIMENSION(6.0,0.52359,0.0,10.0,0.0,0.0,0.0);
#32=MILLING_TECHNOLOGY(0.175,.TCP,.11.0,-5.83568,$,.F.,.F.,.F.,$);
#33=MILLING_MACHINE_FUNCTIONS(.T.,$,,$,.F.,$,(),.T.,$,,$,());
#35=AXIS2_PLACEMENT_3D('Furo com base conica placement',#36,#37,#38);
#36=CARTESIAN_POINT('',(100.0,75.0,30.0));
#46=CONICAL_HOLE_BOTTOM(0.34906,$);
```

Figura 4.8: Mapeamento de Informações Avançadas

A Figura 4.8 exibe a *feature Round\_hole*: “Furo com base cônica” e uma só ferramenta: “center drill”. As caixas (retângulos na Figura 4.8) indicam as informações da *feature*, ferramenta e parâmetros tecnológicos de acordo com as especificações da norma ISO14649 parte 10 e 11(ver sessão 2.6.4). Depois que esta classe mapeia uma ou mais *features*, as informações destas são salvas em uma lista (*ArrayList* tipo *String*) chamada: “*InformaçõesAvançadas*”. Essas informações ficam disponíveis para qualquer classe ou método dentro do algoritmo do adaptador.

As classes anteriores representam os níveis A211 e A212 (Figura 3.5 e Figura 3.7 respectivamente) do diagrama IDEFØ e também representam a primeira parte do diagrama de fluxo da arquitetura do adaptador de STEP-NC a Código G (ver Figura 4.1). Antes de implementar o nível A213 ou a segunda parte do diagrama de fluxo, é necessário conhecer algumas características importantes para a programação NC.

#### 4.2. CARACTERÍSTICAS IMPORTANTES REFERENTES À PROGRAMAÇÃO NC

Para implementar o protótipo do adaptador, é preciso antes conhecer alguns conceitos gerais relacionados com os códigos G e códigos M, programação NC e o conteúdo deste último quanto à formatação e às instruções que descreve a sua sintaxe (ver Figura 2.3).

Tabela 4.1: Conceitos relacionados com a programação NC

|                                    |  |
|------------------------------------|--|
| Trajetórias para a ferramenta.     | Movimentação dos eixos XYZ                               |
|                                    | Compensação de raio de corte.                            |
|                                    | Sistema de coordenadas baseado na função G54             |
|                                    | Deslocamento absoluto ou incremental                     |
| Troca manual de ferramenta.        | Atribuído o Sistema de coordenadas baseado na função G55 |
| Parâmetros de corte do fresamento. | Velocidade de corte.                                     |
|                                    | Velocidade de Avanço.                                    |
|                                    | Profundidade de corte ou de passada.                     |

Levando-se em conta os conceitos apresentados na Tabela 4.1, pode-se criar um programa NC para uma máquina-ferramenta CNC usando a linguagem NC (ver sessão 2.4). Deste modo, por exemplo, a partir das informações extraídas do programa do estudo de caso (como foi exposto na sessão anterior), pode-se gerar basicamente um programa da peça em código G.

Além disso, é necessário ter em conta as observações feitas pelos autores SCHÜTZER et al. [68] e OLIVEIRA [69], que expõem que a programação NC depende não só dos conceitos supracitados, senão também do modelo da máquina e do tipo de controlador NC que executa o programa de peça em código G. Assim, as características da máquina-ferramenta (como avanço, máxima profundidade de corte, entre outras da Router CNC desenvolvido neste projeto e descrito no apêndice C) e do controlador EMC (descrito no apêndice B) devem ser levadas em conta para gerar adequadamente um programa de peça em código G pelo adaptador proposto.

Um programa NC ou programa de peça em código G contém instruções (funções ou comandos G) para a movimentação dos eixos XYZ de uma máquina-ferramenta CNC. Geralmente essa movimentação é baseada na interpolação linear (comando G00 ou G01) e na interpolação circular (comando G02 e G03). Essas instruções especificam a posição em que a ferramenta deve passar para

usinar superfícies simples ou complexas [70], em outras palavras, a trajetória em que a ferramenta deve se movimentar para atender a usinagem especificada nos comandos.

Na Figura 4.9 são ilustrados os dois métodos de interpolação encontrados nos sistemas CAM e, em consequência, no programa NC. Na sessão seguinte explicam-se estas interpolações detalhadamente.

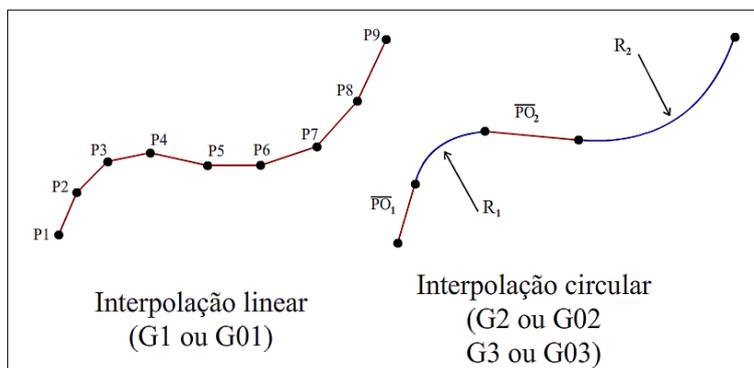


Figura 4.9: Informações Avançadas (Adaptado de [69])

Na criação de um programa de peça em código G ou programa NC, deve ser adotada uma estratégia conforme a adequação ao processo específico de usinagem que sofrerá a peça bruta, a depender da geometria da peça, dos contornos, das áreas planas, côncavas e convexas, ou seja, no contexto do STEP-NC, segundo o tipo de *feature* a ser usinada, as operações usadas e a ferramenta de corte para usiná-la. Para gerar as estratégias de usinagem de uma ferramenta (ou *toolpath*), devem ser usados os códigos G e M, segundo a linguagem NC baseada em [5] e [25].

Nas tabelas 4.2 e 4.3, são apresentados vários dos códigos usados pelo adaptador para criar um programa NC. Os demais códigos podem ser consultados no anexo 1.

Tabela 4.2: Códigos G – Funções preparatórias

| Código   | Descrição  | Parâmetros ou Ação                                    |
|----------|--|---|
| G0       | Deslocamento rápido                                      | Eixos XYZ   |
| G1       | Deslocamento com avanço programado - interpolação lineal | Eixos XYZ-F   |
| G2, G3   | Interpolação circular (Horário e Anti-horário)           | Eixos XY-IJK ou XY-R                                  |
| G17      | Selecionar Plano de trabalho                             | Plano XY  |
| G21      | Unidade de Medida  | Milímetros  |
| G40      | Cancelar compensação de raio de corte                    | Deslocamento normal                                   |
| G41, G42 | Ativar compensação de raio de corte (interno, externo)   | Deslocamento compensando raio de ferramenta           |
| G54-G55  | Selecionar sistema de coordenadas                        | 54 plano de trabalho, 55 Plano de troca de ferramenta |
| G90, G91 | Modo Coordenadas Distancia                               | 90 Absoluto, 91 incremental                           |

Tabela 4.3: Códigos G – Funções miscelâneas e outros códigos auxiliares

| Código  | Descrição  | Parâmetros ou Ação   |
|---------|--|--|
| M03/M04 | Ativar o Spindle   | Sentido CW/ Sentido CCW  |
| M05     | Parar o Spindle  | Stop   |
| M06     | Troca de ferramenta manual   | Parar programa temporalmente, espera confirmação                 |
| M30/M02 | Fim programa NC  | Finalizar execução programa NC                                   |
| F       | Velocidade de avanço   | mm/mim   |
| S       | Velocidade de rotação do <i>Spindle</i> ou velocidade de corte (G96) | rot/min  |
| T       | Seleção de ferramenta  | Número de Ferramenta   |
| D       | Identificador de ferramenta  | Número de identificador de ferramenta para a compensação de raio |

#### 4.2.1. Descrição das funções preparatórias e miscelâneas

Para gerar um programa NC, é necessário, primeiramente, conhecer a linguagem NC descrita pelas normas ISO6983 e RS274. Os códigos que são descritos a seguir são os principais códigos utilizados na programação NC. É essencial saber a utilização correta desses para evitar a redundância de linhas de programação.

Inicialmente expõe-se a linguagem NC referente aos tipos de interpolação, e depois são descritas a compensação de raio de corte, a troca manual de ferramenta, o sistema de coordenadas e, por último, o modo distância ou modo de deslocamento.

- **Interpolação Linear**

Uma interpolação linear só precisa de um ponto de origem qualquer e um ponto final. Nesta interpolação trabalha-se com os eixos XYZ. Na linguagem NC há dois códigos para realizar uma interpolação linear: os códigos G00 e G01. Na Figura 4.10 apresenta-se um exemplo do código G00 e uma representação gráfica do mesmo.

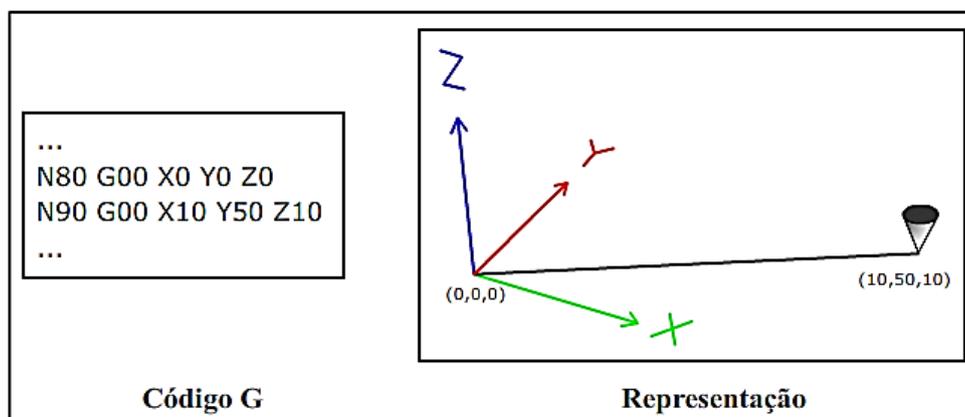


Figura 4.10: Interpolação linear (Adaptado de [71])

O código G00 produz um deslocamento linear rápido dos eixos da máquina de acordo com a sua velocidade máxima de avanço, enquanto o código G01 produz um deslocamento dos eixos da máquina com uma velocidade de avanço programada através do comando F.

- **Interpolação Circular**

A interpolação circular é mais complexa que a linear, pois os parâmetros para gerá-la são mais intrincados. A geração de trajetória com interpolação circular só funciona tendo como referência o plano de usinagem em duas dimensões, por exemplo os planos XY, XZ o YZ [71]. Neste projeto é usado o plano de usinagem XY.

Um arco circular ou helicoidal é especificado usando o comando G02 (arco no sentido horário - CW) ou G03 (arco anti-horário CCW). O sentido do giro (CW, CCW) é visto a partir da extremidade positiva do eixo em torno do qual o movimento circular ocorre.

Existem dois tipos de formatos permitidos para especificar um arco, são elas:

**a. Formato do centro do arco:** este é o formato preferido em programas NC executados pelo EMC pela facilidade de processamento. Neste formato utilizam-se os seguintes parâmetros usando como referência o plano de trabalho XY:

**Eixos- G02 ou G03 I- J-**

Esses parâmetros descrevem uma trajetória de arco tomando como referência o centro do mesmo. O cálculo de arcos feitos à mão pode ser uma tarefa difícil às vezes. Uma opção é desenhar o arco com ajuda de um programa CAD para obter as coordenadas e as compensações. Na Figura 4.11 mostram-se dois exemplos de interpolação circular usando o formato do centro do arco.

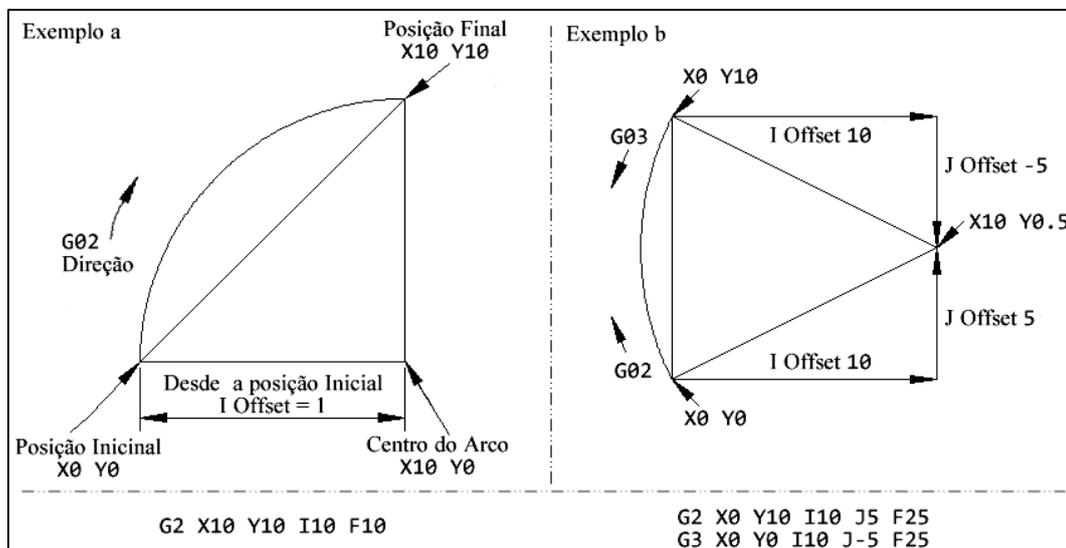


Figura 4.11: Interpolação circular (Adaptado de [72])

**b. Formato do raio do arco:** neste formato as coordenadas do ponto final do arco no plano de trabalho selecionado são especificadas junto com o raio do arco [72].

Os parâmetros necessários para o formato de raio do arco são:

### Eixos- G02 ou G03 R-

Pode-se criar um arco com este formato de maneira similar aos exemplos da Figura 4.11, onde, ao invés de colocar os pontos IJK, coloca-se o raio (R) do arco. Na Figura 4.12, pode-se ver a diferença entre os dois formatos.

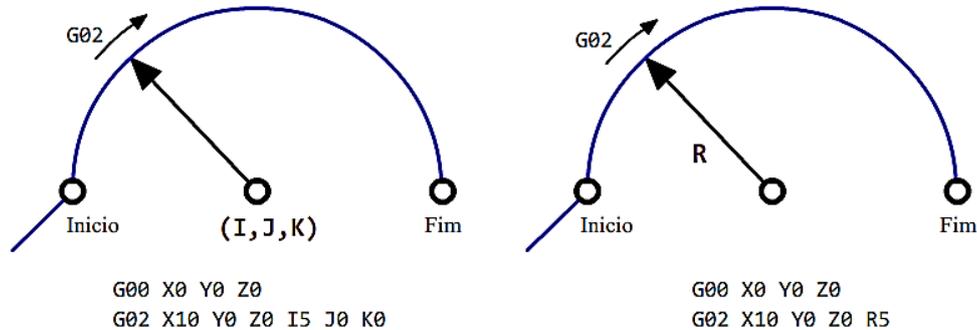


Figura 4.12: Diferenças entre os formatos de interpolação circular (Adaptado de [71])

- **Compensação de raio da ferramenta de corte**

A compensação de raio de corte permite que o programador planeje a trajetória da ferramenta sem saber o diâmetro exato da ferramenta. Quando a compensação é ativada, deve existir pelo menos uma distância tão longa como o raio da quina da ferramenta sendo usada nesse momento. Esta é uma limitação no uso da compensação de raio de corte.

Existem duas trajetórias possíveis que pode ter a ferramenta, enquanto a compensação de raio de corte estiver ativa: a trajetória compensada pode ser à esquerda ou à direita. Os códigos referentes à ativação da compensação de raio de corte são o G41 para o lado esquerdo da trajetória original e o G42 para o lado direito da trajetória original.

O ponto final de cada movimento depende do próximo movimento. Se o movimento seguinte cria uma quina exterior, esta quina será o ponto final da trajetória compensada. Se o movimento seguinte gera uma quina interna, o ponto final da trajetória estará antes de estragar alguma parte da peça. A Figura 4.13 ilustra como são as trajetórias compensadas (esquerda e direita) de uma trajetória exemplo.

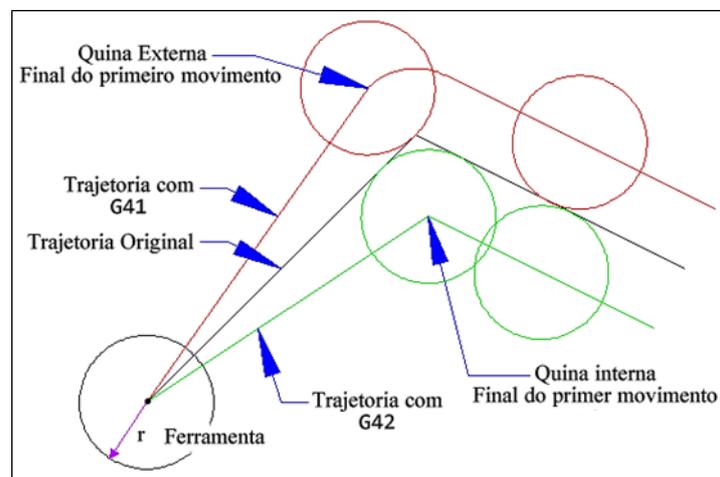


Figura 4.13: Compensação de raio de corte (Adaptado de [72])

Um elemento importante para usar o G41 e o G42 é a tabela de ferramentas, pois a compensação de raio de corte usa os dados contidos na tabela para determinar o *Offset* necessário a fim de compensar a trajetória.

- **Sistema de coordenadas**

Os sistemas de coordenadas são identificados com os códigos G54 e G55, entre outros. Eles definem um ponto coordenado onde iniciam os eixos da máquina, nesse caso os eixos XYZ. Este ponto é a referência para o deslocamento dos eixos dependendo do modo de coordenadas de distância, absoluto G90 ou incremental G91.

Este sistema pode ter vários usos, por exemplo um programa NC que descreve a usinagem de uma *feature* que se repete, ao invés de mudar as coordenadas da trajetória da ferramenta, só se muda o sistema de coordenadas e o mesmo programa é repetido. A Figura 4.14 ilustra um exemplo desta situação.

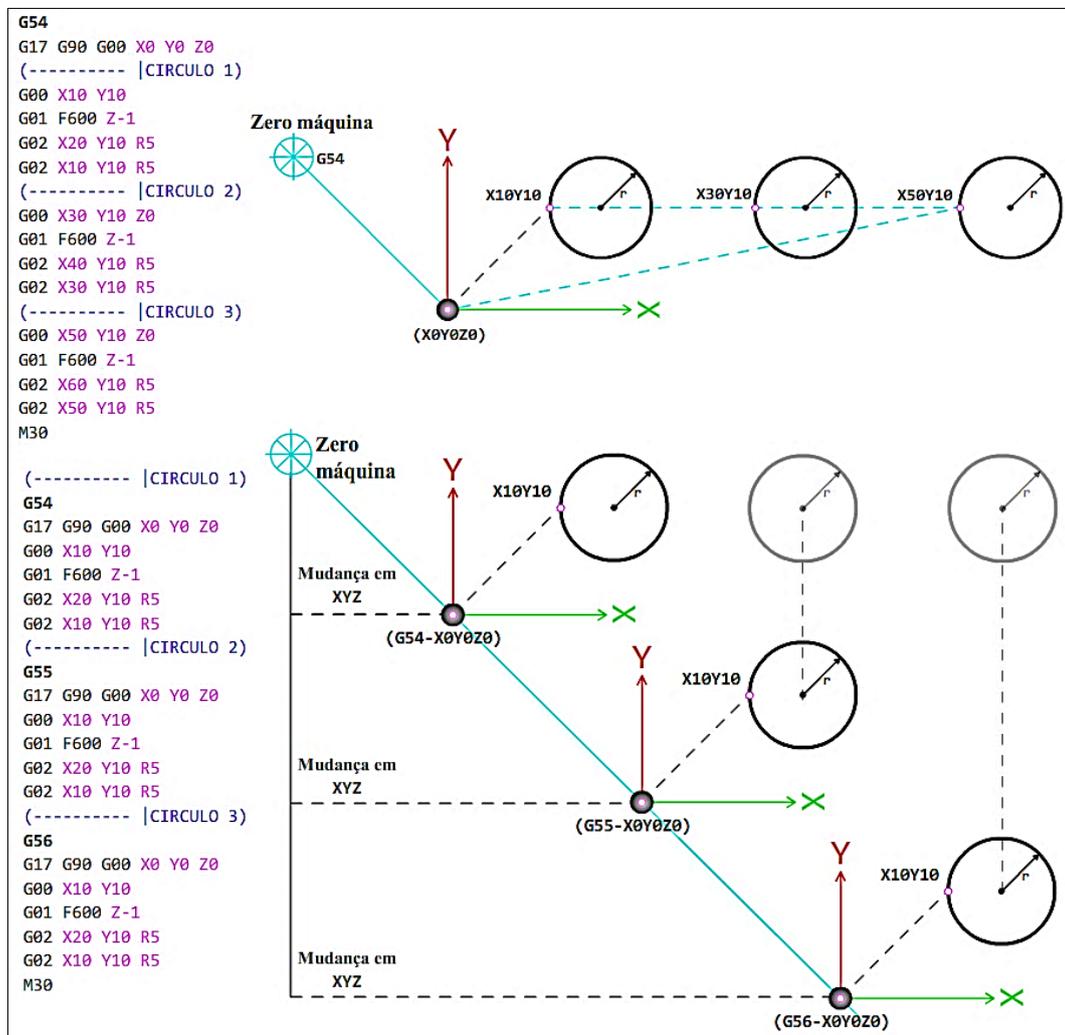


Figura 4.14: Sistemas de coordenadas

O adaptador proposto utiliza o sistema de coordenadas G54 para identificar ponto coordenado XYZ onde fica o zero da peça e o sistema de coordenadas G55 para o ponto de troca de ferramenta manual.

- **Modo coordenadas de distância**

O modo de coordenadas é a movimentação ou a distância que se percorre de uma coordenada a outra. Existem duas funções preparatórias para estabelecer o modo de distância: G90 para coordenadas absolutas e G91 para coordenadas incrementais. As coordenadas absolutas são usadas para descrever uma movimentação partindo desde um único ponto de origem, que é X0, Y0 e Z0 [73]. As coordenadas incrementais são usadas para descrever uma movimentação que tem seu ponto de partida a partir do último ponto coordenado, em outras palavras, este último ponto seria o novo ponto zero X0Y0Z0. Na Figura 4.15 são apresentados dois exemplos de modo de coordenadas de distância, um com G90 e o outro com G91.

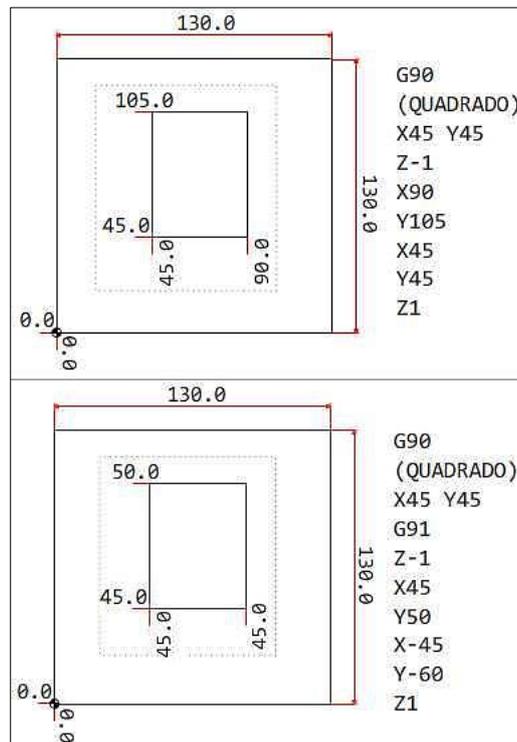


Figura 4.15: Coordenadas de distância G90 e G91 (Adaptador de [73])

A seguir, apresenta-se a função miscelânea utilizada para a troca de ferramenta. As demais funções preparatórias e miscelâneas estão referenciadas nas tabelas do Anexo 1.

- **Troca manual de ferramenta**

A troca manual de ferramenta é uma tarefa realizada pelo usuário da máquina de forma mecânica. Trocar a ferramenta permite usinar distintos tipos de rasgos numa peça com num único programa NC. A linguagem NC contém uma função miscelânea que pausa momentaneamente a execução de um programa NC. Isso deixa um espaço de tempo para que o operário da máquina CNC possa trocar manualmente a ferramenta no fuso (Ver Figura B.16 do apêndice B).

Para ativar a troca de ferramenta manual em um programa NC segundo a norma RS274NGC (ver sessão 2.4), deve-se usar o código M6 mais um parâmetro T acompanhado de um número de identificação da ferramenta. Isto serve para selecionar a ferramenta a ser usada.

O controlador EMC inclui um componente que é ativado com o código M6 chamado *hal\_manualtoolchange* [74]. Este componente é uma janela de aviso que é mostrada na GUI do

EMC, ao mesmo tempo pausa a execução do programa NC, ficando na espera da confirmação da troca da ferramenta para continuar com a execução. A Figura 4.16 mostra a janela de confirmação de troca de ferramenta enquanto o programa NC está pausado.

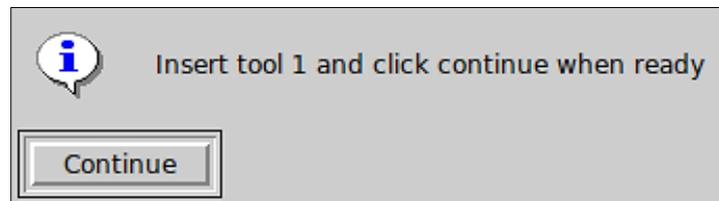


Figura 4.16: Janela: Troca manual de ferramenta [74]

No adaptador proposto é utilizado o sistema de coordenadas G55 para selecionar um ponto coordenado fixo XYZ onde é feita a troca de ferramenta manual.

Depois de descrever alguns dos códigos necessários para criar um programa NC, prossegue-se com a análise geométrica de algumas *features*, as fórmulas matemáticas que descrevem geometricamente uma *feature* e, por fim, a descrição das classes programadas para a geração das trajetórias da ferramenta para o programa NC.

#### 4.2.2. Segmentação geométrica de *features* complexas

A seguir são analisadas as geometrias de algumas das *features* dispostas no *STEP Modeler*. A geometria de algumas *features* são mais complexas que outras em termos de usinagem, sendo, portanto, necessário segmentar uma *feature* complexa em outras mais “simples”.

O autor do *software* *STEP Modeler* [11] implementou algumas regras que foram seguidas para o mapeamento das diferentes *features* de projeto para *machining\_workingsteps*. Seguindo essas regras, pode-se segmentar geometricamente uma *feature* “complexa” em outras mais “simples” de acordo a seus *workingsteps* (WS). Nas figuras 4.17 a 4.23, ilustra-se esse conceito.

- **Feature: Furo com base plana**

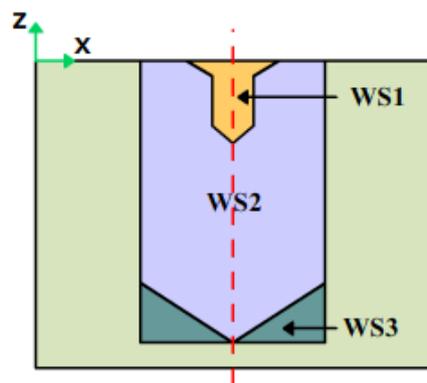


Figura 4.17: *Workingsteps* de um furo com base plana (Adaptado de [11])

- **Feature: Furo com base arredondada**

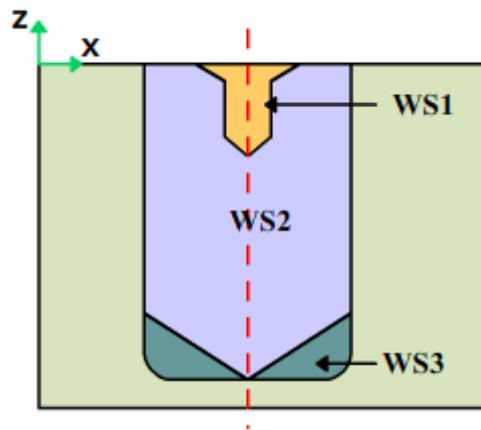


Figura 4.18: *Workingsteps* de um furo com base arredondada (Adaptado de [11])

- **Feature: Furo com base esférica**

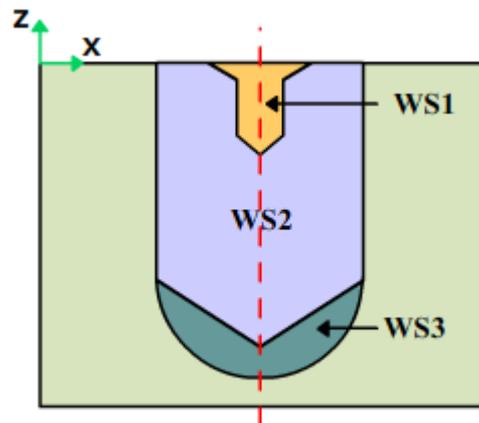


Figura 4.19: *Workingsteps* de um furo com base esférica (Adaptado de [11])

- **Feature: Ranhura perfil quadrado U**

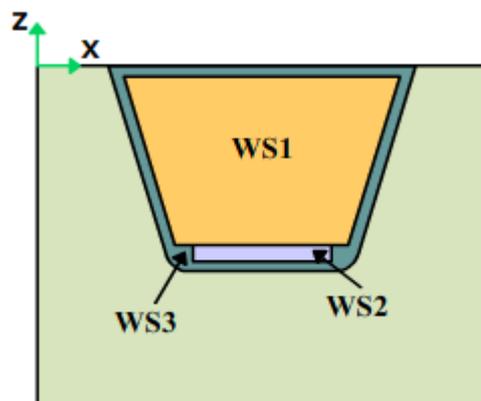


Figura 4.20: *Workingsteps* de uma ranhura perfil quadrado U (Adaptado de [11])

- **Feature: Ranhura perfil parcial circular**

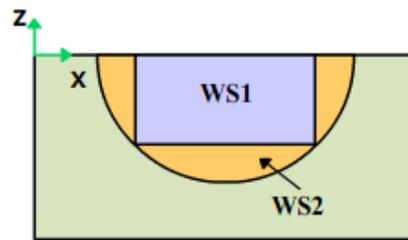


Figura 4.21: *Workingsteps* de uma ranhura perfil parcial circular (Adaptado de [11])

- **Feature: Cavidade com fundo plano**

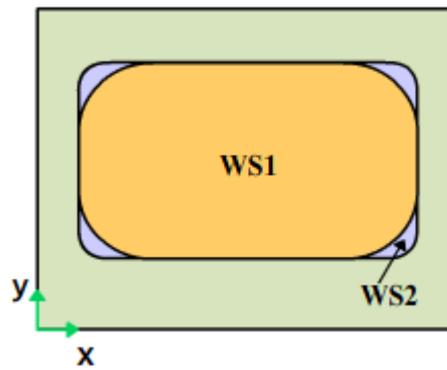


Figura 4.22: *Workingsteps* de uma cavidade com fundo plano (Adaptado de [11]).

- **Feature: Cavidade com fundo arredondado**

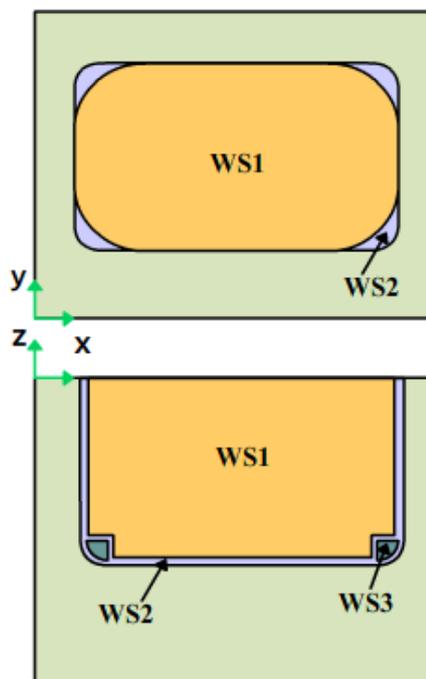


Figura 4.23: *Workingsteps* de uma cavidade com fundo arredondado (Adaptado de [11])

Das figuras 4.17 até a 4.23, nota-se que cada *workingstep* (WSx) refere-se a uma *feature* “simples” que possui uma operação e uma ferramenta. Portanto, cada *feature* simples leva associada a si uma trajetória da ferramenta que descreve a sua geometria.

A trajetória de ferramenta que descreve a geometria de uma *feature* deve ser implementada para os planos de trabalho XY, XZ e YZ. O plano de trabalho XY refere-se à geometria frontal da *feature* e os planos XZ e/ou YZ refere-se à geometria lateral e profundidade da *feature* (ver Figura 4.24). A geração de trajetórias de ferramenta é assunto da sessão seguinte.

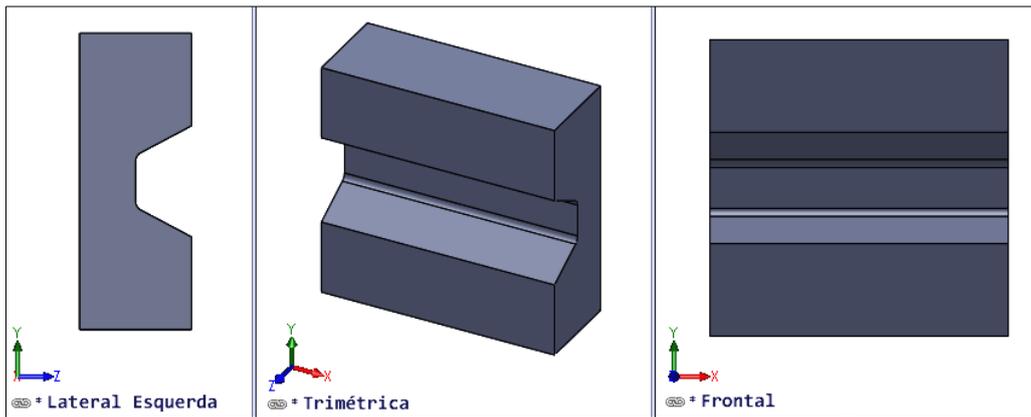


Figura 4.24: Planos de trabalho

### 4.2.3. Geração de trajetórias de ferramenta (*Toolpath*)

A trajetória das ferramentas ou percurso das ferramentas ou ainda o caminho das ferramentas são algumas traduções da expressão inglesa *toolpath*. Portanto, neste documento usa-se esta expressão ou algumas das traduções citadas para tratar o grupo de movimentos a ser executado pela ferramenta de corte que usina uma determinada superfície ou, neste caso, uma determinada *feature* segundo a programação NC.

Este trabalho pressupõe que as geometrias das *features* são do tipo 2½D (*Two5D* em inglês). Para usinar este tipo de *feature*, utiliza-se uma ferramenta rotativa que percorre um caminho definido onde há apenas uma variação da altura (profundidade). O objetivo final é a remoção de cavaco, também conhecida como desbaste.

O autor do trabalho (*STEP Modeler*) [11] deixa disponível quatro tipos gerais de geometrias (*features*), a saber: Furos, Ranhuras, Degraus e Cavidades. Estas geometrias estão dentro da modelagem 2½D.

Segundo OLIVEIRA [75], desde a década de 1990, existem vários autores que se dedicaram a desenvolver diversos tipos de algoritmos para a geração de *toolpaths* de cavidades 2½D visto que este é um tipo comum de geometria nas peças mecânicas fabricadas por usinagem.

Existem geralmente duas etapas na usinagem por fresamento, sendo elas: etapa de desbaste e etapa de acabamento. O *software* *STEP Modeler* gera o programa de peça em STEP-NC com informações apenas para desbaste e, além disso, não contém informações sobre o *toolpath*.

O *toolpath* num programa de peça em STEP-NC é gerado levando-se em conta uma série de informações como *features* de usinagem, estratégia de usinagem, *setup* e informações do plano de segurança.

Existem dois tipos de *toolpath* segundo a norma ISO14649, são eles: Auxiliar (*auxiliary tool-path*) e de Usinagem (*Machining tool-path*). Note-se que, segundo a norma ISO14649, a especificação do *toolpath* é opcional.

O programa de peça em código G que será gerado pelo protótipo de adaptador de STEP-NC a Código G/M incluirá, portanto, apenas a etapa de desbaste.

Quando se usam planos de corte na etapa de desbaste com uma fresa de topo (isto é, plana) para usinar uma *feature* na sua geometria, ficam degraus que só podem ser tirados com o acabamento [75]. A figura 4.25 ilustra este problema.

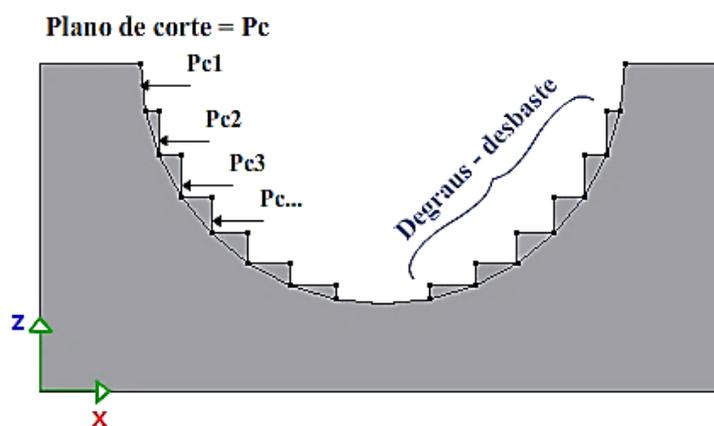


Figura 4.25: Plano de trabalho XZ – A etapa de desbaste gera degraus

Neste trabalho serão apresentados alguns tipos de *toolpath* para desbaste e a suas características. Segundo COSTA [75], os tipos de *toolpaths* podem ser organizados em 4 grupos.

**a. Ziguezague:** Trata-se de deslocamentos da ferramenta rotativa em trajetórias paralelas com um afastamento geralmente uniforme. Este tipo de *toolpath* é muito utilizado em operações de desbaste onde são usadas ferramentas de maior diâmetro e não há grande preocupação com a precisão ou acabamento da superfície [75]. A Figura 4.26 mostra a forma deste *toolpath*.



Figura 4.26: *Toolpath* - Ziguezague

**b. Espiral retangular:** Com o uso voltado para cavidades, este tipo de *toolpath* tem a forma de uma espiral. A usinagem é realizada do centro para o contorno, com trajetórias igualmente espaçadas. A Figura 4.27 mostra a forma deste *toolpath*.

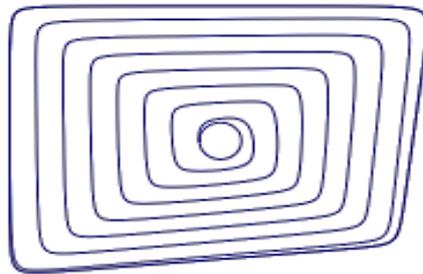


Figura 4.27: *Toolpath* - Espiral retangular com quina circular (Adaptado de [76])

**c. Isoparamétrico:** Usado em superfícies paramétricas, este tipo de *toolpath* é formado por trajetórias em ziguezague criadas através da análise paramétrica da superfície. Para algumas superfícies, este tipo tem o inconveniente de passar por regiões já usinadas devido ao fato que o espaçamento das trajetórias deve ser determinado considerando o diâmetro da ferramenta e a superfície não tem uma forma regular [75]. A Figura 4.28 mostra a forma deste *toolpath*.

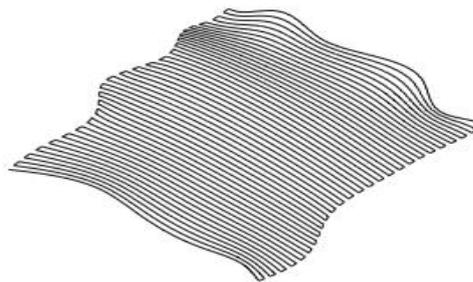


Figura 4.28: *Toolpath* - Isoparamétrico [77].

**d. Preenchimento (*Space-filling*):** O *toolpath* de preenchimento é uma abordagem relacionada ao preenchimento de uma superfície usando as curvas *Hilbert's* [78]. Tais curvas tem sua complexidade adaptada à complexidade da superfície da peça. Este método minimiza o tempo de movimentação da ferramenta sem cortar material e o número de reentradas da ferramenta sobre a peça. A Figura 4.29 mostra um *toolpath* baseado nas curvas *Hilbert*.

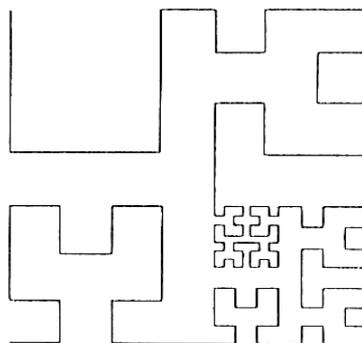


Figura 4.29: *Toolpath* - Preenchimento usando curvas *Hilbert's* [75]

Já a norma ISO14649-11 expõe que para criar um *toolpath* em 2½D, há disponível oito tipos de estratégias (Tabela 4.4).

Tabela 4.4: *Toolpaths* 2½D dispostos na norma ISO14649-11

| Estratégias para fresamento Two5D |                              |
|-----------------------------------|------------------------------|
| <i>Unidirectional</i>             | <i>Contour_bidirectional</i> |
| <i>Bidirectional</i>              | <i>Contour_spiral</i>        |
| <i>Contour_parallel</i>           | <i>Center_milling</i>        |
| <i>Bidirectional_contour</i>      | <i>Explicit_strategy</i>     |

Das estratégias anteriores, é descrita apenas o *Contour\_parallel*, pois ela é escolhida para gerar algumas das trajetórias de ferramenta para usinagem das *features* segundo o *STEP Modeler*. A Figura 4.30 ilustra esta estratégia.

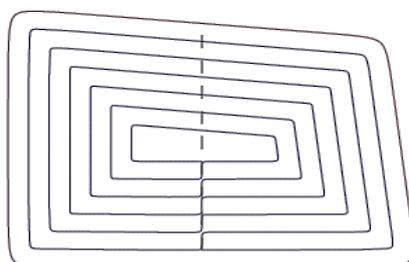


Figura 4.30: *Toolpath* - *Contour parallel* [76]

Assim como existem estes *toolpath* existem muitos outros que não são mencionados neste texto porque obedecem a geometrias mais complexas ou não são aplicáveis na geração de trajetórias das geometrias tratadas neste texto (2½D).

Por outro lado, CHOI [79] propõe um sistema de reconhecimento de forma ou superfícies usináveis a partir das quais são determinados os tipos de geometria e, conseqüentemente, gerar a trajetória de ferramenta. Neste trabalho, é usada a ideia proposta no sistema de [75] para gerar os *toolpaths* partindo do reconhecimento das *features* disponíveis no *STEP Modeler* e das informações extraídas do arquivo de peça em STEP-NC.

#### 4.2.4. Trajetória de ferramenta no plano XY

Os *toolpath* gerados no plano XY devem descrever geometrias de tipo furo, ranhura, degrau e cavidade. Algumas dessas geometrias possuem características similares, o que permite agrupá-las em três conjuntos:

**a. Furos:** os *toolpath* para os furos no plano de trabalho XY são definidos pela relação de características entre o diâmetro da ferramenta e o diâmetro do furo. Existem duas opções para gerar o *toolpath* de um furo:

- Quando o diâmetro da ferramenta é igual ao diâmetro do furo, não é preciso gerar uma trajetória de desbaste no plano XY, mas deve-se avançar no plano Z para a atingir a profundidade.
- Quando o diâmetro da ferramenta é menor do que o diâmetro do furo, é necessário gerar uma trajetória de devaste no plano XY, além de atender a profundidade desejada.

A Figura 4.31 mostra dos tipos de *toolpath* para *features* tipo furo.

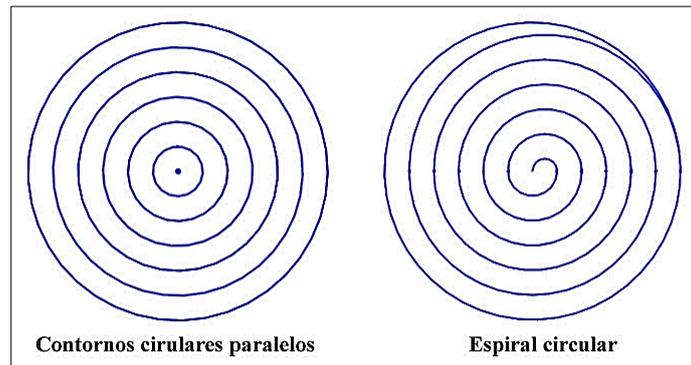


Figura 4.31: *Toolpath* para furos (adaptado de [76]).

**b. Ranhuras e Degraus:** o *toolpath* para esses tipos de geometrias no plano de trabalho XY é definido pela relação de características entre o diâmetro da ferramenta, o comprimento da *feature* e largura da *feature*. Deve-se levar em conta que as extremidades da *feature* são abertas tal como é mostrado na Figura 4.32c. Existem duas opções para gerar o *toolpath* de uma geometria quadrada ou retangular como as ranhuras e/ou degraus, as quais são:

- Quando o diâmetro da ferramenta é igual ao comprimento da largura da *feature* (ranhura ou degrau), é necessário gerar apenas uma trajetória linear horizontal ou vertical segundo seja o caso. A profundidade no plano XZ (horizontal) ou YZ (vertical) é o complemento da *feature* (Figura 4.33).
- Quando o diâmetro da ferramenta é menor que o comprimento da largura da *feature* (ranhura ou degrau), é necessário gerar uma trajetória de ferramenta para o desbaste total dela no plano frontal. As trajetórias úteis para estas *features* podem ser contornos paralelos, espiral ou ziguezague.

A Figura 4.32 ilustra os tipos de *toolpath* para o desbaste de uma ranhura ou degrau.

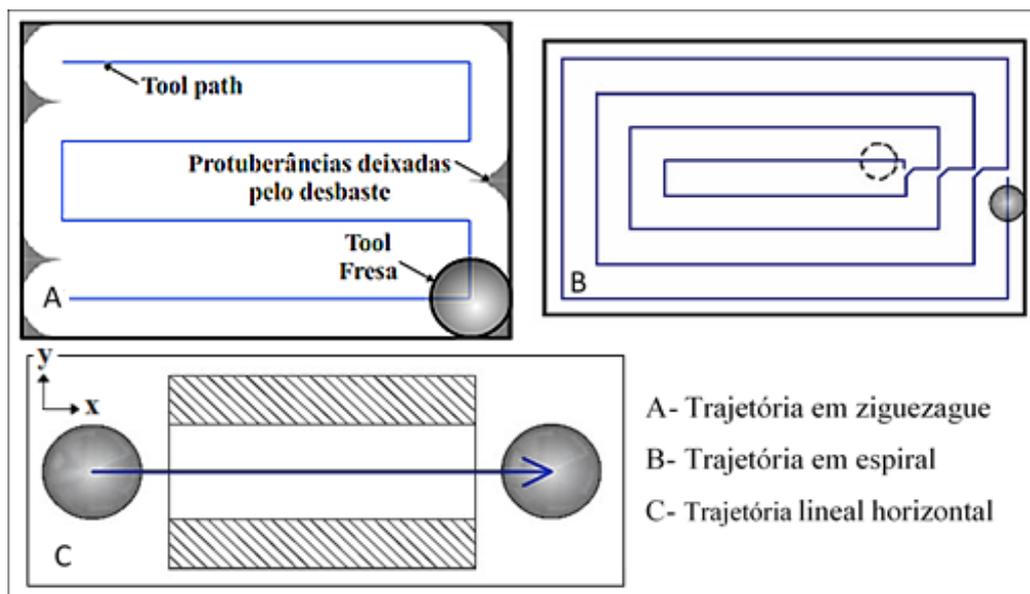


Figura 4.32: *Toolpath* para Ranhuras e Degraus (adaptado A, B de [80] e C de [39])

**c. Cavidades:** o *toolpath* para esses tipos de geometrias no plano de trabalho XY é definido pela relação de características entre o diâmetro da ferramenta, o comprimento, a largura, o raio da circunferência da quina e o raio da base (se houver) da *feature*. Tecnicamente existem várias opções para gerar o *toolpath* de uma geometria tipo cavidade fechada (*closed pocket*), de igual forma como as ranhuras e degraus, mas nesta *feature* deve-se considerar que as quinas possuem um raio ortogonal [36]. As opções para gerar as trajetórias usando uma ou duas ferramentas se dão:

- Quando o diâmetro da ferramenta é igual ao raio ortogonal (quinas da *feature*), é necessário gerar apenas as trajetórias do tipo contorno paralelo, espiral ou zig-zague sem levar em conta a quina da *feature*. Quando o raio da ferramenta é o mesmo da quina da *feature* a trajetória fecha a cavidade.
- Quando o diâmetro da ferramenta é menor que o raio ortogonal (quinas da *feature*), é necessário gerar dois tipos de trajetórias. A primeira é uma trajetória para desbaste que não deve ultrapassar um ponto intermediário entre o raio da ferramenta e o raio ortogonal da quina da *feature*. A segunda trajetória é do tipo contorno paralelo e deve levar em conta o raio ortogonal da *feature* e o raio da ferramenta para fazer o desbaste da geometria desejada.
- Quando o diâmetro da ferramenta é maior que o raio ortogonal (quinas da *feature*), é necessário gerar dois tipos de trajetórias usando duas ferramentas. A primeira trajetória usará a ferramenta de raio maior sendo para desbaste sem ultrapassar o ponto intermediário entre o raio da ferramenta e o raio ortogonal da quina da *feature*. A segunda trajetória usará uma ferramenta de igual ou menor raio que o raio ortogonal da *feature*. Esta última trajetória é do tipo contorno paralelo e deve levar em conta o raio ortogonal da *feature* e o raio da ferramenta para fazer o desbaste.

#### 4.2.5. Equações para cálculo de trajetórias no plano XZ e/ou YZ

Das *features* explicadas na sessão 4.2.2, pode-se observar que, no plano XZ, elas possuem geometricamente formas triangulares básicas, circulares ou uma mistura das duas. Estas formas podem ser modeladas matematicamente usando geometria plana.

A Figura 4.33 mostra os planos de trabalho de uma peça exemplo.

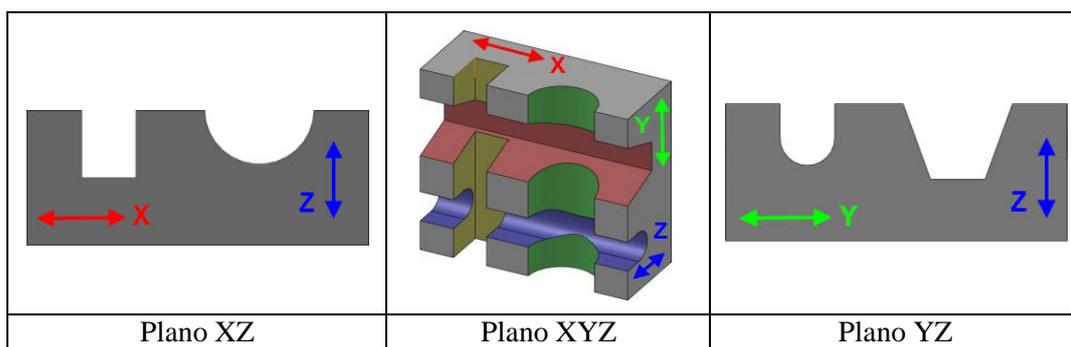


Figura 4.33: Vistas dos planos de trabalho.

Nesta sessão considera-se a necessidade de descrever os métodos matemáticos e geométricos nos quais se apoia o protótipo do adaptador para resolver e obter a representação gráfica das instruções implicadas na geração das trajetórias da ferramenta para a usinagem *two5D*.

Para descrever a trajetória que deve seguir a ferramenta no plano XZ ou YZ, é necessário analisar a *feature* de estudo geometricamente e descrevê-la por meio de equações matemáticas. A seguir são apresentadas diferentes funções ou equações matemáticas usadas na descrição de *features* simples e, portanto, no cálculo de trajetória de ferramenta de corte.

Na Tabela 4.5, é mostrado o tipo de função matemática envolvida e a sua característica.

Tabela 4.5: Equações para descrever a geometria de uma *feature* simples

| Matemática envolvida  | Característica            |
|---|---------------------------|
| Equação da reta   | Forma reduzida            |
| Teorema de Pitágoras  | Triângulo retângulo       |
| Equação das relações trigonométricas do triângulo retângulo | Aplicada à circunferência |

As equações indicadas na Tabela 4.5 são usadas para modelar, por exemplo, uma *feature* tipo *Ranhura com perfil quadrado U* (*feature* disponibilizada no *software STEP Modeler*).

A Figura 4.34 mostra a decomposição da *feature* (*Ranhura com perfil quadrado U*) em diferentes partes geométricas (“geometrias simples”), sendo modeladas matematicamente usando a equação da reta, do triângulo retângulo e da semicircunferência.

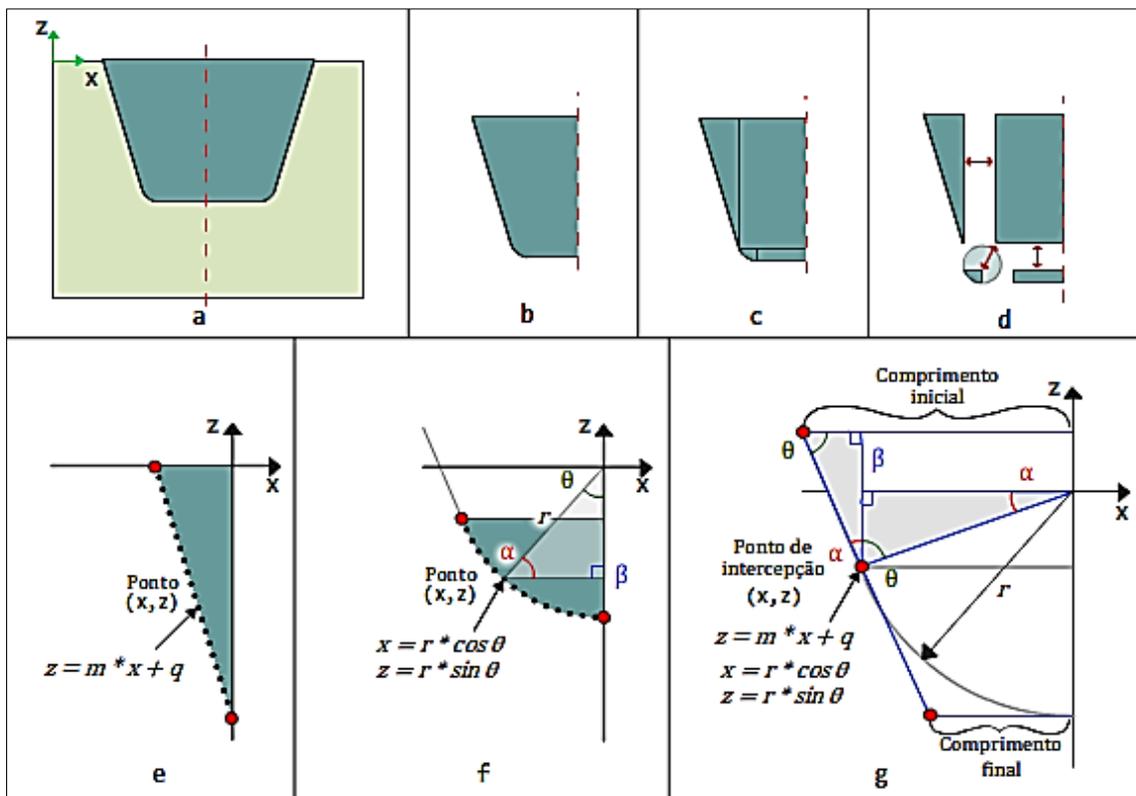


Figura 4.34: Decomposição e análises matemático/geométrico da *feature* tipo ranhura com perfil quadrado U

Na Tabela 4.6 descreve-se cada subdivisão da Figura 4.33.

Tabela 4.6: Seções da ranhura com perfil quadrado U, Figura 4.34

| Seção    | Descrição   |
|----------|---|
| <b>a</b> | <i>Feature</i> tipo ranhura com perfil quadrado U completa.                                 |
| <b>b</b> | Corte à metade da <i>feature</i> .  |
| <b>c</b> | Traços das geometrias que compõem a metade da <i>feature</i> .                              |
| <b>d</b> | Decomposição da geometria da <i>feature</i> em geometrias simples.                          |
| <b>e</b> | Descrição matemática de um ponto que compõe a reta inclinada                                |
| <b>f</b> | Descrição matemática de um ponto que compõe a semicircunferência                            |
| <b>g</b> | Descrição matemática do ponto de intercessão entre a reta inclinada e a semicircunferência. |

No Anexo 7 são descritas as equações usadas para calcular a posição de um ponto coordenado no plano XZ ou YZ correspondente a uma geometria simples.

A seguir é realizado um exemplo em que são usadas algumas das funções matemáticas (equações) descritas no Anexo 7, isto é baseado na *feature* da Figura 4.34 tendo como parâmetros os dados da Tabela 4.7.

Tabela 4.7: Características da *feature* de exemplo.

| Caraterística                      | Valor (mm) |
|------------------------------------|------------|
| Comprimento superior ( $C_{sup}$ ) | 20         |
| Comprimento inferior ( $C_{inf}$ ) | 20         |
| Ângulo da base (raio - $r$ )       | 2          |
| Profundidade ( $prof$ )            | 7          |

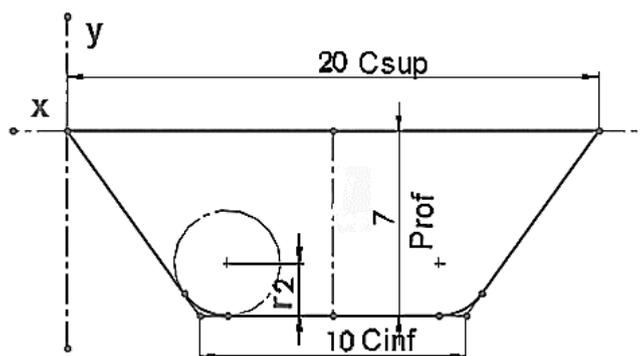


Figura 4.35: Características da *feature* de exemplo (*feature* tipo ranhura com perfil quadrado U)

Primeiramente a *feature* é separada em partes mais simples como mostrado na figura 4.36. Depois modela-se matematicamente cada uma dessas partes.

A modelagem matemática de cada parte geométrica da *feature* mostrada na Figura 4.36 é baseada nos seguintes componentes matemáticos:

- Reta inclinada + triângulo de Pitágoras.

- Ponto de interseção entre a semicircunferência- reta inclinada
- Semicircunferência

O primeiro dos cálculos refere-se ao cálculos da reta inclinada por meio da equação reduzida da reta com o fim de saber o valor do ponto de X variando o valor de Y. Isso fornece os primeiros parâmetros para segmentar a *feature* e criar o código G.

Inicia-se calculando o valor do coeficiente angular  $m$  usando a equação (An7-2) mostrada no Anexo 7 e a figura do triângulo retângulo como exposto na Figura 4.37.

Valores para o cálculo do coeficiente angular:

$$(x_1, y_1); (x_2, y_2) = (0, 0); (5, -7)$$

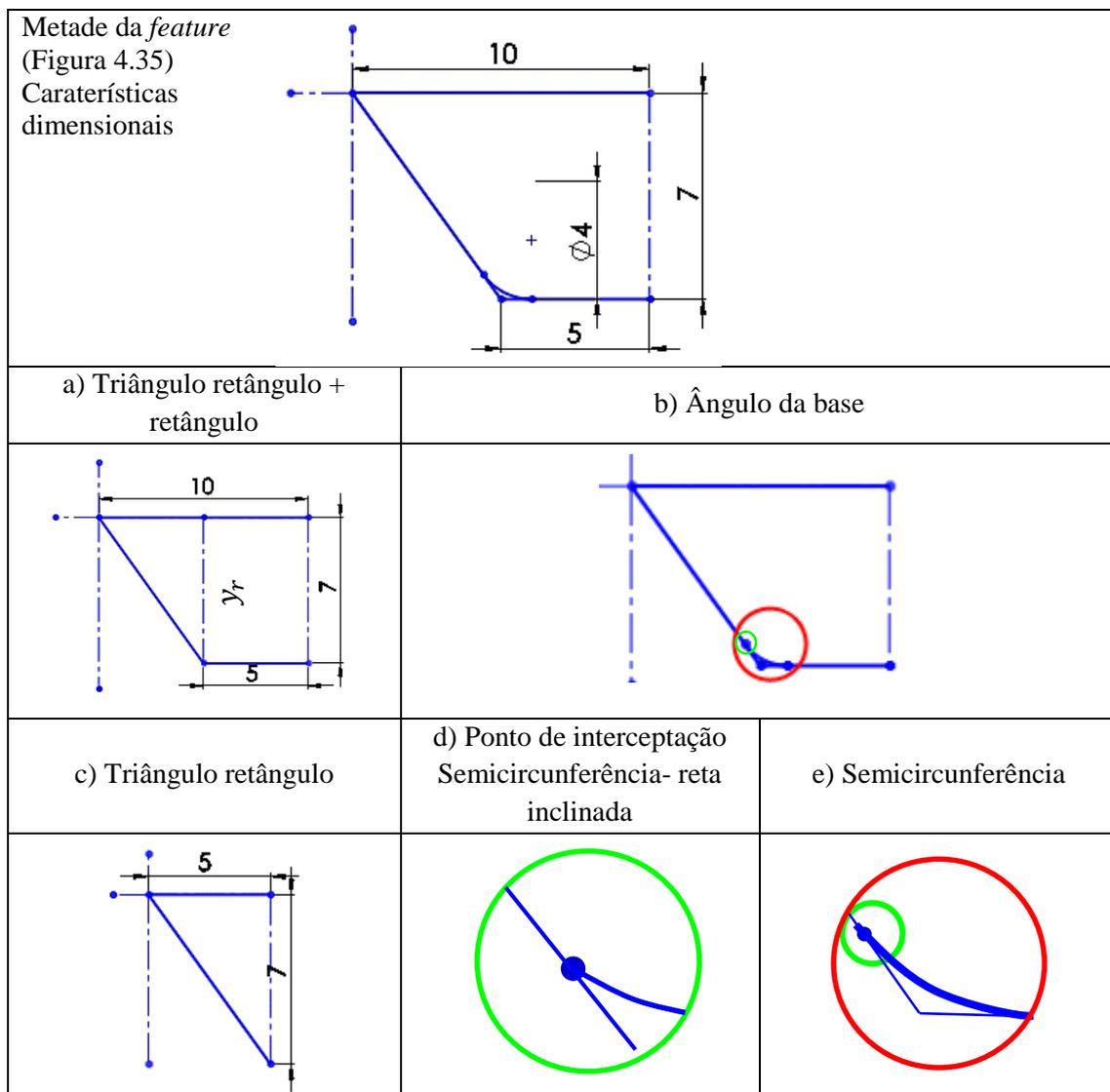


Figura 4.36: Divisão da *feature* em partes mais simples

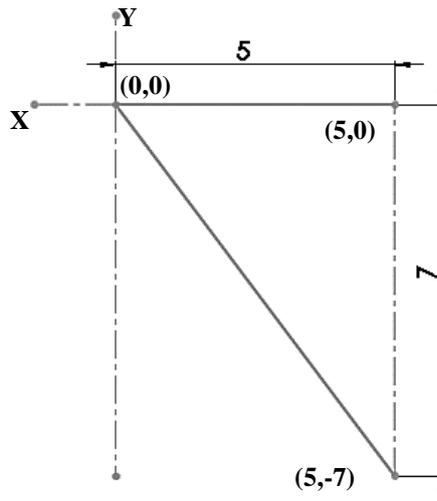


Figura 4.37: Triângulo retângulo parte da *feature* de exemplo

Portanto substituindo os valores para calcular o coeficiente angular, fica:

$$m = \frac{-7-0}{5-0} \rightarrow m = -1,4$$

Tendo-se o valor de  $m$  e utilizando o ponto (5,-7), pode-se obter o valor do intercepto  $b$  usando a equação (An7-3), assim:

$$b = -7 - (-1,4) * 5 \rightarrow b = 0$$

o resultado final com a substituição dos anteriores valores na equação (An7-1), é:

$$y = -1,4 * x + 0 \rightarrow y = -1,4 * x$$

Pode-se isolar a variável  $x$  da anterior equação:

$$x = \frac{y}{-1,4}$$

Mudando o valor de  $y$  em uma sequência (e.g. 1,2,3,...N) pode-se calcular o comprimento da *feature* até o ponto em que se intercepta com a semicircunferência. A Figura 4.38 ilustra esta ideia.

Na Figura 4.38 é apresentada a seguinte equação (4-1):

$$x = \left[ \left( \frac{(-prof + y)}{-1,4} \right) * 2 \right] + (x_r * 2) \quad (4-1)$$

A equação (4-1) permite descrever matematicamente parte da geometria da *feature* tipo ranhura com perfil quadrado U como mostrado na Figura 4.38.

Os dados obtidos a partir da equação (4-1) são mostrados na Figura 4.38. Eles podem ser usados para descrever a trajetória de uma ferramenta de corte (*toolpath*) referente à geometria modelada e, conseqüentemente, a geração do código G dessa parte da *feature*.

O cálculo seguinte é o ponto de intercepção entre a linha inclinada e a semicircunferência. Para calcular o dito ponto, é necessário levar-se a cabo um pouco de geometria analítica. Inicialmente deve-se calcular os ângulos complementares do triângulo retângulo que, neste caso, é o triângulo da Figura 4.37. Para calcular esses ângulos são utilizadas as equações das relações trigonométricas do triângulo retângulo e o teorema de Pitágoras. O teorema de Pitágoras fornecerá o valor da hipotenusa do triângulo. Com este dado é possível calcular os ângulos complementares do triângulo usando as funções seno e cosseno.

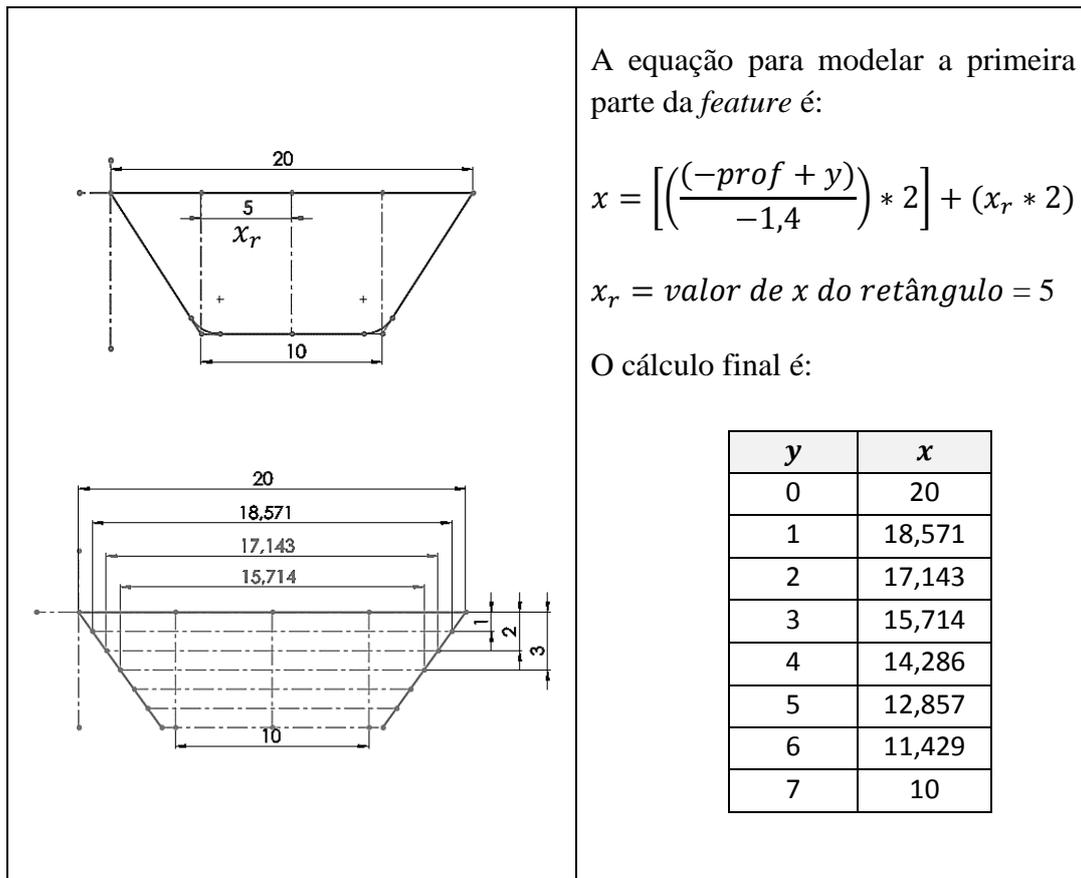


Figura 4.38: Triângulo retângulo mais retângulo, parte da *feature* exemplo

A seguir, descreve-se a metodologia usada, utilizando como referência a Figura 4.38. Começa-se com o cálculo da hipotenusa com a equação (An7-5), assim:

$$c^2 = 5^2 + (-7)^2 \rightarrow c^2 = 74 \rightarrow c = \sqrt{74} \rightarrow c = 8,602$$

Então, o valor da hipotenusa é igual a 8,602mm. Com este dado continua-se com o cálculo de um dos ângulos complementares usando a equação (An7-10).

$$B = \cos^{-1}\left(\frac{5}{8,602}\right) \rightarrow B = 54,46^\circ$$

O valor do ângulo é então:  $54,46^\circ$ . Sabendo-se que a soma dos ângulos de um triângulo retângulo é igual a  $180^\circ$ , pode-se calcular um deles conhecendo-se dois dos três ângulos. Aplicando isso no problema, obtém-se:

$$180 = A(90^\circ) + B(54,46^\circ) + C \rightarrow C = 35,54^\circ$$

A Figura 4.39 mostra a análise geométrica do ponto de interceptação.

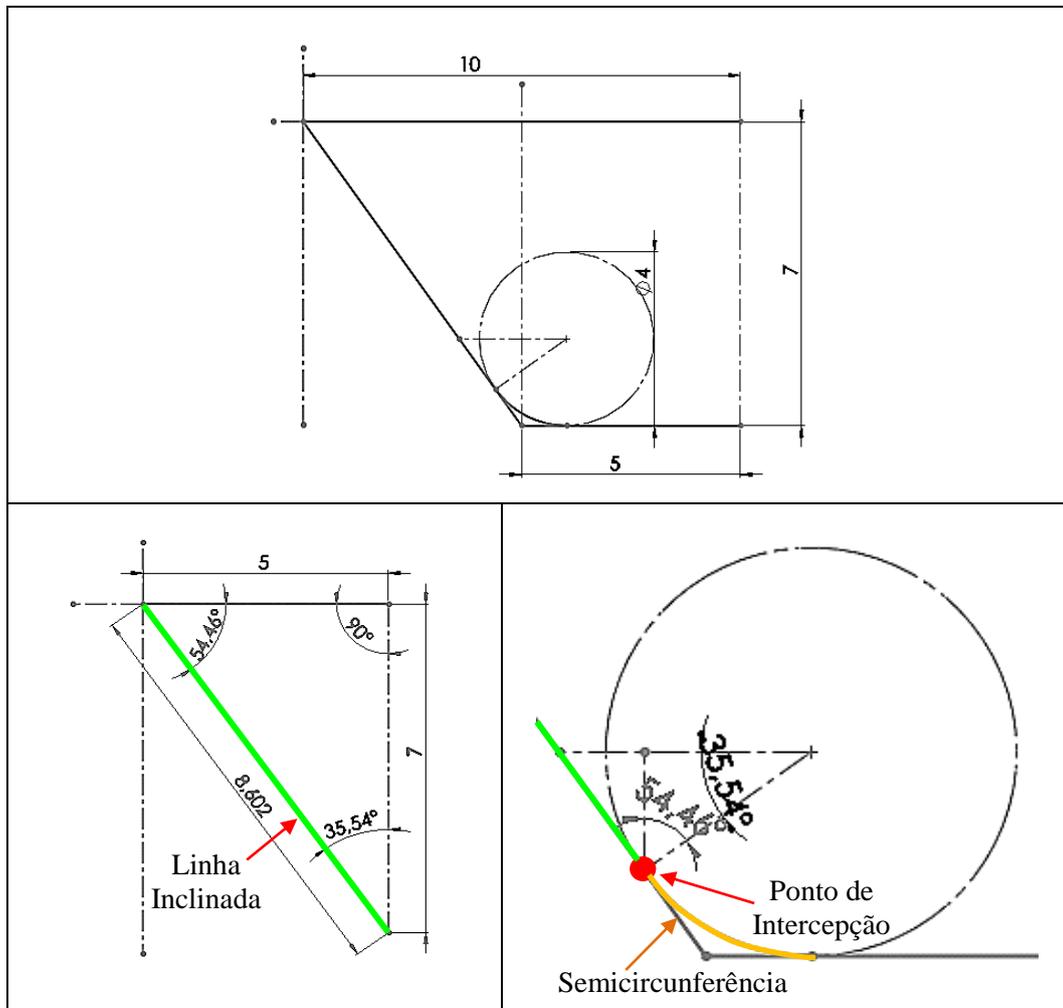


Figura 4.39: Análise geométrica do ponto de interceptação

A análise geométrica na Figura 4.39 mostra que o ponto de interceptação entre a semicircunferência e a linha inclinada possuem algo em comum. Como pode ser observado, o ângulo menor do triângulo retângulo ( $35,54^\circ$ ) é o mesmo ângulo da semicircunferência que descreve o ponto tangencial de interceptação entre a reta inclinada e a semicircunferência. Portanto, com o cálculo dos ângulos complementares do triângulo retângulo anteriormente realizado pode-se determinar que o ponto de interceptação em coordenadas cartesianas é dado pela equação 4-2.

$$y = (prof - r) + (r * \sin C) \quad (4-2)$$

Conhecendo o valor de  $y$  e usando a equação (4-1), pode-se saber o valor do comprimento  $x$  desse ponto, assim:

$$y = (7 - 2) + (2 * \sin 35,54^\circ) \rightarrow y = 5 + (2 * \sin 35,54^\circ) \rightarrow \mathbf{prof_{ptn}} = 6,162$$

então o valor de  $y$  ( $\mathbf{prof_{ptn}}$  profundidade ponto de interceptação) é 6,162 e, conseqüentemente o cálculo do valor de  $x$  ( $\mathbf{comp_{ptn}}$  comprimento ponto de interceptação) é:

$$\mathbf{comp_{ptn}} = \left[ \left( \frac{(-7 + 6,162)}{-1,4} \right) * 2 \right] + (5 * 2) \rightarrow x \cong 11,196$$

Por fim, com os dados obtidos anteriormente, o ponto de interceptação é identificado. Tendo como referência a Figura 4.35, pode-se identificar e marcar este ponto conforma a Figura 4.40.

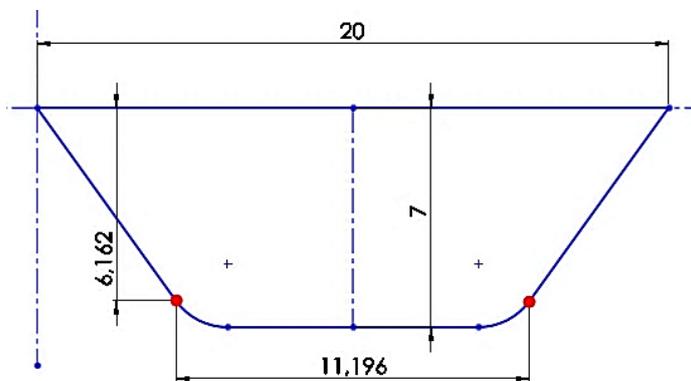


Figura 4.40: ponto de interceptação e a suas medidas.

Efetua-se então o cálculo da porção de semicircunferência mais o retângulo segundo a Figura 4.34d. Esta operação usa o raio e uma sequência de ângulos para modelar a geometria desta parte da *feature*. A Figura 4.41 ilustra esta ideia.

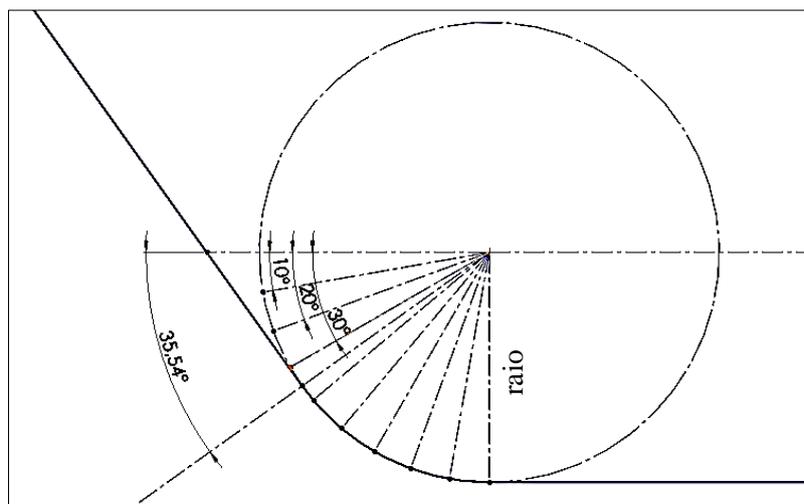


Figura 4.41: Porção de semicircunferência e ângulos em sequência.

Cada ponto da semicircunferência mais o valor de  $x_r$  (ver Figura 4.39) determinam a última parte da *feature*. A Figura 4.42 mostra a geometria desta parte em vários segmentos.

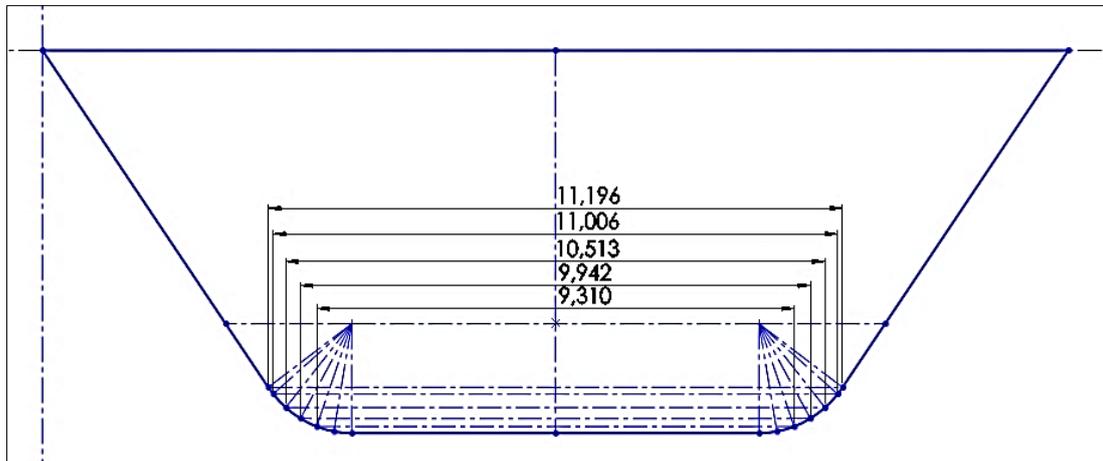


Figura 4.42: Comprimento em sequência

A geometria da figura anterior pode ser modelada usando como referência o cálculo de um ponto tangencial dependendo do ângulo e o raio (ver Figura 4.43) [81]. Esse ponto tangencial está em coordenadas polares e deve ser convertido em coordenadas cartesianas para conseguir-se usá-lo na geração da trajetória da ferramenta e, conseqüentemente, no código G/M. A Tabela 4.8 apresenta as equações (4-3) e (4-4) de conversão de coordenadas. As equações para realizar isso se baseiam na Figura 4.43

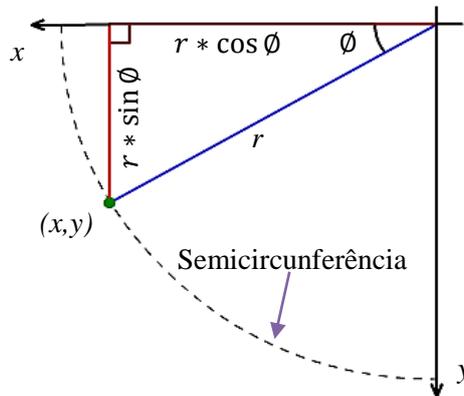


Figura 4.43: Comprimento em sequência (Adaptado de [81])

Tabela 4.8: Conversão de coordenadas

| Coordenadas polares       | Coordenadas polares       |
|---------------------------|---------------------------|
| $r^2 = a^2 + b^2$         | $x = r * \cos \phi$ (4-3) |
| $\tan \phi = \frac{x}{y}$ | $y = r * \sin \phi$ (4-4) |

Usando as equações (4-3) e (4-4) e os valores do ponto de interceptação (*comprimento*, *prof* da Figura 4.40), pode-se modelar as equações (4-5) e (4-6) que descrevem a geometria da semicircunferência desde o ponto de interceptação, assim:

$$\mathbf{comp_{sc}} = ((r * \cos \phi) * 2) + (comp_{ptn} - (r * \cos C)) \quad (4-5)$$

$$\mathbf{prof_{sc}} = (r * \sin \phi) + \mathbf{prof_{ptn}} \quad (4-6)$$

Onde:

***comp<sub>sc</sub>***: é o comprimento (*x* da equação (4-3)) do ponto tangencial da semicircunferência que depende do ângulo  $\phi$ , do ângulo C (triângulo retângulo) e do comprimento do ponto de interceptação.

***prof<sub>sc</sub>***: é a profundidade (*y* da equação (4-4)) do ponto de tangência da semicircunferência que depende do ângulo  $\phi$  e do valor da profundidade do ponto de interceptação. Com os dados da *feature* da Tabela 4.7 e os dados calculados antes do ponto de interceptação, pode-se calcular o comprimento e a profundidade da parte da *feature* que falta (semicircunferência). A Tabela 4.9 mostra os resultados do cálculo usando as equações (4-5) e (4-6). Começando desde o ângulo do ponto de interceptação, seguindo à próxima dezena em sequência até completar 90° (ângulo onde se posiciona o ponto tangencial de interceptação entre a semicircunferência e o limite do comprimento inferior da *feature* como mostrado na Figura 4.36).

Tabela 4.9: Cálculo do comprimento e profundidade da semicircunferência da *feature*

| Ângulo | <i>comp<sub>sc</sub></i> | <i>prof<sub>sc</sub></i> |
|--------|--------------------------|--------------------------|
| 35,54  | 11,196                   | 6,163                    |
| 40     | 11,005                   | 6,286                    |
| 50     | 10,512                   | 6,532                    |
| 60     | 9,941                    | 6,732                    |
| 70     | 9,309                    | 6,879                    |
| 80     | 8,636                    | 6,970                    |
| 90     | 7,941                    | 7,000                    |

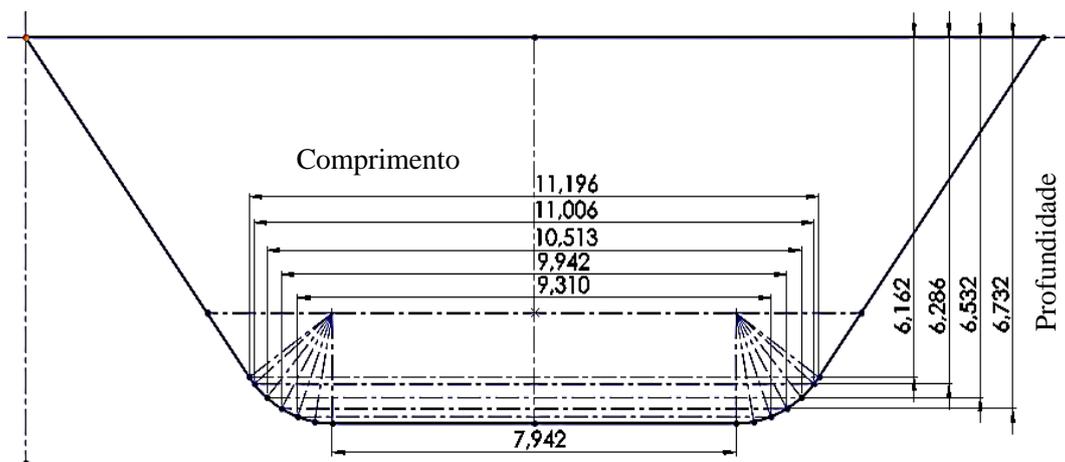


Figura 4.44: Comprimentos e profundidades

A Figura 4.44 representa graficamente os dados (medições dos pontos identificando o comprimento entre pontos e a profundidade) calculados da geometria da *feature* com relação à parte da semicircunferência.

Finalmente, com as equações antes descritas, pode-se criar várias geometrias que tenham um desenho parecido ou igual ao descrito no exemplo de estudo da Figura 4.34. Um exemplo disso é a geometria das *features* descritas na sessão 4.2.2.

### 4.3. GERAÇÃO DE CÓDIGO G & M

Adotando como base o formato do programa NC descrito pela norma ISO6983 parte 1 e parte 2, assim como a identificação dos comandos que o intérprete RS274NGC aceita e todos os conceitos antes referidos, gera-se então o código G/M correspondente às informações extraídas de um programa peça em STEP-NC (arquivo físico p21) informações como os *workingsteps* relacionados às *features* de usinagem com a suas características.

Tendo como referência as características da *feature* exemplo (Tabela 4.7), modela-se essa *feature* no software STEP Modeler, com o objetivo de apresentar a conversão de programa de peça em STEP-NC para Código G/M por meio do adaptador (implementado).

A Tabela 4.10 mostra as características da peça de trabalho, a posição da *feature* e as ferramentas de corte escolhidas para a usinagem.

Tabela 4.10: Informações para criar um exemplo de programa P21

| Item   | Caraterística              | Valor  |
|--|----------------------------|--------|
| Peça de trabalho<br>( <i>workpiece</i> )     | Comprimento X              | 100 mm |
|  | Comprimento Y              | 100 mm |
|  | Comprimento Z              | 20 mm  |
| Posição da <i>feature</i>                    | Comprimento X              | 0 mm   |
|  | Comprimento Y              | 40 mm  |
|  | Comprimento Z              | 0 mm   |
|  | Comprimento <i>feature</i> | 20mm   |
|  | Direção                    | Hor    |
| Ferramenta de corte:<br><i>FaceMill</i>      | Diâmetro do corte          | 6mm    |
|  | Comprimento de corte       | 40 mm  |
|  | Comprimento total          | 70 mm  |
|  | Diâmetro da haste          | 20 mm  |
| Ferramenta de corte:<br><i>Ball End Mill</i> | Diâmetro do corte          | 4 mm   |
|  | Comprimento de corte       | 40 mm  |
|  | Comprimento total          | 70 mm  |
|  | Diâmetro da haste          | 20 mm  |

No apêndice D.1 exemplo b, é apresentada uma sequência de passos (por meio de figuras) para criação de um projeto CAD/CAPP/CAM de uma peça de trabalho usando os dados da Tabela 4.7 e da Tabela 4.10 e tendo como saída o arquivo físico p21.

O arquivo físico p21 (programa de peça em STEP-NC) gerado pelo STEP *Modeler* é mostrado na figura 4.45.

```

ISO-10303-21;
HEADER;
  /*GRIMA*/
FILE_DESCRIPTION(
  /* description */ ('Descricao'),
  /* implementation_level */ ('2;1');
FILE_NAME(
  /* name */ (''),
  /* time_stamp */ ('2013-07-09T10:12:26'),
  /* author */ ('MIGUEL2'),
  /* organization */ ('UFSC'),
  /* preprocessor_version */ ('ST-GENERATOR
5.3'),
  /* originating_system */ (''),
  /* authorization */ ('');
FILE_SCHEMA(('COMBINED_SCHEMA'));
ENDSEC;
DATA;
#1=PROJECT('novo projeto',#87,(#2),$,,$,$);
#2=WORKPIECE('workpiece',#7,50.0,$,$,#9,());
#3=AXIS2_PLACEMENT_3D('workpiece
placement',#4,#5,#6);
#4=CARTESIAN_POINT('',(0.0,0.0,0.0));
#5=DIRECTION('',(0.0,0.0,1.0));
#6=DIRECTION('',(1.0,0.0,0.0));
#7=MATERIAL('SAE 1020','ACO SEM LIGA',(#8));
#8=NUMERIC_PARAMETER('Hardness',250.0,'HB');
#9=BLOCK('piece',#3,100.0,100.0,20.0);
#10=SETUP('setup',#16,#15,(#20));
#11=AXIS2_PLACEMENT_3D('',#12,#13,#14);
#12=CARTESIAN_POINT('',(0.0,0.0,45.0));
#13=DIRECTION('',(0.0,0.0,1.0));
#14=DIRECTION('',(1.0,0.0,0.0));
#15=PLANE('security plane',#11);
#16=AXIS2_PLACEMENT_3D('origin',#17,#18,#19)
#17=CARTESIAN_POINT('',(0.0,0.0,0.0));
#18=DIRECTION('',(0.0,0.0,1.0));
#19=DIRECTION('',(1.0,0.0,0.0));
#20=WORKPIECE_SETUP(#2,#21,$,$,());
#21=AXIS2_PLACEMENT_3D('',#22,#23,#24);
#22=CARTESIAN_POINT('',(0.0,0.0,0.0));
#23=DIRECTION('',(0.0,0.0,1.0));
#24=DIRECTION('',(1.0,0.0,0.0));
#25=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Botto
m And Side Rough
Milling',5.0,#33,#26,#31,#32,$,#34,#35,#38,2
.0,2.25,0.0,0.0);
#26=MILLING_CUTTING_TOOL('SF20',#27,(#29),$,$,
$,$);
#27=FACEMILL(#28,4,.RIGHT,,$,60.0);
#28=MILLING_TOOL_DIMENSION(3.0,0.0,0.0,40.0,
0.0,0.0,0.0);
#29=CUTTING_COMPONENT(70.0,#30,$,$,$);
#30=MATERIAL('P','CARBIDE',());
#31=MILLING_TECHNOLOGY(0.0060,.TCP,.14.5,-
15.38497783221655,$,.F,.F,.F,$);
#32=MILLING_MACHINE_FUNCTIONS(.T,$,$,.F,$,
(),.T,$,$,());
#33=CARTESIAN_POINT('start
point',(0.0,0.0,0.0));
#34=PLUNGE_TOOLAXIS(#36);
#35=PLUNGE_TOOLAXIS(#37);
#36=DIRECTION('approach strategy
direction',(0.0,0.0,1.0));
#37=DIRECTION('retract strategy
direction',(0.0,0.0,-1.0));
#38=CONTOUR_PARALLEL(0.6000000000000001,.T,
.CCW,.CONVENTIONAL);
#39=SLOT('ranhura com perfil
U',#2,(#25,#71),#40,#48,#53,#57,(#68,#69));
#40=AXIS2_PLACEMENT_3D('ranhura com perfil U
placement',#41,#42,#43);
#41=CARTESIAN_POINT('',(0.0,50.0,20.0));
#42=DIRECTION('',(0.0,0.0,1.0));
#43=DIRECTION('',(1.0,0.0,0.0));
#44=AXIS2_PLACEMENT_3D('ranhura com
perfil U depth',#45,#46,#47);
#45=CARTESIAN_POINT('',(0.0,0.0,-7.0));
#46=DIRECTION('',(0.0,0.0,1.0));
#47=DIRECTION('',(1.0,0.0,0.0));
#48=PLANE('ranhura com perfil U depth
location',#44);
#49=AXIS2_PLACEMENT_3D('course of travel
placement',#50,#51,#52);
#50=CARTESIAN_POINT('',(0.0,0.0,0.0));
#51=DIRECTION('',(0.0,0.0,1.0));
#52=DIRECTION('',(1.0,0.0,0.0));
#53=LINEAR_PATH(#49,#54,#56);
#54=TOLERANCED_LENGTH_MEASURE(100.0,#55);
#55=PLUS_MINUS_VALUE(0.05,0.05,4);
#56=DIRECTION('Linear Path Direction -
HORIZONTAL',(1.0,0.0,0.0));
#57=SQUARE_U_PROFILE(#64,#58,#60,0.620249
4859828215,#62,0.6202494859828215);
#58=TOLERANCED_LENGTH_MEASURE(10.0,#59);
#59=PLUS_MINUS_VALUE(0.05,0.05,4);
#60=TOLERANCED_LENGTH_MEASURE(2.0,#61);
#61=PLUS_MINUS_VALUE(0.05,0.05,4);
#62=TOLERANCED_LENGTH_MEASURE(2.0,#63);
#63=PLUS_MINUS_VALUE(0.05,0.05,4);
#64=AXIS2_PLACEMENT_3D($,#65,#66,#67);
#65=CARTESIAN_POINT('',(0.0,0.0,-7.0));
#66=DIRECTION('',(0.0,0.0,1.0));
#67=DIRECTION('',(0.0,-1.0,0.0));
#68=OPEN_SLOT_END_TYPE();
#69=OPEN_SLOT_END_TYPE();
#70=MACHINING_WORKINGSTEP('SquareUprofile
Slot_RGH',#15,#39,#25,$);
#71=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bo
ttom And Side Rough Milling',5.0,#79,#72,
..#77,#78,$,#80,#81,#84,0.2,0.2,0.0,0.0);
#72=MILLING_CUTTING_TOOL('ball end
mill',#73,(#75),$,$,$);
#73=BALL_ENDMILL(#74,4,.RIGHT,,$,60.0);
#74=MILLING_TOOL_DIMENSION(4.0,0.0,0.0,40
.0,2.0,2.0,0.0);
#75=CUTTING_COMPONENT(70.0,#76,$,$,$);
#76=MATERIAL('P','CARBIDE',());
#77=MILLING_TECHNOLOGY(0.068,.TCP,.66.0,-
52.521131220325465,$,.F,.F,.F,$);
#78=MILLING_MACHINE_FUNCTIONS(.T,$,$,.F,
$,(),.T,$,$,());
#79=CARTESIAN_POINT('start
point',(0.0,0.0,0.0));
#80=PLUNGE_TOOLAXIS(#82);
#81=PLUNGE_TOOLAXIS(#83);
#82=DIRECTION('approach strategy
direction',(0.0,0.0,1.0));
#83=DIRECTION('retract strategy
direction',(0.0,0.0,-1.0));
#84=CONTOUR_PARALLEL(0.8,.T,.CCW,.CONVE
NTIONAL);
#85=MACHINING_WORKINGSTEP('SquareUprofile
Slot_RGH',#15,#39,#71,$);
#86=WORKPLAN('workplan setup plano:
XY',(#70,#85),$,#10,$);
#87=WORKPLAN('Main
workplan',(#86),$,$,$);
ENDSEC;
END-ISO-10303-21;

```

Figura 4.45: Arquivo físico p21 gerado pelo STEP *Modeler*

A seguir serão apresentadas algumas janelas capturadas para mostrar a funcionalidade do adaptador e, conseqüentemente, a geração do código G de acordo com o programa de peça em STEP-NC da Figura 4.45.

Deve-se notar que a programação do algoritmo do protótipo de adaptador de STEP-NC para Código G foi realizada usando a linguagem Java e o *software* Eclipse como o ambiente integrado de desenvolvimento (IDE). Também, note-se que a única saída de dados visível que apresenta o adaptador é o código G, e portanto, são apresentadas apenas as informações extraídas do programa peça em STEP-NC de acordo as variáveis que as contêm.

A Figura 4.46 mostra as informações básicas e avançadas extraídas pelo algoritmo de acordo com o exemplo da Figura 4.45.

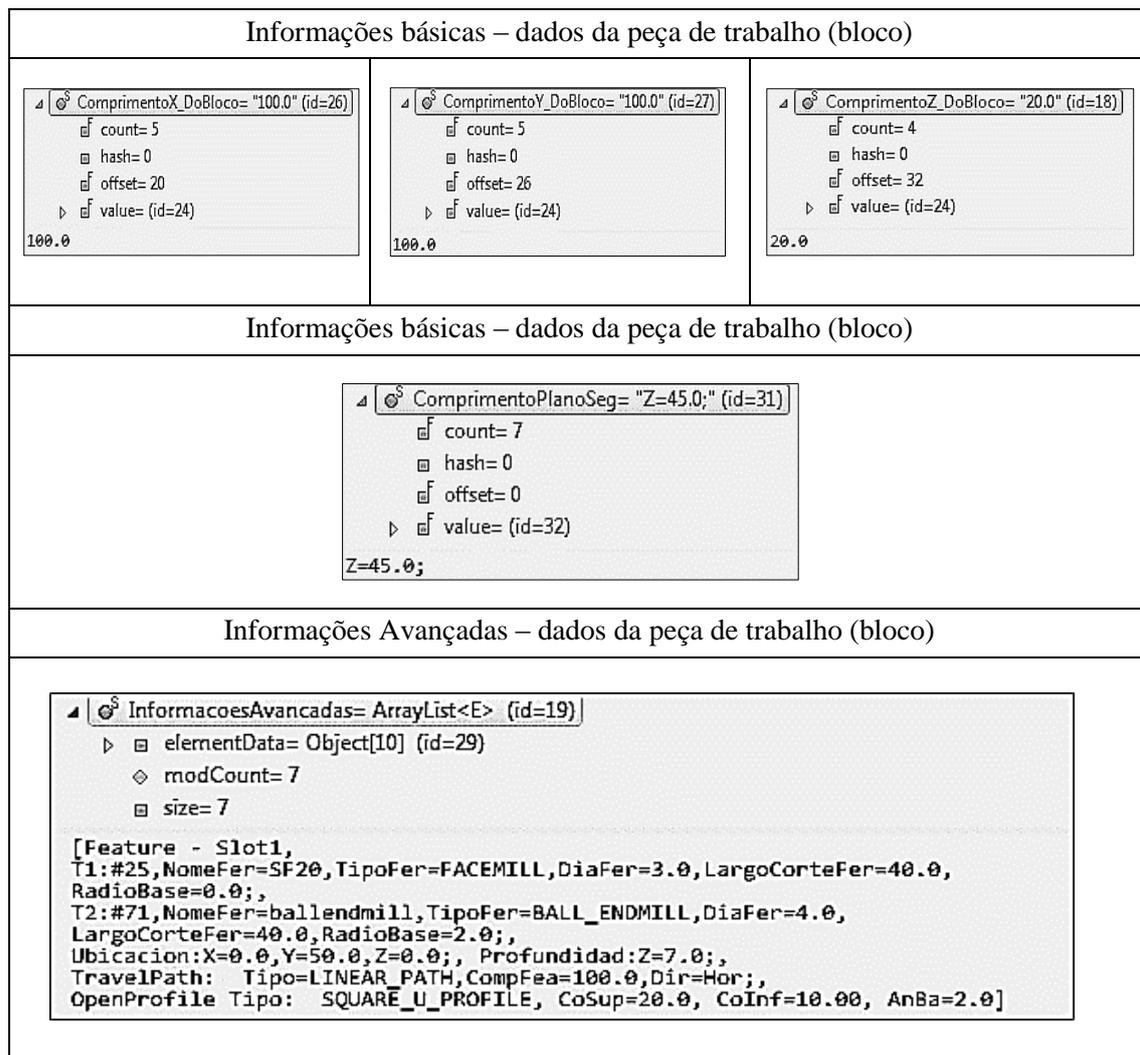


Figura 4.46: Informações extraídas do arquivo físico p21

Por outro lado, de acordo com o tipo de *feature*, é utilizada uma ou várias ferramentas para usin-la. A função seguinte do algoritmo é, então, organizar as ferramentas de todo o programa p21 em uma lista (*ArrayList* listaDadosTrocaFerra), ou seja, uma lista com todas as ferramentas usadas para usinar as *features* contidas no programa p21. Isso evita que sejam salvas ferramentas do mesmo tipo com as mesmas características.

```

listDadosTrocaFerra= ArrayList<E> (id=34)
└─ elementData= Object[10] (id=36)
   └─ modCount= 2
      └─ size= 2

[T1,FACEMILL,3.0,0.0,40.0; , T2,BALL_ENDMILL,4.0,2.0,40.0;]

```

Figura 4.47: Ferramentas do arquivo físico p21

Da figura anterior pode se observar que:

- O tamanho da lista é 2 (*size =2*), portanto, existem duas ferramentas.
- A primeira ferramenta é identificada e indexada como “**T1**”
- O tipo de ferramenta é **FACEMILL**
- O diâmetro da ferramenta é **3.0 mm**
- O ângulo da borda é **0.0 mm**
- O comprimento de corte é **40.0 mm**

Todas as ferramentas são tratadas da mesma forma extraindo-se as mesmas características.

Cada *feature* tem associada uma ou várias ferramentas, no exemplo tem-se duas. Já na geração do código G relacionada à *feature*, as ferramentas que são usadas para usiná-la são comparadas com as ferramentas do programa p21. Isso tem o propósito de colocar a ferramenta que já foi indexada de forma adequada.

Usando como segundo exemplo um programa P21 onde há três ferramentas e duas *features*, pode-se entender como as ferramentas para cada *feature* são ordenadas e indexadas. A Tabela 4.11 mostra as ferramentas do programa P21 e a Tabela 4.12 mostra as ferramentas usadas em cada *feature*.

Tabela 4.11: Ferramentas do programa P21 - segundo exemplo

| Index | Tipo de ferramenta | Diâmetro | Ângulo da base | Comprimento de corte |
|-------|--------------------|----------|----------------|----------------------|
| T1    | FACEMILL           | 6.0      | 0.0            | 40.0                 |
| T2    | FACEMILL           | 10.0     | 0.0            | 40.0                 |
| T3    | BALLENDMILL        | 4.0      | 2.0            | 50.0                 |

Tabela 4.12: *Features* e a suas ferramentas - segundo exemplo

| Index            | Tipo de ferramenta | Diâmetro | Ângulo da base | Comprimento de corte |
|------------------|--------------------|----------|----------------|----------------------|
| <b>FEATURE 1</b> |                    |          |                |                      |
| T1               | FACEMILL           | 6.0      | 0.0            | 40.0                 |
| T2               | BALLENDMILL        | 4.0      | 2.0            | 50.0                 |
| <b>FEATURE 2</b> |                    |          |                |                      |
| T1               | FACEMILL           | 10.0     | 0.0            | 40.0                 |
| T2               | BALLENDMILL        | 4.0      | 2.0            | 50.0                 |

Note-se na Tabela 4.12 que cada *feature* possui as suas ferramentas as quais têm seu próprio índice. Quando o código G enquanto à troca de ferramenta é gerado, as ferramentas de cada *feature* são

comparadas com as ferramentas já organizadas do arquivo físico P21 e, conseqüentemente, cada ferramenta obterá um novo índice (se necessário). Na Tabela 4.13, apresenta-se as ferramentas das *features* do segundo exemplo com os novos índices tendo como referência as ferramentas da tabela 4.11.

Depois que as ferramentas são reindexadas, o algoritmo do adaptador gera uma tabela com as ferramentas do programa p21 seguindo o formato da tabela de ferramentas do controlador EMC (ver apêndice B.4.2).

Tabela 4.13: *Features* e a suas ferramentas rearranjadas – segundo exemplo

| Index            | Tipo de ferramenta | Diâmetro | Ângulo da base | Comprimento de corte |
|------------------|--------------------|----------|----------------|----------------------|
| <b>FEATURE 1</b> |                    |          |                |                      |
| T1               | <i>FACEMILL</i>    | 6.0      | 0.0            | 40.0                 |
| T3               | <i>BALLENDMILL</i> | 4.0      | 2.0            | 50.0                 |
| <b>FEATURE 2</b> |                    |          |                |                      |
| T2               | <i>FACEMILL</i>    | 10.0     | 0.0            | 40.0                 |
| T3               | <i>BALLENDMILL</i> | 4.0      | 2.0            | 50.0                 |

O seguinte passo do algoritmo é selecionar o tipo de *feature*. Isso permite saber as características geométricas da *feature* e seu posterior modelamento matemático. Conseqüentemente é gerado o código G/M dependendo de:

- Informações básicas
- Informações avançadas
- Ferramentas de usinagem
- Tipo de *feature* de usinagem
- Modelamento matemático da geometria da *feature* selecionada
- Trajetórias da ferramenta de corte
- Parâmetros de corte da máquina-ferramenta (arquivo de configuração, ver apêndice G)

Por fim, as características anteriores são traduzidas para códigos G e M levando-se em conta a compensação de raio de ferramenta, o sistema de coordenadas (G54 para zero peça e G55 para troca de ferramenta), o tipo de interpolação e o modo de coordenadas de distância. No anexo 6, mostra-se um fragmento do resultado da conversão do programa de peça em STEP-NC (ver Figura 4.46) para programa peça em código G realizado pelo protótipo de adaptador proposto neste trabalho. A geração de código G/M de outro tipo de *features* é criado de forma similar. Quando um arquivo físico possui vários tipos de *feature*, cada uma é tratada de forma independente no adaptador para a geração do código G.

Com o fim de conferir o código G criado pelo adaptador (lembre-se que só é a operação de desbaste) com relação às trajetórias da ferramenta e a geometria da *feature* (ver Figura 4.49), é usado o *software* de simulação de usinagem CNC chamado Vericut [82]. A Figura 4.48 mostra a peça de trabalho, as trajetórias de ferramenta e a geometria final simuladas no *software* Vericut.

Das vantagens que o Vericut oferece, além da citada anteriormente (simulação de usinagem) são:

- Verificar possíveis batimentos entre a ferramenta de corte, sistema de fixação e partes da máquina.
- Otimizar o programa NC. Isso permite gerar um código G com avanço variável a partir do código G original. Com o código G otimizado e possível mostrar o ganho na produtividade, redução de tempo na usinagem e maximizar a vida útil da(s) ferramenta(s) de corte.

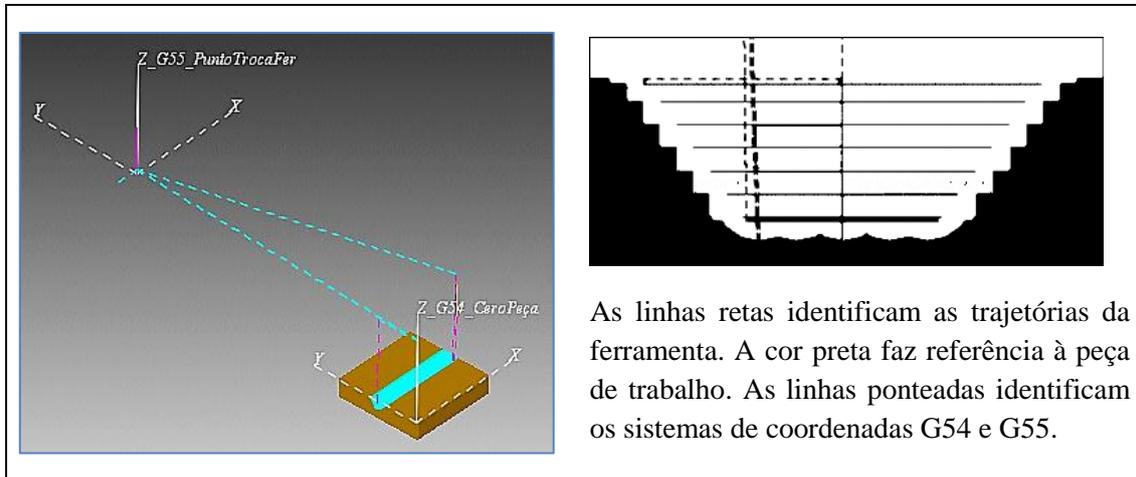


Figura 4.48: Simulação de trajetórias de ferramenta e geometria final (usinagem)

Pode-se observar na Figura 4.48 as diferentes partes usinadas de acordo com o código G simulado pelo Vericut. O código é carregado, logo, seleciona-se a ferramenta que irá usinar a primeira parte da *feature*. Depois de fazer a operação de desbaste da primeira parte da *feature*, prossegue-se com a troca da ferramenta para selecionar a ferramenta seguinte que irá usinar a segunda e última parte da *feature* segundo o exemplo.

Como já foi mencionado, o *software* STEP Modeler fornece uma variedade de *features*, porém não todas as disponibilizadas na norma ISO14640 parte 10. Neste trabalho foram tratadas e testadas algumas das *features* disponíveis nesse *software*, assim sendo, o protótipo de adaptador de STEP-NC a Código G/M funciona somente com as *features* mostradas na Tabela 4.14.

Tabela 4.14: *Features* funcionais no adaptador

| <i>Features</i>  |  |  |
|--|--|--|
| Furos  | Ranhuradas (em horizontal)   | Cavidades  |
| <ul style="list-style-type: none"> <li>• Base plana</li> <li>• Base cônica</li> <li>• Base esférica</li> <li>• Base arredondada</li> <li>• Cônico</li> </ul> | <ul style="list-style-type: none"> <li>• U base plana</li> <li>• U base arredondada</li> <li>• Perfil circular</li> <li>• Perfil U circular</li> </ul> | <ul style="list-style-type: none"> <li>• Retangular com fundo plano</li> <li>• Retangular com fundo arredondado</li> </ul> |

Figura 4.49: Código G gerado a partir do Programa de peça em STEP-NC de exemplo.

|  |   |  |   |  |
|--|---|--|---|--|
| <pre>(Header) (Código G gerado pelo protótipo de Adaptador de STEP-NC a Código G vl.0) (ALGUMAS INFORMAÇÕES IMPORTANTES) (Arquivo: C:/mod/teste01.p21) (Dados da peça ComX:100.0 ComY:100.0 CompZ:20.0) (Plano de Segurança:45.0) (Ferramentas de corte) (T1,FACEMILL,3.0,0.0,40.0;) (T2,BALL_ENDMILL,4.0,2.0,40.0;) (COMENTÁRIOS ATIVADOS)  (Início Programa) N1 G17 G21 G54 G40 G90 (Desliga o Spindle) N2 M05 (Sistema de coordenadas G55) N3 G90 G40 G55 G00 Z5.0000 N4 G01 F800.0000 X0.0000 Y0.0000 N5 G00 Z0.0000 (Troca de ferramenta) N6 M06 T1 N7 G00 Z5.0000 (Liga o Spindle) N8 M03 (Sistema de coordenadas G54) N9 G54 (Primeira parte) N10 G40 G90 G01 F800.0 X0.0000 Y0.0000 N11 G90 G01 F400.0 Y50.0000 N12 G00 Z0.0000 N13 G01 Z-0.2500 N14 G90 G01 X101.5000 N15 Y51.5000 N16 X-1.5000 N17 Y48.5000 N18 X101.5000 N19 G41 D1 N20 G90 Y56.0000 N21 X-1.5000 N22 G90 Y44.0000 N23 X101.5000 N24 G90 Y59.0000</pre> | <pre>N25 X-1.5000 N26 G90 Y41.0000 N27 X101.5000 N28 Y60.0000 N29 X-1.5000 N30 Y40.0000 N31 X101.5000 N32 Z0.0000 N33 G40 G90 X0.0000 Y50.0000 N34 G91 G01 F400.0 Z- 1.0000 N35 G90 G01 X101.5000 N36 Y51.5000 N37 X-1.5000 N38 Y48.5000 N39 X101.5000 N40 G41 D1 N41 G90 Y56.0000 N42 X-1.5000 N43 G90 Y44.0000 N44 X101.5000 N45 G90 Y59.0000 N46 X-1.5000 N47 G90 Y41.0000 N48 X101.5000 N49 Y59.2857 N50 X-1.5000 N51 Y40.7143 N52 X101.5000 N53 G40 G90 X0.0000 Y50.0000 N54 G91 G01 F400.0 Z- 1.0000 N55 G90 G01 X101.5000 N56 Y51.5000 N57 X-1.5000 N58 Y48.5000 N59 X101.5000 N60 G41 D1 N61 G90 Y56.0000 N62 X-1.5000 N63 G90 Y44.0000 N64 X101.5000</pre> | <pre>N65 Y58.5714 N66 X-1.5000 N67 Y41.4286 N68 X101.5000 N69 G40 G90 X0.0000 Y50.0000 N70 G91 G01 F400.0 Z- 1.0000 N71 G90 G01 X101.5000 N72 Y51.5000 N73 X-1.5000 N74 Y48.5000 N75 X101.5000 N76 G41 D1 N77 G90 Y56.0000 N78 X-1.5000 N79 G90 Y44.0000 N80 X101.5000 N81 Y57.8571 N82 X-1.5000 N83 Y42.1429 N84 X101.5000 N85 G40 G90 X0.0000 Y50.0000 N86 G91 G01 F400.0 Z- 1.0000 N87 G90 G01 X101.5000 N88 Y51.5000 N89 X-1.5000 N90 Y48.5000 N91 X101.5000 N92 G41 D1 N93 G90 Y56.0000 N94 X-1.5000 N95 G90 Y44.0000 N96 X101.5000 N97 Y57.1429 N98 X-1.5000 N99 Y42.8571 N100 X101.5000 N101 G40 G90 X0.0000 Y50.0000 N102 G91 G01 F400.0 Z- 1.0000</pre> | <pre>N103 G90 G01 X101.5000 N104 Y51.5000 N105 X-1.5000 N106 Y48.5000 N107 X101.5000 N108 G41 D1 N109 G90 Y56.0000 N110 X-1.5000 N111 G90 Y44.0000 N112 X101.5000 N113 Y56.4286 N114 X-1.5000 N115 Y43.5714 N116 X101.5000 N117 G40 G90 X0.0000 Y50.0000 N118 G91 G01 F400.0 Z- 1.0000 N119 G90 G01 X101.5000 N120 Y51.5000 N121 X-1.5000 N122 Y48.5000 N123 X101.5000 N124 G41 D1 N125 Y55.7143 N126 X-1.5000 N127 Y44.2857 N128 X101.5000 N129 G40 G90 X0.0000 Y50.0000 N130 G91 G01 F400.0 Z- 0.1625 N131 G90 G01 X101.5000 N132 Y51.5000 N133 X-1.5000 N134 Y48.5000 N135 X101.5000 N136 G41 D1 N137 Y55.5982 N138 X-1.5000 N139 Y44.4018 N140 X101.5000 (Desliga o Spindle) N141 M05</pre> | <pre>(Sistema de coordenadas G55) N142 G90 G40 G55 G00 Z5.0000 N143 G01 F800.0000 X0.0000 Y0.0000 N144 G00 Z0.0000 (Troca de ferramenta) N145 M06 T2 N146 G00 Z5.0000 (Liga o Spindle) N147 M03 (Sistema de coordenadas G54) N148 G54 (Segunda parte) N149 G40 G90 G01 F800.0 X0.0000 Y0.0000 N150 G90 G01 F400.0 Y50.0000 N151 G00 Z0.0000 N152 G90 G00 Z-5.0000 N153 G91 G01 Z-2.0000 N154 G90 G01 X102.0000 N155 Y52.0000 N156 X-2.0000 N157 Y48.0000 N158 X102.0000 N159 G41 D2 N160 Y55.5982 N161 X-2.0000 N162 Y44.4018 N163 X102.0000 N164 G90 X104.0000 N165 G40 G90 G00 Z45.0000 (Desliga o Spindle) N166 M05 (Fim programa) N167 M30</pre> |
|--|---|--|---|--|

#### 4.4. GERAÇÃO DO ARQUIVO .TBL

Como foi explicado na modelagem funcional do adaptador de STEP-NC para Código G, há uma função que organiza as ferramentas do programa de peça em STEP-NC. Essa função também gera um arquivo “.tbl” que contém as informações das ferramentas usadas no programa e o salva numa posição específica de acordo com o arquivo de configuração do EMC. O arquivo TBL, o formato do conteúdo e suas características são explicadas no apêndice B.4.2.

#### 4.5. CÓDIGO EM JAVA DO ALGORITMO DO PROTÓTIPO DE ADAPTADOR DE STEP-NC A CÓDIGO G

O código do algoritmo do protótipo de adaptador foi implementado computacionalmente usando-se a linguagem Java. Neste trabalho usa-se a ferramenta gratuita de uso no comercial, “*Google code*” para publicar o código do algoritmo do adaptador de STEP-NC a Código G. O alvo da publicação do código fonte na web é não estender este trabalho em relação à apresentação da implementação do código (na linguagem Java) do algoritmo do adaptador.

O *Google Code* é um site na internet propriedade de Google disponibilizado para desenvolvedores de *software* interessados em alojar códigos fonte abertos [83]. Usando este site, cria-se o projeto “adaptador-de-stepnc-to-gcode-on-linuxcnc” onde é armazenado o código fonte do algoritmo do protótipo de adaptador de STEP-NC para Código G e outros recursos apresentados neste trabalho de mestrado.

O endereço do site web do projeto é: <https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/>. Nesse site encontra-se o código do algoritmo do adaptador no item “*Source*”, assim como exemplos de programas STEP-NC parte 21, imagens de usinagem, fotos, figuras e filmes deste trabalho.

## 5. IMPLEMENTAÇÃO DO CONTROLADOR LINUXCNC (ou EMC) ADERENTE Á NORMA STEP-NC

Neste capítulo apresenta-se a metodologia usada para adicionar duas novas funcionalidades ao controlador LinuxCNC (deve-se levar em conta que no apêndice B é descrito o controlador LinuxCNC, suas características e funções).

A primeira função é a possibilidade de executar, no EMC, de forma nativa o arquivo físico p21 convertendo-o a código G (de maneira automática e transparente para o operador da máquina), por meio do adaptador implementado no capítulo anterior.

A segunda função é a possibilidade de visualizar o programa de peça em STEP-NC que está sendo executado por meio de uma interface gráfica de usuário (GUI) adicional à GUI do controlador EMC. Essa GUI adicional (Visor de código STEP-NC) depende da GUI do controlador EMC para sua execução.

Em seguida, propõe-se a metodologia para aderir o adaptador de STEP-NC a Código G/M na GUI *AXIS* do EMC.

### 5.1. METODOLOGIA PARA ADIÇÃO DO ADAPTADOR DE STEP-NC PARA CÓDIGO G AO CONTROLADOR LINUXCNC

O adaptador implementado no capítulo anterior permite converter o arquivo físico P21 para código G. Esta função deve ser adicionada ao controlador LinuxCNC para que, de forma nativa, ele consiga carregar programas de peça em STEP-NC.

Note-se que o intérprete (RS274NGC) do LinuxCNC só pode executar programas de peça em código G. Portanto, o adaptador é usado para converter de forma automática e transparente para o operador da Router CNC o programa P21 para código G.

A metodologia usada descreve a forma em que se implementa a função de conversão de um formato a outro por meio da execução do adaptador. Só é possível por meio da modificação do arquivo .INI (ver apêndice B.4.1) da configuração entre a Router e o controlador LinuxCNC.

Antes de explicar quais são as modificações feitas no arquivo INI, deve-se entender que existem vários tipos de GUI's usados pelo LinuxCNC, sendo que cada uma delas pode ter diferentes formatos de arquivo INI [72]. Portanto neste trabalho é usada a GUI padrão do LinuxCNC, i.e. a interface *AXIS* (ver sessão 5.2.1).

No arquivo INI da interface *AXIS*, pode-se agregar programas adicionais por meio da seção [*FILTER*] (filtro). Esses programas (filtros) têm a capacidade de adotar o arquivo aberto pelo EMC e tratá-lo segundo suas necessidades. O filtro pode fazer qualquer tarefa desejada, como por exemplo ter certeza que o arquivo termina com M30, ou algo tão complicado como detectar se a entrada é uma imagem de profundidade, e gerar código G para usinar a forma definida na imagem [72].

A seção [*FILTER*] do arquivo INI controla como os filtros funcionam. Em primeiro lugar, para cada tipo de arquivo, há que escrever uma linha *PROGRAM\_EXTENSION*. Em seguida, há que especificar o programa a ser executado para cada tipo de arquivo. Este programa recebe o nome do arquivo de entrada como seu primeiro argumento e deve escrever código RS274NGC para a saída

padrão, i.e. código G. Esta saída é o que será exibida na área de texto da GUI *AXIS* e será executado pelo LinuxCNC [84].

Na Figura 5.1 são mostradas duas configurações da seção *[FILTER]*, uma com a configuração original e outra com o adaptador adicionado como um filtro.

| Configuração Original   | Configuração de um novo filtro   |
|---|--|
| <pre>[FILTER] PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image PROGRAM_EXTENSION = .py Python Script png = image-to-gcode gif = image-to-gcode jpg = image-to-gcode py = python</pre> | <pre>[FILTER] PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image PROGRAM_EXTENSION = .py Python Script PROGRAM_EXTENSION = .p21,.STP Iso14649 p21 to G-code png = image-to-gcode gif = image-to-gcode jpg = image-to-gcode py = python p21 = java -jar "./AdapterSTEPNC_GCode.jar" STP = java -jar "./AdapterSTEPNC_GCode.jar"</pre> |

Figura 5.1: Ferramentas e suas características

Pode-se observar na Figura 5.1 que foram adicionadas três novas linhas:

- a. **PROGRAM\_EXTENSION** = .p21,.STP Iso14649 p21 to G-code
- b. **p21** = java -jar “./AdapterSTEPNC\_GCODE.jar”
- c. **STP** = java -jar “./AdapterSTEPNC\_GCODE.jar”

A primeira linha é para identificar os arquivos com as extensões “.p21” e “.STP”. A segunda e terceira linha relacionam os arquivos de entrada com o protótipo do adaptador de STEP-NC a Código G, que neste caso é um executável Java.

O executável Java do adaptador foi obtido por meio da opção de exportar executável no IDE Eclipse. Esse executável deve estar alojado na mesma pasta onde estão os arquivos de configuração do LinuxCNC (normalmente é *Área de trabalho/”nome pasta”*).

## 5.2. METODOLOGIA PARA CRIAR UMA GUI PARA VISUALIZAR O ARQUIVO FÍSICO P21 JUNTO COM A GUI *AXIS* DO LINUXCNC

Nesta sessão é apresentada a metodologia de implementação de uma GUI que permitirá visualizar o arquivo físico p21. Este arquivo é o arquivo original que o operador da máquina carrega no LinuxCNC. A GUI de visualização é uma GUI adicional que será agregada como uma nova função da GUI *AXIS* do LinuxCNC.

Antes de implementar a GUI de visualização, deve-se conhecer algumas características importantes da GUI *AXIS* do LinuxCNC.

### 5.2.1. A GUI *AXIS* do LinuxCNC

*AXIS* é uma interface gráfica de usuário (GUI) para LinuxCNC que conta com uma pré-visualização ao vivo tanto do código G como da trajetória da ferramenta. O código G é visualizado numa janela que mostra o código (texto) e a trajetória da ferramenta é visualizada numa janela de forma gráfica. *AXIS* está escrita em Python e utiliza *Tcl/Tk* e *OpenGL* para exibir sua interface de usuário. A Figura 5.2 mostra uma janela que representa a GUI *AXIS* [72].

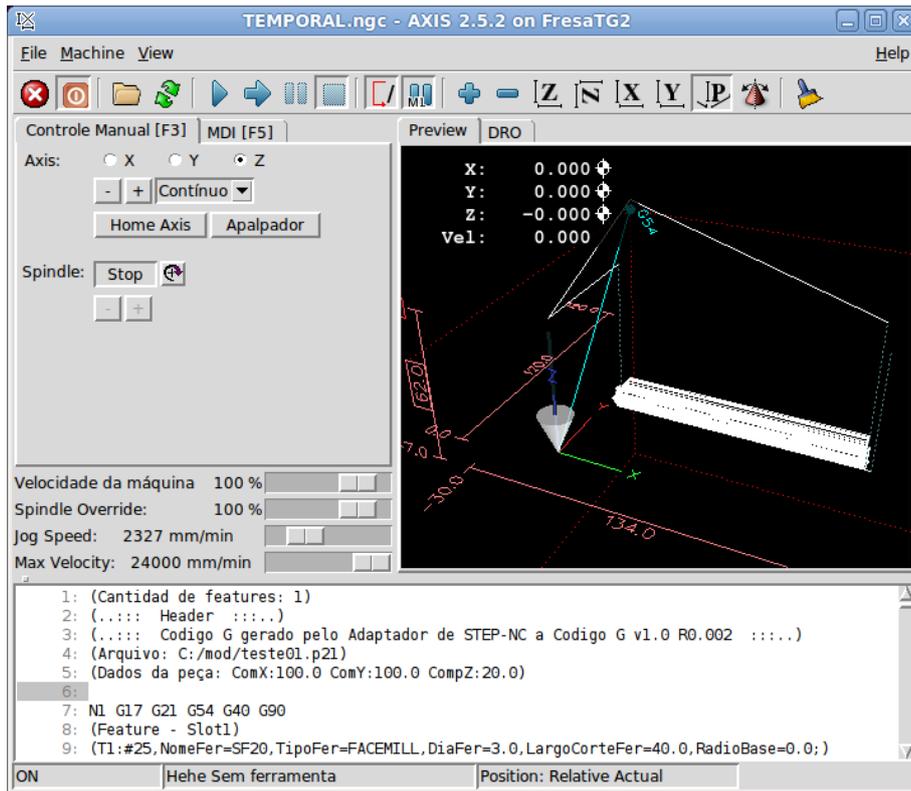


Figura 5.2: GUI AXIS do LinuxCNC

Para inserir uma nova interface junto com a GUI AXIS existem três formas, PyVCP, GladeVCP e Glade com Python.

- O Python - Painel de Controle Virtual ou PyVCP (*Python Virtual Control Panel*) é projetado para dar ao operador a capacidade de personalizar a interface AXIS com botões e indicadores (elementos chamados *widgets*) para realizar tarefas especiais. A Figura 5.3 mostra os *widgets* disponíveis no PyVCP [74].

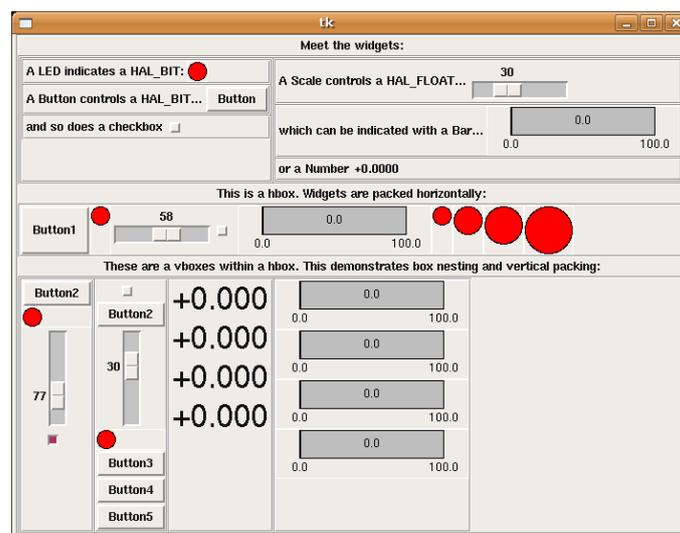


Figura 5.3: Widgets do PyVCP.

O PyVCP não possui um *widget* para mostrar grandes quantidades de texto (e.g. um *TextBox*), portanto este tipo de painel não pode ser usado para visualizar o código STEP-NC.

- O GladeVCP é um componente do LinuxCNC que adiciona a capacidade de inserir um novo painel de interface do usuário (GUI) para interfaces de usuário do LinuxCNC como *AXIS* ou *TOUCHY*. Ao contrário de PyVCP, GladeVCP não está limitado para exibir e definir componentes HAL. Portanto, o GladeVCP fornece ao operário a capacidade de construir uma GUI completa para o LinuxCNC, apoiando-se na linguagem Python [72]

O GladeVCP possui uma grande quantidade de *widgets* entre eles o “*EMC SourceView*” que permite visualizar o código que está sendo executado pelo LinuxCNC. Como o arquivo físico P21 não é executado (mas sim o código G convertido pelo adaptador a partir do arquivo físico p21), apenas visualizado, esse *widget* não resolve a necessidade antes mencionada.

### 5.2.2. Implementação de uma GUI para visualizar o arquivo físico p21 usando Glade

O Glade Interface Designer ou Glade é uma ferramenta de desenvolvimento visual de interfaces gráficas por meio de GTK/GNOME. É independente da linguagem de programação e de forma predeterminada não gera código fonte, senão um arquivo XML.

Por meio do uso de *GtkBuilder*, os arquivos XML de Glade podem ser usados em várias linguagens de programação como C, C++, Java, Perl, Python entre outros. Glade é um *software* livre publicado com licença GNU/GPL [85].

Com o uso de Glade é implementada uma GUI para visualizar o arquivo físico p21. Usando os *widgets* encontrados normalmente em Glade (não GladeVCP), pode-se criar uma GUI que basicamente permita visualizar o programa peça em STEP-NC. O protótipo da GUI deve ter seguir os seguintes parâmetros:

- Visualizar o arquivo físico P21 que foi convertido pelo adaptador a código G.
- Ocultar o texto do arquivo físico p21.
- Ter duas janelas para realizar o proposto nos dois itens anteriores.
  - A primeira janela deve ter dois botões um para carregar o arquivo físico p21 e outro para ocultá-lo. Quando o botão de carregar é pressionado o programa peça em STEP-NC é mostrado na janela, e depois o botão de carregar desaparece e o botão de ocultar aparece. Quando o botão de ocultar é pressionado, a primeira janela se oculta e a segunda janela é invocada e mostrada.
  - A segunda janela deve estar vazia e ter um botão para invocar a primeira janela. Quando o botão é pressionado a segunda janela se oculta e a primeira janela é invocada e mostrada.

As janelas propostas anteriormente são apresentadas na Figura 5.4.

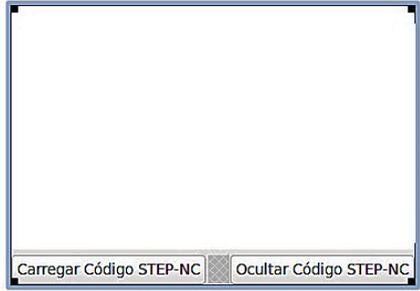
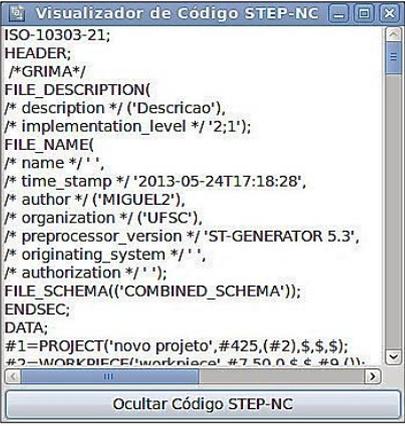
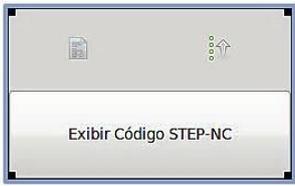
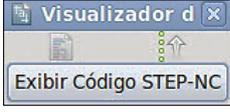
|  | Versão em Glade   | Versão no LinuxCNC  |
|--|---|---|
| Primeira Janela onde se visualiza o arquivo físico p21 |  |   |
| Segunda Janela que oculta o arquivo visualizado        |  |  |

Figura 5.4: GUI para visualização do arquivo físico p21

Por último, resta fazer a programação das funções da GUI usando a linguagem Python para implementá-las.

### 5.2.3. Implementação das funções da GUI que visualiza o arquivo físico p21

Na sessão anterior, foi modelada a GUI que permitirá visualizar o arquivo físico p21, que consiste apenas, na interface gráfica. Agora é preciso implementar o algoritmo (programa) que permite dar funcionalidade aos *widjets* dessas GUI. Para isto, utiliza-se a linguagem Python que trabalha em conjunto com Glade.

O algoritmo proposto segue o diagrama de fluxo da Figura 5.5. O código da implementação do programa em Python é apresentado no apêndice E. Esse código também pode ser visto no site web do projeto <https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/> do Google *code*.

Tendo todos os componentes necessários, pode-se continuar com a inclusão da GUI modelada nos itens anteriores na GUI AXIS do LinuxCNC, sendo esse o último passo na implementação do CNC aderente à norma STEP-NC.

## 5.3. METODOLOGIA PARA AGREGAR A GUI DE VISUALIZAÇÃO NA GUI AXIS DO LINUXCNC

Como já foi explicado antes, a GUI AXIS do LinuxCNC está escrita em Python e utiliza Tcl/Tk e OpenGL. O arquivo que pertence à GUI do LinuxCNC pode ser reconhecido como “axis.tcl” e é encontrado dentro da pasta do controlador. A extensão do arquivo *axis* é *tcl*. O *tcl* (*Tool Command Language*), é usado principalmente para o desenvolvimento rápido de protótipos, aplicações e interfaces gráficas [86]. A GUI *AXIS* usa a combinação de Tcl/Tk (*Tool Kit*) que são fundamentalmente os encarregados da GUI e todos seus componentes.

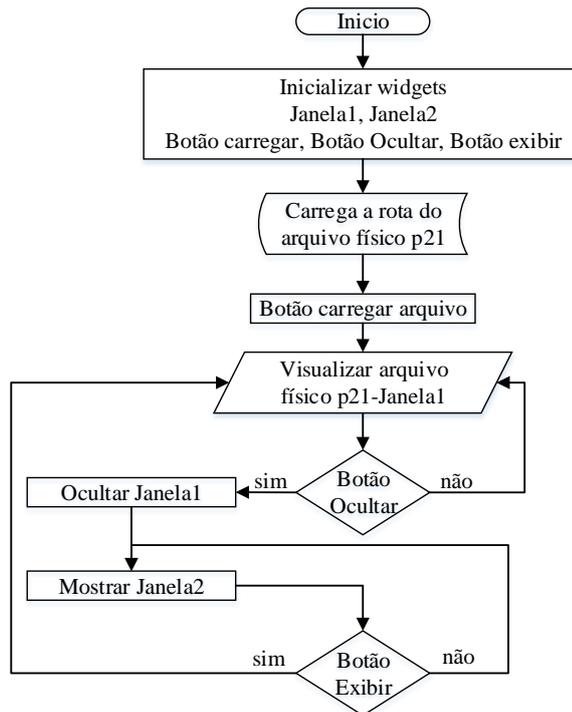


Figura 5.5: Diagrama de fluxo do algoritmo da GUI de visualização.

Para agregar a GUI de visualização do arquivo físico p21 na GUI *AXIS*, escolhe-se o menu “*View*” para acrescentar uma nova função. Essa função tem como objetivo executar o programa Python da GUI de visualização e conseqüentemente poder visualizar o programa de peça em STEP-NC nessa GUI. Isso é possível modificando-se o arquivo *axis.tcl* (encontrado na rota “*/usr/share/axis/tcl*” no Ubuntu) e adicionando algumas linhas na linguagem tcl. Na Figura 5.6 apresenta-se um fragmento original que descreve o menu *View* da GUI *AXIS* e outro fragmento onde é feita a adição da nova função, a execução do programa em Python da GUI de visualização.

| Fragmento original do código da GUI <i>AXIS</i>  | Fragmento modificado do código da GUI <i>AXIS</i> – Agregado: executar programa Python   |
|--|--|
| <pre> # ----- .menu.view add radiobutton \   -command set_view_z \   -variable view_type \   -value 1 \   -accelerator V setup_menu_accel .menu.view end [ "_Top view" ]  .menu.view add radiobutton \   -command set_view_z2 \   -variable view_type \   -value 2 \   -accelerator V setup_menu_accel .menu.view end [ "_Rotated Top view" ] </pre> | <pre> # ----- .menu.view add command \   -command {exec python /home/mod/GUI/CodViewerGui.py &amp;} setup_menu_accel .menu.view end [ "_STEP-NC Code Viewer" ]  .menu.view add separator  .menu.view add radiobutton \   -command set_view_z \   -variable view_type \   -value 1 \   -accelerator V setup_menu_accel .menu.view end [ "_Top view" ]  .menu.view add radiobutton \   -command set_view_z2 \   -variable view_type \   -value 2 \   -accelerator V setup_menu_accel .menu.view end [ "_Rotated Top view" ] </pre> |

Figura 5.6: Fragmento original e modificado da GUI *AXIS*

As linhas agregadas que permitem executar o programa Python se baseiam na lógica seguinte:

- Adicionar uma nova opção (função) tipo comando no menu *View*
- Criar um comando de execução de um programa tipo Python.
- Executar o programa Python na rota especificada.
- Acrescentar um novo elemento no menu *View* relacionado com o comando com o nome: “STEP-NC Code Viewer”
- Adicionar uma linha tipo separador

Depois de modificar o arquivo *axis.tcl*, o último passo é substituí-lo pelo original na pasta do controlador LinuxCNC.

Na Figura 5.7 apresenta-se duas janelas, uma com o menu *View* original e outra com esse menu modificado com o comando que executa o programa da GUI de visualização do arquivo físico p21.

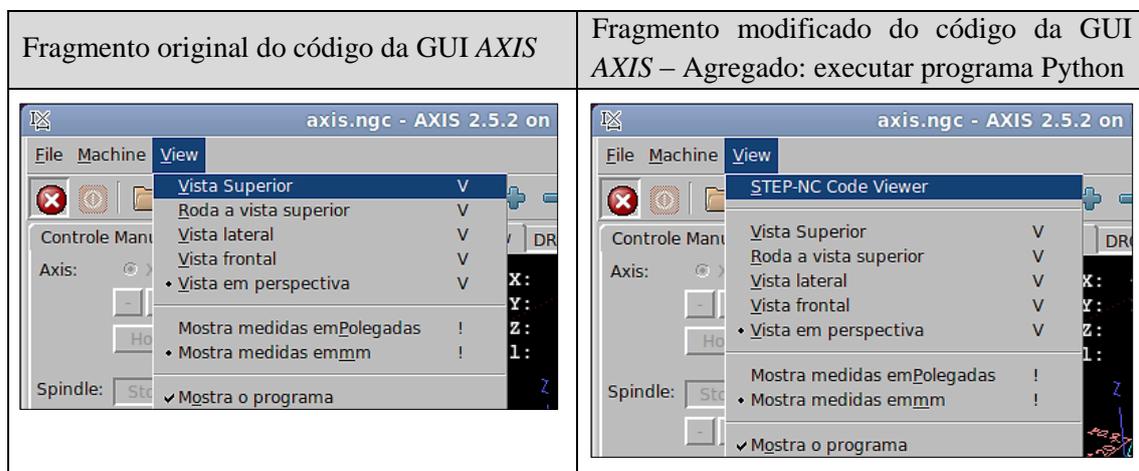


Figura 5.7: Menu “View” original e modificado da GUI AXIS.

## 6. ESTUDO DE CASOS

Neste capítulo é apresentada a validação do controlador LinuxCNC em relação à aderência à norma ISO14649 por meio de um estudo de caso.

### 6.1. ESTUDO DE CASO – CNC ADERENTE A STEP-NC

Este estudo de caso está relacionado com o controlador LinuxCNC usando o adaptador de STEP-NC para Código G para executar programas de peça em STEP-NC *part 21* de forma nativa. Em outras palavras, é verificada a aderência do controlador LinuxCNC à norma ISO14649 por meio da usinagem uma peça teste de madeira, mostrando que os *workingsteps* do programa p21 corresponde com o código G gerado pelo adaptador. Este estudo de caso tem o seguintes objetivos:

- Projetar uma peça no *software STEP Modeler* (usar algumas das *features* funcionais descritas na sessão 4.3), exportando (gerando) o arquivo físico p21, correspondente ao modelo CAD/CAPP/CAM.
- Realizar a conversão do arquivo físico p21 a código G usando o LinuxCNC
- Visualizar o arquivo físico p21 (o texto) na GUI AXIS do LinuxCNC depois da conversão.
- Simular no software Vericut o código G gerado pelo adaptador a partir do programa de peça em STEP-NC para encontrar erros (se houver).
- Usinar na Router CNC uma peça teste para o estudo de caso, usando como entrada um arquivo físico p21 (antes gerado) no controlador LinuxCNC.
- Comparar de forma geral as medidas do desenho CAD com a peça usinada pela Router CNC.
- Fazer manualmente um código G de acordo com as *features* projetadas no *software STEP Modeler*.
- Validar o código G gerado pelo adaptador a partir do arquivo físico p21 de entrada, comparando-o com o código G feito manualmente para ver semelhanças e concordâncias.

#### 6.1.1. Esboço da peça teste do estudo de caso

Realiza-se um esboço da peça teste para ser projetada no *software STEP Modeler*. Isto permite realizar duas ações: validar as *features* disponibilizadas no STEP Modeler e gerar um arquivo físico p21 de uma peça projetada a partir de um esboço. A Figura 6.1 mostra o esboço da peça desejada. As medidas da peça apresentadas no esboço possuem uma tolerância dimensional de  $\pm 0,05$  mm.

As características da peça de trabalho e das ferramentas de corte usadas no projeto da peça teste são apresentadas na Tabela 6.1.

Tabela 6.1: Itens e suas características para o projeto da peça

| Item   |             | Caraterística (em mm) |
|--|-------------|-----------------------|
| Informações da geometria da peça de trabalho |             | 120 x 120 x 18,25     |
| Ferramentas                                  | CenterDrill | Ø6 x TLO10            |
|  | TwisDrill   | Ø6 x TLO40            |
|  | FaceMill    | Ø6 x TLO40            |
|  | BallEndMill | Ø4 x TLO40            |

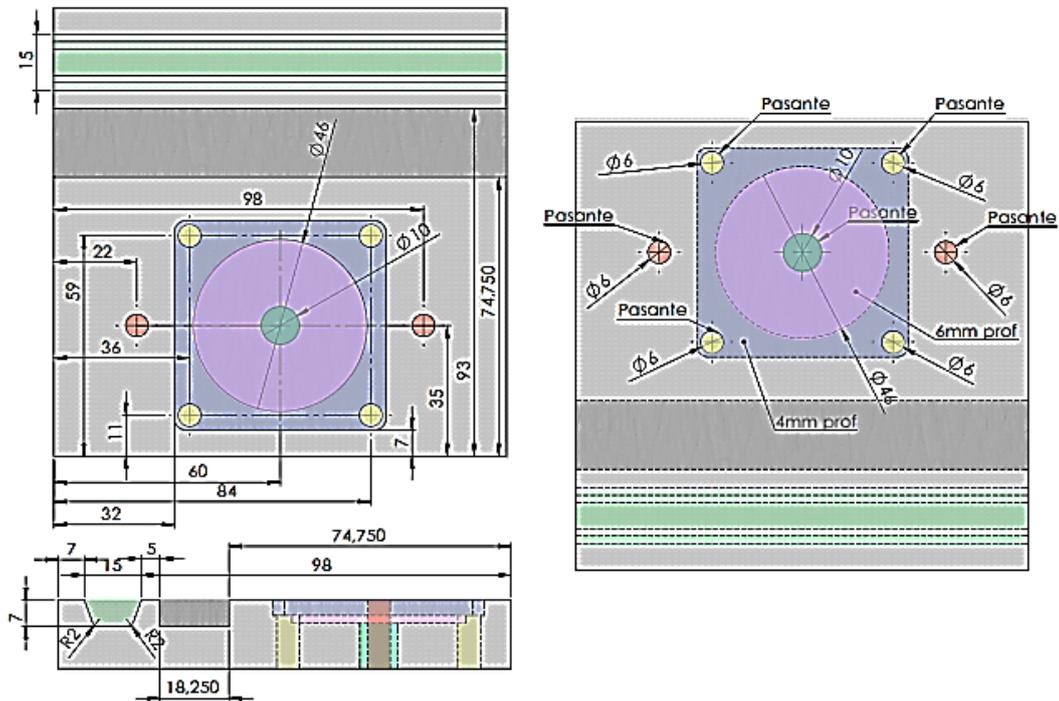


Figura 6.1:Esboço da peça teste do estudo de caso

### 6.1.2.Gerar um arquivo físico p21 no STEP Modeler de acordo com o esboço da peça teste

Nas Figuras 6.2 a 6.5 são mostradas algumas janelas capturadas do processo do projeto da peça teste no *software* STEP Modeler e, conseqüentemente, da geração do programa peça em STEP-NC parte 21.

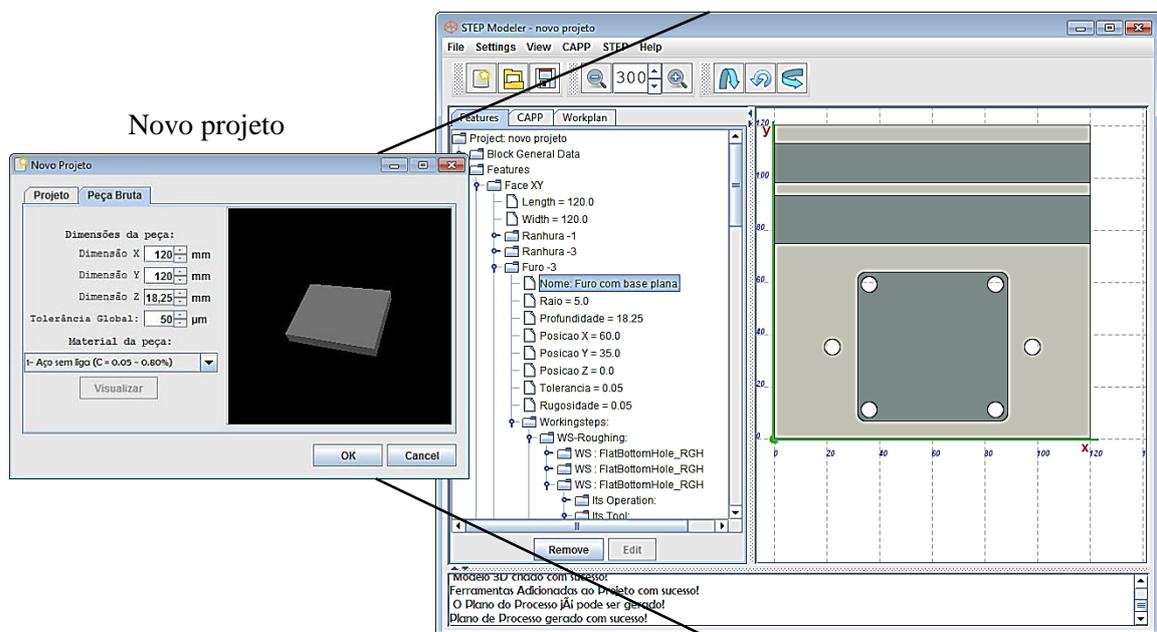


Figura 6.2:Projeto CAD/CAPP/CAM da peça teste no *software* STEP Modeler.

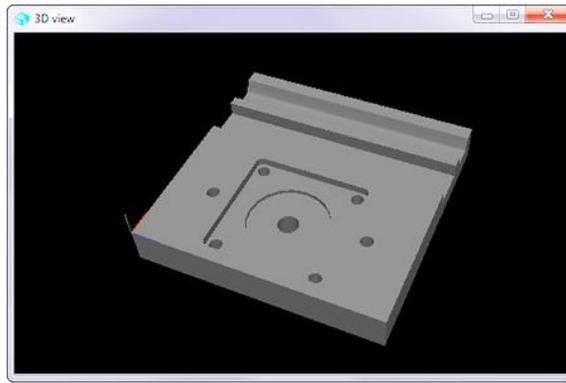


Figura 6.3: Modelo 3D da peça projetada gerado pelo STEP Modeler

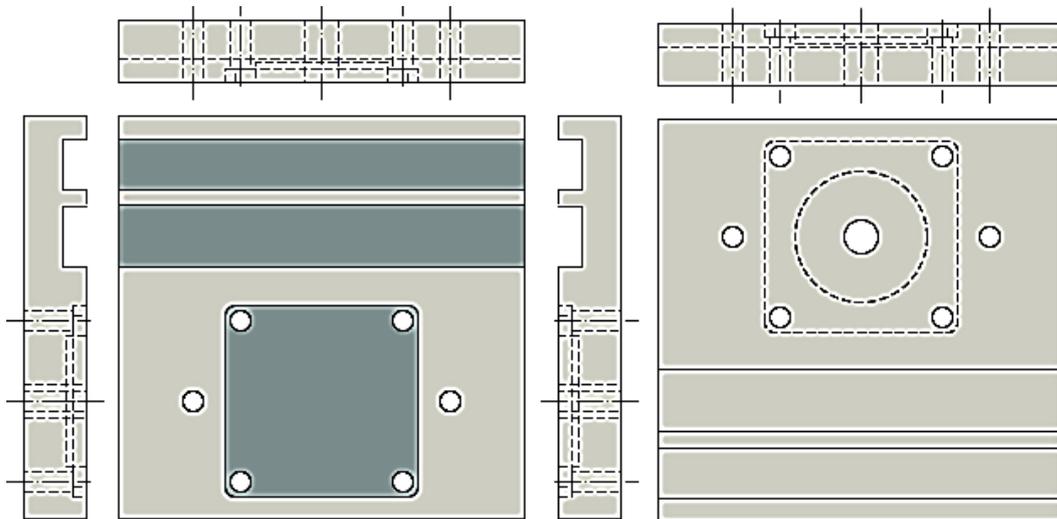


Figura 6.4: Diferentes vistas 2D da peça modelada no STEP Modeler

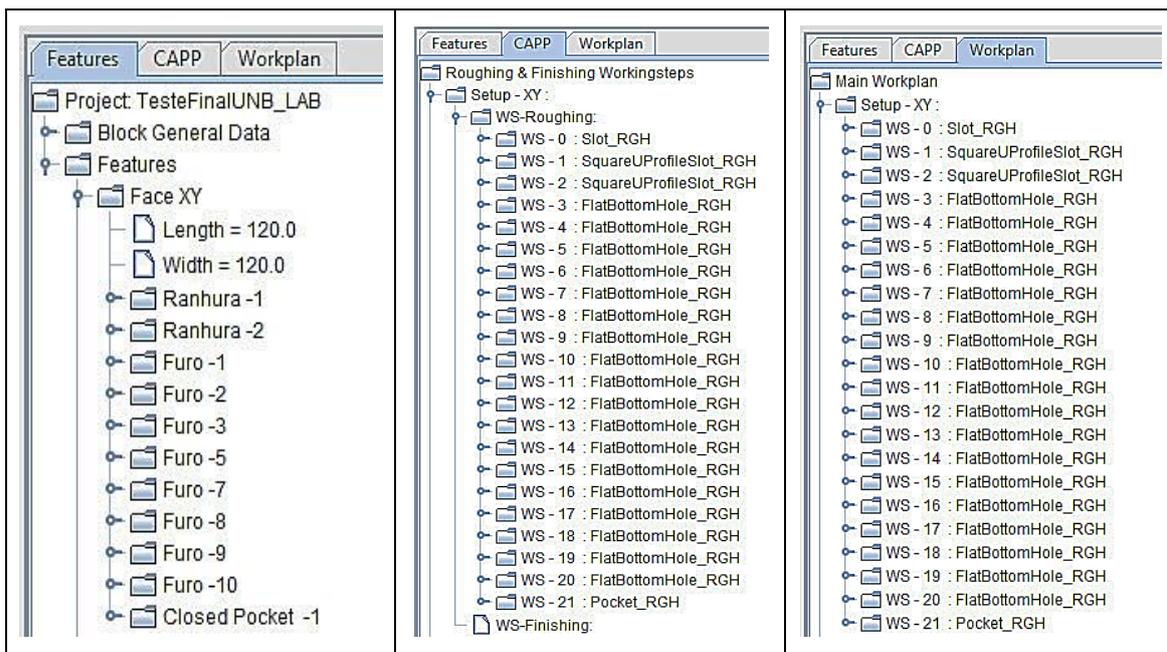


Figura 6.5: Features, CAPP e workplan do projeto

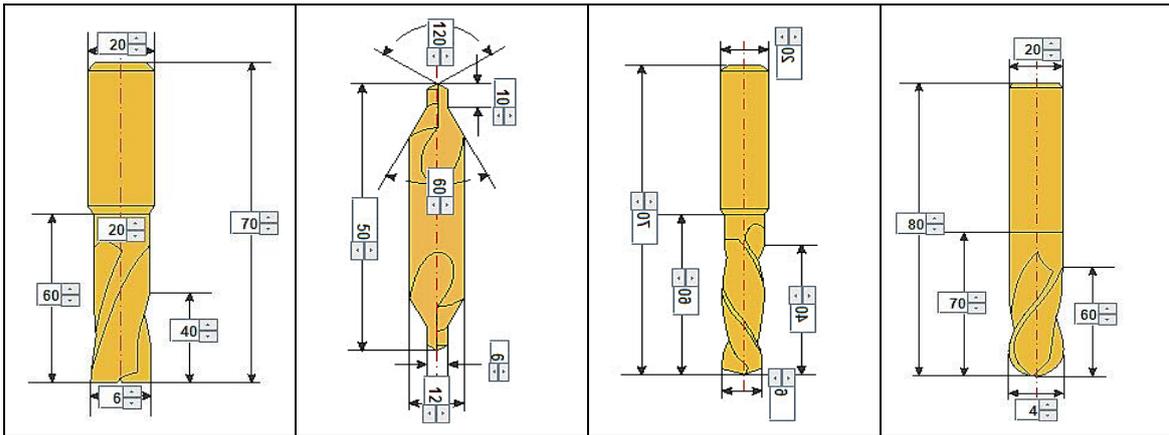


Figura 6.6: Ferramentas utilizadas no projeto

Na figura são mostradas as ferramentas disponíveis para usinar a peça teste. Um fragmento do arquivo físico gerado pelo STEP Modeler de acordo com a peça teste projetada é apresentado na Figura 6.7

```

ISO-10303-21;
HEADER;
  /*GRIMA*/
FILE_DESCRIPTION(
  /* description */ ('Descricao'),
  /* implementation level */ '2;1');
....
DATA;
#1=PROJECT('novo projeto',#468,(#2),$, $, $);
#2=WORKPIECE('workpiece',#7,50.0,$, $, #9, ());
....
#9=BLOCK('piece',#3,120.0,120.0,18.25);
#10=SETUP('setup',#16,#15,(#20));
#11=AXIS2_PLACEMENT_3D('',#12,#13,#14);
#12=CARTESIAN_POINT('',(0.0,0.0,43.25));
....
#164=FACEMILL(#165,4,.RIGHT,$,60.0);
#165=MILLING_TOOL_DIMENSION(6.0,0.0,0.0,40.0,0.0,0.0,0.0);
#166=CUTTING_COMPONENT(70.0,#167,$, $, $);
....
#424=TWIST_DRILL(#425,2,.RIGHT,$,60.0);
#425=MILLING_TOOL_DIMENSION(6.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#426=CUTTING_COMPONENT(70.0,#427,$, $, $);
....
#467=WORKPLAN('workplan setup plano:
XY',(#66,#112,#127,#150,#161,#176,#199,#210,
#225,#262,#285,#296,#319,#330,#353,#364,#387,#398,#421,#432,#455,#466)
,$,#10,
$);
#468=WORKPLAN('Main workplan',(#467),$,$,$);
ENDSEC;
END-ISO-10303-21;

```

Figura 6.7: Fragmento do arquivo físico p21 gerado pelo STEP Modeler para a peça teste

Tendo o programa de peça em STEP-NC gerado pelo STEP Modeler, pode-se continuar com o processo de usinagem da peça teste na Router CNC. Para isso, são realizados os passos a seguir.

### 6.1.3. Carregar um arquivo físico p21 no LinuxCNC

A próxima etapa é carregar o programa peça em STEP-NC (chamado “TesteFinalUNBLAB.p21”) no Controlador LinuxCNC. Isso permite que o adaptador abra esse programa e realize a conversão (fonte-fonte) do arquivo físico p21 para código G, além de criar o arquivo TBL com as ferramentas necessárias para a correta execução do código G. Nas Figuras 6.8 a 6.10, são apresentadas algumas janelas do dito processo.

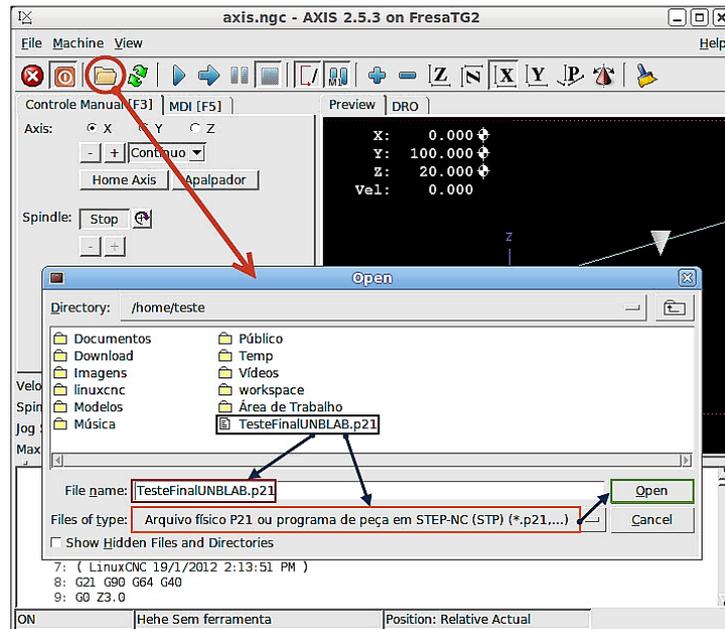


Figura 6.8: Abrindo o arquivo físico p21

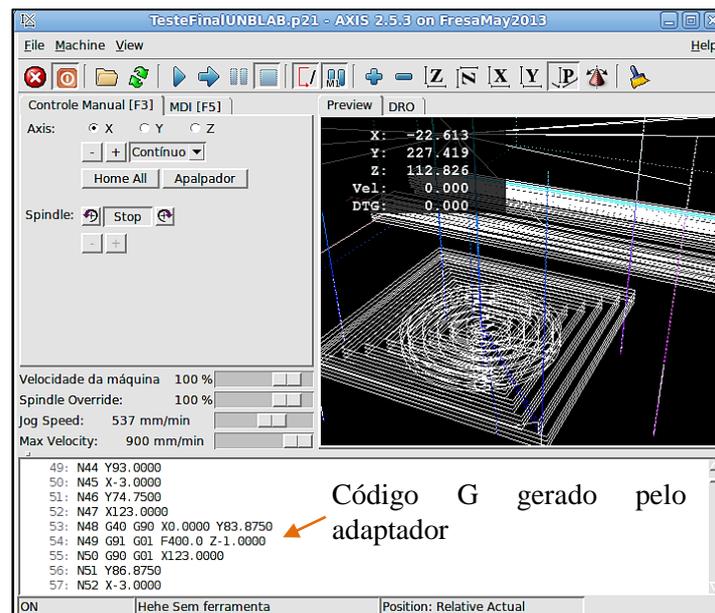


Figura 6.9: Trajetórias geradas pelo adaptador visualizadas na GUI AXIS

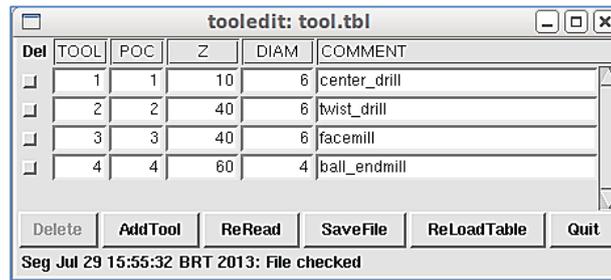


Figura 6.10: Editor de ferramentas com as ferramentas usadas no Código G

O processo de conversão fonte-fonte do arquivo físico p21 a código G e a criação do arquivo “tool.tbl” é realizado de forma automática e transparente. Com isso, o operador da Router CNC pode abrir no LinuxCNC esse tipo de arquivo de forma nativa. A Figura 6.11 exibe a GUI AXIS e a GUI de visualização do arquivo físico p21.

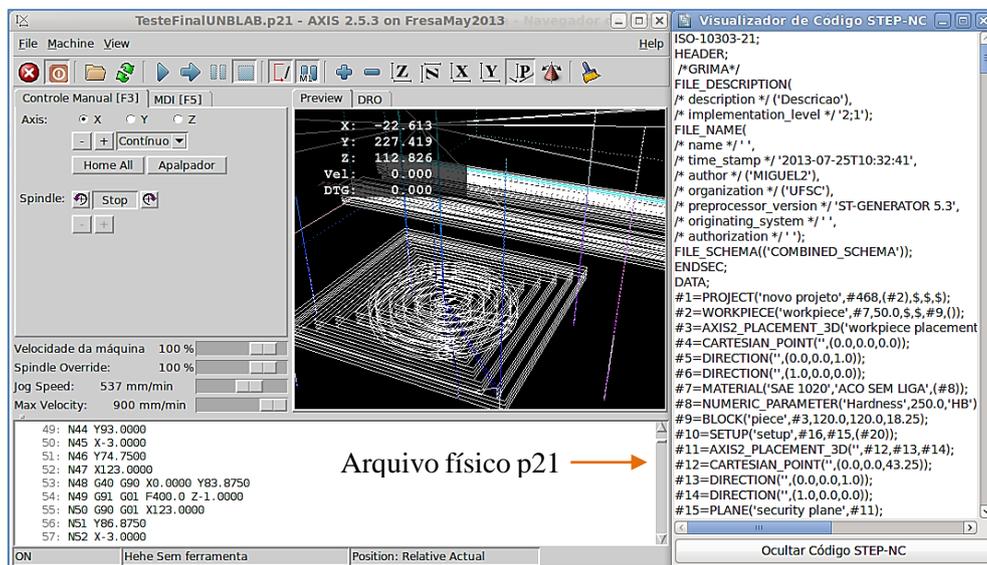


Figura 6.11: GUI AXIS e a GUI de visualização

Os passos mostrados nas figuras anteriores revelam que o adaptador funciona sem nenhum problema em relação à conversão de um programa STEP-NC parte 21 a um programa peça em código G (RS274NGC). O passo seguinte é efetivar a usinagem da peça teste do estudo de caso. Antes de realizar a usinagem, pode-se verificar que não existem erros nas trajetórias da ferramenta ou outros erros, procedimento este feito usando-se o software Vericut de simulação de usinagem CNC.

#### 6.1.4. Simulação do código G gerado pelo adaptador de STEP-NC a código G em Vericut

A seguir é apresentado o resultado da simulação (no *software* Vericut) do código G gerado pelo adaptador a partir do arquivo físico TesteFinalUNBLAB.p21.

Para realizar a simulação mais real, foi implementada a simulação 3D da Router CNC nesse software. Foi usado então para a usinagem o código G gerado pelo adaptador a partir do arquivo físico TesteFinalUNBLAB.p21. A execução do adaptador foi feita de forma independente do

LinuxCNC, i.e. o adaptador foi executado usando o IDE Eclipse. O programa NC da peça teste foi carregado no Vericut. O resultado da simulação da usinagem CNC é exibido nas Figuras 6.12 e 6.13. Foi também gravado um filme deste processo, o qual está disponibilizado no site *web Youtube*. O link para visualizar o vídeo é <http://youtu.be/3DCRe0UtoR8>.

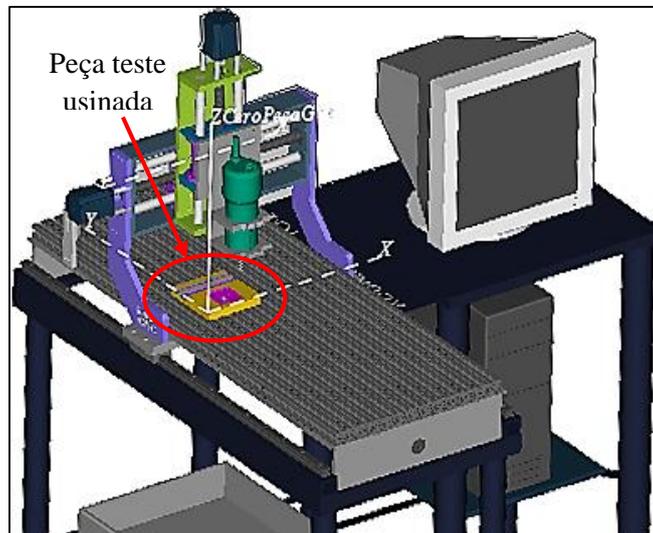


Figura 6.12: Resultado da simulação da Router CNC com a peça teste usinada

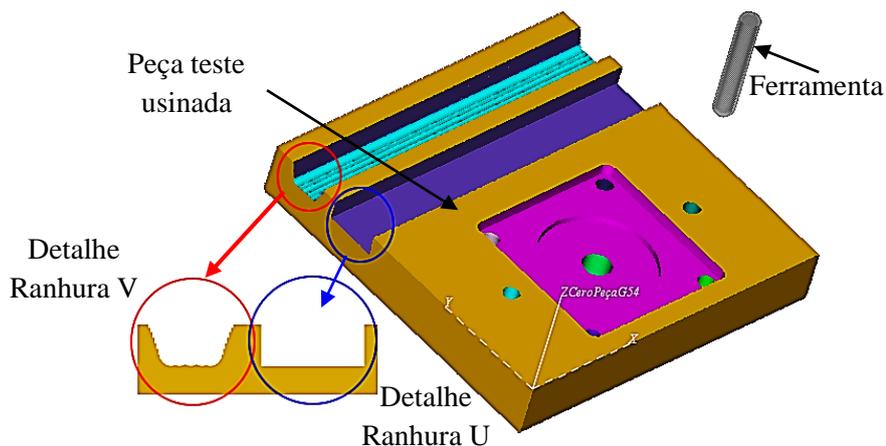


Figura 6.13: Resultado da simulação da peça teste usinada

Observando-se o processo de simulação do código G da peça teste no software Vericut, pode-se concluir que não há erros no código G gerado pelo adaptador. Portanto, continua-se com o processo real de usinagem da peça teste do estudo de caso.

### 6.1.5. Usinagem da peça de trabalho

Inicialmente deve-se realizar um procedimento para preparar a máquina para a usinagem. Isso é conseguido colocando-se os eixos (XYZ) da máquina nos pontos “Home” (a origem das coordenadas de cada eixo da máquina). Em seguida deve-se fixar a peça bruta na mesa de trabalho da Router CNC, levando em consideração que nenhum tipo de batimento entre o sistema de fixação e a ferramenta de corte pode ocorrer. Depois deve-se colocar a referência do “zero-peça” usando o sistemas de coordenadas G54. Por último é colocada a referência para o ponto para troca de ferramenta usando o sistema de coordenadas G55. A Figura 6.14 mostra a fixação.

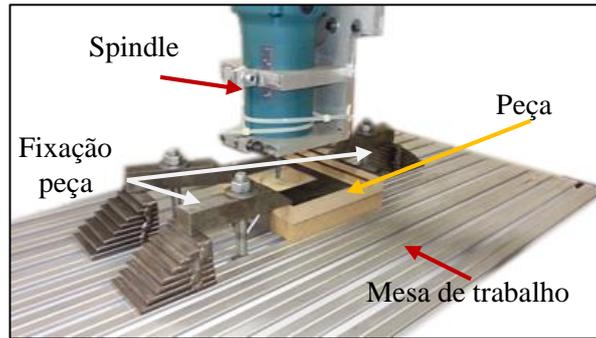


Figura 6.14: Fixação da peça teste na mesa de trabalho

Depois de realizar as instruções anteriores, pode-se usinar a peça teste com o código G gerado pelo adaptador, devendo antes recarregar as ferramentas do arquivo TBL. A seguir é apresentado o resultado final da usinagem da peça teste na Figura 6.15.

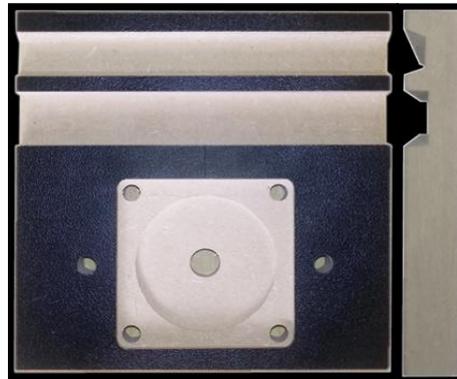


Figura 6.15: Resultado da usinagem da peça do segundo estudo de caso

Terminado o processo de usinagem da peça teste, segue-se com a comparação das medidas teóricas (segundo a Figura 6.1) com as práticas (obtidas da peça teste usinada). Isso é realizado com o objetivo de validar o código G gerado pelo adaptador com os *workingsteps* do arquivo físico p21 de entrada. É validado também o arquivo TBL gerado, correspondente ao formato usado pelo EMC. Na Figura 6.16 são identificadas as features com letras e um número.

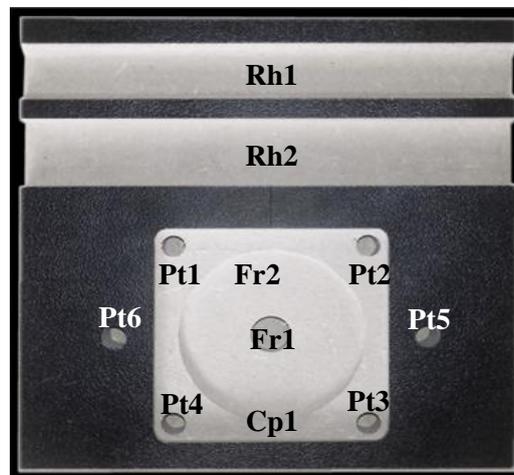


Figura 6.16: Identificação das *features* da peça usinada

### 6.1.6. Comparação entre as medidas teóricas (esboço peça teste) com as práticas (peça teste usinada)

A seguir são comparadas as medidas teóricas e práticas da peça teste. As medidas práticas (6 amostras por *feature*) são obtidas usando o procedimento descrito no estudo de capacidade da Router CNC (apêndice F figuras F.7E e F.7F), ou seja, é usada uma máquina de medição por coordenadas para obter as amostras (medições). Os pontos de referência para a medição das características dimensionais das features da peça teste são apresentados na Figura 6.12.

Já uma medida prática é tomada como média das amostras encontradas na peça teste usinada para cada medição levando-se em conta os erros geométricos e de posição da Router (ver apêndice F). Por fim, a comparação das características dimensionais teóricas com as práticas das *features* da peça teste é realizada. A Tabela 6.2 mostra a medições encontradas na peça teste usinada segundo a Figura 6.17

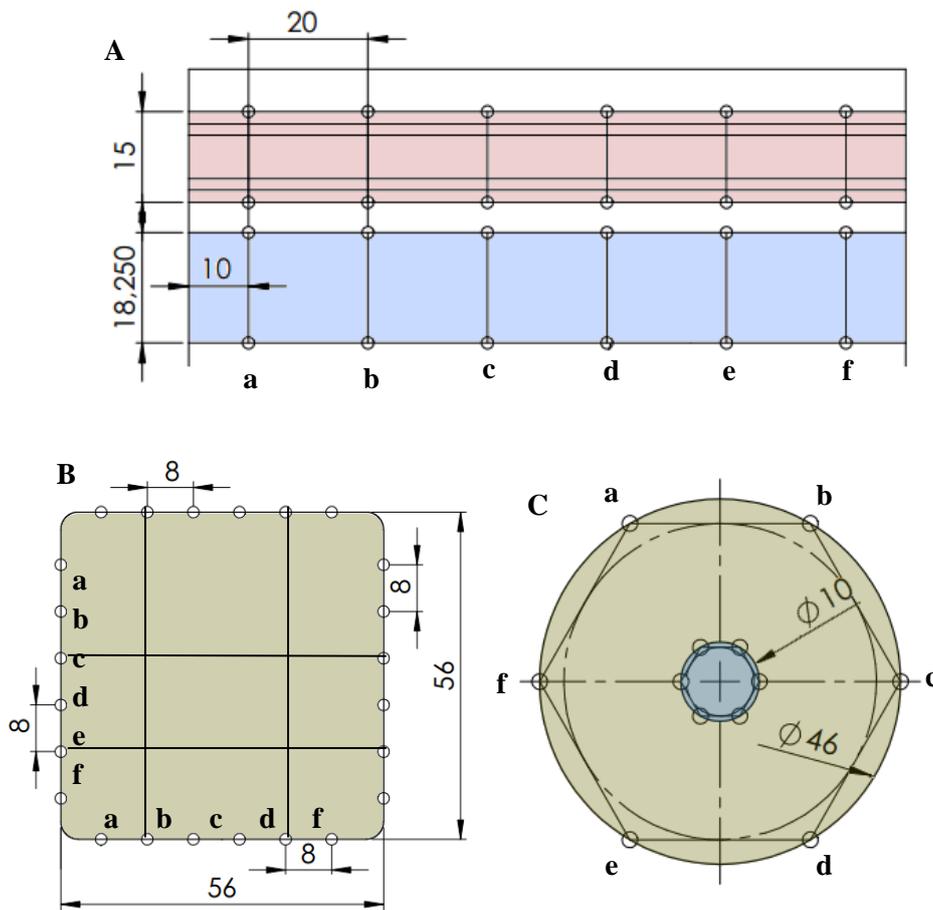


Figura 6.17: Divisão das features da peça teste – pontos para medição

Na Figura 6.17, pode-se observar a individualização das features da peça teste. Isso permite identificar melhor os pontos de referência onde são tomadas as amostras. No caso das ranhuras Rh1 e Rh2 são medidos os comprimentos das linhas, e.g. Rh1  $a=18,25$ ,  $b=18,251$  e assim sucessivamente. No *pocket* Cp1 as medições são, por exemplo, para horizontal Cp1  $h_a=56,012$  e para vertical Cp1  $av=56,012$  e assim sucessivamente. Já no furo (circunferência), o medidor solicita 5 ou mais pontos de referência para realizar a medição. No final, o mesmo medidor proporciona a medida de circularidade. As medições para o furo 1 são e.g. Fr1  $ac=46,012$  e assim sucessivamente.

Tabela 6.2: Medidas encontradas na peça teste usinada

| <i>Feature</i>                          | Operação   | Medidas encontradas                                      |  |  |  |
|---|------------|--|--|--|--|
|   |            | Compr. (X) mm  | Altura (Y) mm  | Largura (Z) mm                                     | Diâmetro mm  |
| Ranhura1 – Rh1                          | Fresamento | 120  | 15,002<br>15,012<br>15,010<br>15,016<br>15,009<br>15,019 | 5,009<br>5,019<br>5,017<br>5,004<br>5,002<br>5,011 | Não se aplica  |
| Ranhura2 – Rh2                          | Fresamento | 120  | 18,253<br>18,264<br>18,267<br>18,255<br>18,251<br>18,262 | 7,009<br>7,015<br>7,001<br>7,018<br>7,012<br>7,005 | Não se aplica  |
| Furos – P1 a P6                         | Furação    | Não se aplica  | Não se aplica  | Passante   | 6 mm   |
| Furo – F1                               | Fresamento | Não se aplica  | Não se aplica  | Passante   | 10,010<br>10,007<br>10,002<br>10,013<br>10,007<br>10,017 |
| Furo base plana – F2                    | Fresamento | Não se aplica  | Não se aplica  | 6,008<br>6,006<br>6,007<br>6,009<br>6,002<br>6,016 | 46,007<br>46,019<br>46,001<br>46,017<br>46,014<br>46,013 |
| Cavidade retangular 1 fundo plano – Cp1 | Fresamento | 56,002<br>56,012<br>56,017<br>56,006<br>56,014<br>56,019 | 56,010<br>56,002<br>56,008<br>56,017<br>56,013<br>56,005 | 4,008<br>4,010<br>4,008<br>4,014<br>4,008<br>4,009 | Raio ortogonal 3 mm                                      |

Na Tabela 6.3 é apresentada a média das amostras para cada feature.

Tabela 6.3: Medidas encontradas na peça teste usinada

| <i>Feature</i> | Operação   | Médias das amostras |               |                |               |
|----------------|------------|---------------------|---------------|----------------|---------------|
|                |            | Compr. (X) mm       | Altura (Y) mm | Largura (Z) mm | Diâmetro mm   |
| Rh1            | Fresamento | 120                 | 15,012        | 5,011          | Não se aplica |
| Rh2            | Fresamento | 120                 | 18,253        | 7,01           | Não se aplica |
| Pt1 a Pt6      | Furação    | Não se aplica       | Não se aplica | Passante       | 6             |
| Fr1            | Fresamento | Não se aplica       | Não se aplica | Passante       | 10,009        |
| Fr2            | Fresamento | Não se aplica       | Não se aplica | 6,008          | 46,012        |
| Cp1            | Fresamento | 56,012              | 56,009        | 4,009          | Raio ort 3    |

A comparação dos valores teóricos (ver Figura 6.1 e Figura 6.17) com os práticos (Tabela 6.3) é apresentada na tabela 6.4.

Tabela 6.4: Comparação entre medidas teóricas e práticas da peça teste

| Feat         | Compr. (X) mm |         | Altura (Y) mm   |         | Largura (Z) mm |          | Diâmetro mm |         |
|--------------|---------------|---------|-----------------|---------|----------------|----------|-------------|---------|
|              | Teórico       | Prático | Teórico         | Prático | Teórico        | Prático  | Teórico     | Prático |
| Rh1          | 120           | 120     | 15±0,05         | 15,012  | 7±0,05         | 5,011    | NA          | NA      |
| Rh2          | 120           | 120     | 18,25<br>± 0,05 | 18,253  | 7±0,05         | 7,01     | NA          | NA      |
| Pt1 a<br>Pt6 | NA            | NA      | NA              | NA      | Passante       | Passante | 6           | 6       |
| Fr1          | NA            | NA      | NA              | NA      | Passante       | Passante | 10±0,05     | 10,009  |
| Fr2          | NA            | NA      | NA              | NA      | 6±0,05         | 6,008    | 46±0,05     | 46,012  |
| Cp1          | 56±0,05       | 56,012  | 56±0,05         | 56,009  | 4±0,05         | 4,009    | Rort 3      | Rort 3  |

Deve ser elucidado que a Ranhura1 não tem a parte de desbaste da base arredondada (2mm raio) de acordo com o *workinstep* que usa a ferramenta “*BallEndMill*” de 4mm de diâmetro, já que essa ferramenta não estava disponível no momento da usinagem. O raio ortogonal (Rort) da cavidade foi colocado propositalmente para 3mm, pois a ferramenta usada tem um diâmetro de 6mm. Isso permitiu ganhar um pouco de tempo na usinagem da peça teste. Note-se que na Tabela 6.4 usa-se a abreviação NA que significa Não Aplica.

Já o resultado da comparação numérica mostra que os valores teóricos estão próximos dos práticos (note-se que não estão incluídos os erros geométricos e posicionamento da Router) e é, portanto, possível concluir que a Router desenvolvida neste trabalho atende os valores de tolerância na usinagem da peça. Consequentemente, pode-se inferir que o código G gerado sim corresponde com os *workingsteps* do arquivo físico TesteFinalUNBLAB.p21. Já uma comparação visual entre a peça teste usinada (Figura 6.15) e o modelo 3d da peça teste (Figura 6.3) oferece uma ideia de que o resultado da usinagem concorda com o desenho CAD de acordo com o projeto da peça no STEP Modeler.

### 6.1.7. Comparação entre código G feito manualmente e o código G gerado pelo adaptador

A realização do código G feito manualmente foi uma tarefa que levou bastante tempo e muito cuidado a fim de evitar erros, uma vez que muitas trajetórias devem ser programadas. Somado a isso, deve-se levar em conta a troca de ferramenta, *feedrate*, a velocidade de corte e a profundidade de corte e a compensação de raio de corte, entre outros. O código G final é muito extenso para incluí-lo neste texto, portanto é disponibilizado no site web do projeto deste trabalho de mestrado: <https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/>.

A comparação é realizada por segmentos. Esses são referentes aos *workingsteps* (WS). Na figura 6.5 podem-se identificar 21 WS. Assim, para esta comparação são tomadas apenas duas *features* (ver Figura 6.16) Rh2 e Fr1 com um total de 4 WS.

Na Figura 6.18 são apresentados dois resultados de simulação de usinagem no Vericut. O primeiro é o código G feito manualmente e o segundo é o código G gerado pelo adaptador.

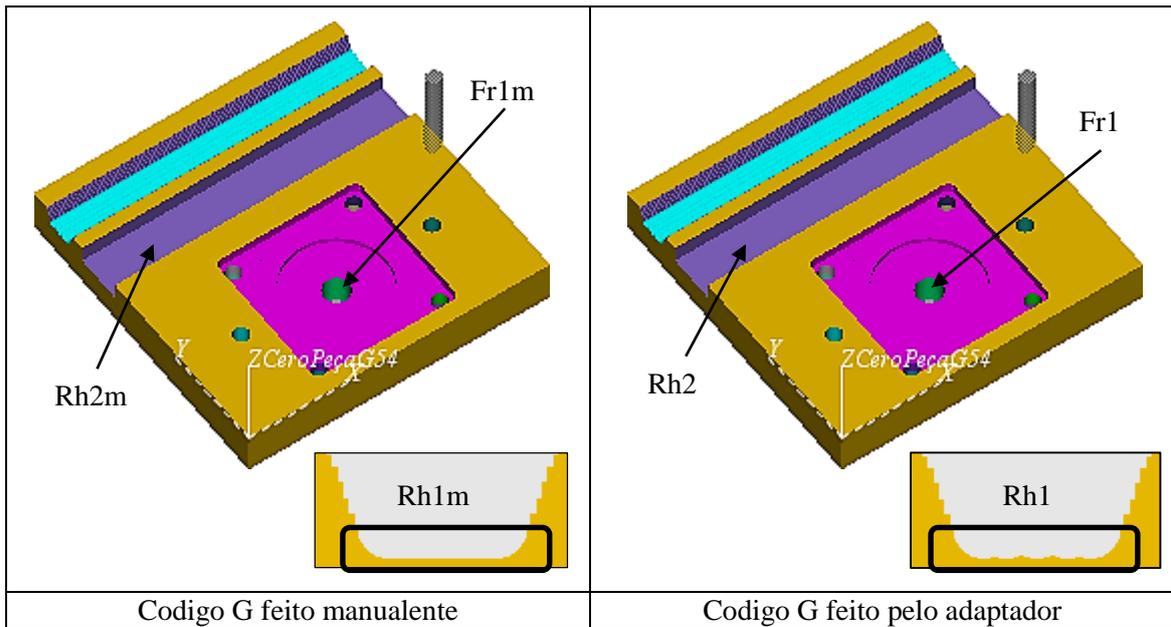


Figura 6.18: Simulação de usinagem CNC no Vericut

Como pode ser observado na figura anterior, o resultado é basicamente o mesmo. Foi utilizado o sistema de inspeção do Vericut para medir as features e encontrou-se que as medidas são as mesmas. A diferença mais clara (embora seja sutil) é o desbaste da base da Ranhura Rh1.

Para identificar a ranhura e o furo do código G manual, adiciona-se uma letra “m” na identificação da feature, i.e. Rh2m e Fr1m.

A comparação entre as medidas de Rh2m e Fr1m e as medidas de Rh2 e Fr1 é apresentada na Tabela 6.5. Os dados são obtidos do sistema de inspeção do Vericut. Todas as medidas têm um erro associado à medição de  $\pm 0.2\text{mm}$  segundo o sistema de inspeção do Vericut.

Tabela 6.5: Medidas encontradas nas peças simuladas usinadas

| Ranhuras( $\pm 0.2\text{mm}$ ) |       | Furos( $\pm 0.2\text{mm}$ ) |     |
|--------------------------------|-------|-----------------------------|-----|
| Rh2m                           | Rh2   | Fr1m                        | Fr1 |
| 18,25                          | 18,25 | 10                          | 10  |

A Figura 6.19 mostra uma montagem de duas peças teste usinadas e um motor.



Figura 6.19: Exemplo de montagem com duas peças teste

## 7. CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões do desenvolvimento de uma Router CNC aderente a norma STEP-NC assim como sugestões de trabalhos futuros.

### 7.1. CONCLUSÕES DO TRABALHO

Neste trabalho foi concebido e desenvolvido um adaptador de STEP-NC/RS274 (NGC) para o controlador LinuxCNC tornando-o aderente à norma ISO14649 em plataforma Linux Ubuntu. Também foi projetada e fabricada uma máquina-ferramenta CNC tipo Router onde foi validado o controlador LinuxCNC aderente à norma ISO14649. Portanto conseguiu-se que a Router CNC, tendo como controlador o LinuxCNC, execute arquivos físicos no formato STEP-NC parte 21. O controlador LinuxCNC ficou aderente à norma STEP-NC (sendo do tipo I) de acordo com o exposto por SUH et al. [31] sobre controladores aderentes à norma STEP-NC. Foi feita uma revisão extensa sobre manufatura integrada, sistemas CAD/CAPP/CAM/CNC e o modelo de dados fornecido pelo padrão STEP-NC. Como resultado, concluiu-se que o padrão STEP-NC é o novo modelo de dados para o intercâmbio de informações entre os sistemas CAD/CAPP/CAM e CNC, pois possibilitou a manufatura integrada por meio do *STEP Modeler* e o CNC aderente a STEP-NC desenvolvido neste trabalho. O padrão ISO14649 substituirá o antigo padrão ISO6983 (Código G) convertendo-se na nova linguagem de programação para máquinas NC, sendo a Router CNC aderente a norma STEP-NC desenvolvida neste trabalho um exemplo disso. A revisão dos projetos relacionados com a norma ISO14649 permitiu identificar algumas das funções que deveria ter o controlador NC. Este conhecimento forneceu as bases para a modelagem do CNC aderente a STEP-NC.

Na modelagem foi possível identificar dois componentes que fazem parte do CNC aderente a STEP-NC: um adaptador de STEP-NC a Código G e o controlador LinuxCNC, no qual foi adicionado o anterior. Além disso, foi modelado um sistema geral que abrange o ciclo de vida do projeto de uma peça prismática desde o projeto CAD até a sua usinagem. Nesse modelo, contemplou-se o uso do *software STEP Modeler* para criar o projeto CAD/CAPP/CAM. A validação em relação ao funcionamento desse software foi feita por meio de testes. Na Tabela 4.10 da sessão 4, foi apresentado um resumo das *features* tratadas pelo adaptador segundo o arquivo STEP-NC parte 21 gerado pelo *STEP Modeler*.

Embora tenham sido feitos muitos testes do arquivo físico p21 gerado pelo *STEP Modeler*, um estudo muito mais amplo e detalhado do uso das *features* em conjunto deve ser realizado e, conseqüentemente, testar a geração e coerência do arquivo físico p21 enquanto as informações contidas nele.

Um arquivo físico p21 gerado no *STEP Modeler* foi usado como entrada no CNC aderente a STEP-NC para se executar o processo de usinagem. Embora não tenha sido implementado, foi idealizado o monitoramento da Router CNC usando o padrão *MTConnect*. Desse modo, do modelo do sistema geral (chamado “SISTEMA INTEGRADO CAD/CAPP/CAM/CNC aderente a STEP-NC”) foram implementados apenas os dois componentes: implementação do adaptador de STEP-NC para código G e implementação da aderência do controlador LinuxCNC à norma ISO14649 por meio do adaptador. Como já foi mencionado, o controlador LinuxCNC aderente a STEP-NC é então do tipo I (sessão 2.10) de acordo com o exposto por SUH et al. [31].

A implementação do adaptador de STEP-NC para código G significou um completo desafio para o autor, pois a programação do algoritmo foi bastante complexa e extensa. A implementação do algoritmo do adaptador foi feita usando-se a linguagem Java.

Embora tenham sido consideradas várias formas de atender às necessidades que a modelagem funcional do adaptador exigia como a criação de um compilador, um intérprete ou um conversor fonte-fonte para adaptar o programa de peça em STEP-NC a um programa de peça em código G/M, concluiu-se que a solução mais prática foi a conversão fonte-fonte, pois o arquivo físico p21 gerado pelo *STEP Modeler* já possui uma análise léxica, sintática e semântica desse arquivo. Isso permitiu que, na implementação do adaptador, fosse apenas necessário ler as informações (que o autor chama no texto de informações básicas e avançadas) contidas no arquivo físico p21. A partir dessas informações, foram gerados o código G e o arquivo *tool.tbl*, que contém as ferramentas de corte (e suas características) usadas no processo de usinagem. O adaptador funciona apenas com as *features* mostradas na Tabela 4.1.

O adaptador usa um arquivo de configuração externo que pode ser modificado pelo operador da Router. Nesse arquivo são colocados os parâmetros de corte da Router CNC e as rotas onde serão salvos o arquivo TBL e o repositório do adaptador. Por último, o protótipo do adaptador foi exportado como uma aplicação Java (IDE Eclipse, opção exportar “*Runnable JAR file*”). Essa aplicação (programa) foi usada como filtro no LinuxCNC, o que permitiu a aderência à norma STEP-NC.

A implementação da aderência do controlador LinuxCNC à norma ISO14649 por meio do adaptador requereu muita revisão de literatura em relação à estrutura do controlador e aos arquivos importantes, tais como o arquivo INI, o arquivo TBL e o arquivo da GUI do controlador chamado “*AXIS.tcl*”.

Identificou-se que o arquivo INI é o encarregado de guardar todas as configurações do controlador LinuxCNC e da Router. Foi neste arquivo que agregou-se a função de abrir o arquivo físico p21 (ou programa de peça em STEP-NC STP) usando o adaptador como um filtro (programa executável Java) para realizar essa tarefa. O filtro (adaptador) carrega o programa de peça em STEP-NC e o converte a código G de forma automática e transparente para o operador da Router CNC. Além disso, o adaptador gera o arquivo *tool.tbl* e o substitui pelo arquivo original que se encontra na pasta dos arquivos de configuração do LinuxCNC e da Router.

Em relação à GUI *AXIS*, foi agregado um novo elemento no menu “*View*”, chamado “*STEP-NC Code Viewer*”. Esse elemento ou comando se relaciona com a adição de uma nova função no LinuxCNC. Essa função é encarregada de abrir uma janela (GUI de visualização) onde é exibido o arquivo físico p21 que foi convertido a código G pelo adaptador. A GUI de visualização foi implementada com o *software* GLADE e a linguagem de programação Python.

Como resultado final, foi possível criar a GUI de visualização do arquivo físico p21. Essa GUI é executada em conjunto com a GUI *AXIS* e permite que o operador da Router CNC confira o programa de peça em STEP-NC.

Além disso, foi criada uma máquina CNC virtual no software Vericut baseada no projeto CAD da Router CNC desenvolvida neste trabalho. Essa máquina CNC permite simular e otimizar o código G gerado pelo adaptador. O projeto da máquina CNC em Vericut está disponibilizado no site [web https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/](https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/).

A validação do adaptador e, conseqüentemente, do controlador LinuxCNC foi feita por meio de um estudo de caso de uma peça teste usinada na Router CNC aderente à norma STEP-NC. Esse estudo demonstrou o funcionamento do adaptador relacionado à concordância do código G gerado com os *workingsteps* do programa STEP-NC *part 21* gerado pelo STEP *Modeler*. Foi analisada também a comparação entre a usinagem simulada no Vericut de duas peças. Uma com código G feito manualmente e outra com código G gerado pelo adaptador a partir de um arquivo físico p21 gerado no STEP *Modeler*. O código G gerado manualmente seguiu o plano de processo usado no projeto CAD/CAPP/CAM do STEP *Modeler* da peça teste. O resultado foi basicamente o mesmo, obtendo-se, na usinagem manual, um maior detalhamento do desbaste numa das features usadas. Um fator importante e não medido na simulação da usinagem foi a determinação do tempo de usinagem da peça teste com código G manual e do tempo de usinagem da peça teste com código G gerado pelo adaptador. Ainda assim, foi validado o adaptador usado no LinuxCNC permitindo verificar a aderência à norma STEP-NC.

Por último e não menos importante, está o projeto e fabricação de uma Router de três eixos. Inicialmente foram estudadas algumas máquinas-ferramenta relacionadas com o tipo de máquina que o projeto precisava. Fresadoras CNC e Routers CNC foram a base do projeto de desenvolvimento da Router neste trabalho. Usou-se o programa de simulação CAD SolidWorks Premium 2010 para o desenho CAD da Router. Esse *software* forneceu o necessário para simular desde as peças até os movimentos e limites da máquina. Por fim, as peças que compõem a Router, inicialmente desenhadas em CAD, foram fabricadas no laboratório de Mecânica do SG9 da Universidade de Brasília. A fabricação das peças foi realizada em parceria com os técnicos do laboratório e o autor. Conclui-se, assim, que o projeto, a fabricação e a montagem da Router é um trabalho muito meticuloso, uma vez que todos os componentes devem ser montados de forma correta a fim de evitar erros para que a máquina funcione tal qual como projetada.

A Router foi configurada com o controlador LinuxCNC aderente a STEP-NC. A validação da Router CNC foi feita por meio do estudo de capacidade da máquina. Isto permitiu encontrar os erros de retilinearidade, paralelismo, perpendicularidade e posicionamento dos eixos XY assim como erro de posicionamento do eixo Z.

Finalmente, deve-se notar que este trabalho de mestrado conta com um projeto em Google *code*. O endereço do site web do projeto é: <https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/>. Nesse site, encontra-se o código do algoritmo do adaptador no item “*Source*”, assim como exemplos de programas STEP-NC *part 21*, imagens de usinagem, fotos, figuras, estudo de casos e filmes do processo de usinagem e ciclo de vida da peça, simulação no Vericut, entre outros itens usados neste trabalho. Os filmes também estão disponíveis no canal de Youtube <http://www.youtube.com/user/meduag>

## 7.2. SUGESTÕES PARA TRABALHOS FUTUROS

A implementação computacional do adaptador de STEP-NC a código G é a parte principal da aderência do LinuxCNC à norma ISO14649. Uma primeira versão do adaptador foi gerada, mas ainda há muito para evoluir. Devido às limitações temporais para conclusão deste trabalho, não foi possível implementar várias features das disponibilizadas no STEP *Modeler* e que não foram tratadas. A seguir, é apresentada uma lista de trabalhos futuros que poderão complementar o adaptador e, conseqüentemente, a aderência do controlador LinuxCNC à norma STEP-NC.

1. Realizar uma parceria com o projeto do STEP Modeler para corrigir erros no arquivo físico P21 de acordo as features da tabela 7.1.
2. Acrescentar as features do STEP *Modeler* não tratadas pelo adaptador para a geração das trajetórias.
3. Dotar de inteligência o adaptador para realizar otimização do código G gerado, usando algoritmos de otimização de trajetórias de ferramenta.
4. Implementar no adaptador um sistema de informação de erros do formato p21.
5. Atualizar o adaptador em termos da aquisição de informações (básicas e avançadas) de acordo com a norma STEP-NC AP238
6. Realizar uma interface de usuário individual para o adaptador, permitindo que seja usado independentemente do tipo de controlador e acrescentando a opção de gerar código G em diferentes formatos a partir de código STEP-NC AP 238.
7. Elaborar a documentação da implementação computacional do algoritmo do adaptador a fim de que seja mais fácil mantê-lo e modificá-lo por outros programadores.
8. Divulgar e disponibilizar o adaptador na comunidade LinuxCNC.org para que seja adotado como uma nova funcionalidade do controlador, o que permitirá a disseminação do padrão ISO14649 (AP21 e AP238) substituindo o antigo código G (RS274NGC) e comportando a integração da manufatura entre sistemas CAD/CAPP/CAM e CNC.
9. Implementar o padrão *MTConnect* no Router com LinuxCNC aderente a STEP-NC.
10. Realizar um estudo detalhado da capacidade da Router CNC.
11. Melhorar o sistema de acoplamento mecânico entre o motor de passo e o parafuso (aplica para todos os eixos).
12. Implementar no adaptador a possibilidade de compensação de erros geométricos e de posicionamento da máquina no código G para melhorar a capacidade da Router CNC e, conseqüentemente, o processo de usinagem.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] S. J. KEMMERER, "STEP The Grand Experience. Manufacturing Engineering Laboratory. National Inst. of Standards and Technology", Gaithersburg (MD), 2009.
- [2] R. UBERTINO JR e N. STEPHEN T, "Estrutura de Dados para Sistemas CAD/CAM aderentes à STEP", VI Congresso Ibero-Americano de Engenharia Mecânica - Universidade de Coimbra, Coimbra, Portugal, 15-18, 2003.
- [3] ISO10303-1, "A integration –Product Data Representation and Exchange-Part 1: Overview and fundamental principles", ISO, 1994.
- [4] T. R. KRAMER, F. M. PROCTOR e E. R. MESSINA, "The NIST RS274NGC Interpreter - Version 3", 2010.
- [5] 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.
- [6] ISO/FDIS/DIS , "14 Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers", 2002.
- [7] ISO 14649-1, "Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers -- Part 1 -- Overview and fundamental principles", 2003.
- [8] A. J. ALVARES, "Uma Metodologia para Integração CAD/CAPP/CAM Voltada para Manufatura", Florianópolis, 2003.
- [9] M. W. a. J. Wolf., STEP-NC. "The STEP compliant NC Programming Interface: Evaluation, Ascona/Switzerland, 2001.
- [10] I. 10303-21, "Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure", 2002.
- [11] TICONA JULIO C. B., "Um Sistema Para O Projeto E Fabricação De Peças Mecânicas A Distância Via Internet Aderente À Norma ISO 14649 (STEP-NC)", Trabalho de Doutorado em Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, 223f, 2011.
- [12] H. WANG, "New Control Strategy for CNC Machines Via STEP-NC", Intelligent and Interoperable Manufacturing Systems (IIMS), Department of Mechanical Engineering, The University of Auckland, New Zealand, 2009.

- [13] L. J. KRAJEWSKI e L. P. RITZMAN, "Administración de operaciones: estrategia y análisis", Pearson Educación, 2000, pp 157-160.
- [14] K. SEROPE, S. STEVEN R. e R. ULISES, "Manufactura, ingeniería y tecnología", Pearson Educación, 2002, pp. 1075-1077.
- [15] M. GROOVER, "Automation, Production Systems, and Computer-Integrated Manufacturing", Third Edition, Prentice Hall, 2007.
- [16] R. HERNANDEZ OMAR, M. NEGRON DAVID e S. ROMERO, "Introduction to Engineering: An Industrial Approach", Cengage Learning Editores, 2006, pp. 195-198.
- [17] X. XU, "Chapter X - Integration of CAD/CAPP/CAM/CNC," em *Integrating Advanced Computer - Aided Desing, Manufacturing, and Numerical Control: Pinciples and Implementations*, Hershey, IGI Global, 2009, pp. 231-245.
- [18] X. XU e J. MAO, "Development of the client tier for a STEP compliant CAPP," em *Proceedings of the 6th International Conference on Frontiers of Design and*, Xi'an, 2004, pp. 21-23.
- [19] US\_PRO, "Initial Graphics Exchange Specification 5.3", MerAI Productions, 2006.
- [20] X. XU, "Chapter XI - Integration Based on STEP Standars," em *Integrating Advanced Computer - Aided Desing, Manufacturing, and Numerical Control: Pinciples and Implementations*, Hershey, IGI Global, 2009, pp. 246-265.
- [21] X. XU, M. H. J., S. NEWMAN e K. M., "STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC", *International Journal of Production Research*, Vol. 43, pp.3703–3743, 2005.
- [22] S. T. NEWMAN, R. D. ALLEN e J. R. ROSSO, "CAD/CAM solutions for STEP-compliant CNC," *INT. J. Computer Integrated Manufacturing*, vol. 16, nº 7–8, p. 590–597, 2003.
- [23] "ABOUT ISO", International Organization for Standardization, Tech. Rep. [Online]. Disponível: <http://www.iso.org/iso/home/about.htm>, 2013.
- [24] 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.
- [25] 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.
- [26] W. MAEDER, K. NGUYEN VAN e R. JACQUES, "Standardisation of the Manufacturing Process : the IMS STEP-NC project", ESPRIT Project, Tech. Rep. [Online]. Disponível: [http://www.step-nc.org/data/IPLnetWorkshop2002\\_paper\\_27082002.pdf](http://www.step-nc.org/data/IPLnetWorkshop2002_paper_27082002.pdf), set, 2002.

- [27] R. S. U. ROSSO JÚNIOR, "STEP Compliant CAD/CAPP/CAM System For Rotational Asymmetric Parts", 175 f. Thesis (Doctoral) - Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, 2005.
- [28] R. S. PRESSMAN, "Ingeniería de Software, un enfoque práctico 5ed", Mc Graw Hill, 2002.
- [29] SCRA, STEP APPLICATION HANDBOOK ISO 10303, VERSION 3, SCARA Collaborating To Advanced Tecnology , 2006.
- [30] NIST, "STEP The Grand Experience", Manufacturing Engineering Laboratory. National Institute of Standards and Technology. Gaithersburg (MD), USA, 1999.
- [31] S. SUK-HWAN, K. SEONG-KYOON, C. DAE-HYUK e IA, "Theory and Desing of CNC Systems", Springer, 2008.
- [32] X. XU e N. ANDRE, "Advanced Desing and Manufacturing Based on STEP", STEP in a Nutshell , Springer- Verlag London Lirnited, 2009.
- [33] ISO, "ISO10303-11, EXPRESS (data modeling language)", Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual.
- [34] A. MARK, "Conéctese al STEP NC", Artículo de Revista, Metalmecánica, Vol. 7 No. 6, pp. 32-34, Otubre 2002.
- [35] 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.
- [36] ISO14649-10, "Industrial automation systems and integration -- Physical device control, ISO 14649 Data model for Computerized Numerical Controllers, Part 10: General Process Data", ISO, 2003.
- [37] R. S. U. ROSSO JÚNIOR e S. T. NEWMAN, “Estrutura de Dados para Sistemas CAD/CAM aderente à STEP,” em *Proceedings of VI Congresso Ibero-Americano de Engenharia Mecânica - Universidade de Coimbra*, Coimbra, 2003.
- [38] X. XU e Q. HE, "Striving for a totalintegration of CAD, CAPP, CAM and CNC", obotics and Computer-Integrated Manufacturing 20, pp 101–109, New Zealand: ELSEVIER, 2004.
- [39] ISO 14649-11, "Industrial automation systems and integration -- Physical device control, ISO 14649 Data model for Computerized Numerical Controllers, Part 11: Process Data For Milling", ISO, 2002.
- [40] ISO 14649-111, "Industrial automation systems and integration -- Physical device control, ISO 14649 Data model for Computerized Numerical Controllers,Part 111: Tools For Milling",ISO, 2002.

- [41] STEP TOOLS, Super Model Project, Tech. Rep. [Online]. Disponível: <http://www.steptools.com/>, STEP Tools, 1999.
- [42] STEP TOOLS, STEP-NC MACHINE, Tech. Rep. [Online]. Disponível: <http://www.steptools.com/products/stepncmachine/>, STEP Tools, 2013.
- [43] MASTERCAM, "STEP-NC and Mastercam — The Future is Only a Few", Tech. Rep. [Online]. Disponível: <http://www.mastercam.com/camzone/Newsletter/Archives/3rd2003-final.pdf>, Mastercam Notes, 2003.
- [44] IMS STEP-NC, step-nc,official STEP-NC page, Tech. Rep. [Online]. Disponível: <http://www.step-nc.org/index.htm>, 2001.
- [45] P. IMS, STEP-NC, STEP-compliant Data Interface for Numerical, IMS Project No. 01009, 2003.
- [46] ISO-14649-TOOLKIT, Google code, Toolkit for ISO 14649 / STEP-NC, Tech. Rep. [Online]. Disponível: <http://code.google.com/p/iso-14649-toolkit/>, 2009.
- [47] SEBN2PARS, Google code, A tool for building STEP Part 21 file parsers, Tech. Rep. [Online]. Disponível: <http://code.google.com/p/sebnf2pars/>, 2009.
- [48] NIST SCL, STEP Class Library (SCL), NIST, STEP Class Library Components, Tech. Rep. [Online]. Disponível: <http://www.nist.gov/el/msid/scl.cfm>, 1999.
- [49] X. XU, H. WANG, J. MAO, S. T. NEWMAN, T. R. KRAMER, F. M. PROCTOR e J. L. MICHALOSKI, "STEP—Compliant NC Research: the Search for Intelligent CAD/CAPP/CAM/CNC Integration", International Journal of Product Research, 2004.
- [50] H. K. DENKENA, J. TONSHOFF, A. S. S. SELLE, S. HEUSINGER e G. ROGERS, "Offline-Berechnung der Zerspankräfte in der NC-Programmierung, Predição de forças de corte no fresado de acabamento HSC", 2002.
- [51] SCRA, "Defining the Suite of STEP Application Protocols and the implementation architecture for STEP-enabled parts production within commercial and defense applications", White paper, 2004.
- [52] R. W. Paper, Driving Down the Cost of Spares Provisioning. LSC Group, Warship Support Agency, 2002.
- [53] STEP-NC, STEP-NC Research, Tech. Rep. [Online]. Disponível: [http://mech-web.mech.auckland.ac.nz/manfSys/IIMS/html/international\\_research.html](http://mech-web.mech.auckland.ac.nz/manfSys/IIMS/html/international_research.html), 2012.
- [54] J. GARRIDO CAMPOS, "Standard function blocks for product oriented programmed process data access", Vigo: Springer Link, 2013.

- [55] M. MINHATA, X. XU e V. VYATKIN, "STEPNCMillUoA: a CNC system based on STEP-NC and Function Block architecture", *Int. J. Mechatronics and Manufacturing Systems*, Vol. 2, Nos. 1/2, Auckland: Inderscience Publishers, 2009.
- [56] M. MINHATA, V. VYATKIN, X. XU, S. WONG e Z. AL-BAYAA, "A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks", *Robotics and Computer-Integrated Manufacturing*, Auckland: ScienceDirect, 2008.
- [57] E. HARBS, G. JARENTCHUK, G. H. NEGRI, R. S. U. ROSSO JR., M. DA SILVA HOUNSELL, F. H. HUMEL LAFRATTA e J. C. ESPÍNDOLA FERREIRA, "CNC servoacionado aderente às normas ISO 14649 e IEC 61499", 7º CONGRESSO BRASILEIRO DE ENGENHARIA DE FABRICAÇÃO, Santa Catarina: ABCM, 2013.
- [58] S.-H. SUH e S.-J. SHIN, "Transformation method of G-Code into STEP-NC part program", Patent No US 8,041,445 B2, United States Patent, 2011.
- [59] S.-H. SUH e S.-U. CHEON, "Method for automatically generating part program for use in STEP-NC", Patent No US 6,795,749 b2, United States Patent, 2002.
- [60] S. SUH, "Architecture and implementação of a shop-floor programming system for STEP-compliant CNC", ELSEVIER, 2003.
- [61] ISO10306-AP238, "Industrial automation systems and integration -- Product data representation and exchange - Part 238: Application protocol: Application interpreted model for computerized numerical controllers", ISO, 2007.
- [62] J. GARRIDO CAMPOS e M. HARDWICK, "Manufacturing traceability automation using features and nc-functions", *International Journal of Computer Integrated Manufacturing*, Vigo: Scholarone Manuscript Central, 2010.
- [63] STEP TOOLS, STEP-NC MACHINE, Tech. Rep. [Online]. Disponível: [http://www.steptools.com/library/stepnc/tech\\_resources/ap238is\\_review](http://www.steptools.com/library/stepnc/tech_resources/ap238is_review), NY: STEP Tools, 2006.
- [64] A. VIJAYARAGHAVAN, W. SOBEL, A. FOX, D. DORNFELD e P. WARNDORF, "*Improving machine tool interoperability using standardized interface protocols: MTConnect*", Atlanta, GA, USA: Proceedings of ISFA, International Symposium on Flexible Auto-mation., 2008.
- [65] MTCONNECT, The MTConnect Institute, Tech. Rep. [Online]. Disponível: [http://mtconnect.org/index.php?option=com\\_frontpage&Itemid=1](http://mtconnect.org/index.php?option=com_frontpage&Itemid=1), 2011.
- [66] R. FIRMAN e X. XU, "*Advanced CNC system with in-process feed-rate optimisation*", New Zealand: Robotics and Computer-Integrated Manufacturing - SciVerse ScienceDirect, 2012.
- [67] LINUXCNC, "Tool Table", Enhanced Machine Controller, Tech. Rep. [Online]. Disponível: [http://www.linuxcnc.org/docs/2.4/html/gcode\\_tool\\_compensation.html](http://www.linuxcnc.org/docs/2.4/html/gcode_tool_compensation.html), 2012.

- [68] K. SCHÜTZER e A. L. HELLENO, “Importância do recurso de associatividade entre os sistemas de CAD/CAMna geração do programa NC.,” em *O Mundo da Usinagem*, São Paulo, 2005, pp. 27-32.
- [69] A. C. OLIVEIRA, “Sistema CAM para o fresamento em altas velocidades de moldes e matrizes.,” em *O Mundo da Usinagem*, São Paulo, 2003, pp. 12-17.
- [70] O. WRUBLAK, L. A. PILATTI e B. PEDROSO, “Parâmetros e métodos de usinagem e sua relação com os custos do,” 4to Encontro de Engenharia e Tecnologia dos Campos Gerais, Campos Gerais, 2008.
- [71] J. M. CARBONEL, “Simulador de Mecanizados en Código ISO-6983, 156 f. Thesis (graduação) - UNIVERSIDAD POLITECNICA DE VALENCIA, ESCUELA TECNICA DE INFORMATICA,” Valencia, 2011.
- [72] LINUXCNC/DOCS, “LinuxCNC,” Linuxcnc.org, 13 12 2011. [Online]. Available: <http://www.linuxcnc.org/docs/html/gcode/main.html>. [Acesso em 15 Abril 2013].
- [73] CURSOSGURU, “CadGuru,” 2013 04 05. [Online]. Available: <http://www.cadguru.com.br/cursos/cnc/curso-cnc-basico-programacao-cnc-aplicada-torno-fresa/programacao-cnc-coordenadas-absolutas-incrementais/>.
- [74] LINUXCNC, “GUI Axis - Manual Tool Change,” 01 11 2012. [Online]. Available: <http://linuxcnc.org/docs/html/gui/axis.html#sec:Manual-Tool-Change>. [Acesso em 04 2013].
- [75] L. COSTA DE OLIVEIRA, "Um Sistema de Geração de Trajetórias de Ferramentas em 3 Eixos", Porto Alegre: Universidade Federal do Rio Grande do Sul, Instituto de Informatica, 1997.
- [76] R. CARRILLO e A. LOPEZ, "Tiempo de corte estimado en operaciones de vaciado", Asociación Española de Ingeniería Mecánica, XIX CONGRESO NACIONAL DE INGENIERÍA MECÁNICA, Universidad de Jaén, 2002.
- [77] S. YUWEN e G. DONGMING, "Iso-parametric tool path generation from triangular meshes for free-form surface machining", Springer-Verlag London Limited, 2005.
- [78] J. G. GRIFFITHS, "Toolpath based on Hilbert's curve", Computer-Aided Design, Surrey, England, v. 26, n. 11, p. 839-844, Nov. 1994..
- [79] B. CHOI, M. M. BARASH e D. C. ANDERSON, "Automatic recognition of machined surface from a 3D solid model", Computer Aided Design. March, vol. 16: pp. 81-86, Butterworth & Co Ltd, 1984.
- [80] J. C. ESPINDOLA FERREIRA, “Tecnologia de controle numérico - programação de fresadoras CNC (peças prismáticas),” 2013. [Online]. Available: [http://www.grima.ufsc.br/cnc/transparencias/Aula5\\_Programacao\\_Parte4.pdf](http://www.grima.ufsc.br/cnc/transparencias/Aula5_Programacao_Parte4.pdf).

- [81] E. STEINER, "Coordenadas polares," em *Matemáticas para las ciencias aplicadas*, Oxford University Press Inc. New York. USA, REVERTE, 2003, pp. 59-62.
- [82] V. CGTECTH, Vericut - CNC Machine Simulation, Tech. Rep. [Online]. Disponível: <http://www.vericut.com.br/>, CGTECTH, 2013.
- [83] GOOGLE, "Google code - Google developers," google, Novembro 2012. [Online]. Available: <https://code.google.com>. [Acesso em Julho 2013].
- [84] LINUXCNC, "Introducción a EMC2," LinuxCNC.org, 2011. [Online]. Available: [http://www.linuxcnc.org/docs/2.4/html/common\\_user\\_intro.html](http://www.linuxcnc.org/docs/2.4/html/common_user_intro.html). [Acesso em maio 2013].
- [85] GLADEPROJECT, "Glade - A User Interface Designer," The Glade Project, Março 2013. [Online]. Available: <https://glade.gnome.org/>. [Acesso em julho 2013].
- [86] TCL/TK, "Tcl8.5.14/Tk8.5.14 Documentation," Tcl/Tk, 2008. [Online]. Available: <http://www.tcl.tk/man/tcl8.5/contents.htm>. [Acesso em Dezembro 2012].
- [87] K. AGUANNO, "PMP®, MAPM, Managing Agile Projects", first edition, Ontario: Multimedia Publications Inc., 2005.
- [88] J. HIGHSMITH, "Agile Software Development Ecosystems", Addison Wesley Inc., 2002.
- [89] S. GALVES ROJAS e M. A. MORA MATA, "Compiladores, Traductores y compiladores con Lex/Yacc, JFlex/Cup y JavaCC", Malaga: Universidade de Malaga, 2005.
- [90] A. AHO e J. ULLMAN, "Compiladores: Princípios, Técnicas e Ferramentas", Rio de Janeiro: Guanabara Koogan, 1995.
- [91] URICER, "Apostila de COMPILADORES", Erechim: Universidade Regional Integrada do Alto Uruguai, 2001.
- [92] LYRA, PABLO V. A., "*Desenvolvimento de uma Máquina Fresadora CNC Didática*", Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, Brasília DF, 123f, 2010.
- [93] A. ALAIN, "Understanding CNC Routers", Quebec: FPIInnovations™, Forintek, 2008, pp. 2-4.
- [94] DUTRAMAQUINAS, "Tupia Makita Ref 3709," 2012. [Online]. Available: <http://www.factoryauthorizedoutlet.com/makita/makita-1-4-fixed-base-laminate-trimmer>. [Acesso em 08 2012].
- [95] H. MUNKTELL, "LinuxCNC manual tool change with tool length offset," 2011. [Online]. Available: <http://www.youtube.com/watch?v=9H86iNWPYds>.
- [96] P. D. HOOD e J. K. FLOYD, "Build Your Own CNC Machine", New York: Springer-Verlag and Apress, 2009.

- [97] GRAVIA, "Gravia Industria de Perfilados," 2013. [Online]. Available: <http://www.gravia.net.br/industria-de-perfilados/download/catalogo-tecnico-gravia.pdf>.
- [98] H. PARKER, "Tecnologia Eletromecânica, Apostila 1600.231.01 BR", Jacaré, SP: Parker Hannifin Ind. Com. Ltda, 2003.
- [99] A. OVERAY, "CNC MACHINING Handbook - Building, Programing and Implementation," em *The CNC Controller*, New York, McGraw-Hill, 2011, pp. 69-90.
- [100] H. BRASIL, "SMC-B-PRO GUIA DE UTILIZAÇÃO", Belo Horizonte: HobbyCNC Brasil, 2011.
- [101] CIMM, "Variáveis e parâmetros de corte, CIMM - Cento de Informação Metal Mecânica," 2012. [Online]. Available: [http://www.cimm.com.br/portal/material\\_didatico/4861-variaveis-e-parametros-de-corte#.UZpfdqLVBmx](http://www.cimm.com.br/portal/material_didatico/4861-variaveis-e-parametros-de-corte#.UZpfdqLVBmx). [Acesso em maio 2013].
- [102] N. D. SOUSA JUNIOR, "Qualidade de superfície na produção de peças S2S no setor de usinagem de uma industria de molduras de pinus", Curitiba: Universidade Federal do Paraná, 2009.
- [103] G. WESTERMANN VERLAG e BRAUNSSCHWEIG, "HERRAMIENTAS - MAQUINAS - TRABAJO", Barcelona: Reverté S.A., 1998.
- [104] S. MILLÁN GÓMEZ, "Procedimientos de Mecanizado", Madrid: Editorial Paraninfo, 2006.
- [105] P. COCA REBOLLERO e J. R. JIMENEZ, "TECNOLOGIA MECANICA Y METROTECNIA (8ª ED.)", España: Piramide, 2005.
- [106] T. TURCATO, "Removendo cavaco", Biblioteca Virtual do Estudante Brasileiro - TC2000 - Profissionalizante, 2000.
- [107] R. RAMISCH, Schawnt's Outline: Precalculus 2/Ed. - Pré-Cálculo, New York: McGraw-Hill, 2009, pp. 79-82.
- [108] L. CUNHA SALLES, "MANUAL PRÁTICO DO MECÂNICO," em *GEOMETRIA*, Florianópolis, HEMUS, 2007, pp. 546-570.
- [109] H. J. FINKE, "Controle numérico computadorizado aderente às normas STEP-NC e IEC 61499", Universidade Do Estado De Santa Catarina UDESC, Centro De Ciências Tecnológicas CCT, Departamento De Ciência Da Computação DCC, 2010.
- [110] ROZENFELD, H, Gestão de desenvolvimento de produtos: uma referência para a melhoria do processo., São Paulo: Saraiva, 2006. 542p..
- [111] Spirsin, "Mesa Giratoria SP-MF de fresado", Spirsin, tecnología en rotación, Tech. Rep. [Online]. Available: <http://www.spirsin.com/mesas/ver/mesa-giratoria-de-fresado-sp-mf>.

- [112] S. CARVALHO e W. CAMPOS, Estatística Básica Simplificada, Rio de Janeiro: Elsevier, 2008.
- [113] R. METROLOGY, ROMER, Absolute Arm ROMER - Metrology to go Hexagon Metrology, Tech. Rep. [Online]. Available:[http://www.romer.eu/ROMER-Absolute-Arm\\_711.htm#.UgWtGNLVBmw](http://www.romer.eu/ROMER-Absolute-Arm_711.htm#.UgWtGNLVBmw), ROMER, 2013.
- [114] ISO14649-1, "Industrial automation systems and integration — Physical device control — Data model for computerized numerical", 2003.
- [115] SCARA, "STEP APPLICATION HANDBOOK ISO 10303 VERSION 3", JUNE: SCARA COLLABORATING TO ADVANCE TECHNOLOGY, 2006.
- [116] P. DEITEL e H. DEITEL, "JAVA como programar - Oitava Edição", São Paulo: Pearson, 2010.

# APÊNDICE A. METODOLOGIA DE TRABALHO E CONCEITOS SOBRE TRADUTOR DE LINGUAGEM PARA A MODELAGEM DO CNC ADERENTE A NORMA STEP-NC

## A.1. METODOLOGIA DE TRABALHO

O adaptador proposto é um *software* modelado e implementado fazendo uso de metodologias para o desenvolvimento de *software*. Estas metodologias incorporam novas ideias de desenvolvimento e evoluíram dentro do contexto da engenharia de *software*. Dentre as principais metodologias, tem-se as metodologias ágeis.

### A.1.1. Metodologias ágeis para desenvolvimento de *software*

Os conceitos de desenvolvimento ágil de produtos emergiram nos anos 1980 com os fabricantes de automóveis japoneses. Tais conceitos rapidamente se estenderam para os fabricantes de automóveis norte-americanos, e posteriormente para seus departamentos de tecnologia da informação no final dos anos 1980. Uma vez que essas ideias passaram a ser aplicadas no âmbito da tecnologia da informação, elas se abriram rapidamente para outras organizações e indústrias através da exposição em congressos e periódicos de desenvolvimento de *software* [87].

As metodologias ágeis visam o desenvolvimento de *software* de maneira que as necessidades de velocidade e de flexibilidade sejam satisfeitas usando-se os recursos disponíveis da maneira mais eficiente [88]. Alguns dos métodos ágeis de desenvolvimento de *software* são:

- Programação extrema (*Extreme Programming - XP*)
- *Scrum*
- *Adaptive Software Development (ASD)*.
- *Agile Unified Process (AUP)*.
- Métodos Cristal (*Crystal\_Clear*).
- *Essential Unified Process (EssUP)*.
- *Feature Driven Development (FDD)*.

Estas metodologias têm alguns princípios em comum que são adotados no presente trabalho, alguns deles são:

- *Desenvolvimento iterativo e incremental*: pequenas melhoras no código do adaptador, uma após da outra.
- *Correção de todos os erros antes de adicionar uma nova funcionalidade*: fazer que o código seja modular ao tempo que são feitas as correções necessárias antes de adicionar um novo módulo.
- *Lançamentos*: o projeto principal está dividido em vários módulos, os quais realizam tarefas específicas sendo lançadas em sequência.
- *Testes*: há partes do programa que são de grande importância, e.g. a programação dos módulos de leitura de arquivo físico STP. É necessário verificar que os métodos retornem os valores corretos, no formato correto.

Existem outros princípios destas metodologias ágeis que não são considerados, pois só são aplicados para práticas de programação de equipes numerosas de desenvolvimento de *software*.

## A.2. TRADUTOR DE LINGUAGEM

Um tradutor é definido como um programa que traduz ou converte a partir de um texto ou um programa escrito numa linguagem fonte para um texto ou um programa equivalente escrito numa linguagem distinta, se possível, lida com mensagens de erro [89], assim como mostrado na Figura A.1.

Os tradutores abarcam tanto os compiladores (em que a língua de saída é geralmente o código de máquina) como os intérpretes (em que a língua de saída é composta de ações que podem ser executadas no intérprete) [89] e os Conversores fonte-fonte (C. f-f, onde a linguagem de saída é uma linguagem distinta à original).

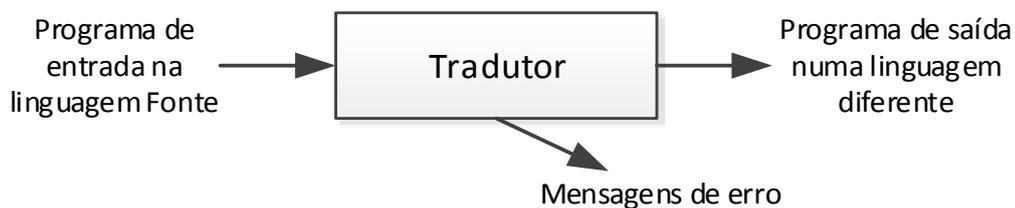


Figura A.1: Desenho preliminar de um tradutor (Adaptado de [89])

A seguir, apresentam-se alguns tipos de tradutores com suas definições e a suas características.

### A.2.1. O Compilador

Um compilador é um *software* que lê um programa escrito numa linguagem (a linguagem fonte) e o traduz num programa equivalente numa outra linguagem (a linguagem alvo) [90]. Uma parte importante desse processo de tradução são as mensagens de erro que o compilador relata para o usuário [89]. Num compilador a entrada é uma sentença numa linguagem formal (e.g. C, Python) e na saída tem-se um arquivo executável, em outras palavras, o compilador realiza uma tradução de um código de alto nível para código de máquina. Um compilador também pode ter como saída um arquivo objeto em lugar de um executável. Algumas das características gerais de um compilador são:

- Análise Léxica, Sintática e Semântica
- Geração de código intermediário
- Optimização de código
- Geração de código final
- Tratamento de erros

### A.2.2. O Intérprete

Um intérprete funciona de forma parecida a um compilador, mas a saída dele é uma execução. O programa de entrada se reconhece e executa ao mesmo tempo. Não se produz um resultado físico (o código de máquina) mas sim lógico (uma execução) [89]. A principal vantagem é a fácil depuração mas também há algumas desvantagens, como, por exemplo: uma execução lenta, consumo de recursos, entre outros. Algumas das características gerais de um intérprete são:

- Interpretação pura

- Execução divaga
- Fácil depuração

### A.2.3. O Conversor Fonte - Fonte

O conversor F-F permite traduzir desde uma linguagem de alto nível a outra linguagem de alto nível ou ainda outro nível, com isto consegue-se uma maior portabilidade nos programas de alto nível. Segundo GALVES et al. [89], um programa resultante de uma conversão F-F, pode requerer retoques manuais devido a vários motivos, como:

- Situações em que a linguagem-alvo carece de importantes características com relação à linguagem de origem. Como, por exemplo, um conversor de Java a C, pois C não tem coletor de lixo.
- Situações em que a tradução não é inteligente e os programas finais ou saída são altamente ineficientes.

Neste projeto, utiliza-se principalmente os conceitos do conversor fonte – fonte e algumas características dos compiladores e intérpretes para modelar o adaptador de STEP-NC a Código G/M. Uma dessas características é o analisador léxico. Um analisador léxico tem a função de identificar os *tokens* (Trata-se de um bloco de texto categorizado, e.g., um sinal de pontuação, um operador, um identificador, um número, etc.) que compõem a linguagem original, ou seja, identifica-se as menores unidades de informação que constituem a linguagem em questão. Assim, pode-se dizer que o analisador léxico é responsável pela leitura dos caracteres da entrada, agrupando-os em palavras, que são classificadas em categorias [91]. Estas categorias podem ser, basicamente, as seguintes:

- Palavras reservadas: palavras que devem aparecer literalmente na linguagem, sem variações. Algumas palavras reservadas do arquivo físico STEP são: *HEADER*, *FILE\_SCHEMA*, *ENDSEC*.
- Identificadores: palavras que seguem algumas regras de escrita, porém podem assumir diversos valores. São definidos de forma genérica. Geralmente, as regras de formação de identificadores são as mesmas utilizadas para a formação de palavras reservadas. Nesse caso, é necessário algum mecanismo para decidir quando um *token* forma um identificador ou uma palavra reservada.
- Símbolos especiais: sequências de um ou mais símbolos que não podem aparecer em identificadores nem palavras reservadas. São utilizados para composição de expressões aritméticas ou lógicas, comando de atribuição, etc. São exemplos de símbolos especiais: “;” (ponto-e-vírgula), “:” (dois pontos), “:=” (atribuição).
- Constantes: podem ser valores inteiros, valores reais, caracteres ou literais.
- Comentário: qualquer cadeia de caracteres iniciando com e terminando com símbolos delimitadores, utilizada na documentação do programa fonte.

Neste trabalho se usará a conversão fonte-fonte, convertendo a linguagem de alto nível STEP-NC à linguagem de baixo nível Código G/M. Também, observe-se que o reconhecimento de tokens que descrevem o programa de peça em STEP-NC estão governados e referenciados pelos atributos e entidades dispostas nas normas que compõem o padrão STEP-NC (*part 1*, 10, 11 e 111) para fresamento.

## APÊNDICE B. O CONTROLADOR EMC SUAS CARACTERÍSTICAS E FUNÇÕES

### B.1. O CONTROLADOR LINUXCNC

O LinuxCNC, também chamado EMC2 (*Enhanced Machine Controller*, EMC), é um controlador (*software*) *FLOSS* (*Free/Open-Source Software*) com licença GNU/Linux que implementa a capacidade de Controle Numérico (NC) usando qualquer computador pessoal (PC) para controlar máquinas-ferramenta CNC de 3 a 9 eixos. O LinuxCNC (ver sessão 2.5) usa Códigos G/M segundo a norma RS274NGC [72].

O LinuxCNC atualmente é desenvolvido e mantido por um grande número de programadores voluntários do projeto LinuxCNC.org. A arquitetura aberta do controlador NC permite modificar ou adicionar novas funcionalidades ao EMC2.

A seguir apresenta-se o projeto LinuxCNC.org e o *software* LinuxCNC. Em relação ao EMC2, é descrita sua instalação, características e configurações.

### B.2. PROJETO LINUXCNC.ORG

Como já foi mencionado, o LinuxCNC.org é um projeto desenvolvido atualmente por uma comunidade de programadores voluntários. Este projeto mantém o *software* LinuxCNC, usado para controlar máquinas CNC [84].

O *software* EMC é de domínio público e foi originalmente desenvolvido pelo NIST (*National Institute of Standards and Technology*). O desenvolvimento desse controlador teve vários patrocinadores e participantes como: SOSAS, NCMS e UEFA dos EUA. No ano de 1993 o controlador passou a ser chamado ECA (*Enhanced Machine Controller Architecture*), sendo depois nomeado EMC (*Enhanced Machine Controller*).

O EMC foi a implementação de uma referência de "fornecedor-neutral" da linguagem padrão da indústria para controle numérico de operações de usinagem usando a norma RS-274D (*Gcode*). O *software* inclui o intérprete RS274NGC para controlar: o planejador de trajetória de movimento, os *drivers* motor/atuador em tempo real e a interface de usuário. Ele demonstrou a viabilidade de um sistema de controle numérico avançado usando, fora do hardware PC (com FreeBSD ou Linux), diversas interfaces de hardware de sistemas de controle de movimento (*drivers*) [72].

O projeto de demonstração foi muito bem sucedido e criou uma comunidade de usuários e colaboradores voluntários. Em junho de 2000, NIST mudou o código fonte para sourceforge.net (site web de colaboração para projetos de *software* de código livre) sob a licença Domínio Público, a fim de permitir que colaboradores externos fizessem alterações.

Em 2003, a comunidade reescreveu algumas partes do EMC, reorganizou e simplificou outras partes e, em seguida, deu-lhe um novo nome, EMC2. O EMC2 ainda está sendo desenvolvido ativamente, agora licenciado sob GNU (*General Public License*).

A adoção do novo nome EMC2 foi motivada por várias mudanças importantes. Primeiramente, uma nova camada conhecida como HAL (*Hardware Abstraction Layer* ou camada de abstração de hardware) foi introduzida para interligar funções facilmente sem alterar o código C ou recompilar.

E devido à necessidade de movimentações muito pequenas, também chamadas de grão fino, é preciso que o controle do movimento da máquina seja em tempo real. Portanto, o EMC requer uma plataforma (Sistema Operacional OS) com recursos de computação em tempo real. O EMC usa o kernel do Linux Ubuntu com extensões em tempo real (RTAI) [72].

### B.3. INSTALAÇÃO DO LINUXCNC

Para fazer a instalação do EMC2, o projeto LinuxCNC.org disponibiliza no seu site [www.linuxcnc.org](http://www.linuxcnc.org) o sistema operacional Linux Ubuntu com o controlador LinuxCNC pré-instalado em um único arquivo com extensão “.ISO” para ser instalado em um PC. Desse modo, um novo sistema operacional (OS) deve ser instalado no computador. A versão do OS Linux Ubuntu atualmente disponibilizada no site é a 10.04 LTS e a versão do EMC2 pré-instalado é a 2.5.2.

No site também pode-se encontrar e acessar todas as informações para instalação, configuração, atualização e manuseio do programa.

A instalação do Linux Ubuntu é simples, devendo-se apenas seguir as instruções do assistente de instalação do sistema operativo. Depois de instalado o OS, é possível configurar o EMC2 para o perfil da Router CNC desenvolvida neste projeto (ver apêndice C - Projeto de Desenho e fabricação de uma Router CNC de três eixos para peças prismáticas).

A configuração do EMC2 é feita por meio de um assistente de configuração chamado “*LinuxCNC Stepconf Wizard*”. Na Figura B.1 pode-se observar a tela de boas vindas do programa *LinuxCNC Stepconf Wizard*.

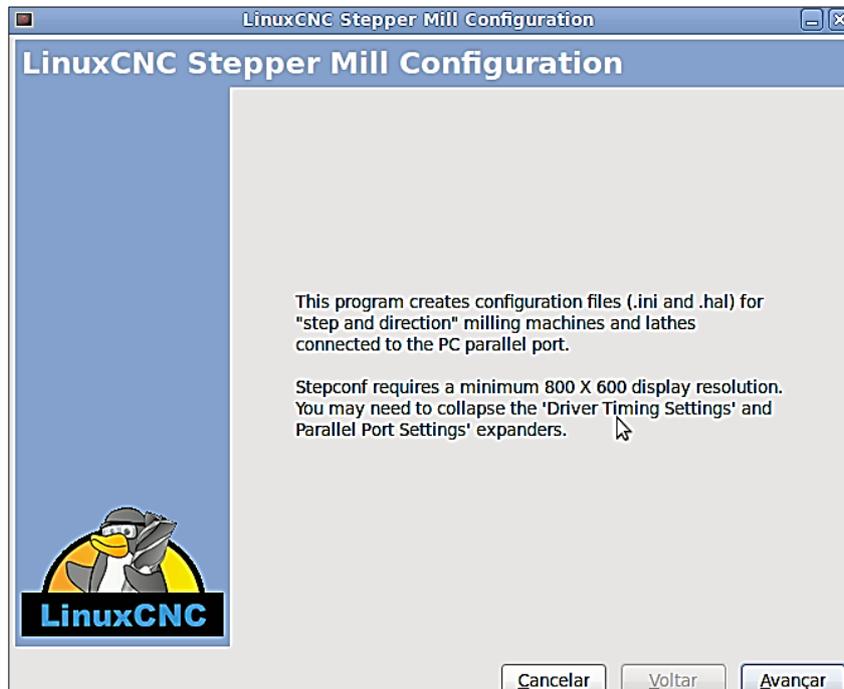


Figura B.1: Tela de início do programa LinuxCNC *Stepconf Wizard*

Na sessão seguinte é explicada a configuração do EMC2 para a Router CNC desenvolvida neste trabalho.

## B.4. CONFIGURAÇÃO DO CNC TIPO ROUTER COM O EMC2

Na sessão anterior foi referenciado o programa LinuxCNC *Stepconf Wizard* (assistente de configuração do EMC2) que vem junto com o *software* do EMC2. Esse programa facilita a configuração do EMC2 para uma máquina-ferramenta específica, como a Router CNC desenvolvida neste projeto.

A primeira tela do assistente informa que será criado dois arquivos, um com extensão “.ini” e outro com extensão “.hal”. Nestes arquivos estão disponíveis os parâmetros de configuração da Router CNC que receberá os comandos de controle por meio da porta paralela (LPT).

Pressionando o botão “Avançar”, dirige-se para a criação de uma nova configuração ou modificar uma existente. Na seguinte Figura B.2, pode-se observar essa tela.

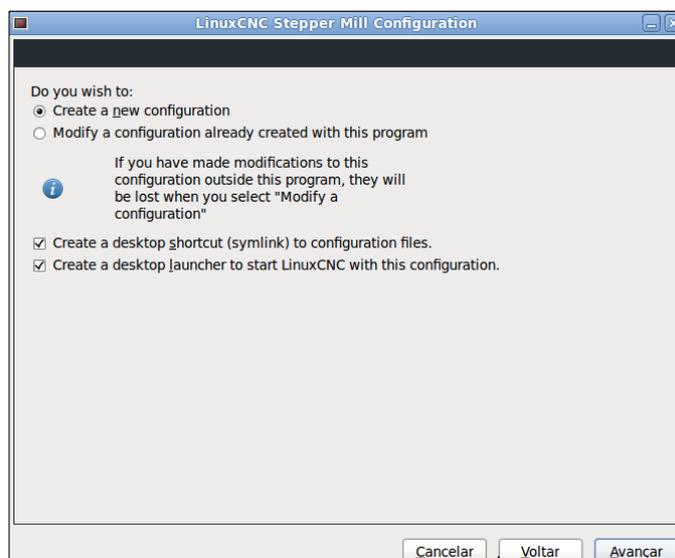


Figura B.2: Tela de escolha configuração do EMC2

Da tela mostrada na Figura B.2, é selecionada a opção de criar uma nova configuração e, imediatamente após, continua-se com a terceira tela.

Como pode-se observar na Figura B.3, adiciona-se as informações básicas a respeito da Router CNC, do PC, da porta LTP junto à placa de comunicação e, por último, os *drivers* de potência que controlarão os motores de passo.

A primeira lacuna é o nome da máquina. Depois, escolhem-se quantos eixos a máquina possui, neste caso três: XYZ. Na terceira lacuna seleciona-se a unidade dimensional, escolhe-se milímetros. O próximo passo é referente aos dados técnicos da controladora e todos esses dados são fornecidos pelo “Guia de Utilização” do fabricante (ver anexo 4). No campo “Tipo de *Driver*”, seleciona-se um dos *drivers* conhecido pelo EMC2 ou então a opção “Outros”. No caso deste trabalho, opta-se por “Outros”, visto que a controladora SMC-B-PRO não está na lista do assistente. Continuando com os dados técnicos da controladora SMC-B-PRO, o próximo item é o “*Step Time*”. Este parâmetro é o tempo de duração do sinal de passo em nível alto. Na especificação do fabricante da controladora, encontra-se o valor de 10.0  $\mu$ s, que corresponde a 10000 ns. O campo seguinte é o “*Step Space*”. Esse é o tempo de duração do sinal de passo no nível baixo. Na Figura B.3, o valor do “*Step Space*” vale 10000 ns (ver anexo 4). O parâmetro “*Direction Hold*” refere-se ao tempo de duração do sinal

de direção depois que o próximo sinal de passo foi emitido. E o “*Direction Setup*” é o tempo de duração do sinal de direção antes do sinal de passo. Segundo o guia de utilização do *driver*, o valor destes últimos dois dados é de 20000 ns (ver anexo 4). Na Figura B.4 pode-se observar para melhor entendimento, uma ilustração dos sinais de comando segundo o assistente.

Figura B.3: Tela de Informações básicas da máquina

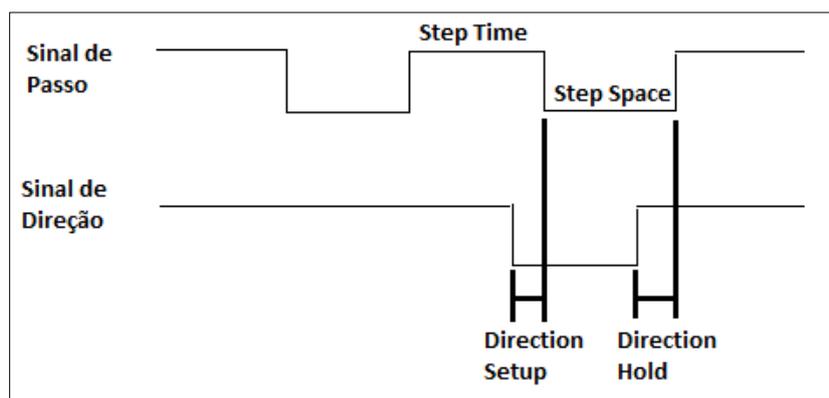


Figura B.4: Sinais do *Driver Timing Settings* [92]

O próximo campo compreende com “Endereço base LPT”. Este campo é referente ao endereço de memória da porta paralela do PC cujo EMC2 foi instalado. Como pode ser visto no apêndice A em relação ao sistema eletrônico de controle (segundo componente - placa de comunicação via porta paralela), o endereço padrão da porta paralela LPT1 é 0x378 (em hexadecimal). Neste projeto usou-se a porta LPT1 como porta de comutação e acesso aos drives de potência.

Por fim, o último campo nesta tela (ver Figura B.3), refere-se ao teste de latência do computador. A latência é o tempo que o computador precisa para responder a um pedido externo. Neste projeto, o pedido externo é o tempo necessário para a geração de pulsos de passos. Assim, quanto menor a latência, mais rápido é o trem de pulsos.

É importante notar que deve-se usar o OS em tempo real para garantir que o controle dos motores seja executado num intervalo de tempo adequado. Portanto, o EMC2 é um *software* RT (*real time*). Segundo LYRA [92], isso garante que as tarefas serão executadas num intervalo de tempo pré-estabelecido e o teste de latência fornecerá este parâmetro de relógio para garantir que o motor de passo não perca passos.

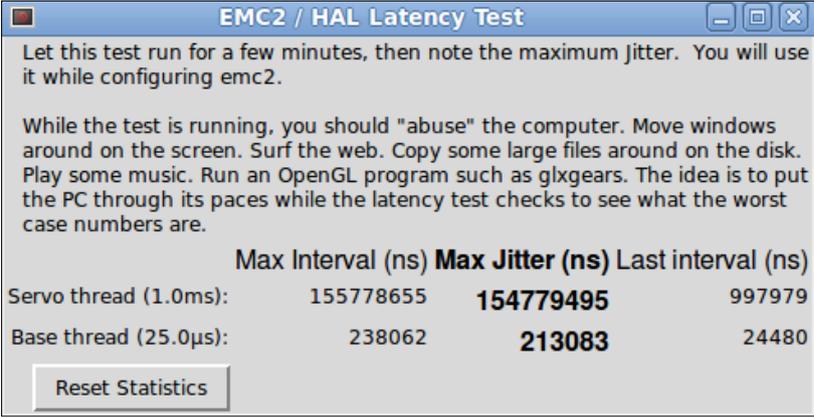
- **Teste de Latência**

A latência é o tempo que o PC tarda para deter o que está fazendo e responder um pedido externo. Segundo o manual do LinuxCNC [72], quanto menor for a latência, mais rápidos e suaves serão os pulsos de passo.

A latência é muito mais importante que a velocidade da CPU. A CPU não é o único fator para determinar a latência. Outros componentes como Placa base, placa de vídeo, portas USB, processador e um número de outras coisas podem interferir na latência.

Segundo o manual do assistente [72], enquanto é executado o teste, deve-se “abusar” do computador, em outras palavras, por uma grande carga computacional no PC, por exemplo movimentar janelas pela tela, navegar pela web, copiar alguns arquivos de grande tamanho no disco e, também, executar um programa *OpenGL* como *glxgears*. Não se deve, no entanto, usar o EMC2 em quando o teste estiver sendo feito.

Quando a prova de latência (ver Figura B.5) é feita, durante um minuto ou mais, são capturados os eventos que poderiam ocorrer em intervalos menos frequentes.



|                       | Max Interval (ns) | Max Jitter (ns)  | Last interval (ns) |
|-----------------------|-------------------|------------------|--------------------|
| Servo thread (1.0ms): | 155778655         | <b>154779495</b> | 997979             |
| Base thread (25.0µs): | 238062            | <b>213083</b>    | 24480              |

Figura B.5: Janela do Teste de latência [72]

Antes de fazer o teste de latência, deve-se conhecer as características do hardware do PC disponibilizado para instalar o LinuxCNC. Esse PC tem as seguintes características:

Tabela B.1: Hardware do PC usado na Router CNC

| Componente     | Caraterística          |
|----------------|------------------------|
| Placa Mae      | Intel 865GV            |
| Placa de vídeo | Intel <i>Extreme 2</i> |
| Disco rígido   | 40Gb                   |
| Memoria        | 1Ghz                   |
| Processador    | Pentium IV de 2.8Ghz   |

O teste de latência específico para este PC foi feito utilizando o “EMC2/Hal Latency Test”, que é executado quando se pressiona o botão “Test Base Period Jitter” da janela de informações básicas da máquina. O teste teve como resultado um valor de 24763ns (Last Interval). Na Figura B.3, pode-se observar o valor de latência.

Ao final da tela de informações básicas da máquina também está a opção de visualizar na janela do EMC2 uma janela para pedir a troca de ferramenta. Esta opção é selecionada, pois será usada posteriormente quando for necessária a troca de ferramenta manual invocada pelo programa de peça em código G.

Depois de preencher todos os requisitos da janela de informações básicas da máquina, pode-se avançar à tela seguinte, chamada de opções de configuração avançada (ver Figura B.6). Há duas opções para escolher uma interface de usuário adicional ao EMC2. Essas interfaces podem conter, mostrar ou adicionar novas funcionalidades ao controlador LinuxCNC. Neste projeto não se usa nenhuma dessas funcionalidades.

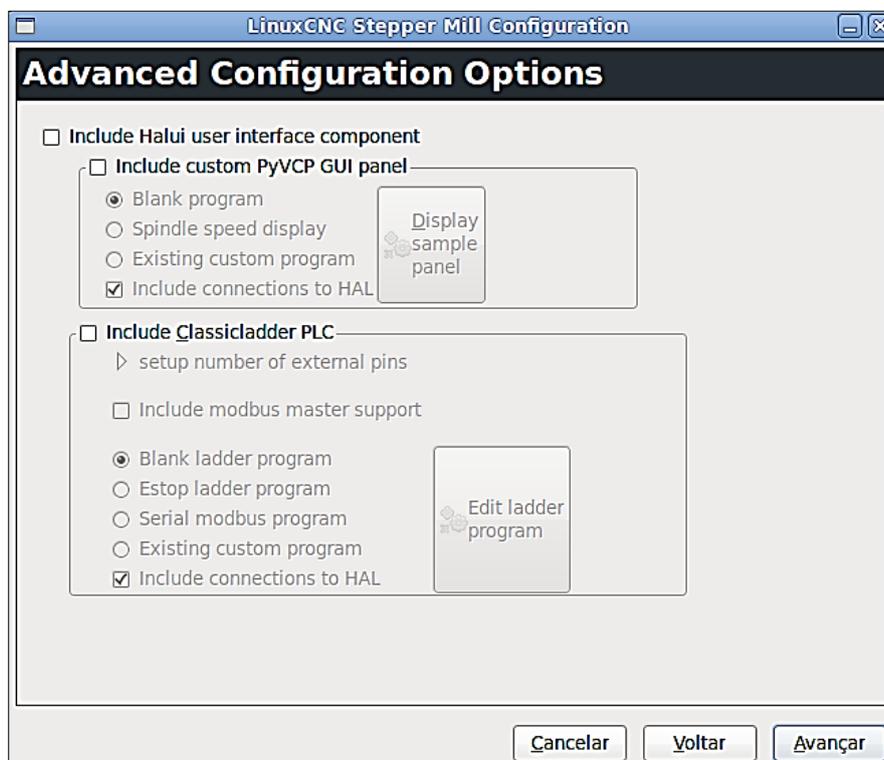


Figura B.6: Opções de configuração avançada do LinuxCNC [72]

A tela seguinte mostra os requisitos para a configuração da porta paralela. Baseando-se no guia de usuário do BCP-PRO (placa de comunicação via porta paralela), a configuração de cada pino da porta paralela resultante para a Router CNC é como mostrado na Figura B.7.

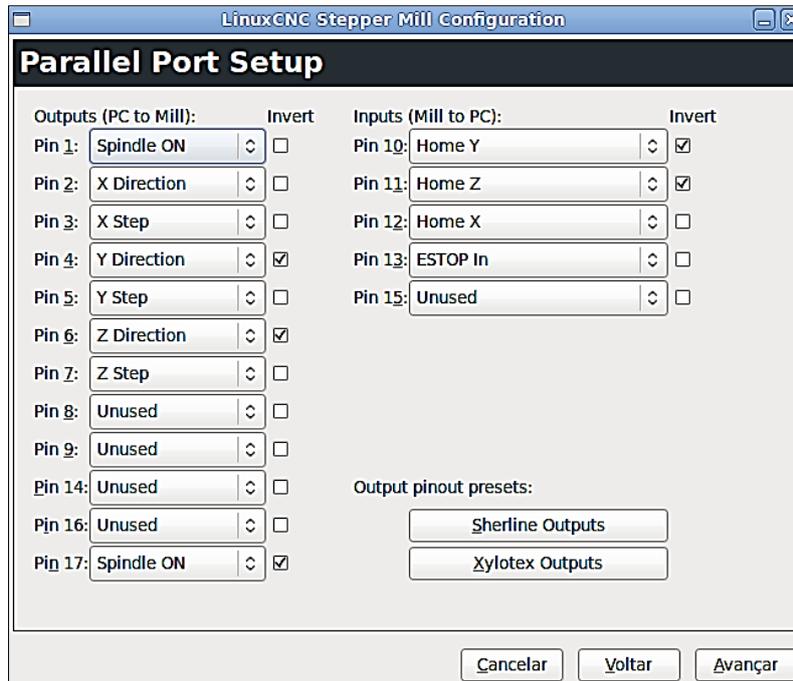


Figura B.7: Configuração dos pines da porta paralela

Nas próximas telas (Figuras B.8 até B.10) são apresentadas as configurações de cada eixo. A Router CNC desenvolvido neste projeto, conta com três eixos, X, Y e Z. Embora o EMC permita a configuração de mais eixos, só serão configurados os eixos supracitados.

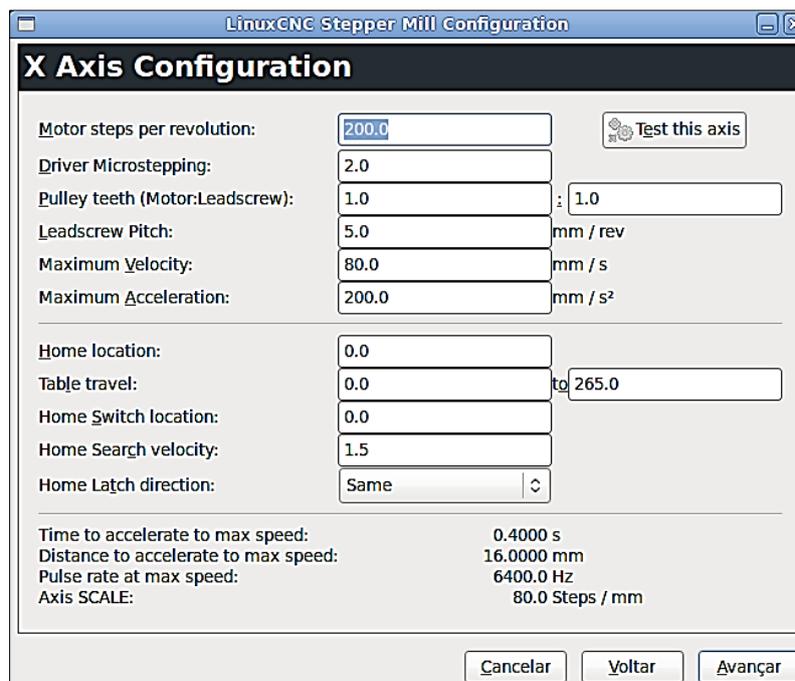


Figura B.8: Configuração Eixo X

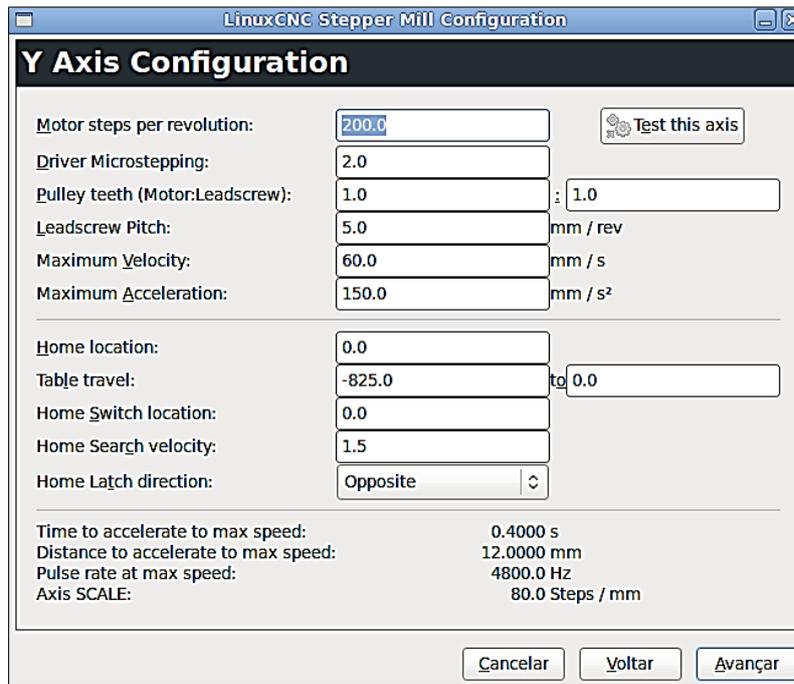


Figura B.9: Configuração Eixo Y

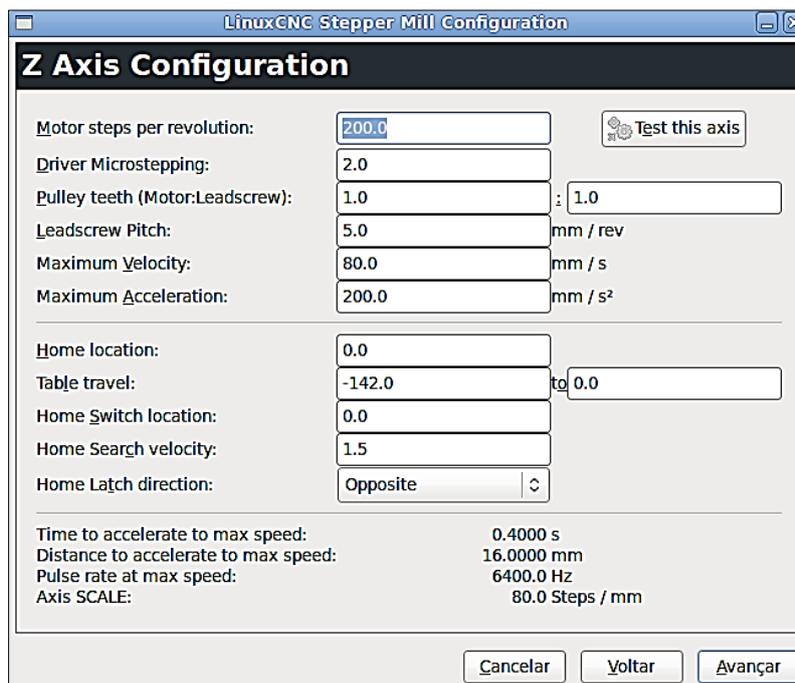


Figura B.10: Configuração Eixo Z

Os valores iniciais como passo do motor por revolução (*Motor steps per revolution*), micro-passo por *driver* (*Driver Microstepping*) e passo do fuso (*leadscrew Pitch*) foram obtidos das características tanto do motor como do fuso de esferas. Os outros dados foram obtidos de maneira experimental graças ao teste do eixo que permite conferir a configuração estabelecida nesse momento.

Por fim, a última janela é para confirmar a configuração entre a máquina e o controlador NC. Ao terminar, o assistente cria na “Área de Trabalho” do OS, um atalho do executável do controlador

LinuxCNC é uma pasta onde se encontram alguns arquivos de configuração da mesma. Entre os arquivos mais importantes da configuração da máquina estão o arquivo com extensão “.INI” e o arquivo com extensão “.TBL”. Esses arquivos são explicados detalhadamente a seguir.

#### **B.4.1. O arquivo .INI**

Este arquivo não apenas contém as informações inseridas na configuração anterior (configuração da máquina com o controlador), mas também as informações relacionadas com a configuração interna da GUI do controlador e a configuração da HAL [72].

O arquivo “.INI” (ou arquivo INI) está composto basicamente por três “camadas”, comentário, seções e variáveis.

Uma linha de comentário começa com um caractere tipo ; ou #. Quando uma linha do arquivo INI inicialmente contém esses caracteres, o resto da linha é ignorada pelo *software*. Os comentários podem ser usados para descrever o funcionamento de um elemento. O arquivo INI está separado em seções. O nome delas está entre colchetes e.g. [SEÇÃO]. A ordem das seções não é importante. As seções começam pelo nome de uma seção e termina no próximo nome de outra seção.

As seções a seguir são usadas pelo controlador LinuxCNC:

- [EMC] Informações gerais
- [DISPLAY] Configurações relacionadas à interface gráfica do usuário
- [FILTER] Configurações de programas de filtragem de entrada
- [RS274NGC] Configurações usadas pelo intérprete *g-code*
- [EMCMOT] Configurações utilizadas pelo controlador de movimento em tempo real
- [TASK] Configurações utilizadas pelo controlador de tarefa
- [HAL] Especifica arquivos “.hal”
- [HALUI] Comandos utilizados pelo MDI HALUI
- [TRAJ] Configurações adicionais utilizados pelo controlador de movimento em TR
- [AXIS\_n] Variáveis de eixos individuais
- [EMCIO] Configurações usadas pelo controlador de I/O

Por último estão as variáveis. Uma linha que contém uma variável se identifica por estar composta do nome de variável, um sinal de igual (=), e um valor, e.g. MACHINE = Minha Máquina.

Para este trabalho a seção mais importante é a chamada [FILTER], nesta seção será implementado ou adicionado o adaptador desenvolvido na sessão 3.4. Isto será explicado depois.

#### **B.4.2. O arquivo .TBL**

O arquivo TBL salva as informações das ferramentas. Este arquivo obedece os parâmetros de configuração da tabela de ferramentas (*tool table*) do controlador LinuxCNC [74].

A tabela de ferramentas é um arquivo de texto que contém informações sobre cada ferramenta. O arquivo está localizado no mesmo diretório que a sua configuração e é chamado *tool.tbl*. As ferramentas podem estar em um trocador de ferramentas automático ou apenas trocar a ferramenta manualmente.

O arquivo TBL pode ser modificado por meio de um editor de texto ou usando-se o Editor de Ferramenta (*Tool Editor*) disponibilizado pelo EMC. Depois de modificar o arquivo, é necessário certificar-se de recarregar a tabela de ferramentas na GUI do controlador.

O formato do arquivo tbl compõe-se de uma tabela onde são alojadas as informações ou características das ferramentas usadas pelo código G. Na Figura B.11, apresenta-se um exemplo do conteúdo de um arquivo TBL usando o Editor de Ferramenta.

| Del | TOOL  | POC   | X | Y | Z     | A | B | C | U | V | W | DIAM   | FRONT | BACK | ORIEN | COMMENT         |
|-----|-------|-------|---|---|-------|---|---|---|---|---|---|--------|-------|------|-------|-----------------|
| ┆   | 1     | 1     |   |   | 0.511 |   |   |   |   |   |   | 0.125  |       |      |       | 1/8 end mill    |
| ┆   | 2     | 2     |   |   | 0.1   |   |   |   |   |   |   | 0.0625 |       |      |       | 1/16 end mill   |
| ┆   | 3     | 3     |   |   | 1.273 |   |   |   |   |   |   | 0.201  |       |      |       | #7 tap drill    |
| ┆   | 99999 | 99999 |   |   | 0.1   |   |   |   |   |   |   |        |       |      |       | big tool number |

Dom Jul 14 17:28:20 BRT 2013: File checked

Figura B.11: Edição de um arquivo “.tbl” com o Editor de Ferramentas

Em geral, a tabela possui os seguintes dados:

- TOOL - número da ferramenta, 0-99999 (números de ferramenta deve ser único)
- POC - número bolso, 1-99999 (números de bolso deve ser único)
- X, W - deslocamento no eixo específico da ferramenta - de ponto flutuante
- DIAM - diâmetro da ferramenta - de ponto flutuante, o valor é absoluto
- I - ângulo frontal (apenas para torno) - ponto flutuante
- J - ângulo do encosto (apenas para torno) - ponto flutuante
- Q - orientação da ferramenta (apenas para torno) – inteiro, 0-9
- Uso do caráter “;” Início de comentário ou observação - texto

Nas versões anteriores do LinuxCNC, havia dois formatos diferentes da tabela de ferramentas, um para fresamento e outro para torneamento, mas desde a versão 2.4.x um só formato de tabela ferramenta é usado para todas as máquinas, sendo apenas necessário ignorar as partes da tabela de ferramentas que não pertencem à máquina configurada.

As unidades utilizadas para o comprimento, diâmetro, etc, são em unidades de máquina (milímetros ou polegadas).

Apenas três, dos dados para máquinas fresadoras, são obrigatórios, TOOL, POC e DIAM. Os demais são opcionais.

No formato de texto (tabela) cada linha representa uma ferramenta, os dados são separados por espaços simples e no comentário usando ponto e vírgula. A seguir apresenta-se as informações no editor de ferramentas (Figura B.12) do arquivo TBL do exemplo da Figura 4.50, depois mostra-se essas mesmas informações em um editor de texto plano (*gedit* do Linux na Figura B.13).

| Del | TOOL | POC | X | Y | Z  | A | B | C | U | V | W | DIAM | FRONT | BACK | ORIEN | COMMENT      |
|-----|------|-----|---|---|----|---|---|---|---|---|---|------|-------|------|-------|--------------|
| └   | 1    | 1   |   |   | 40 |   |   |   |   |   |   | 3    |       |      |       | facemill     |
| └   | 2    | 2   |   |   | 40 |   |   |   |   |   |   | 4    |       |      |       | ball_endmill |

Dom Jul 14 23:16:20 BRT 2013: File checked

Figura B.12: Ferramentas e suas características no Editor de Ferramenta

```

tool.tbl ✕
T1 P1 Z40.0 D3.0;FACEMILL
T2 P2 Z40.0 D4.0;BALL_ENDMILL

```

Figura B.13: arquivo TBL no Editor de texto

## APÊNDICE C. PROJETO DE DESENHO E FABRICAÇÃO DE UMA ROUTER CNC DE TRÊS EIXOS PARA PEÇAS PRISMÁTICAS

Este apêndice descreve o desenvolvimento de um protótipo de Router de comando numérico computadorizado (CNC) para validar o controlador EMC (LinuxCNC) aderente a STEP-NC. São discutidos detalhes de seu desenho, construção e operação. São tratadas também as várias possibilidades e capacidades deste protótipo validado mediante testes de ajustes.

### C.1. COMPONENTES DE UMA ROUTER CNC

Antes de fazer o desenho CAD da Router, deve-se saber o que é e quais são os seus componentes. Assim sendo, uma Router CNC é uma máquina ferramenta que contém uma mesa com um portal no qual está montado um Spindle (fuso). Um controlador NC executa um programa NC que contém as instruções que movimenta o Spindle ao longo de três eixos (XYZ) [93].

Uma máquina ferramenta CNC tipo Router também é conhecida como fresadora por causa da suas semelhanças. O desenho da Router CNC está baseado nas seguintes partes ou componentes escolhidos:

➤ **Controlador NC:** O controlador é o cérebro da máquina CNC. O termo controlador geralmente se refere não apenas ao computador, mas também a todos os dispositivos eletrônicos e elétricos que dão movimento à máquina. Existem dois tipos de controladores disponíveis em máquinas de hoje. Controladores proprietários e de arquitetura aberta, baseados no computador pessoal (PC) [93]. Este projeto usa o controlador de arquitetura aberta EMC ou LinuxCNC aderente à norma ISO14649, para controlar o protótipo da Router.

➤ **Mesas de trabalho:** Atualmente existem muitos tipos de mesas de trabalho disponíveis. Estas variam dependendo do tipo de trabalho a ser feito e dos métodos de fixação que estão sendo utilizados. Segundo ALAIN [93], há atualmente cinco tipos de mesas de trabalho para Router CNC, são elas: mesa simples, mesa com T-entalhe, mesa transversa, vácuo universal e mesa tipo matriz. Para escolher uma mesa apropriada para a Router foram considerados os seguintes requisitos, fácil fixação da peça bruta, alta resistência mecânica e manutenção simples. Dado isso, foi escolhida a construção de uma mesa tipo ranhura T com a união de perfis em alumínio extrudado. A Figura C.1 mostra a mesa montada com os perfis de alumínio.

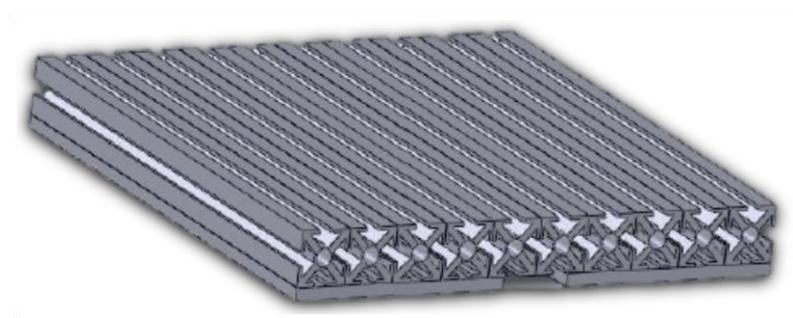


Figura C.1: Mesa de trabalho para a Router

Este tipo de mesa de trabalho feita de perfis de alumínio é de fácil montagem, possui baixo peso, alta resistência mecânica e pela sua geometria permite uma fácil fixação da peça prismática para usinagem.

➤ **Movimento dos eixos:** a movimentação ou o deslocamento dos eixos (neste caso três XYZ) da Router é realizado por meio de motores de passo acoplados a fusos de esferas recirculantes. Assim sendo, usa-se um fuso e um motor para cada eixo. A Tabela C.1 mostra as características dos fusos de esferas recirculantes e a Tabela C.2 mostra as características do motor de passo.

Tabela C.1: Fusos de esferas recirculantes

| Comprimento mm | Passo mm/360° | Precisão mm | Diâmetro mm | Aplica ao Eixo |
|----------------|---------------|-------------|-------------|----------------|
| 420            | 5             | 0.001       | 20          | <b>X</b>       |
| 1020           | 5             | 0.001       | 20          | <b>Y</b>       |
| 320            | 5             | 0.001       | 20          | <b>Z</b>       |

Tabela C.2: Características do Motor de passo KTC-HT23-400

| Característica                  | Valor               |
|---------------------------------|---------------------|
| Torque Estático                 | 15 Kg x cm          |
| Peso                            | 1.0 Kg              |
| Resolução                       | 1.8° x passo        |
| Tipo de controle ou acionamento | Unipolar ou Bipolar |

De acordo com as características dos fusos (comprimento) é possível ter uma noção da área de trabalho efetiva que terá a Router, sendo 400X x 1000Y x 300Z em teoria aproximadamente. Deve-se, ainda, notar que todos os eixos utilizam o mesmo tipo de motor de passo. No Anexo 2 são apresentadas as características gerais deste tipo de motor. Na Figura C.2, apresenta-se o motor de passo e sistema de movimentação.

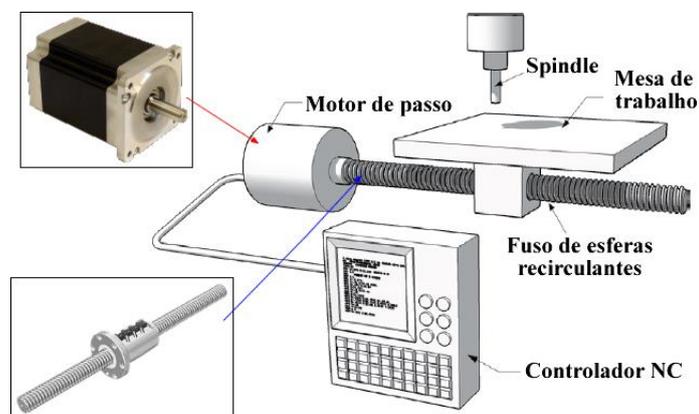


Figura C.2: Acople de motor de passo com fuso de esferas (Adaptado de [93])

➤ **Spindle:** este componente é responsável pela rotação da ferramenta de corte na máquina. Segundo ALAIN [93], essencialmente existem dois tipos diferentes de Spindles em uso hoje em dia, Spindle acionado por correias e Spindle de motor de alta velocidade [93]. Para rotar a ferramenta de corte foi escolhido o segundo tipo de Spindle por sua facilidade de manuseio e sua disponibilidade no mercado nacional. Para escolher um Spindle de motor apropriado para a Router, foram considerados os seguintes requisitos: tem que ser um motor elétrico de CA de 220V (de acordo com a voltagem local), possuir uma geometria regular cilíndrica para fácil fixação no eixo Z (cabeçote),

possuir tamanho reduzido, alta potência e, por último, ter que ser preferivelmente voltado para usinagem de madeira (material que será usado para fazer os testes e estudo de caso da Router CNC). O tamanho reduzido do motor é necessário, visto que o conjunto de componentes do cabeçote (no eixo Z) ocupa parte do comprimento útil do eixo, logo, quanto maior o tamanho do motor, menor seria a área de trabalho efetiva do eixo Z. além disso, quando um Spindle possui uma alta velocidade rotacional, isto favorece as velocidades de corte e de avanço da máquina.

No mercado local, pode-se encontrar vários tipos de motores (ferramentas para usinagem) que atendem um ou vários dos requisitos supracitados, são estes: furadeiras, tupias (router) e retificadoras. Mesmo com a diversidade de modelos e marcas reconhecidas no mercado de furadeiras e retificadoras, a Tupia mostrou-se ser a opção mais adequada para o projeto. A Tabela C.3 resume as características de algumas das ferramentas comumente usadas como Spindle em máquinas-ferramenta CNC.

Tabela C.3: Resumo das características de ferramentas elétricas.

|                   | <b>Furadeira Black e Decker KR505</b> | <b>Retificadora Bosch Gss 28</b> | <b>Tupia Makita 3709</b> |
|-------------------|---------------------------------------|----------------------------------|--------------------------|
| <b>Potencia</b>   | 500W                                  | 500W                             | 530W                     |
| <b>Peso</b>       | 1.6 Kg                                | 1.4 Kg                           | 1.5 Kg                   |
| <b>Geometria</b>  | Irregular                             | Cilíndrica Regular               | Cilíndrica Regular       |
| <b>Velocidade</b> | Baixa-variável 3000 rpm               | Media 25000 rpm                  | Média-alta 30000 rpm     |
| <b>Custo</b>      | Baixo                                 | Médio                            | Médio                    |

Das furadeiras encontradas no mercado, julgou-se que geralmente essas possuem velocidades de rotação baixa da ordem de 3000 rpm, e uma geometria irregular em forma de “L”, características que prejudicam à máquina tanto na fixação no cabeçote quanto na usinagem. As retificadoras embora possuam dimensões intermediárias entre as furadeiras e tupias, não são muito boas para usinagem de madeira e seu uso é mais voltado para trabalhos manuais delicados. Portanto, a tupia considerada (Tupia marca Makita, modelo 3709 [94]) se adequou ao projeto devido, principalmente, à sua geometria regular cilíndrica que facilita a fixação no cabeçote do eixo Z. Esta ferramenta também possui maior velocidade e maior torque quando comparada com as outras ferramentas da Tabela A.3. Por último, as tupias são indicadas pelos fabricantes para uso em madeira, característica que se sobressai das outras ferramentas. A Figura C.3 mostra a Tupia escolhida.



Figura C.3:Tupia Makita 3709 [94]

➤ **Troca de ferramenta:** na usinagem de uma *feature* numa peça, às vezes é necessário o uso de várias ferramentas. Isto quer dizer que a máquina-ferramenta deve permitir a troca da ferramenta no Spindle sem perder a referência do sistema de coordenadas ou por exemplo do zero peça. Há dois tipos de troca de ferramenta, manual e automática. Na troca de ferramenta manual, o Spindle é desligado, o sistema de fixação da ferramenta de corte é liberado e a ferramenta é trocada. Em alguns

casos a troca de ferramenta é comandada pelo programa NC e muitas vezes, porém, um programa NC necessita ser dividido em várias partes, iniciando e finalizando em cada troca de ferramenta. Em todos os casos de troca de ferramenta manual há intervenção do operador da máquina. Na troca de ferramenta automática, o que realmente existe é um espaço coordenado especialmente projetado para este fim. A troca de ferramenta é feita completamente pela máquina sem nenhuma ou pouca intervenção do operador da máquina. A Figura C.4 ilustra estes dois tipos de troca de ferramenta.

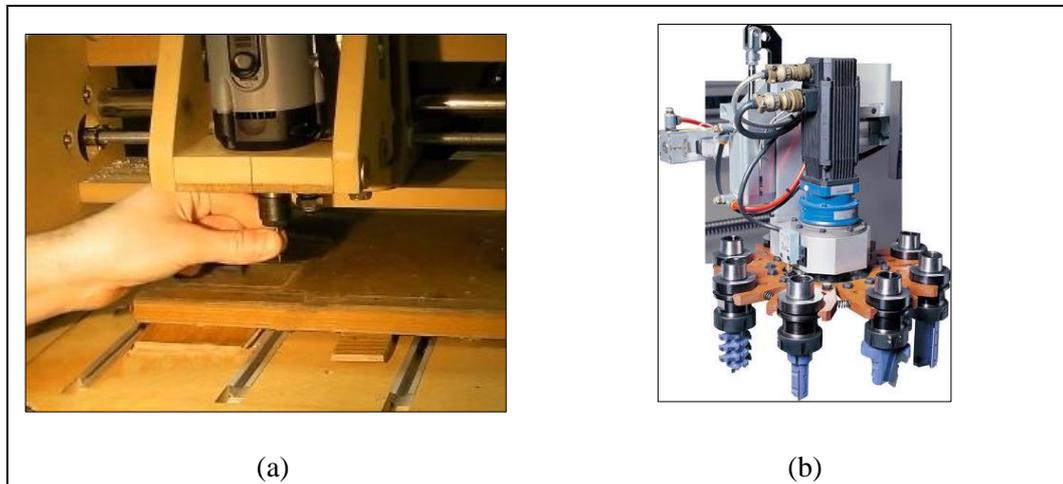


Figura C.4:Troca de ferramenta: (a) Manual [95], (b) Automática [93]

➤ **Estrutura da máquina:** a estrutura da máquina é o esqueleto, o suporte onde os demais componentes da máquina estão montados. Esta estrutura mecânica pode ser feita de diferentes tipos e formas de material. Para escolher o material apropriado para estrutura da Router, foram considerados os seguintes requisitos, fácil usinagem e alta resistência mecânica. Assim sendo, pode-se listar vários tipos de materiais como chapa de madeira, chapa de aço e chapa de alumínio. A Tabela C.4 mostra as vantagens e desvantagens desses materiais.

Tabela C.4: Matérias para a estrutura da Router

| Tipo de Material  | Vantagem  | Desvantagem  |
|-------------------|---|--|
| Madeira           | - Fácil Usinagem<br>- Fácil Montagem<br>- Leve            | - Baixa resistência mecânica                         |
| Chapa de Aço      | - Alta resistência mecânica                               | - Difícil Usinagem<br>- Difícil Montagem<br>- Pesado |
| Chapa de Alumínio | - Alta resistência mecânica<br>- Leve<br>- Media Montagem | - Media Usinagem                                     |

O melhor material, escolhido por sua relação de vantagens e desvantagens, foi a chapa em alumínio, a qual é menos densa e mais fácil de usinar que o aço. Além disso, a chapa de alumínio possui alta resistência mecânica, característica importante principalmente pela geometria das peças que serão usinadas e que conformam a Router.

A Figura C.5 mostra as chapas de alumínio de vários calibres.



Figura C.5: Chapas de alumínio, vários calibres

Estas chapas de alumínio estão formadas pela liga normalizada ALZINTOK 75-T651. As características gerais desse material estão descritas no Anexo 3.

Até aqui foram apresentados os componentes (físicos) básicos necessários para começar com o projeto CAD do desenho da Router. Posteriormente serão apresentados os componentes eletrônicos que permitem controlar os motores de passo (*drivers* de potência) e também a placa de comunicação que serve de interface entre os *drivers* dos motores e o PC. A comunicação entre essa interface e o computador é por meio da porta paralela. Em seguida, descreve-se o projeto CAD de desenho e simulação da Router CNC.

## C.2. PROJETO CAD DE DESENHO E SIMULAÇÃO DA ROUTER CNC

Este projeto CAD descreve o desenho da Router CNC baseado nos componentes supracitados (Spindle, motores de passo, fusos de esferas recirculantes e mesa de trabalho), para criar a estrutura da máquina onde são montados estes elementos.

É importante mencionar que inicialmente este projeto contava com alguns componentes (materiais) em estoque. A seguinte tabela mostra esses elementos.

Tabela C.5: Componentes para o Projeto

| Componentes   | Quantidade |
|---|------------|
| Parafuso de esferas recirculantes Ø20 mm x 320 mm                           | 1          |
| Parafuso de esferas recirculantes Ø20 mm x 420 mm                           | 1          |
| Parafuso de esferas recirculantes Ø20 mm x 1020 mm                          | 1          |
| Barra guia linear de Ø 20 mm x 320 mm                                       | 2          |
| Barra guia linear de Ø 20 mm x 420 mm                                       | 2          |
| Trilho de 1000 mm com patim de carga  | 2          |
| Castanha simples de Ø 20 mm   | 4          |
| Motor de passo de 1.9N, Marca KTC-HT23-400                                  | 3          |
| <i>Driver</i> para motor de passo, Marca Hobby CNC                          | 3          |
| Placa de comunicação entre porta paralela e <i>driver</i> , Marca Hobby CNC | 1          |
| Transformador 220vac - 24vac x 5amp   | 1          |

As características geométricas e dimensionais desses elementos somados aos escolhidos anteriormente marcam o ponto de partida para o desenho CAD da máquina. Este projeto usa a

metodologia de construção por módulos proposta por [96] para a construção de máquinas CNC. Usando essa metodologia, pode-se dividir o desenho da máquina em várias partes (módulos), ficando menos difícil a montagem final tanto no desenho CAD como na fabricação da Router. Cada eixo (XYZ) compõe um módulo da máquina e possui suas próprias características e detalhes, mas também há características similares que são compartilhadas entre eles, por exemplo o sistema de acoplamento entre o fuso de esferas recirculantes e o motor de passo. Também podem existir outros módulos e.g. a mesa de trabalho. A Figura C.6 ilustra os módulos separados dos eixos X, Y e Z.

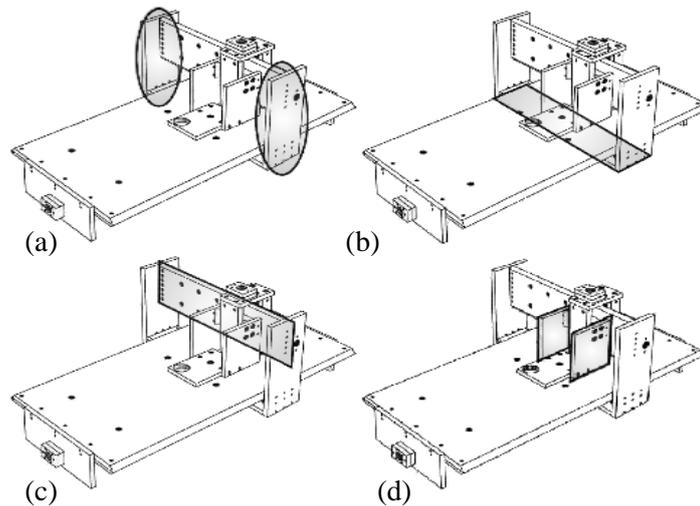


Figura C.6: Módulos: eixo Y (a e b), eixo X (c) e eixo Z (d) (Adaptado de [96])

O desenho CAD da máquina começa com o módulo do eixo Z, pois deve-se levar em consideração o espaço aproximado que é ocupado pelo cabeçote, visando assim otimizar o espaço para os demais módulos. Depois o desenho que segue é o módulo do eixo X, prossegue-se com o eixo Y, com a mesa de trabalho e, por último, com o módulo referente ao suporte da máquina que inclui o espaço para o sistema de controle. A Figura C.7 apresenta uma proposta do protótipo terminado da Router CNC.

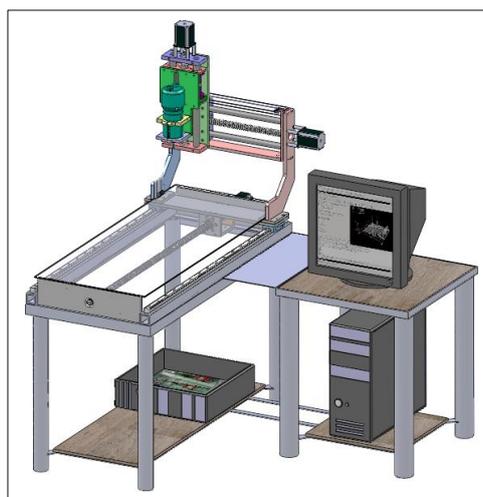


Figura C.7: Protótipo da Router CNC

Os módulos e demais itens que compõem a máquina são brevemente descritos a seguir.

➤ **Desenho CAD do módulo do eixo Z:** deve-se, primeiramente, listar os componentes que inicialmente estão disponíveis para este módulo (Tabelas C.6 e C.7), a partir das características geométricas e dimensionais desses elementos, são desenhadas as peças necessárias para completar o modelo CAD do módulo do eixo Z.

Tabela C.6: Componentes mecânicos para o Módulo do eixo Z

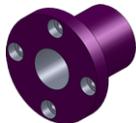
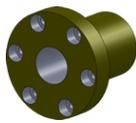
| Componentes                                      | Quant. | Caraterística geométrica  |
|--|--------|---|
| Fuso de esferas recirculantes Ø20 mm x 320 mm    | 1      |   |
| Barra guia linear de Ø 20 mm x 320 mm            | 2      |   |
| Castanha simples para guia linear de Ø 20 mm     | 4      |   |
| Castanha simples para fuso de esferas de Ø 20 mm | 1      |  |

Tabela C.7: Componentes eletromecânicos para o Módulo do eixo Z

| Componentes                               | Quant. | Caraterística geométrica   |
|---|--------|--|
| Motor de passo de 1.9N, ref. KTC-HT23-400 | 1      |  |
| Tupia Makita ref. 3709                    | 1      |  |

O desenho CAD deste módulo está dividido em duas partes, são elas, o cabeçote (ver peças que compõem o cabeçote na Figura C.8) e o grupo de movimentação do eixo Z (ver peças que compõem o grupo na Figura C.9).

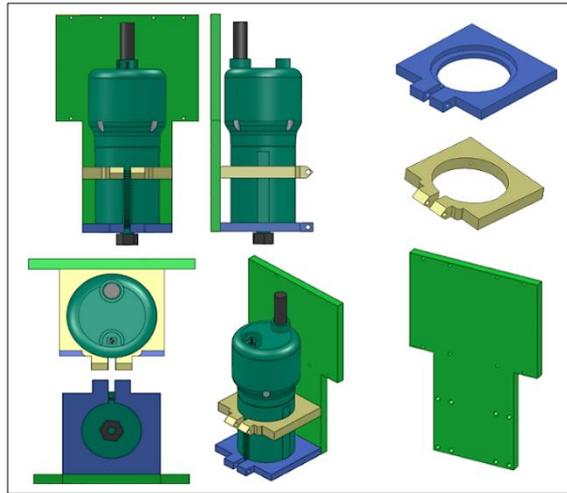


Figura C.8: Desenho CAD - Peças do Cabeçote

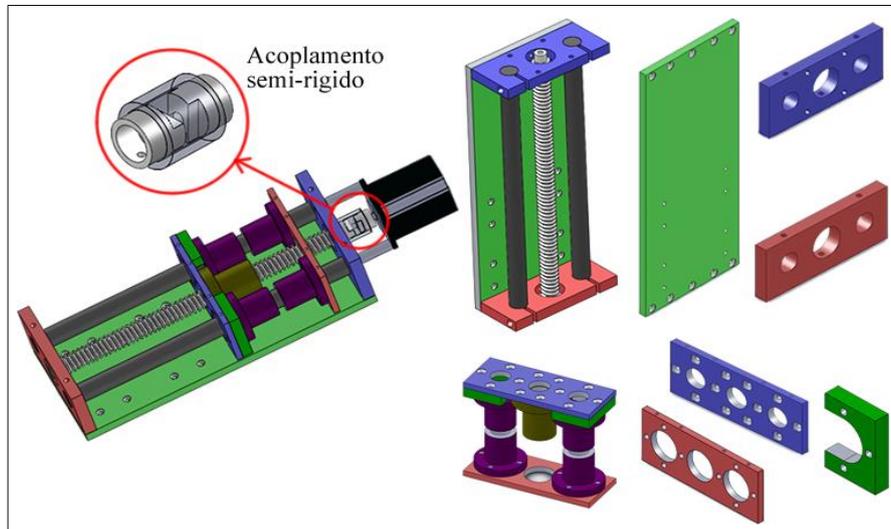


Figura C.9: Desenho CAD - Peças do Grupo de movimentação do eixo Z

Pode-se observar nas Figuras C.8 e C.9 que foi necessário criar ou desenhar novas peças para fazer o desenho CAD da montagem do módulo Z. A Figura C.10 mostra todas as peças deste módulo montadas.

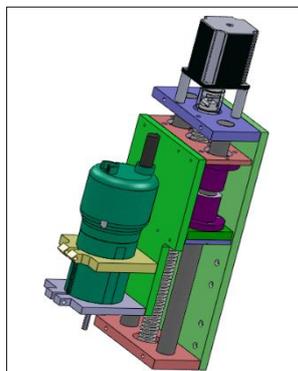


Figura C.10: Desenho CAD - Módulo do eixo Z completo

➤ **Desenho CAD do módulo do eixo X:** deve-se, primeiramente, listar os componentes que inicialmente estão disponíveis para este módulo (Tabelas C.8 e C.9). A partir das características geométricas e dimensionais destes elementos, são desenhadas as peças necessárias para completar o modelo CAD do módulo do eixo X.

Tabela C.8: Componentes mecânicos para o Módulo do eixo X

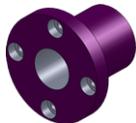
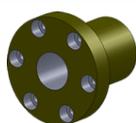
| Componentes                                      | Quant. | Caraterística geométrica  |
|--|--------|---|
| Fuso de esferas recirculantes Ø20 mm x 420 mm    | 1      |   |
| Barra guia linear de Ø 20 mm x 420 mm            | 2      |   |
| Castanha simples para guia linear de Ø 20 mm     | 4      |   |
| Castanha simples para fuso de esferas de Ø 20 mm | 1      |  |

Tabela C.9: Componentes eletromecânicos para o Módulo do eixo X

| Componentes                               | Quant. | Caraterística geométrica   |
|---|--------|--|
| Motor de passo de 1.9N, ref. KTC-HT23-400 | 1      |  |

O desenho CAD deste módulo herda várias características do grupo de movimentação do eixo Z (ver Figura C.9), com a diferença de que se faz necessário colocar duas chapas de suporte, uma superior e outra inferior, para melhorar a perpendicularidade e paralelismo do mecanismo. A Figura C.11 mostra todas as peças deste módulo montadas. Pode-se observar que foi necessário criar ou desenhar novas peças para fazer o desenho CAD da montagem do módulo Z.

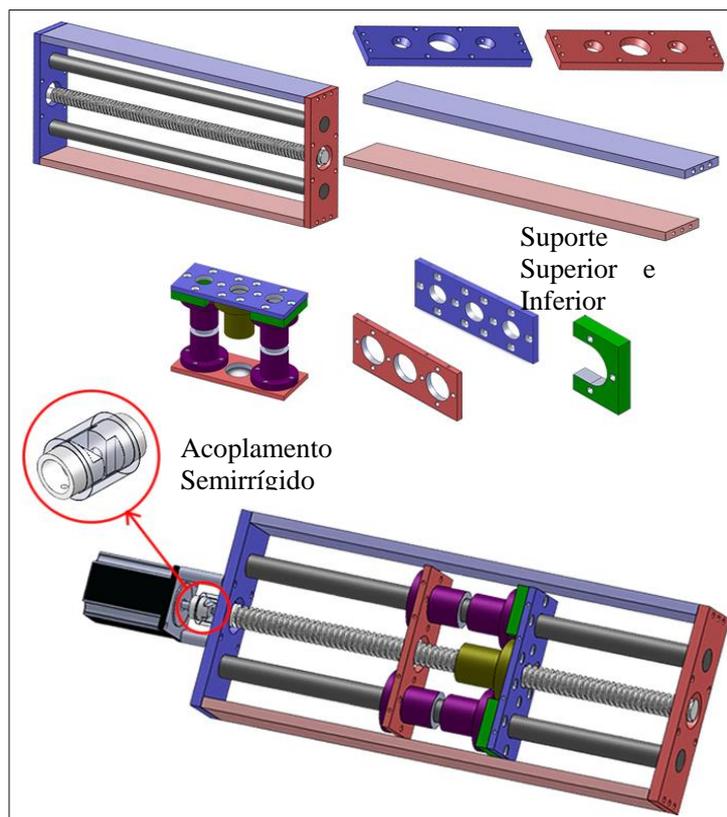


Figura C.11: Desenho CAD - Peças do módulo do eixo X

➤ **Desenho CAD do módulo do eixo Y (Portal):** deve-se, primeiramente, listar os componentes que inicialmente estão disponíveis para este módulo (Tabelas C.10 e C.11). A partir das características geométricas e dimensionais destes elementos, são desenhadas as peças necessárias para completar o modelo CAD do módulo do eixo X.

Tabela C.10: Componentes mecânicos para o Módulo do eixo Y

| Componentes                                      | Quant. | Caraterística geométrica |
|--|--------|--------------------------|
| Fuso de esferas recirculantes Ø20 mm x 1000 mm   | 1      |                          |
| Trilho linear 1000 mm                            | 2      |                          |
| Patim de carga para trilho linear                | 2      |                          |
| Castanha simples para fuso de esferas de Ø 20 mm | 1      |                          |

Tabela C.11: Componentes eletromecânicos para o Módulo do eixo Y

| Componentes                                  | Quant. | Caraterística geométrica   |
|--|--------|--|
| Motor de passo de 1.9N,<br>ref. KTC-HT23-400 | 1      |  |

O desenho CAD deste módulo é totalmente diferente dos já apresentados, pois possui dois trilhos que serão os encarregados de ajudar a suportar o peso e a deslocar os módulos X e Z. Para este desenho foi necessário criar duas chapas em forma de L para conectar os trilhos/patins com o módulo X (ver Figura C.13). Logo, o desenho da estrutura deste módulo tem como referência a forma de um portal. Há duas partes que formam o módulo do eixo Y ou portal, são elas: a base do portal (ver peças que conformam a base na Figura C.12) e o portal (ver peças que compõem o portal na Figura C.13),

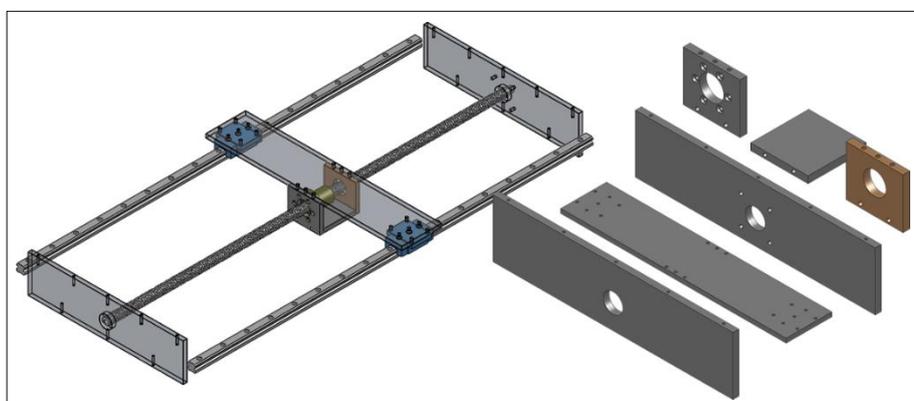


Figura C.12: Desenho CAD – Peças da base portal do modulo do eixo Y

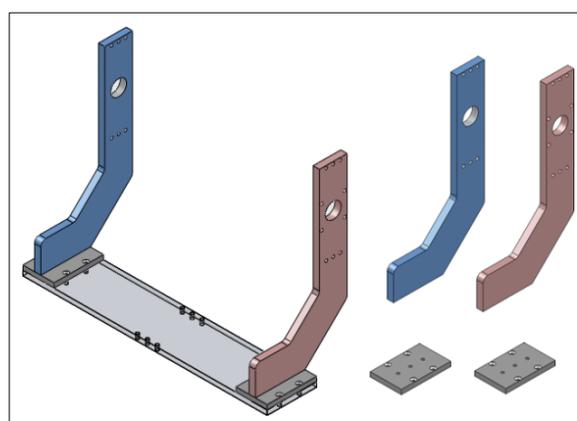


Figura C.13: Desenho CAD - Peças do Portal do modulo do eixo Y

Como pode-se observar nas Figuras C.12 e C.13, há peças novas que foram criadas no desenho CAD para se conseguir fazer a montagem do módulo Y (portal). A Figura C.14 mostra todas as peças deste módulo montadas.

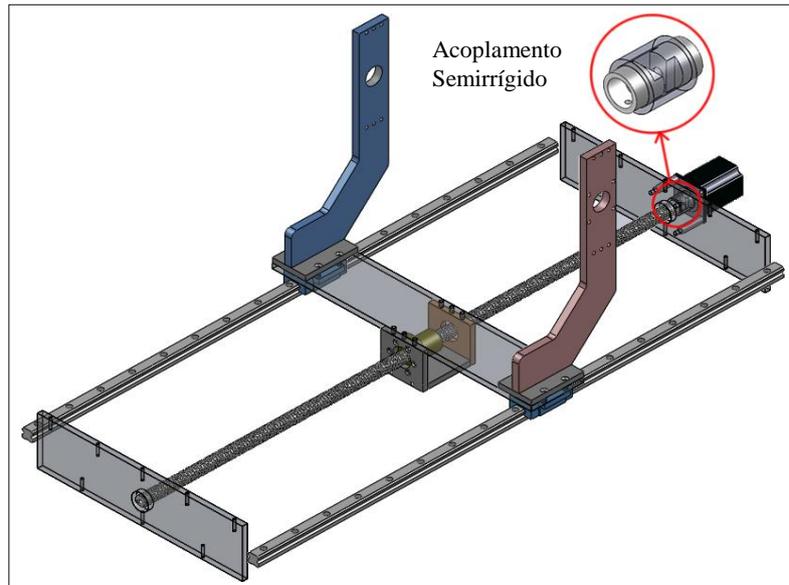


Figura C.14: Desenho CAD - Módulo do eixo Y (portal) completo

➤ **Desenho CAD do suporte:** O suporte da máquina deve ser apropriado para suportar as vibrações da Router no momento do seu funcionamento e, além disso, deve fornecer uma estrutura sólida para a fixação de todos os componentes da máquina, incluindo, o sistema de controle, além de possuir uma adequada ergonomia para o usuário final. Para desenhar o suporte da máquina, toma-se como referência um projeto feito na faculdade de tecnologia da UNB/GRACO sobre o desenvolvimento de uma fresadora CNC didática e o uso de perfisados industriais [92]. Os perfis metálicos usados como base para o desenho CAD são tubos quadrados, tubos redondos, cantoneiras dobrada de abas iguais e barras chatas. Esses perfis seguem os padrões da indústria de perfilados. Na Tabela C.12 são ilustrados esses materiais.

Tabela C.12: Perfis Industriais [97].

| Componentes                       | Caraterística geométrica |
|-----------------------------------|--------------------------|
| Tubo Industrial redondo           |                          |
| Tubo Industrial quadrado          |                          |
| Cantoneira dobrada<br>Abas iguais |                          |
| Ferro chato                       |                          |

Há três partes que compõem o suporte, são elas: suporte para a Router CNC, suporte para o sistema de controle e suporte para o Controlador. O sistema de Controle é descrito no próximo item e o Controlador é descrito no capítulo 5. A Figura C.15 mostra o desenho CAD do suporte.

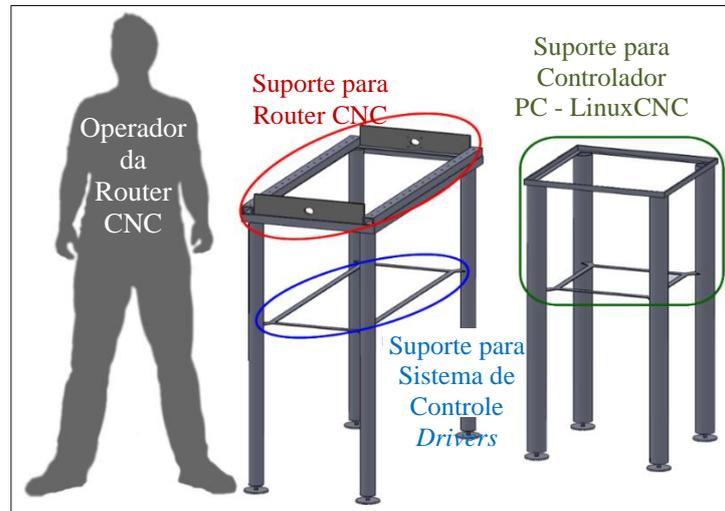


Figura C.15: Desenho CAD Suporte

Depois de descrever os módulos e demais componentes da Router CNC, continua-se com a montagem final da máquina. Em seguida, é apresentada detalhadamente a Router CNC com seus componentes hardware e *software*. Na Figura C.18 apresentam-se os desenhos CAD com múltiplas vistas da Router CNC.

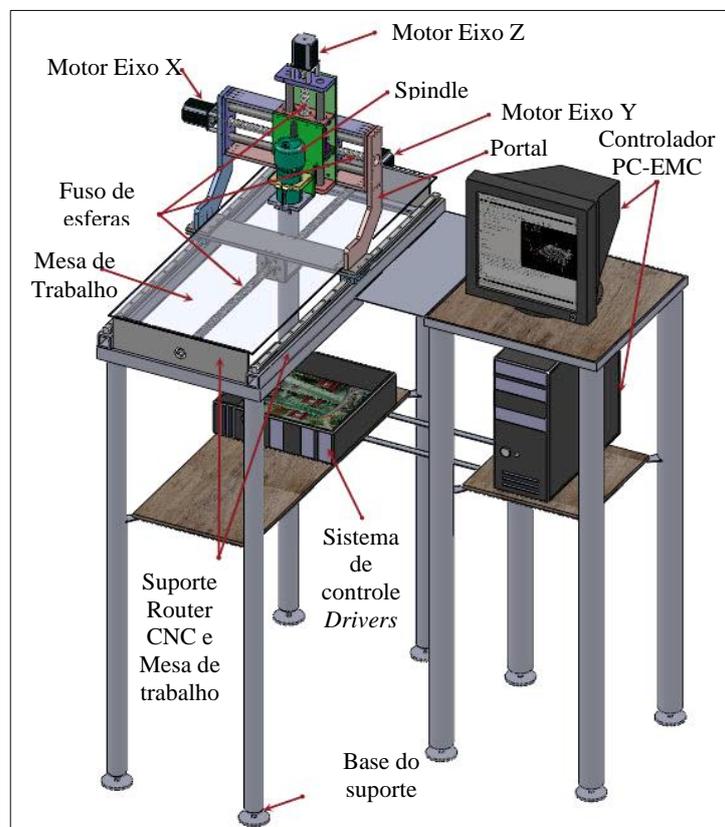


Figura C.16: Desenho CAD Router CNC completo

➤ **Desenho CAD do aparelho para troca de ferramenta Manual:** Este dispositivo é o encarregado de “memorizar” mecanicamente o ponto coordenado (XYZ) de referência da ferramenta de corte para ajudar na troca de ferramenta de forma manual. Este aparelho deve ser flexível e deve se fixar na mesa de trabalho sem atrapalhar o espaço efetivo de trabalho do Router CNC. Pode-se observar o desenho CAD do aparelho na Figura C.17. Os componentes deste aparelho são móveis o que permite a modificação das suas articulações e, desta forma, fixar e “memorizar” temporalmente a posição da ferramenta de corte nesse ponto XYZ.

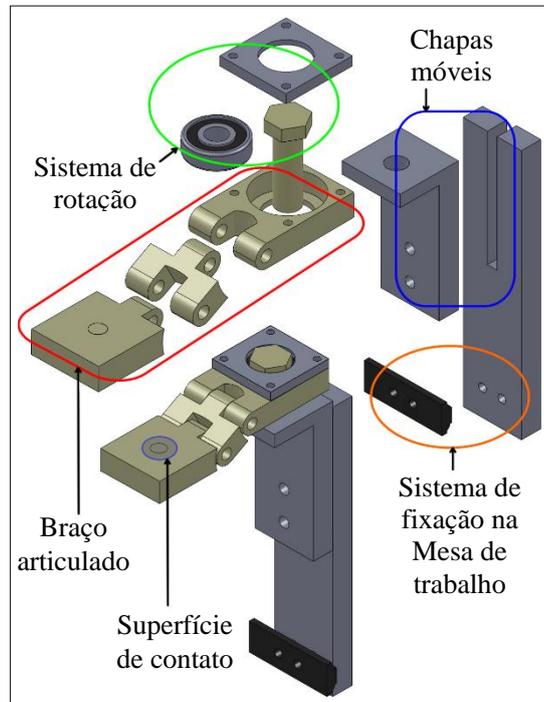


Figura C.17: Desenho CAD Aparelho para troca de ferramenta

Como observado na Figura C.17 o desenho CAD do aparelho para a troca de ferramenta está dividido geralmente em 5 partes, são elas:

- a. **Sistema de fixação na Mesa de trabalho:** este consiste em duas peças que ajudam a fixar a base do aparelho na mesa de trabalho.
- b. **Chapas móveis:** estas chapas permitem modificar rapidamente a altura do braço articulado do dispositivo e, conseqüentemente, a superfície onde a ponta ou as arestas da ferramenta de corte faz contato. Desta forma, a modificação da altura não oferece precisão.
- c. **Sistema de rotação:** consiste em quatro peças, incluindo um rolamento que ajuda a rotar o braço articulado a fim de não atrapalhar o espaço de trabalho efetivo do Router CNC.
- d. **Braço articulado:** este consiste em três peças articuladas que permitem modificar a altura da superfície de contato de forma mais precisa que as chapas móveis.
- e. **Superfície de contato:** esta superfície é essencialmente o ponto coordenado XYZ no qual a ponta ou as arestas da ferramenta de corte faz contato.

O uso de este dispositivo é simples. Primeiramente deve-se colocar o cabeçote em um ponto de referência XYZ o mais perto possível da superfície de contato do braço articulado, relacionando-o a um sistema de coordenadas (e.g. G55, G56 ou outro). Logo, deslocar manualmente o braço do aparelho até conseguir que a superfície de contato toque a ponta ou as arestas da ferramenta de corte. Desta forma, pode-se “memorizar” mecanicamente esse ponto (posição de contato) para depois usá-

lo como referência para a troca de ferramenta. Resumindo, o desenho CAD do Router CNC permite identificar todos os componentes necessários para a sua construção, evitando o desperdício de material. Na Figura C.18 apresenta-se o desenho CAD terminado do Router CNC com todos seus componentes. A Tabela C.13 mostra a lista dos materiais necessários para a fabricação do Router CNC.

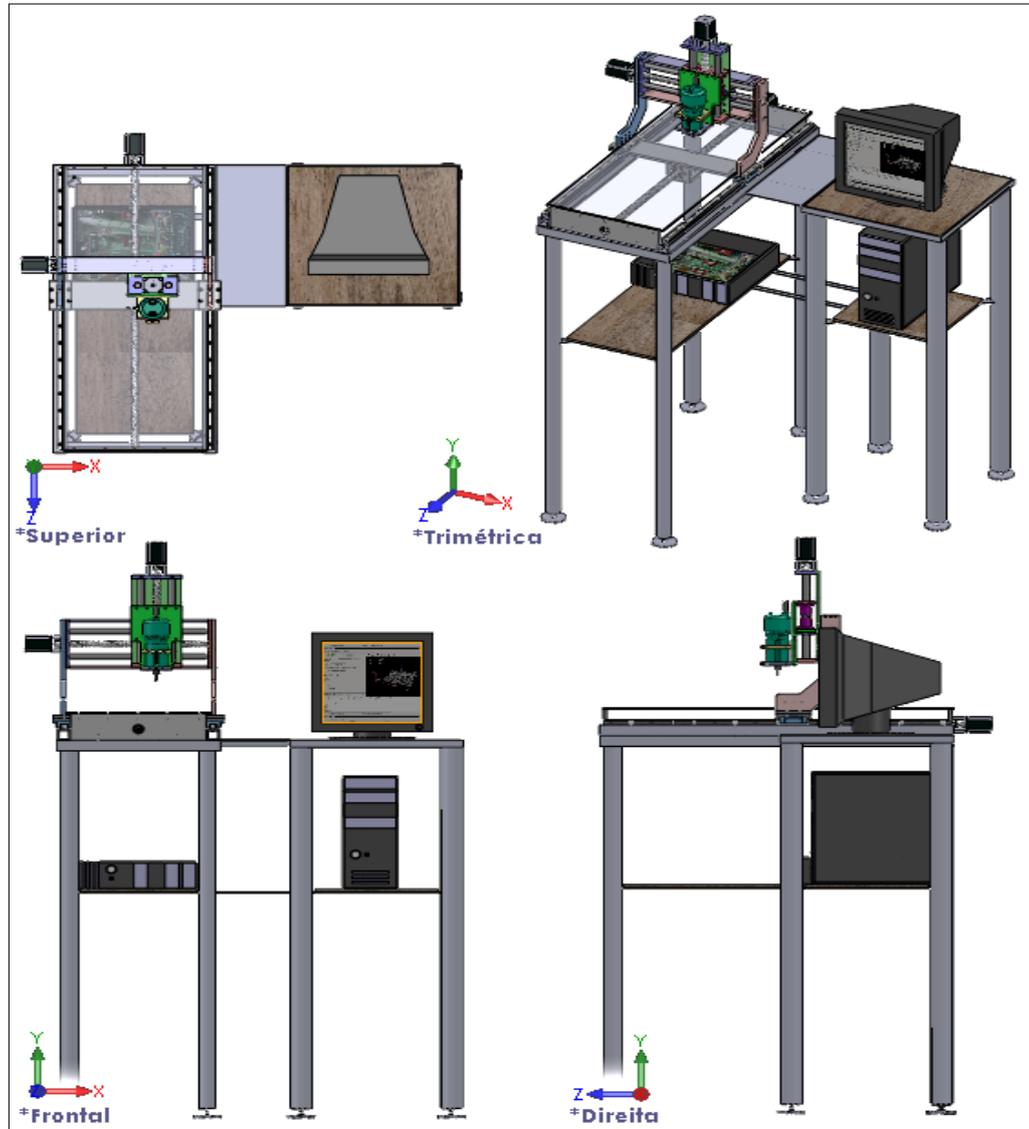


Figura C.18: Desenho CAD Router CNC múltiplas vistas.

Como experiência de pré-fabricação, uma parte do desenho CAD da máquina foi impresso em 3D para conferir o modelo do Router CNC projetado e ter-se uma ideia de como seria o resultado final da implementação do Router CNC. Na Figura C.19 pode-se visualizar o resultado da impressão 3D em plástico ABS.

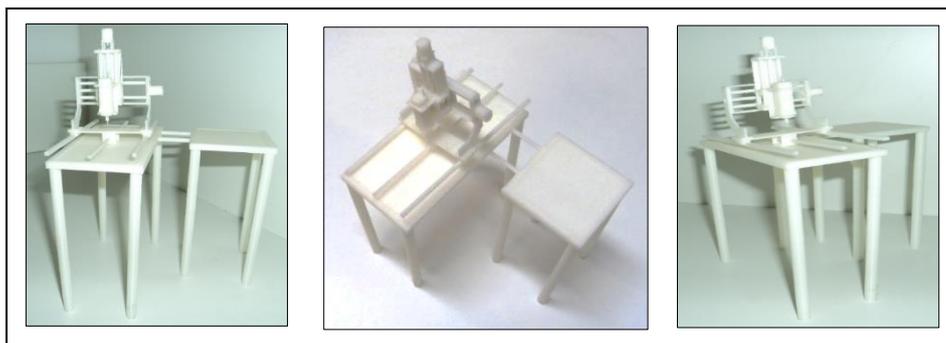


Figura C.19: Impressão 3D do Router CNC, escala 1:10

Tabela C.13: Lista de novos materiais

| Quant. | Material                         | TIPO | Comprimento mm                 |
|--------|----------------------------------|------|--------------------------------|
| 36     | Parafuso allen cabeça cilíndrica | M5   | 20                             |
| 24     | Parafuso allen cabeça cilíndrica | M5   | 25                             |
| 24     | Parafuso allen cabeça cilíndrica | M5   | 35                             |
| 48     | Parafuso allen cabeça cilíndrica | M6   | 65                             |
| 12     | Parafuso allen cabeça cilíndrica | M6   | 70                             |
| 24     | Parafuso allen cabeça chata      | M4   | 20                             |
| 24     | Parafuso allen cabeça chata      | M5   | 25                             |
| 12     | Parafuso allen cabeça chata      | M5   | 30                             |
| 36     | Parafuso allen cabeça chata      | M5   | 35                             |
| 12     | Parafuso allen cabeça chata      | M5   | 45                             |
| 100    | Porca auto-travante              | M5   | M5                             |
| 100    | Arruela de pressão               | M5   | M5                             |
| 7      | Rolamento de esfera radial       |      | Dint:15 x Dext:32 x Largura:10 |
| 1      | Chapa de alumínio 6,35mm         | 1/4" | 1500x1500                      |
| 1      | Chapa de alumínio 9,53mm         | 3/8" | 1000x1000                      |
| 1      | Chapa de alumínio 12,7mm         | 1/2" | 1000x1000                      |

### C.3. USINAGEM E MONTAGEM DAS PEÇAS DO ROUTER CNC

Neste item é apresentado o resultado final da usinagem das peças do desenho CAD do Router CNC, assim como a montagem dos módulos, mesa de trabalho e suporte. A usinagem das peças foi feita no laboratório de mecânica do SG9 (serviços gerais 9) da Universidade de Brasília. É importante ressaltar que esse laboratório não possui máquinas de controle numérico para usinar metal. Logo as peças desenhadas para o Router CNC foram feitas manualmente, usando a experiência dos técnicos do laboratório e a experiência própria.

Os desenhos nos quais foi baseada a usinagem das peças projetadas para o Router CNC podem ser consultadas no site web do projeto deste trabalho de mestrado: <https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/>.

Na montagem e fixação dos componentes do Router CNC, foram usados diferentes tipos de parafusos, arruelas de pressão, porcas auto-travantes e rolamentos. A seguir, são mostradas as peças usinadas, a montagem dos módulos, a mesa de trabalho e o suporte (Figuras C.20 até C.28).



Figura C.20: Montagem do Cabeçote (peças e tupa)

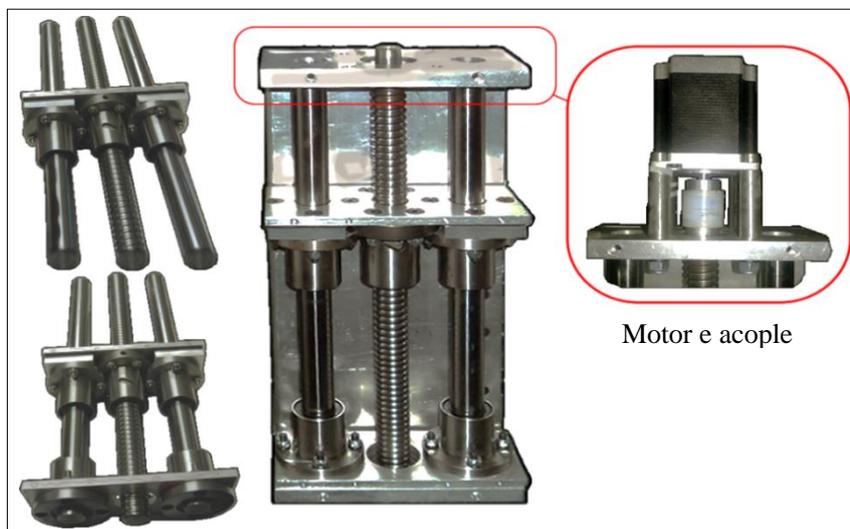


Figura C.21: Montagem grupo de movimentação eixo Z (peças, motor e acople)



Figura C.22: Módulo do Eixo Z



Figura C.23: Módulo do Eixo X

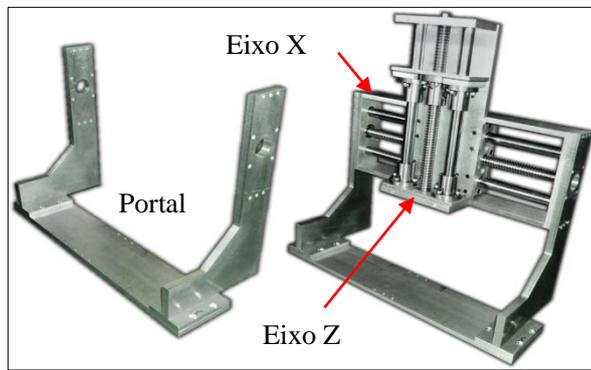


Figura C.24: Portal com o módulo do eixo X e Z

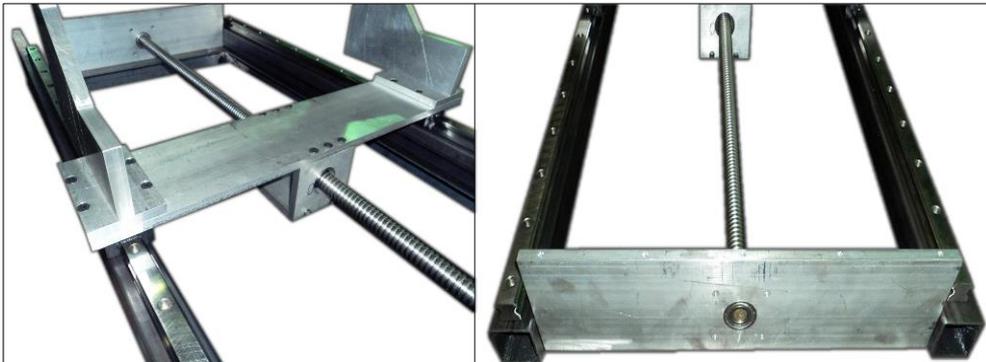


Figura C.25: Base do Portal

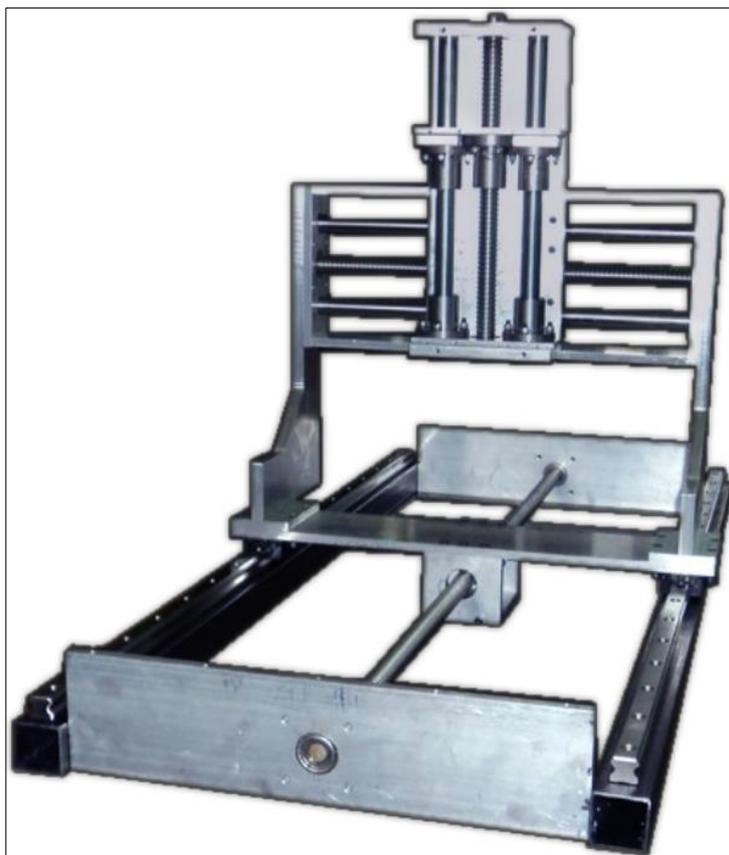


Figura C.26: Montagem do módulo do eixo Y com o módulo do eixo X e Z

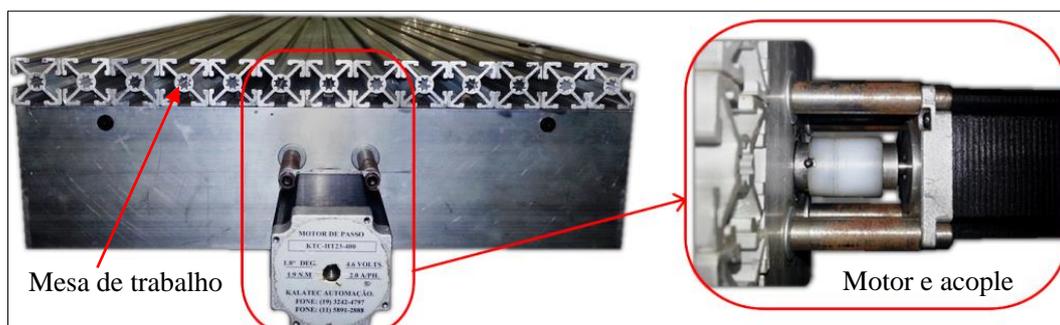


Figura C.27: Mesa de trabalho e motor do módulo do eixo Y

#### C.4. DISPOSITIVO DE TROCA DE FERRAMENTA

Em seguida, na Figura C.29, apresenta-se o dispositivo de troca de ferramenta totalmente montado e fixado na mesa de trabalho. A Figura C.30 mostra um exemplo de uso deste aparelho onde se unem a ferramenta de corte e a superfície de contato do dispositivo de troca de ferramenta.

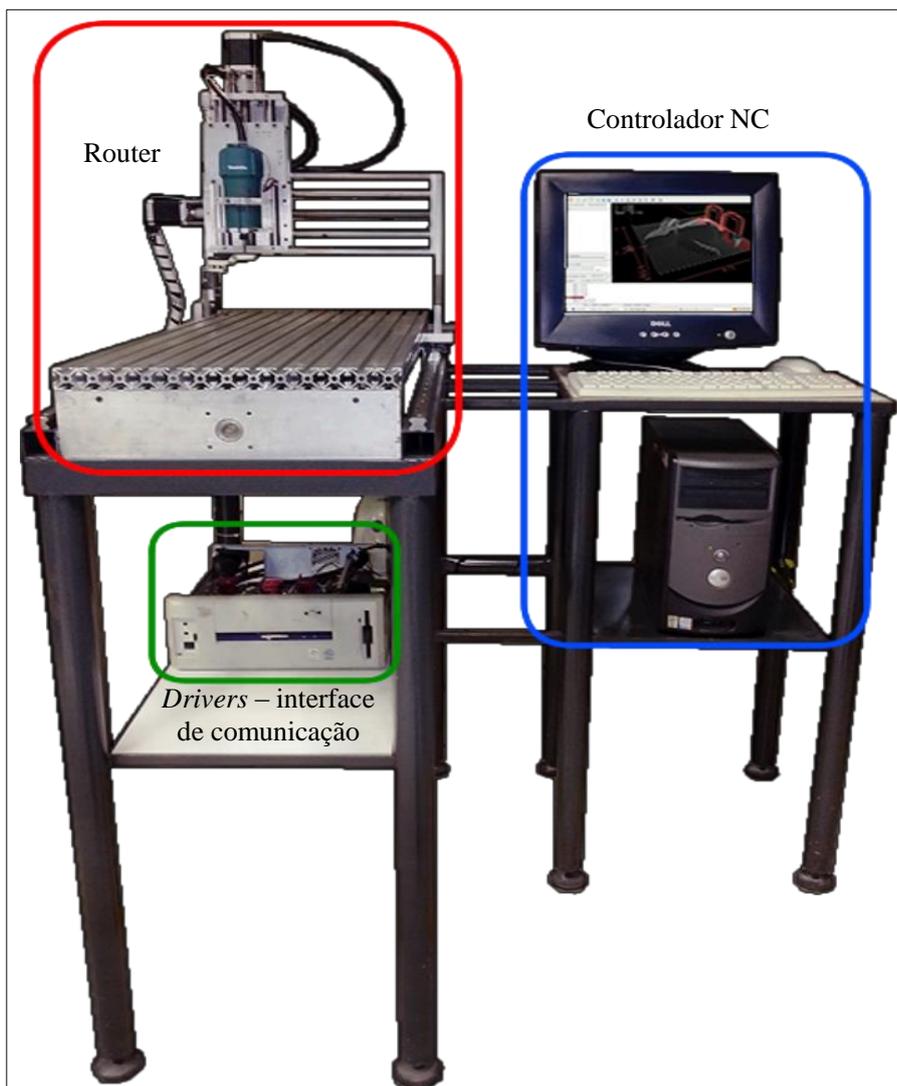


Figura C.28: Montagem completa da Router CNC, mesa de trabalho e suporte

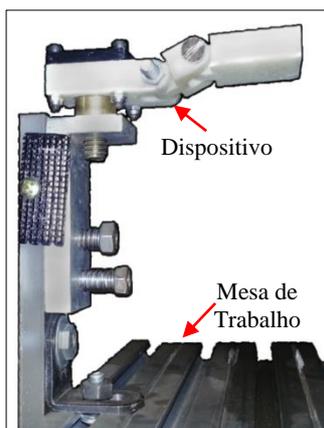


Figura C.29: Montagem do dispositivo de troca de ferramenta

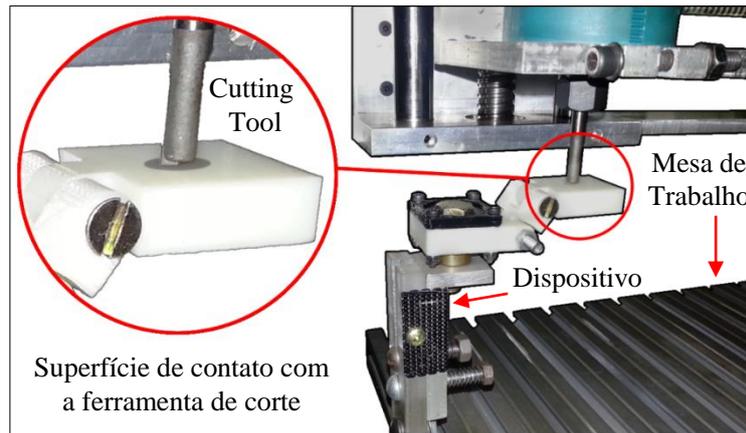


Figura C.30: Dispositivo de troca de ferramenta: Superfície de contato com ferramenta de corte do Router CNC.

### C.5. MOTORES DE PASSO

O projeto inicialmente contava com os motores de passo modelo KTC-HT23-40 (ver anexo 2). Esses são otimizados para uso em aplicações de *Micro Passo*, portanto, com a existência desses motores é apenas necessário analisar a suas características e saber se são apropriados para usá-los no Router CNC. Na Tabela C.2, foram referidas suas características.

Os motores disponibilizados podem ser acionados no modo unipolar ou bipolar. O correto modo de acionamento deve ser baseado no torque exigido ao motor. A partir do conhecimento do torque que deve atingir o motor, pode-se escolher melhor a controladora dos motores de passo.

Para o cálculo do torque que deve atingir o motor, é necessário realizar algumas considerações e obter alguns dados técnicos a respeito do fuso de esferas recirculantes, do motor de passo e da carga que será submetida ao fuso [92]. Primeiramente considera-se o motor de passo acoplado ao fuso que por sua vez desloca uma carga (e.g. o Spindle), como mostra a Figura C.31.

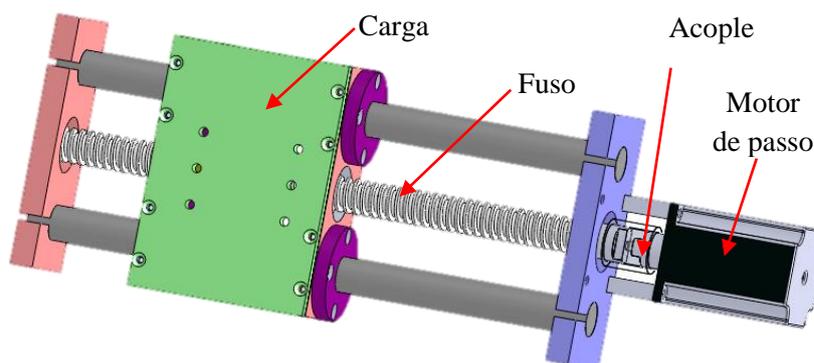


Figura C.31: Desenho CAD da integração entre motor, fuso e carga (Adaptado de [92])

Segundo PARKER [98], o torque necessário ao motor é definido pela equação C-1:

$$T = 2 \times \pi \times J \times \text{rot}/\text{seg}^2 \quad (\text{C-1})$$

A equação (C-1) descreve que o torque é o produto da inércia e da aceleração, T em  $N.m$  e J em  $Kg.m^2$ .

A inércia (J) é referente à soma das inércias do rotor, do fuso e da carga. A aceleração é uma estimativa do valor esperado para a aplicação e, portanto, escolhida pela experiência do projetista.

Segundo o Anexo 1, o rotor tem inércia igual à  $480 g.cm^2$ . Este valor é equivalente à  $48.10 - 6 Kg.m^2$ .

Segundo PARKER [98], a inércia do fuso pode ser estimada pela equação C-2:

$$Jc = 761 \times D^4 \times L \quad (C-2)$$

A equação C-2 descreve a inércia de uma barra cilíndrica de aço, onde D em m, L em m e J em  $Kg.m^2$ .

Se:  $D = 25.10 - 3 m$  e:  $L = 0,45 m$ , temos:  $Jc = 133,77.10 - 6 kg.m^2$ .

A carga também é estimada de acordo com a experiência do projetista. Tomando o pior caso, pode-se considerar igual a  $60 Kg$ . Segundo PARKER [98], para um sistema de parafuso, a inércia referente à carga é estimada pela equação C-3:

$$Jw = W \times p^2 \div 4 \times 10^7 \quad (C-3)$$

A equação C-3 descreve que a inércia é igual ao produto da massa, em Quilogramas (Kg), e o passo do fuso em milímetros (mm).

Portanto,  $Jw = 37,5.10 - 6 Kg.m^2$ , visto que  $W = 60Kg$  e  $p = 5mm$ .

A inércia total é igual à soma das três parciais já calculadas pelas equações C-2 e C-3 e ainda a inércia especificada na nota técnica do motor de passo. Portanto,  $J = 219,27.10 - 6 Kg.m^2$ .

Considera-se ainda que a eficiência do fuso, o coeficiente de atrito dos rolamentos e o desalinhamento entre o eixo do motor e fuso, aumente a inércia em até 70%. Com isso tem-se uma inércia total (Jt):  $Jt = 350,83.10 - 6 Kg.m^2$

O último parâmetro que ainda resta encontrar é a aceleração. Conhecendo as máquinas CNC do mercado nacional, encontram-se uma velocidade máxima de trabalho na ordem de 900 até 2000 mm/min para motores semelhantes ao KTC-HT23-400. Esse valor corresponde a 5 mm em 85ms, ou seja, 1 passo a cada 85ms. Sabendo que o movimento do eixo possui um perfil trapezoidal, ou seja, parte da velocidade zero e aumenta linearmente até a velocidade máxima, mantém depois a velocidade constante e posteriormente começa a decair linearmente até a parada, é possível determinar a aceleração pela equação C-4. Na Figura C.32, mostra-se o gráfico do movimento com perfil trapezoidal.

Segundo PARKER [98], a aceleração para um movimento com perfil trapezoidal é definida pela seguinte equação:

$$Ac = 4,5 \times D \div t^2 \quad (C-4)$$

A equação C-4 descreve a aceleração para um movimento com perfil trapezoidal, com D em rotações e t em segundos.

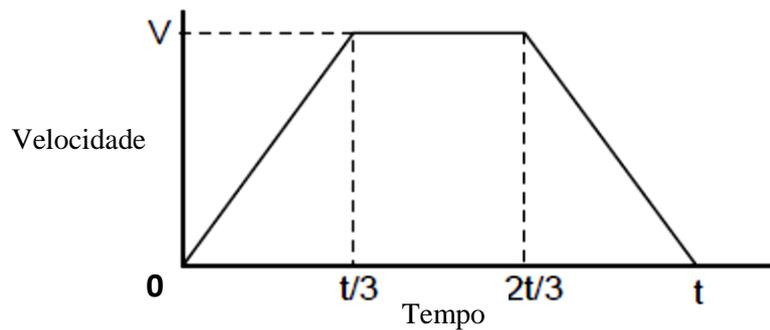


Figura C.32: Perfil trapezoidal (Adaptado de [98])

Se a carga deve mover 5mm em 85ms, temos:  $A_c = 622,83 \text{ rot/s}^2$ .

Substituindo os valores encontrados na Equação C-1, encontra-se o valor do torque estático exigido ao motor:

$$T = 1,37 \text{ N.m}$$

“ $T$ ” é o valor estimado do torque necessário no motor para cumprir com os requisitos da máquina. Em conclusão analisou-se que os motores disponíveis atendem às necessidades do projeto, visto que a necessidade do projeto é de 1,37 N.m e a capacidade do motor em modo bipolar é 1,82 N.m como se pode olhar no anexo 2. Os motores disponibilizados podem ser acionados no modo bipolar ou no modo unipolar. A decisão correta do modo de acionamento deve ser baseada no torque exigido ao motor. A partir do conhecimento do torque calculado anteriormente, pode-se escolher adequadamente a melhor controladora de motores de passo, neste caso para modo bipolar.

## C.6. SISTEMA ELETRÔNICO DE CONTROLE DO ROUTER CNC

O controle físico de máquinas-ferramenta é tarefa de vários componentes necessários para a conversão de comandos de *software* em sinais elétricas [99]. Esses sinais geralmente controlam os motores de cada eixo da máquina, além de outros componentes como o *Spindle*, dispositivos para troca automática de ferramentas, entre outros. Neste projeto, o sistema de controle está encarregado de controlar a movimentação dos eixos do Router CNC. Esse sistema está dividido em duas partes, a saber: componentes eletrônicos de potência (*Hardware*) e controlador NC (*Software*). Neste item só é apresentado os componentes eletrônicos. O controlador NC é descrito detalhadamente no apêndice B.

Os componentes eletrônicos são um conjunto de placas controladoras de motores de passo e uma placa de comunicação via porta paralela. Existem três placas controladoras de motores de passo, uma para cada motor de cada eixo (XYZ) do Router. Essas placas, também chamadas controladoras de potência ou ainda *drivers*, são basicamente uma fonte de corrente controlada. Seu objetivo é fornecer uma corrente adequada ao motor de passo [92].

Para controlar os motores de passo do Router CNC, foram utilizadas os *drivers* SMC-B-PRO fabricada por HobbyCNC Brasil. Esses *drivers* permitem o acionamento de motores de passo bipolares e a movimentação do motor por meio de micropassos.

Segundo LYRA [92], o acionamento bipolar aumenta em aproximadamente 40% o torque em relação ao acionamento unipolar. Portanto, o acionamento do motor de passo KTC-HT23 em forma bipolar permite utilizar o maior torque que o motor pode oferecer. A Figura C.29 exibe o *driver* SMC-B-PRO e a Figura C.34 ilustra o esquema de conexão da fonte de alimentação e motor junto ao *driver*.

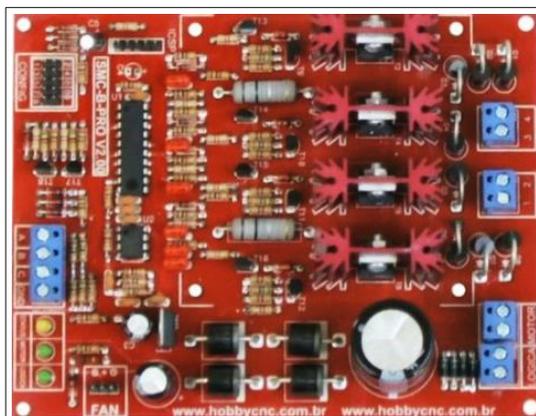


Figura C.33: *Driver* SMC-B-PRO [100]

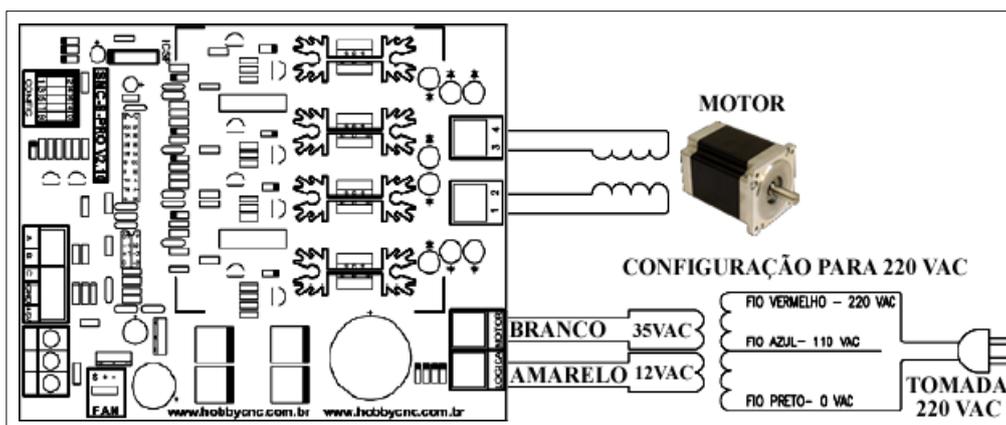


Figura C.34: Esquema de conexão *Driver*/Motor/Fonte de alimentação (Adaptado de [100])

Além disso, segundo o guia de utilização do SMC-B-PRO (ver Anexo 4), essa placa tem entrada de controle padrão STEP/DIR/ENA (Passo / Direção / Habilitação). A configuração de corrente e a resolução de passos é feita via *software* (controlador NC). Logo, é possível, através do computador (PC), controlar velocidade, sentido de rotação e passo com grande precisão.

O segundo componente eletrônico é a placa de comunicação via porta paralela. Neste projeto foi usada a interface opto-isolada BPC-PRO para comunicar os *drivers* dos motores de passo com o PC via porta paralela, assim como colocar os limites (home) de cada eixo e o botão de parada de emergência. Na Figura C.35, apresenta-se interface opto-isolada BPC-PRO.

Segundo o guia de utilização do BCP-PRO (ver Anexo 5), a interface BPC possui entradas opto-isoladas para controle de motores de passo com os sinais STEP/DIR/ENA, entradas para fins de curso e entrada para sinal de parada de emergência. Na Figura C.36, mostra-se a identificação dos pontos de conexão do BPC-PRO.



Figura C.35: Interface opto isolada BPC-PRO [100]

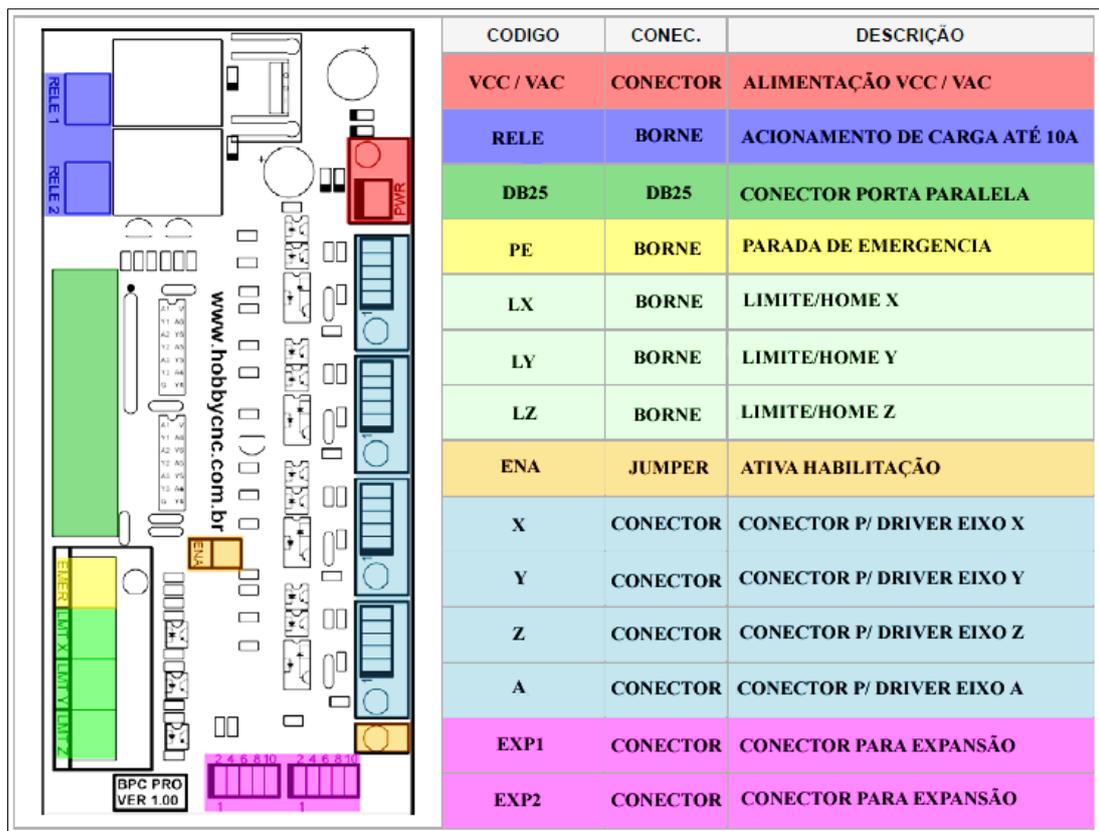


Figura C.36: Identificação dos pontos de conexão do BPC-PRO [100]

Na Figura C.37, pode-se observar o projeto completo de conexão de todos os componentes eletrônicos do sistema de controle juntos, assim: motores de passo, *drivers* de potência, fins de carreira, botão de parada de emergência e um PC com LinuxCNC.

A Figura C.38 apresenta a montagem do sistema de controle em relação aos componentes eletrônicos deste projeto. Por fim, como mostrado na Figura C.28, apresenta-se o Router CNC montado completamente.

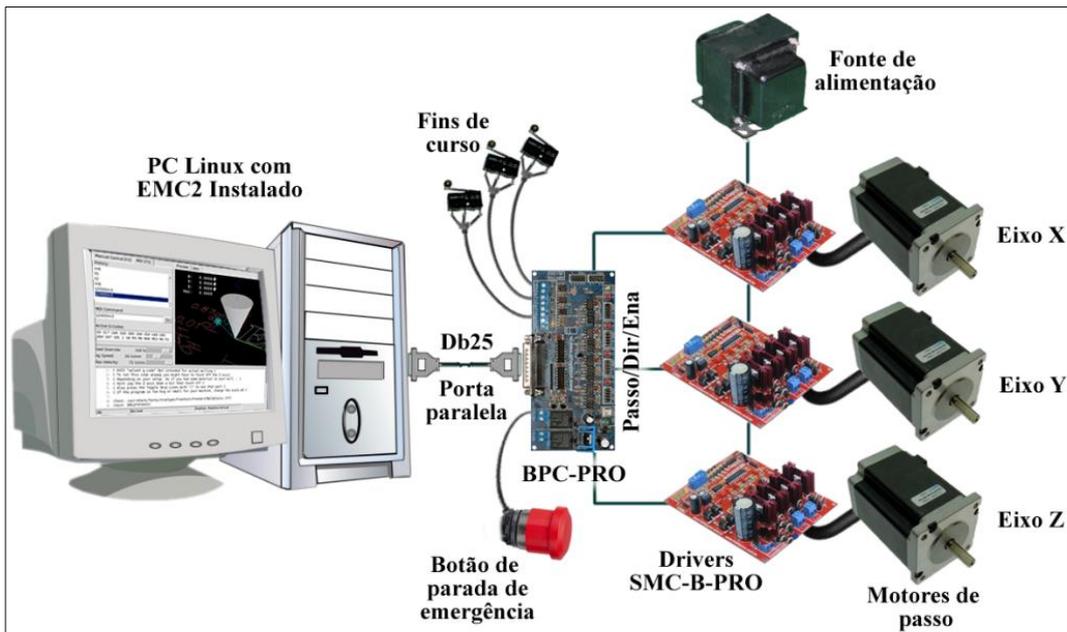


Figura C.37: Sistema de controle do Router CNC (Adaptado de [84])

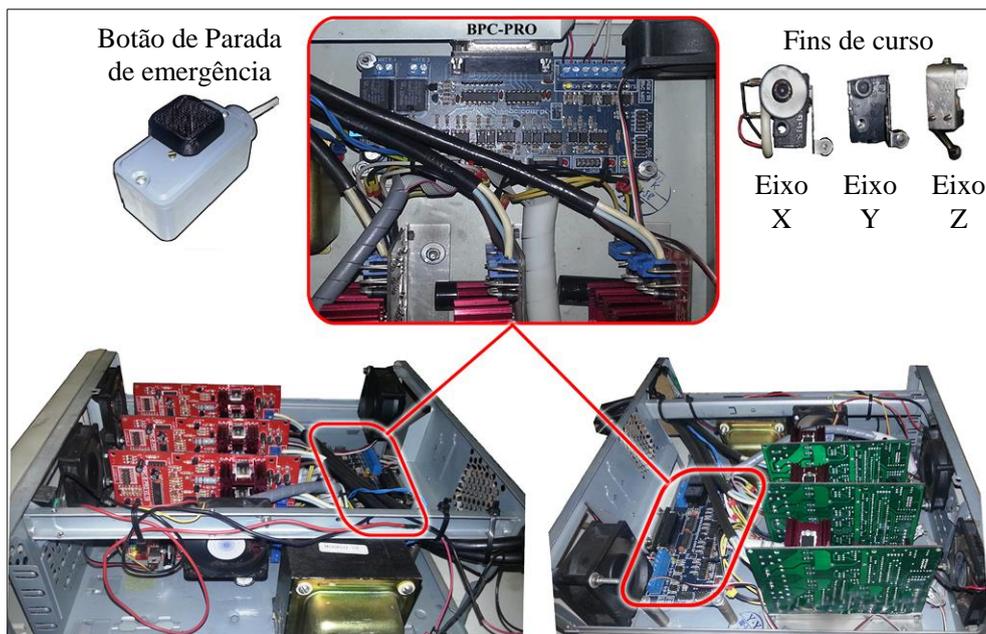


Figura C.38: Montagem do sistema de controle do Router CNC

## C.7. PARÂMETROS DE CORTE DO ROUTER CNC

O Router CNC pertence às máquinas-ferramenta que permitem levar a cabo o processo de usinagem tipo fresamento, existem, porém, uma série de importantes parâmetros de corte a considerar. Esses parâmetros descrevem quantitativamente os movimentos, as dimensões e outras características da operação de corte.

Os parâmetros que descrevem o movimento da ferramenta e/ou peça são: velocidade de corte e velocidade de avanço. As dimensões do corte são: profundidade de corte e penetração de trabalho.

Outros parâmetros são: diâmetro da ferramenta e seu número de dentes (arestas principais), taxa de remoção de material e o tempo de corte [101].

Para definição e medição dos ângulos da ferramenta e outros parâmetros é usado um ponto selecionado sobre a aresta como referência, como mostrado na Figura C.39.

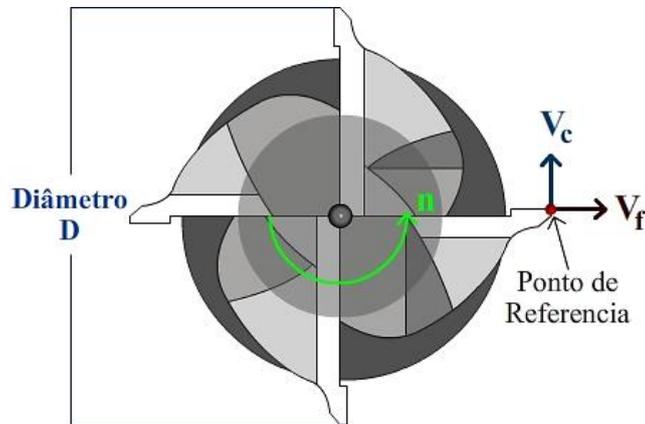


Figura C.39: Referencias da Ferramenta de corte.

A seguir são explicados os diferentes parâmetros supracitados, aplicando-os no processo de fresamento do Router CNC desenvolvido neste projeto.

➤ **Velocidade de Corte:** é a velocidade com que os dentes da ferramenta de corte tocam a peça a ser usinada [102]. O movimento de corte no fresamento é circular. Portanto, para um diâmetro da ferramenta  $d$  em mm o percurso da aresta numa revolução é  $d * \pi$  mm e em  $n$  revoluções por minuto será [103] (ver equações C-5 e C-6):

$$V_c = d * \pi * n \quad [mm] \quad (C-5)$$

$$V_c = \frac{d * \pi * n}{1000} \quad [m] \quad (C-6)$$

Segundo VERLAG et al. [103], apesar de ter, teoricamente, a equação que define a velocidade de corte, a verdade é que ela está regulada por diversos fatores, tais como: o tipo de material da ferramenta, do tipo de material da peça, pelo procedimento de trabalho (desbaste, acabamento, etc.) pelo meio de refrigeração, pelo funcionamento estável do Router CNC, pela fixação da peça bruta, e também pela largura e profundidade da operação de fresamento. Em suma, o cálculo da velocidade de corte pode se tornar uma tarefa manual, feita diretamente pelo operador no manuseio da máquina.

➤ **Velocidade de avanço:** O avanço ou a velocidade de avanço no fresamento é a velocidade relativa entre a ferramenta de corte e a peça, ou seja, a velocidade com que o corte progride. O raio da ponta ou também a aresta da ferramenta de corte e o avanço são os dois fatores mais importantes dos quais depende a rugosidade da superfície obtida no processo de fresamento [104]. Cada ferramenta pode cortar adequadamente numa variedade de velocidades de avanço por cada revolução dada. Isto é chamado de avanço por revolução, identificada com o índice  $f$ . Essa variedade depende fundamentalmente do número de arestas de corte ou dentes da ferramenta de corte, do tamanho de cada um e da profundidade de corte, do tipo de material da peça e o tipo de corte. Essa variedade de velocidades é determinada experimentalmente ou são dadas pelo fabricante da

ferramenta de corte. Além disso, a velocidade de avanço está limitada pela fixação da peça, da ferramenta de corte e pela potência do motor, que se reflete no avanço da máquina [105]. A espessura máxima do cavaco em mm é o indicador mais importante da limitação da ferramenta de corte na operação de fresamento. Segundo CIMM [101], o avanço por revolução ( $f$ ) é a distância linear percorrida por um conjunto de dentes ou aresta de corte que compõe uma ferramenta durante uma rotação completa da mesma, medido no plano de trabalho. O avanço por revolução  $f$  é o produto do avanço por dente ( $f_z$ ) pelo número de dentes ou aresta de corte ( $z$ ) da ferramenta (ver equação C-7).

$$f = f_z * z \left[ \frac{mm}{rev} \right] \quad (C-7)$$

Por fim, a velocidade de avanço é o produto do avanço por revolução pela velocidade de rotação da ferramenta (ver equação C-8).

$$v_f = f * n \left[ \frac{mm}{min} \right] \quad (C-8)$$

Onde,  $f$  é avanço por revolução e  $n$  é o número de revoluções por minuto da ferramenta.

Estas equações foram testadas no desenvolvimento do adaptador para o cálculo de velocidade de avanço e de corte dos programas NC gerado por esse adaptador. Testes de usinagem foram feitos para experimentar o resultado do cálculo teórico das velocidades, mas esse cálculo não deu bons resultados na prática, portanto não serão usadas diretamente essas equações no cálculo da velocidade de avanço ou de corte no adaptador.

Como resultado final, a velocidade de avanço usada no Router CNC foi achada de forma experimental, obtendo-se os valores mostrados na Tabela C.14.

Tabela C.14: Velocidades de avanço

|                             | Eixo X       | Eixo Y       | Eixo Z       |
|-----------------------------|--------------|--------------|--------------|
| Velocidade máxima de Avanço | 400,2 mm/min | 390,5 mm/min | 409,8 mm/min |

Experimentalmente também, foi possível encontrar a aceleração máxima de avanço por meio do programa de configuração do EMC. Os resultados obtidos se apresentam na Tabela C.15:

Tabela C.15: Aceleração de avanço

|                             | Eixo X                    | Eixo Y                  | Eixo Z                    |
|-----------------------------|---------------------------|-------------------------|---------------------------|
| Aceleração máxima de Avanço | 299,8 mm/min <sup>2</sup> | 250 mm/min <sup>2</sup> | 299,8 mm/min <sup>2</sup> |

➤ **Profundidade de corte:** é a quantidade de penetração da ferramenta de corte na peça, medida no plano de trabalho na direção do eixo dessa ferramenta. Calcular a profundidade de corte permite saber quantas passadas a ferramenta de corte deve dar sobre a peça a fim de retirar o cavaco e deixar a peça no tamanho desejado. Segundo TURCATO [106], na prática, a profundidade de corte depende muito da experiência do operador, quem identifica a resistência e robustez da máquina-ferramenta. No entanto explica que na prática, a máxima profundidade de corte adotada para fresamento é de até 1/3 do comprimento de corte da fresa. A Figura C.40 ilustra esta ideia.

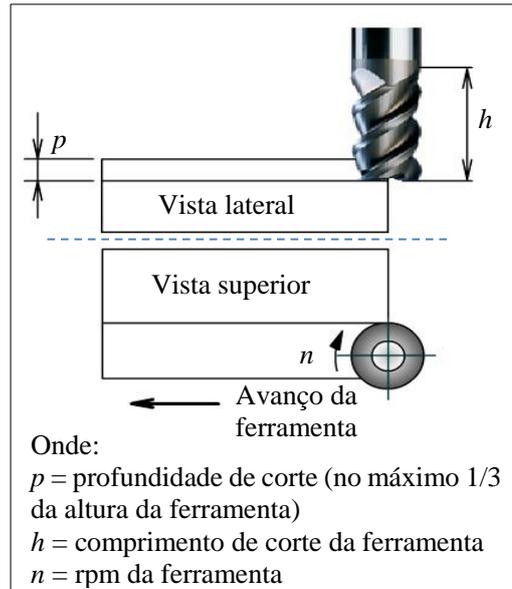


Figura C.40: Profundidade de Corte (Adaptado de [106])

Já o manual de operação do *Spindle* (Tupia Makita 3709), elucida sobre a profundidade de corte adequada para a operação de fresamento. Segundo esse manual de operação, dado que a excessiva operação de corte pode causar sobrecarga do motor ou dificuldade em controlar a ferramenta, a profundidade de corte não deve ser mais do que 3 mm por passagem. Quando precisar atender uma profundidade de corte com mais de 3 mm, deve-se fazer várias passagens progressivamente até chegar à profundidade desejada. Embora esta explicação da profundidade de corte seja para fresamento manual, pode ser usada para fresamento numa máquina CNC.

## APÊNDICE D. CRIAÇÃO DE PROGRAMAS STEP-NC PART 21 USANDO O STEP *MODELER*

### D.1. Exemplo a) Programa de peça em STEP-NC: Peça com uma *feature* (“simples”) tipo furo com base cônica gerado pelo *STEP Modeler*

Caraterísticas iniciais:

Tabela D.1: Informações da peça bruta

|                          |                   |
|--------------------------|-------------------|
| <b>Dimensões da peça</b> | <b>X</b> = 200 mm |
|                          | <b>Y</b> = 150 mm |
|                          | <b>Z</b> = 30 mm  |
| <b>Tolerância Global</b> | 50 $\mu$ m        |

Tabela D.2: Informações da *feature*

| <b>Tipo: Furo com base cônica</b> |                   |
|-----------------------------------|-------------------|
| <b>Posição do furo</b>            | <b>X</b> = 100 mm |
|                                   | <b>Y</b> = 75 mm  |
|                                   | <b>Z</b> = 0 mm   |
| <b>Profundidade</b>               | 10 mm             |
| <b>Ângulo da base</b>             | 20°               |
| <b>Diâmetro</b>                   | 10 mm             |
| <b>Offset Z</b>                   | 0 mm              |

Tabela D.3: Informações das ferramentas

| <b>Tipo: Center Drill</b>   |       |
|-----------------------------|-------|
| <b>Comprimento do corte</b> | 10 mm |
| <b>Diâmetro de corte</b>    | 6 mm  |
| <b>Comprimento total</b>    | 50 mm |
| <b>Diâmetro de haste</b>    | 12 mm |
| <b>Ângulo da base</b>       | 70°   |
| <b>Tipo: Twist Drill</b>    |       |
| <b>Comprimento do corte</b> | 40 mm |
| <b>Diâmetro de corte</b>    | 10 mm |
| <b>Comprimento total</b>    | 70 mm |
| <b>Diâmetro de haste</b>    | 20 mm |
| <b>Ângulo da base</b>       | 70°   |

Usando o sistema integrado CAD/CAPP/CAM da UFSC/GRIMA, cria-se o projeto da peça adicionando-se a ele a *feature* e as ferramentas, tendo como base as características das tabelas anteriores. Na Figura D.1, apresenta-se um conjunto de imagens do *software STEP Modeler* que contém as informações da peça bruta (a), a *feature* (b, c) e as ferramentas (d).

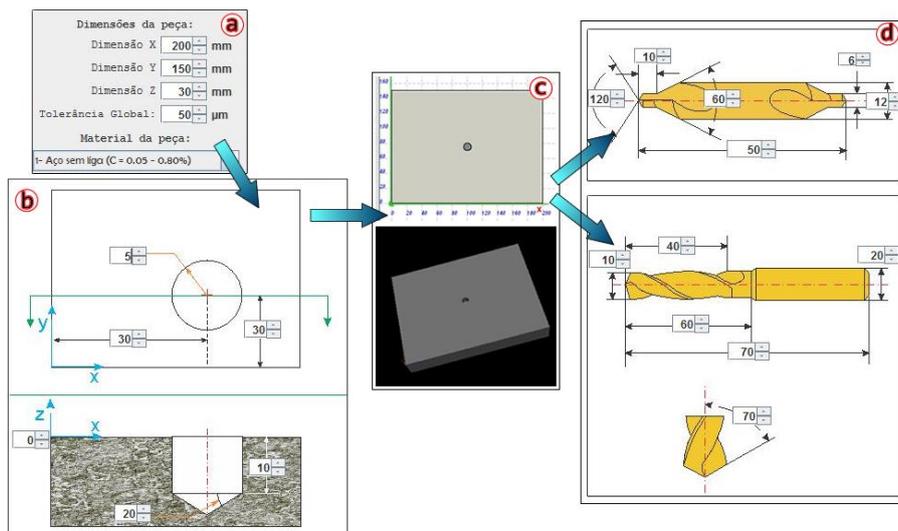


Figura D.1: Peça bruta, *feature* e ferramentas

**D.2. Exemplo b) Programa de peça em STEP-NC: Peça com uma feature tipo Ranhura com perfil em U gerada pelo STEP Modeler**

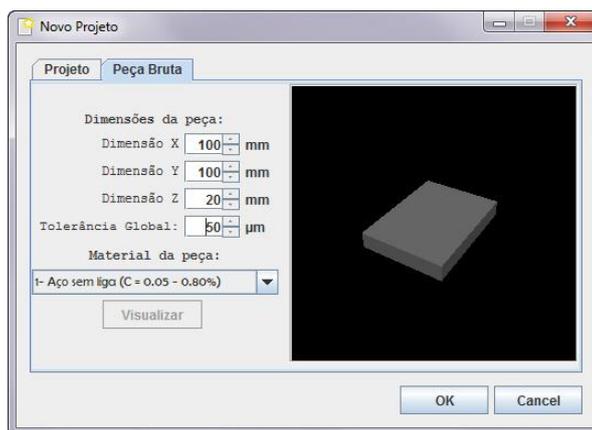


Figura D.2: Características da Peça bruta

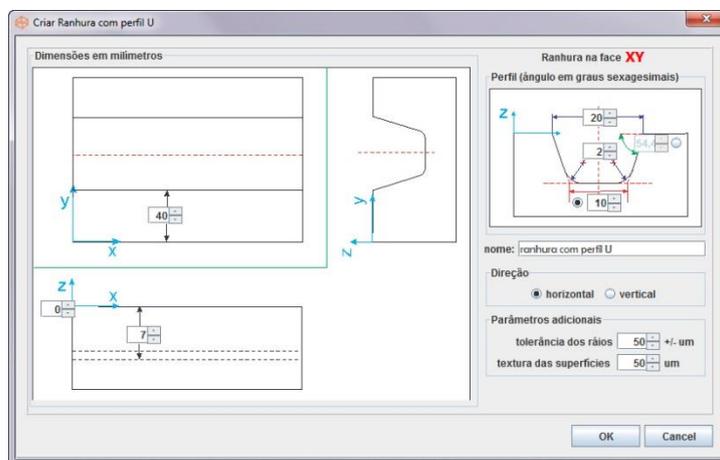


Figura D.3: Características da *feature*

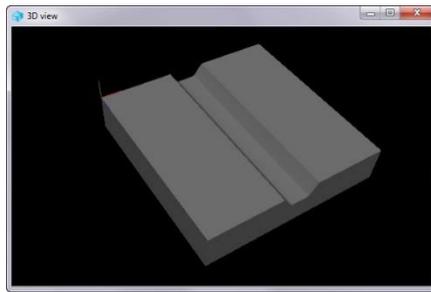


Figura D.4: Vista 3D da *feature*

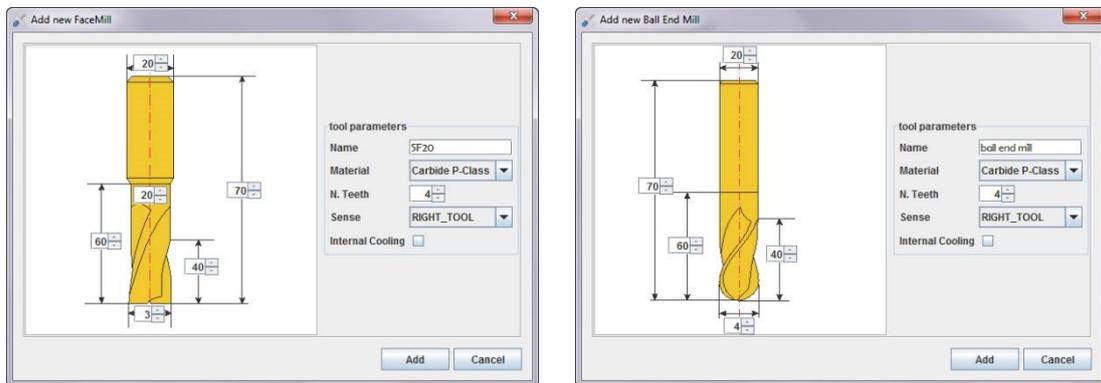


Figura D.5: Características das ferramentas

## APÊNDICE E. IMPLEMENTAÇÃO DO ALGORITMO DA GUI DE VISUALIZAÇÃO USANDO A LINGUAGEM PYTHON

```
#!/usr/bin/env python

try:
    import pygtk
    pygtk.require('2.0')
except:
    pass
try:
    import gtk
    import gtk.glade
except:
    print('GTK not available')
    sys.exit(1)

class texto:

    def __init__(self):
        self.gladefile = "/home/mod/GUI/GUI_STEP-NC_Viewer.ui"
        self.builder = gtk.Builder()
        self.builder.add_from_file(self.gladefile)
        self.window1 = self.builder.get_object("window1")
        self.window2 = self.builder.get_object("window2")
        self.textview1 = self.builder.get_object("textview1")
        self.button1 = self.builder.get_object("button1")
        self.button2 = self.builder.get_object("button2")
        self.button3 = self.builder.get_object("button3")
        self.builder.connect_signals(self)
        self.buffertexto = gtk.TextBuffer()
        self.lista = []
        self.archivo_name = "/home/mod/GUI/Code_original.txt"
        self.lista = open(self.archivo_name, "r").readlines()
        for i in range(len(self.lista)):
            self.buffertexto.insert_at_cursor(self.lista[i])
        self.textview1.set_buffer(self.buffertexto)
        self.window1.show()
        self.window2.hide()
        self.button2.hide()

    def on_button1_clicked(self, widget):
        self.archivo_name = "/home/mod/GUI/Code_original.txt"//pasta temporal modificar
        self.lista = open(self.archivo_name, "r").readlines()
        for i in range(len(self.lista)):
            self.buffertexto.insert_at_cursor(self.lista[i])
        self.textview1.set_buffer(self.buffertexto)
        self.button1.hide()
        self.button2.show()

    def on_button2_clicked(self, widget):
        self.window2.show()
        self.window1.hide()

    def on_button3_clicked(self, widget):
        self.window1.show()
        self.window2.hide()

    def on_window1_destroy(self, object, data=None):
        #print "quit with cancel"
        gtk.main_quit()

    def on_window2_destroy(self, object, data=None):
        #print "quit with cancel"
        gtk.main_quit()

if __name__ == "__main__":
    main = texto()
    gtk.main()
```

## APÊNDICE F. ESTUDO DE CAPABILIDADE DA ROUTER CNC

Neste apêndice é apresentado o estudo da capacidade da Router CNC que permitirá conhecer o desempenho da máquina (com relação aos eixos X e Y) em relação a retilidade, paralelismo, perpendicularidade, circularidade, posicionamento por meio da usinagem de uma peça teste. Apenas um estudo de posicionamento do eixo Z será feito e apresentado no final deste apêndice.

Para a realização das medições, utilizou-se como medidor eletrônico uma máquina de medição por coordenadas marca *Mitutoyo* (ver modelo e certificado de calibração no Anexo 8). Essa máquina possui uma resolução de 0,0001 mm, uma incerteza de 0.003 mm e um nível de confiança de aproximadamente 95%. Para usinar a peça teste (na Router com o controlador LinuxCNC), é usado um código G (ver Figura F.2) feito manualmente de acordo com as geometrias da Figura F.1.

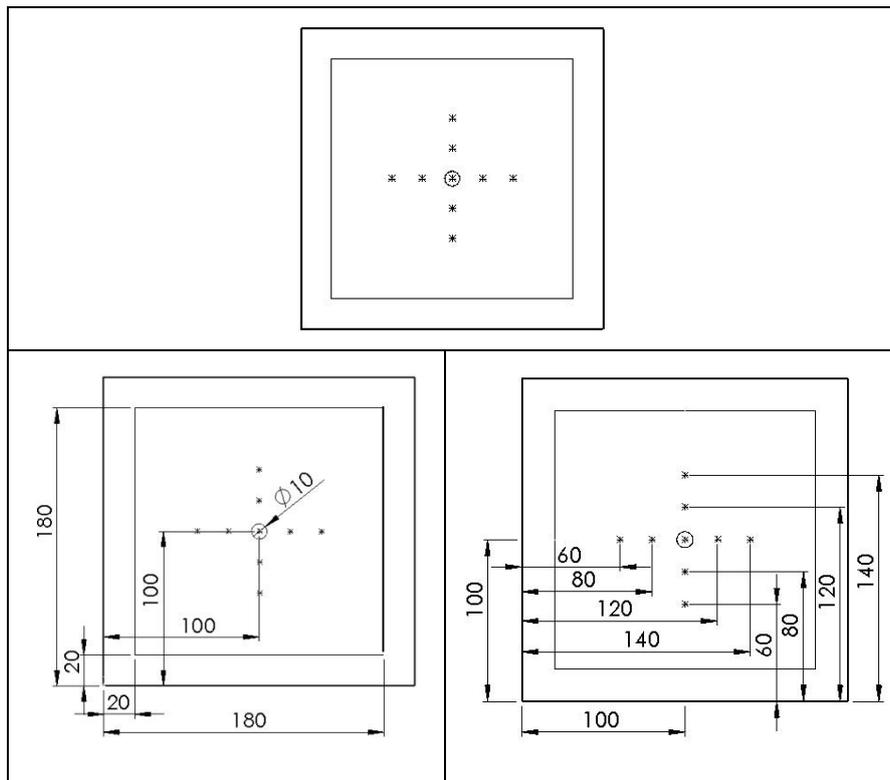


Figura F.1: Esboço de referência da peça teste

|                 |              |           |              |
|-----------------|--------------|-----------|--------------|
| G21 G17 G54 G90 | X60 Y100     | Z10       | Z-4          |
| M06 T1          | Z-4          | X140 Y100 | (FURO 9)     |
| M03             | (FURO 2)     | Z-4       | Z10          |
| G01 F400 X0 Y0  | Z10          | (FURO 6)  | X100 Y140    |
| Z10             | X80 Y100     | Z10       | Z-2          |
| (QUADRADO)      | Z-4 (FURO 3) | X100 Y60  | Z10          |
| X20 Y20         | Z10          | Z-4       | X110 Y100    |
| Z-4             | X100 Y100    | (FURO 7)  | Z-4          |
| X180            | Z-4          | Z10       | (CIRCULO)    |
| Y180            | (FURO 4)     | X100 Y80  | G03 X90 R10  |
| X20             | Z10          | Z-4       | G03 X110 R10 |
| Y20             | X120 Y100    | (FURO 8)  | G01 Z10      |
| (FURO 1)        | Z-4          | Z10       | G00 X0 Y0    |
| Z10             | (FURO 5)     | X100 Y120 | M30          |

Figura F.2: Código G da peça teste.

Na Figura F.3 , é apresentada a simulação do código G da peça teste no *software Vericut*. Para a simulação da usinagem, usou-se uma ferramenta tipo *Facemill* de  $\varnothing 6$  mm.

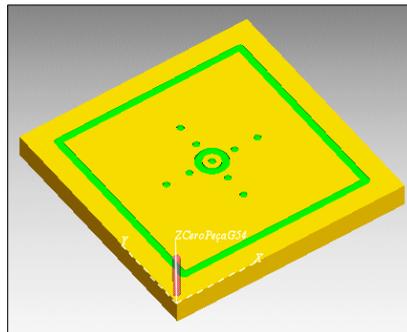


Figura F.3: Resultado da simulação de usinagem da peça teste no *software Vericut*

Na Figura F.4, são mostradas as trajetórias de ferramenta do código G da peça teste representadas graficamente na GUI do controlador LinuxCNC.

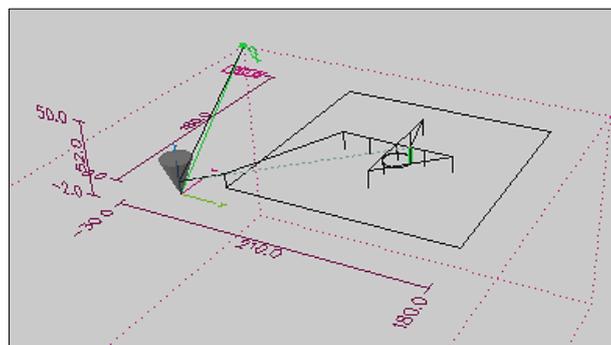


Figura F.4: Trajetórias da ferramenta de corte (linhas pretas) mostradas pelo LinuxCNC de acordo com o código G da peça teste

Depois de observar a simulação e comprovar que não há erros no código G, usinou-se a peça teste usando a Router CNC desenvolvida neste trabalho, utilizando como ferramenta de corte uma “fresa de  $\varnothing 6$  mm”. O resultado obtido é mostrado na Figura F.5.

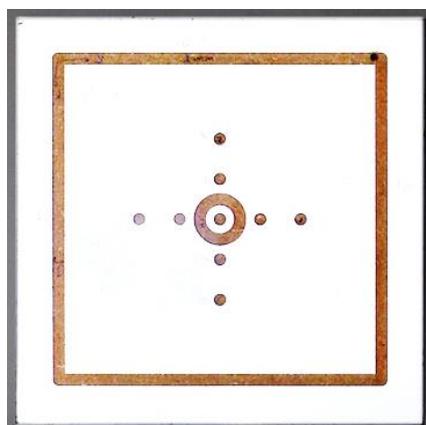


Figura F.5: Resultado da peça teste usinada na Router CNC

Note-se que o código G exposto na Figura F.2 não possui compensação de raio de ferramenta. Idealmente usa-se uma ferramenta com um diâmetro muito pequeno mas o resultado da usinagem mostrado na Figura F.5 exibe alterações das medições (ver esboço da Figura F.1) criadas pelo uso de uma ferramenta com diâmetro maior, neste caso  $\varnothing 6$  mm. Na Figura F.6, são apresentadas as novas características dimensionais. É esperada uma tolerância de  $\pm 0,05$  (mm ou graus) tanto para tolerância geométrica quanto para tolerância dimensional.

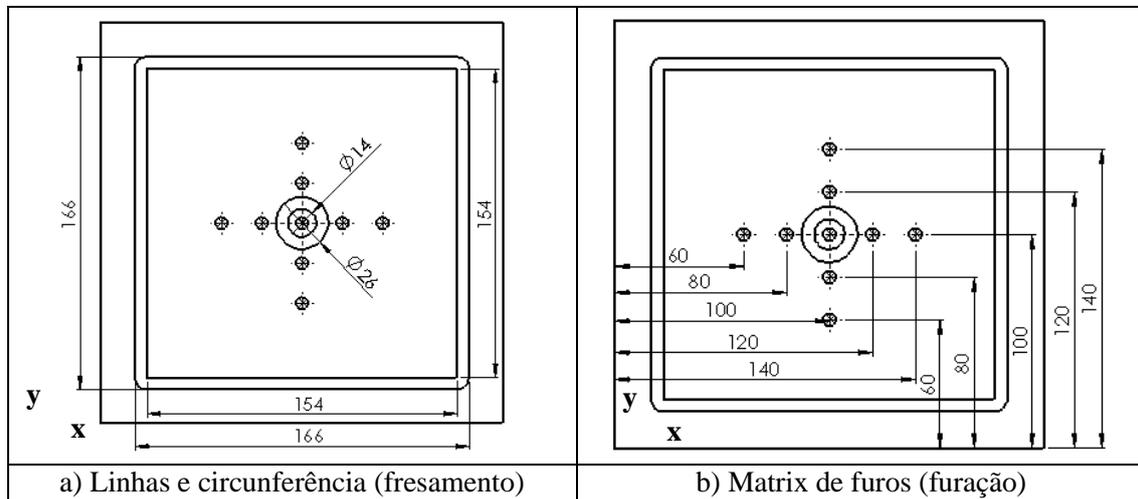


Figura F.6: Novo esboço de referência da peça teste

Já comparando as figuras F.1 e F.6 (a e b), pode-se entender a necessidade de criar um novo esboço da peça teste com as dimensões ideais de acordo com o resultado da usinagem.

Usinando a peça teste, é possível identificar experimentalmente os parâmetros de corte mais adequados para a Router CNC (ver apêndice C.7). Esses parâmetros são apresentados na Tabela F.1.

Tabela F.1: Parâmetros de corte da Router CNC encontrados experimentalmente

| Parâmetro                  | Operação   |                                   |
|----------------------------|------------|-----------------------------------|
|                            | Fresamento | Furação                           |
| Velocidade de Corte máx.   | 400 mm/min | 400 mm/min                        |
| Velocidade de avanço máx.  | 900 mm/min | 900 mm/min                        |
| Profundidade de corte máx. | 2 mm       | Depende do diâmetro da ferramenta |

Tendo como base a peça teste usinada, é possível determinar a capacidade da Router CNC de forma indireta estudando a retilidade, paralelismo, perpendicularidade, circularidade, posicionamento de seus eixos XY por meio de toma de amostras (medições). Deve-se frisar que apenas é feito o estudo de posicionamento o eixo Z.

Como já foi mencionado, para a obtenção de amostras, utiliza-se como medidor eletrônico uma máquina de medição por coordenadas (CMM). Essa máquina permite obter automaticamente os valores da retilidade, do paralelismo, da perpendicularidade, da circularidade e posicionamento. As medições foram feitas pelo pessoal técnico do laboratório de metrologia no setor de serviços gerais (SG9) da UNB. O relatório dos resultados das medições (em uma planilha EXCEL) está

disponível no site web do projeto <https://code.google.com/p/adaptador-de-stepnc-to-gcode-on-linuxcnc/>. A apresentação dos resultados é feita usando simbologia de tolerâncias.

## F.1. PROCEDIMENTOS PARA OBTENÇÃO DE AMOSTRAS E APRESENTAÇÃO DE RESULTADOS

Para poder identificar o erro da retilidade, do paralelismo, da perpendicularidade, da circularidade e de posicionamento é necessário tomar algumas amostras (medições) das características geométricas da peça teste usinada. São mostrados tres procedimentos usados para a obtenção das amostras na peça teste usinada, utilizando o CMM. Depois de cada procedimento, são apresentados os resultados obtidos.

### F.1.1. Procedimento para obtenção de amostras para identificar o valor do erro de retilidade, paralelismo e perpendicularidade

Usando a CMM, é possível determinar a retilidade de uma reta, a perpendicularidade e o paralelismo entre duas retas. Neste caso há varias retas presentes na peça teste usinada mas apenas serão usadas as retas que formam o quadrado da parte interna como mostrado na Figura F.7. Nessa figura, há vários pontos em cada linha do quadrado. Os pontos estão separados a cada 10 mm. Cada linha tem um comprimento de 154mm por tanto há um total de 16 pontos. Esses pontos são os locais onde são tomadas as amostras.

A CMM possui um apalpador (ver Figura F.8) que contém uma ponta esférica de 2mm. Essa ponta é colocada a uma profundidade de aproximadamente 2,5mm com relação a face superficial da peça, i.e. a ponta é colocada dentro da ranhura sem tocar o fundo ou as bordas (paredes internas). Depois manualmente aproxima-se a ponta até fazer contato com a face interna da ranhura de acordo com cada ponto. Na Figura F.9, esse procedimento é ilustrado.

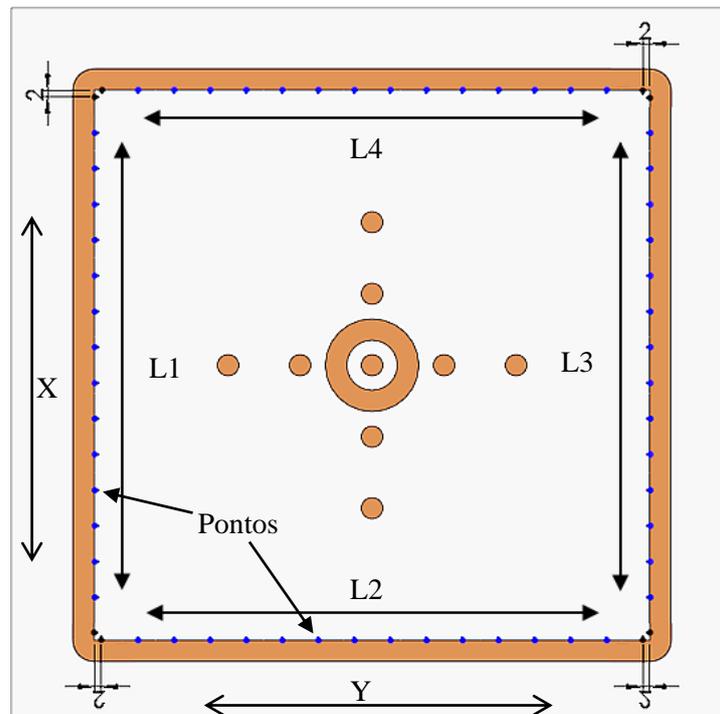


Figura F.7: Pontos de referência para medição da peça teste

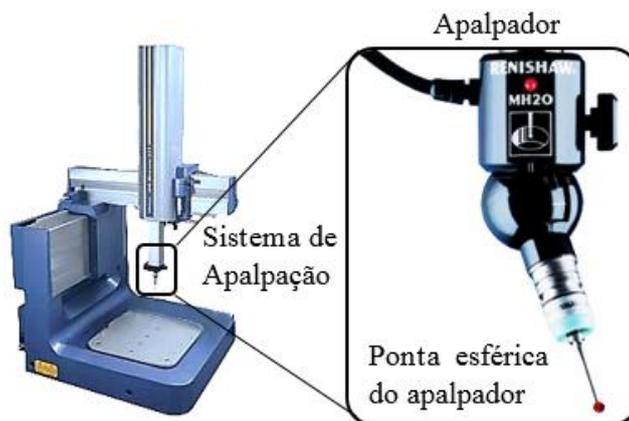


Figura F.8: CMM QM-353 e sistema de palpação

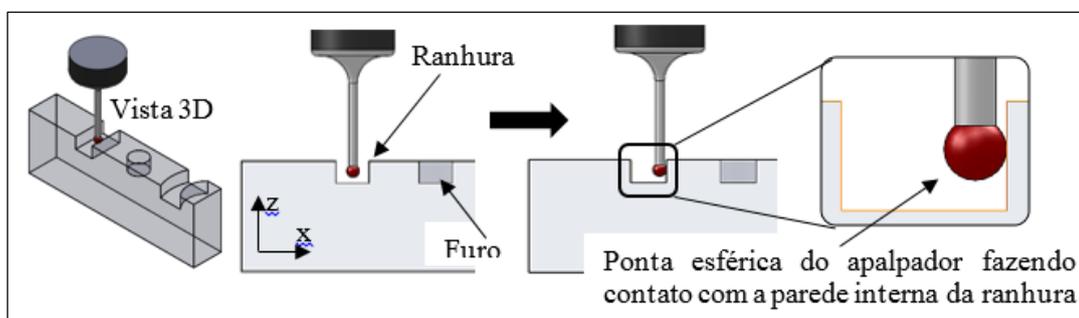


Figura F.9: Procedimento para medição em um ponto

Para se ter uma ideia mais clara do procedimento, pode-se assistir ao vídeo “Procedimento da toma de amostras com CMM (QM-353) peça teste 01” no canal <http://www.youtube.com/user/meduag>.

Note-se que existem diferentes fontes de erro (e.g. erro direcional da ferramenta de corte) que podem influenciar na usinagem e refletir-se na medição. No entanto, é levada em conta apenas a incerteza de medição relacionada com a CMM. Um estudo mais detalhado da capacidade da Router CNC poderia ser um projeto para trabalhos futuros.

### F.1.2. Resultados dos erros de forma macrogeométricos da peça teste relacionados com retilidade, paralelismo e perpendicularidade

Os resultados dos erros de forma macrogeométricos da peça teste obtidos por meio da CMM são apresentados nas seguintes tabelas:

Tabela F.2: Resultados do ensaio de retilidade

| Linha/Eixo | Amplitude        | Tolerância           | Incerteza de medição |
|------------|------------------|----------------------|----------------------|
| L1/X       | 19 $\mu\text{m}$ | $\pm 50 \mu\text{m}$ | 3,0 $\mu\text{m}$    |
| L2/Y       | 59 $\mu\text{m}$ |                      |                      |
| L3/X       | 57 $\mu\text{m}$ |                      |                      |
| L4/Y       | 63 $\mu\text{m}$ |                      |                      |

Tabela F.3: Resultados do ensaio de paralelismo (180°)

| Referência | Entre Linha# e Linha# | Amplitude | Tolerância | Incerteza de medição |
|------------|-----------------------|-----------|------------|----------------------|
| 180°       | L1/L3                 | 0,019°    | ±0,05°     | 3,0 μm               |
|            | L2/L4                 | 0,134°    |            |                      |

Tabela F.4: Resultados do ensaio de perpendicularismo (90°)

| Referência | Entre Linha# e Linha# | Amplitude | Tolerância | Incerteza de medição |
|------------|-----------------------|-----------|------------|----------------------|
| 90°        | L1/L2                 | 0,093°    | ±0,05°     | 3,0 μm               |
|            | L2/L3                 | 0,093°    |            |                      |
|            | L3/L4                 | 0,136°    |            |                      |
|            | L4/L1                 | 0,137°    |            |                      |

- **Retilidade:** a seguir são apresentados graficamente os resultados da Tabela F.2 com relação a retilidade.

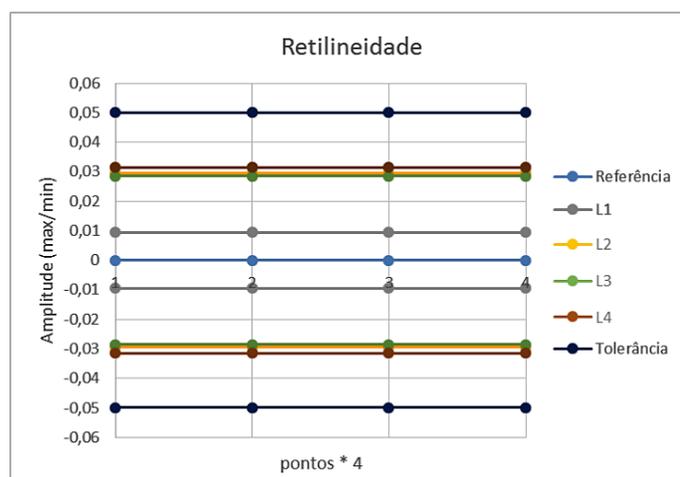


Figura F.10: Apresentação gráfica dos resultados do ensaio de retilidade

Como pode-se observar na figura anterior, a amplitude de cada linha está dividida por dois para encontrar o valor máximo e mínimo do erro de retilidade, e.g. o valor da amplitude de L1 segundo a Tabela F.2 é 19μm, portanto o valor máximo é 9,5μm e o valor mínimo é -9,5μm (este método de divisão também é usado nas próximas gráficas). Pode-se concluir então que as linhas estão dentro do valor de tolerância esperado. Outra observação que pode ser feita de acordo com a Figura F.7 é que as linhas L1 e L3 representam o erro de retilidade do eixo X e as linhas L2 e L4 representam o erro de retilidade do eixo Y.

- **Paralelismo:** Os dados apresentados na Tabela F.3 mostram os resultados dos ensaios de paralelismo na peça teste usinada. O procedimento para a obtenção desses dados (resultados das medições) é feito medindo-se o ângulo formado entre duas retas paralelas. Esse ângulo tem um valor teórico de 180°. Seguindo as referências mostradas na Figura F.7, pode-se calcular o ângulo formado entre L1/L3 e L2/L4 respectivamente. A Figura F.11 apresenta graficamente os resultados da Tabela F.3.

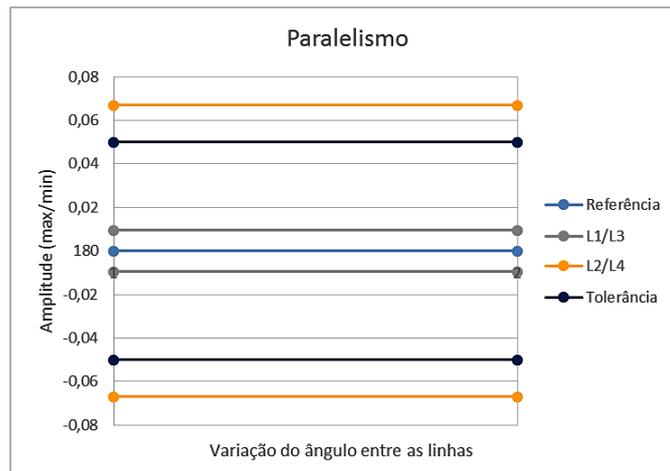


Figura F.11: Apresentação gráfica dos resultados do ensaio de paralelismo

Como explicado no ensaio de retilidade, as linhas L1 e L3 representam o erro de paralelismo do eixo X e as linhas L2 e L4 representam o erro de paralelismo do eixo Y. Pode-se observar na Figura F.11 que o erro de paralelismo entre as linhas L1 e L3 relacionadas com o eixo X está dentro da tolerância esperada mas o paralelismo entre as linhas L2 e L4 relacionadas com o eixo Y encontra-se mais longe do esperado, no entanto o seu valor do erro é  $\pm 0.067^\circ$  e ainda fica na casa das centésimos de grau. Pode-se concluir então que primeiramente deve ser feita uma inspeção mecânica do eixo Y para melhorar e corrigir o paralelismo nesse eixo, depois procurar outras fontes de erro e tentar corrigi-las também.

- Perpendicularidade:** Os dados apresentados na Tabela F.3 mostram os resultados dos ensaios de perpendicularidade na peça teste usinada. O procedimento para a obtenção desses dados (resultados das medições) é feito medindo-se o ângulo formado entre duas retas perpendiculares. Esse ângulo tem um valor teórico de  $90^\circ$ . Seguindo as referências mostradas na Figura F.7, pode-se calcular o ângulo formado entre L1/L2, L2/L3, L3/L4 e L4/L1 respectivamente. A Figura F.12 apresenta graficamente os resultados da Tabela F.4.

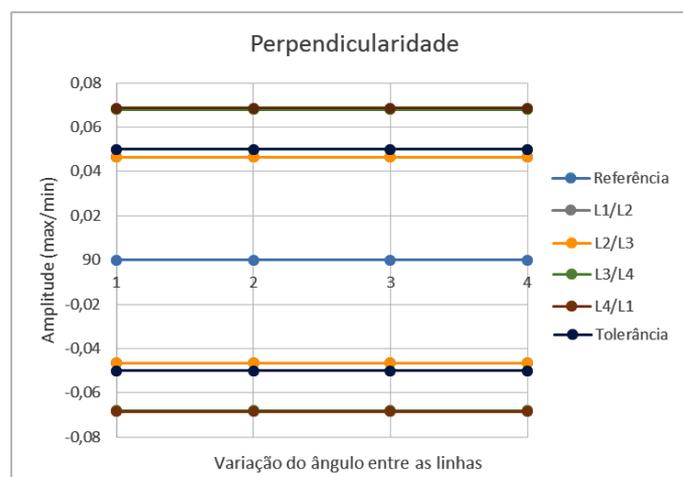


Figura F.12: Apresentação gráfica dos resultados do ensaio de perpendicularismo

Os dados apresentados na figura anterior mostram que o perpendicularismo entre as linhas L1/L2 e L2/L3 está dentro da tolerância esperada (com valores iguais), mas o perpendicularismo entre as

linhas L3/L4 e L4/L1 (com valores muito próximos) é maior. Isso permite identificar junto ao erro da retilidade dos eixos X e Y que a máquina possui um erro geométrico maior quando se afasta do início das coordenadas (*Home* ou *Zero\_máquina*) da Router.

### F.1.3. Procedimento de toma de amostras para identificar o valor do erro de circularidade

Usando a CMM, é possível determinar a circularidade de uma circunferência. Neste caso, há duas delas presentes na peça teste usinada. Essas circunferências estão no centro da peça (ver Figura F.7). A CMM precisa de três ou mais pontos para recriar uma circunferência. Na Figura F.13, há um fragmento da Figura F.7, nela há seis pontos ao redor da circunferência externa. Os pontos estão separados a cada 60°. Esses pontos são os locais de onde são tomadas as amostras usando o procedimento ilustrado na Figura F.9.

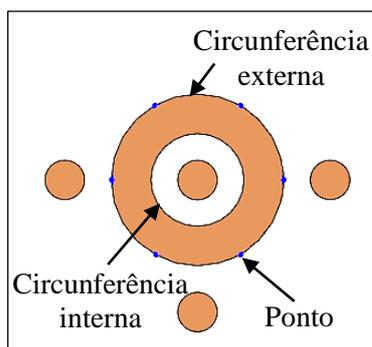


Figura F.13: Fragmento da peça teste – Centro da peça

### F.1.4. Resultado do erro de forma macrogeométrico da peça teste relacionado com circularidade

Foi medida a circularidade da circunferência interna e externa de acordo com os pontos mostrados na Figura F.13. O procedimento de obtenção das amostras aplica para as duas circunferências.

Os resultados dos erros de forma macrogeométrico relacionado com a circularidade das circunferências (externa e interna) da peça teste são apresentados na Tabela F.5.

Tabela F.5: Resultados do ensaio de circularidade

| Referencia | Circunferência | Amplitude | Tolerância | Incerteza de medição |
|------------|----------------|-----------|------------|----------------------|
| Ø26        | Externa        | 34 µm     | ±50 µm     | 3,0 µm               |
| Ø14        | Interna        | 31 µm     |            |                      |

Na Figura F.14, pode-se observar que os valores máximo e mínimo da circularidade das duas circunferências estão dentro da tolerância esperada. O valor da referência da circunferência externa é pequeno relacionado com o comprimento das linhas. Portanto, há um erro pequeno que possivelmente não mostre a verdadeira amplitude do erro da circularidade. Ou seja, esse erro de circularidade para este estudo básico da capacidade da Router CNC é aceitável.

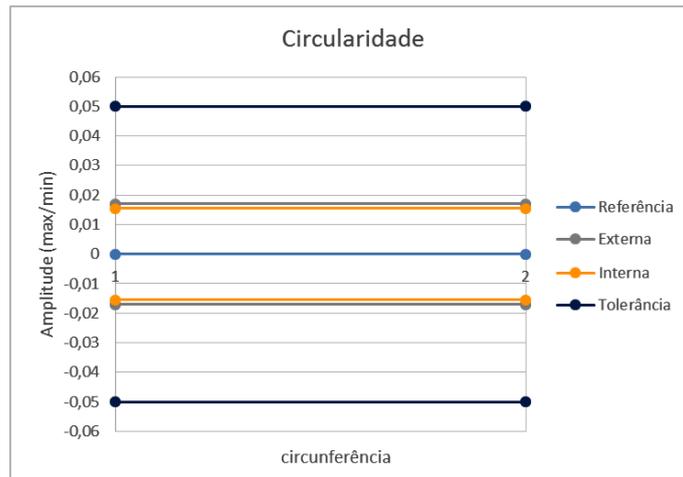


Figura F.14: Apresentação gráfica dos resultados do ensaio de circularidade

### F.1.5. Procedimento de toma de amostras para identificar o valor do erro de posicionamento dos eixos X, Y e Z

Usando a CMM, é possível determinar o erro de posicionamento dos eixos X, Y e Z. O erro de posicionamento dos eixos X e Y é achado usando a matriz de furos mostrada na Figura F.15a. Cada furo é relacionado a um ponto. Cada ponto identifica o centro do furo. Para identificar o centro de cada furo, toma-se três amostras (de acordo com os pontos de medição) ao redor da circunferência do furo. A Figura F.15b ilustra esta ideia. Finalmente, mede-se a distância entre um furo e outro, e.g. entre F1/F2, F2/F3 e assim sucessivamente. A distância teórica entre dois furos segundo o esboço da Figura F.1 é de 20mm com uma tolerância de  $\pm 0,05\text{mm}$ . Para o eixo X são medidos os furos em horizontal e para o eixo Y são medidos em vertical de acordo com a matriz da Figura F.15a.

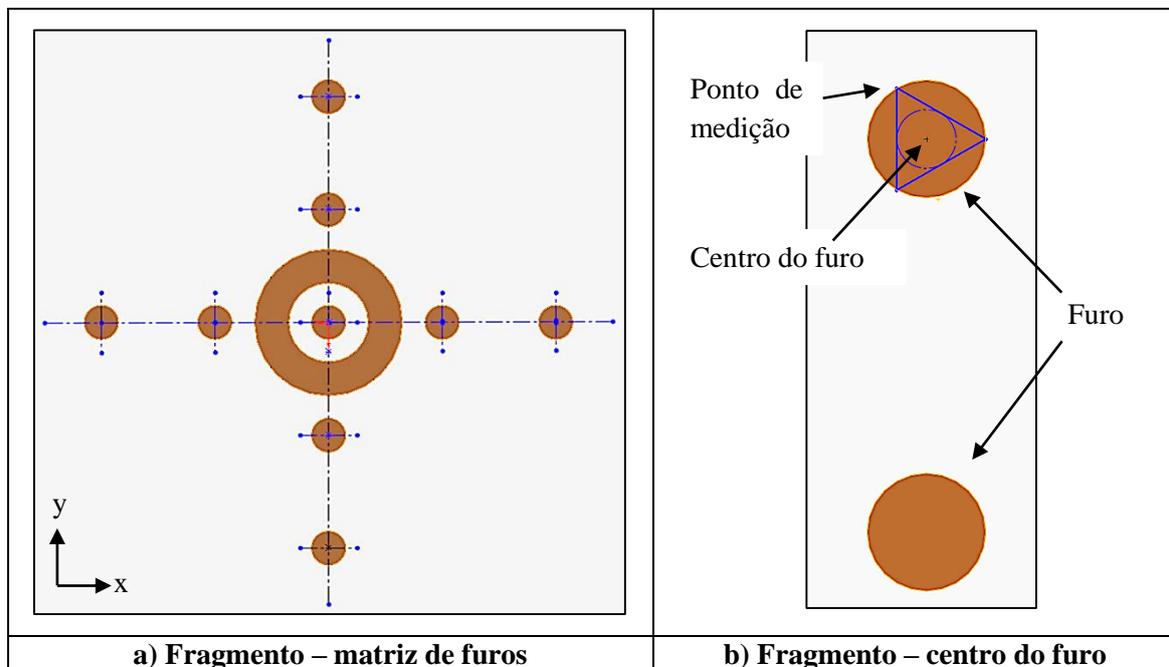


Figura F.15: Fragmento da peça teste – Centro da peça teste

Já o procedimento para determinar o erro de posicionamento do eixo Z é diferente dos eixos X e Y. Neste caso, a medição é levada a cabo por meio da criação de dois planos de referência. O primeiro plano faz referência à face superficial da peça. O segundo plano faz referência à face da base (profundidade) das ranhuras e furos. Usando a CMM são criados esses planos e medida a distância entre eles. O valor de referência é de 4mm com uma tolerância de  $\pm 0,05\text{mm}$ . A Figura F.16 ilustra os planos de referência na peça teste usinada.

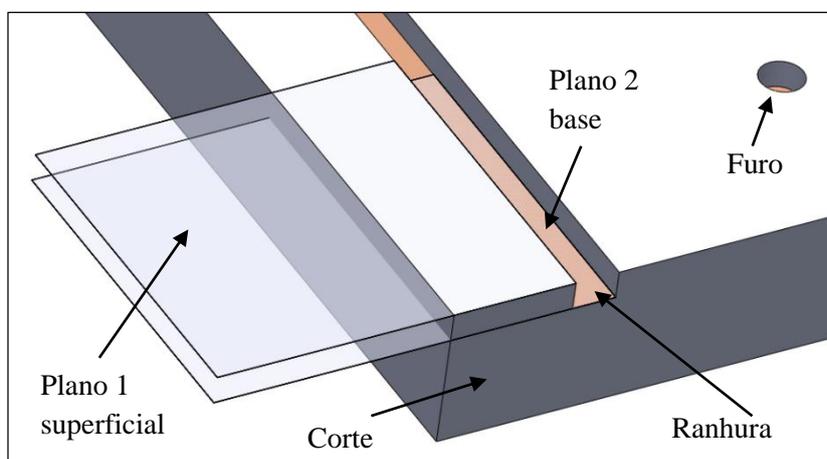


Figura F.16: Fragmento da peça teste – planos de referencia

#### F.1.6. Resultado do erro de posicionamento da peça teste relacionado com os eixos X, Y e Z da Router CNC

Os resultados dos erros de posicionamento dos eixos X, Y e Z são apresentados nas Tabela F.6.

Tabela F.6: Resultados do ensaio posicionamento eixos X, Y e Z

| Referencia | Eixo | Media              | Tolerância           | Incerteza de medição |
|------------|------|--------------------|----------------------|----------------------|
| 20mm       | X    | 16,1 $\mu\text{m}$ | $\pm 50 \mu\text{m}$ | 3,0 $\mu\text{m}$    |
|            | Y    | -21 $\mu\text{m}$  |                      |                      |
| 4mm        | Z    | 189 $\mu\text{m}$  |                      |                      |

Segundo os resultados apresentados na tabela anterior, o valor da média do erro de posicionamento dos eixos X e Y encontram-se dentro da tolerância esperada, mas o valor da média do erro de posicionamento do eixo Z a ultrapassou. Isso pode ter sido causado pela superfície rugosa da peça teste. Deve-se frisar, no entanto, que é necessário um estudo mais detalhado da capacidade da Router CNC.

Apesar das limitações da Router CNC por ter sido construída de forma manual, os resultados apresentados pelo estudo básico e indireto da capacidade da máquina são satisfatórios. A maioria dos erros de forma macrogeométricos e de posicionamento estão perto da tolerância esperada. Algumas correções poderiam ser realizadas na mecânica da máquina, principalmente no que se refere ao erro de posicionamento apresentado pelo eixo Z.

## APÊNDICE G. ARQUIVO DE CONFIGURAÇÃO USADO PELO ADAPTADOR DE STEP-NC A CÓDIGO G.

Neste apêndice, é apresentado o formato que deve ter o arquivo de configuração usado pelo adaptador de STEP-NC a código G.

Os dados devem ser formatados da seguinte maneira:

**“Repositorio=”** Rota do repositório usado pelo adaptador.

**“Code\_p21=”** Rota onde irá ser guardado temporalmente uma cópia do arquivo físico p21 usado pelo adaptador.

**“ComentariosInf=”** habilitar comentários sobre obtenção de informações no código G gerado pelo adaptador. Opções *true/false*.

**“ComentariosCod=”** habilitar comentários sobre informações das *features* de usinagem, ferramenta e condições de usinagem no código G gerado pelo adaptador. Opções *true/false*.

**“Pla\_Seg=”** é o valor do plano de segurança em caso que o valor fornecido pelo arquivo físico p21 seja muito grande, nesse caso o valor máximo é 55mm e corresponde ao eixo Z.

**“VelAvaMaq=”** Velocidade de máxima de avance da Router CNC. Valor predefinido em 900 mm/min. Se precisar mudar tem de ser menor que esse valor.

**“VelAvaCorte=”** Velocidade de máxima de avance de corte da Router CNC. Valor predefinido em 400 mm/min. Se precisar mudar, deve ser menor que esse valor.

**“AvaProfCorte=”** Avanço da profundidade de corte da Router CNC. Valor predefinido em 2 mm para operação de fresamento. Se precisar mudar, deve ser menor que esse valor.

**“AvaAngSemiCir=”** Avanço do ângulo da semicircunferência para segmentação da *feature* e, conseqüentemente, a usinagem por camadas. Valor predefinido em 10°. Se precisar mudar, deve ser menor que esse valor.

**“PorFerEspiral=”** porcentagem da ferramenta de corte para o avanço da usinagem de um furo com diâmetro maior que a ferramenta de corte usada. O furo é usinado em espiral. O passo de uma circunferência a outra depende deste valor predefinido em 50%. Se precisar mudar, deve ser maior que esse valor.

**“AvaLinhaProf=”** Avanço da profundidade de corte para segmentar uma *feature* em V com relação a sua inclinação versus profundidade. Valor predefinido em 1mm. Se precisar mudar, deve estar entre [0.5 - 2]mm.

**“tooltbl=”** Rota da pasta de configuração da Router CNC.

O arquivo de configuração deve ser nomeado **“arqConf.txt”** e ser salvo na pasta de configuração da Router CNC. A seguir é apresentado o arquivo gerado para a Router CNC desenvolvida neste trabalho.

| arqConf.txt  |
|--|
| Repositorio=/home/fresa/Temp/<br>Code_p21=/home/fresa/Temp/<br>ComentariosInf=false<br>ComentariosCod=false<br>Pla_Seg=55<br>VelAvaMaq=900<br>VelAvaCorte=400<br>AvaProfCorte=1<br>AvaAngSemiCir=10<br>PorFerEspiral=50<br>AvaLinhaProf=1<br>tooltbl=/home/fresa/FresaTG2/ |

Figura G.1: Exemplo do arquivo de configuração usado pelo adaptador

## ANEXO 1. LISTA DE CÓDIGOS G/M PARA FRESAMENTO

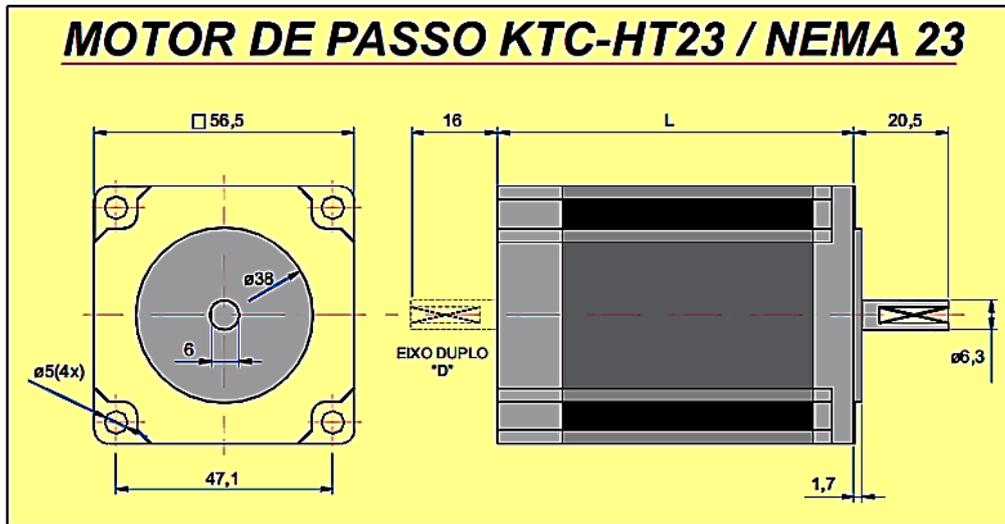
| <b>Código G</b> | <b>Significado</b>   |
|-----------------|--|
| G0              | Posicionamento rápido                                      |
| G1              | Interpolação linear  |
| G2              | Circular / helicoidal interpolação (no sentido horário)    |
| G3              | Circular / helicoidal interpolação (anti-horário)          |
| G4              | Dwell  |
| G10             | Definição origem do sistema de coordenadas                 |
| G17             | Seleção do plano XY  |
| G18             | Seleção do plano XZ  |
| G19             | Seleção do plano YZ  |
| G20             | Seleção do sistema de entrada em polegada                  |
| G21             | Seleção do sistema de entrada em milímetros                |
| G28             | Retorno para home  |
| G30             | Retorno à casa secundária                                  |
| G38.2           | Sensor em linha reta                                       |
| G40             | Cancelar a compensação do raio de corte                    |
| G41             | Ativar compensação de raio de corte à esquerda             |
| G42             | Ativar compensação do raio de corte à direita              |
| G43             | Compensar comprimento da ferramenta (Plus)                 |
| G49             | Cancelar a compensação do comprimento da ferramenta        |
| G53             | Movimento no sistema de coordenadas                        |
| G54             | Uso do sistema de coordenadas de trabalho programada 1     |
| G55             | Uso do sistema de coordenadas de trabalho programada 2     |
| G56             | Uso do sistema de coordenadas de trabalho programada 3     |
| G57             | Uso do sistema de coordenadas de trabalho programada 4     |
| G58             | Uso do sistema de coordenadas de trabalho programada 5     |
| G59             | Uso do sistema de coordenadas de trabalho programada 6     |
| G59.1           | Uso do sistema de coordenadas de trabalho programada 7     |
| G59.2           | Uso do sistema de coordenadas de trabalho programada 8     |
| G59.3           | Uso do sistema de coordenadas de trabalho programada 9     |
| G61             | Modo de controle de trajetória: trajetória exata           |
| G61.1           | Modo de controle de trajetória: parada exata               |
| G64             | Modo de controle de trajetória: contínuo                   |
| G80             | Cancelar modo de movimento (incluindo qualquer ciclo fixo) |
| G90             | Modo de distância absoluto                                 |
| G91             | Modo de distância incremental                              |
| G93             | Modo inverso de tempo do feed rate                         |
| G94             | Modo de unidades por minuto para feed rate                 |

| <b>Código M</b> | <b>Significado</b>                     |
|-----------------|--|
| M0              | Parada do programa                     |
| M1              | Parada do programa opcional            |
| M2              | Fim do programa                        |
| M3              | Ativar Spindle no sentido horário      |
| M4              | Ativar Spindle no sentido anti-horário |
| M5              | Para giro do Spindle                   |
| M6              | Troca de ferramenta                    |
| M7              | Ativar refrigeração em neblina         |
| M8              | Ativar refrigeração a jato             |
| M9              | Cancelar refrigeração                  |
| M30             | Fim do programa e reset do programa    |

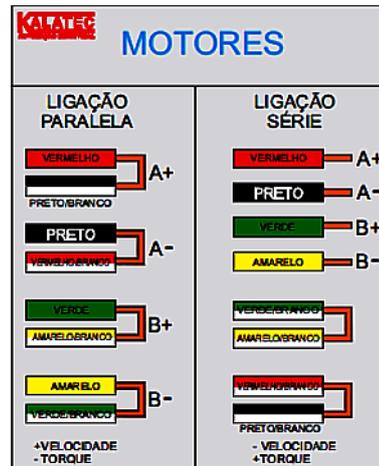
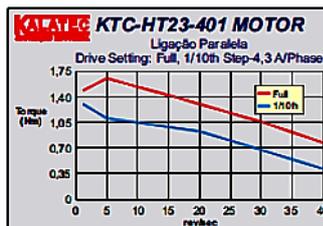
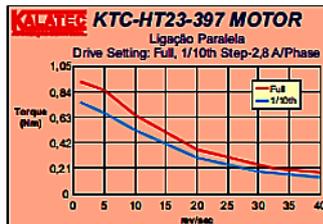
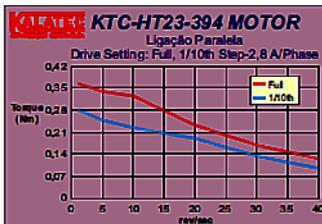
## ANEXO 2. CARACTERÍSTICAS GERAIS DO MOTOR DE PASSO KTC-HT23-400, KALATEC®



www.kalatec.com.br



| MODELO       | LIGAÇÃO DO MOTOR<br>1=SÉRIE<br>2=PARALELO<br>3=UNIPOLAR | COMPRIMENTO "L"<br>(mm) | MINIMO TORQUE ESTÁTICO (N.m) | FIOS | STEP (ÂNGULO) | VOLTS | AMPS | OHMS | mH   | ROTOR INÉRCIA (g.cm <sup>2</sup> ) | MOTOR PESO (Kg) |
|--------------|---|-------------------------|------------------------------|------|---------------|-------|------|------|------|------------------------------------|-----------------|
| KTC-HT23-394 | 1   | 40,6                    | 0,54                         | 8    | 1,8           | 4,0   | 1,41 | 2,8  | 5,6  | 120                                | 0,45            |
|              | 2   |                         |                              |      |               | 2,0   | 2,38 | 0,7  | 1,4  |                                    |                 |
|              | 3   |                         | 2,8                          |      |               | 2,00  | 1,4  | 1,4  |      |                                    |                 |
| KTC-HT23-397 | 1   | 59                      | 1,25                         | 8    | 1,8           | 5,1   | 1,41 | 3,6  | 10,0 | 300                                | 0,69            |
|              | 2   |                         | 2,5                          |      |               | 2,38  | 0,9  | 2,5  |      |                                    |                 |
|              | 3   |                         | 3,6                          |      |               | 2,00  | 1,8  | 2,5  |      |                                    |                 |
| KTC-HT23-400 | 1   | 76                      | 1,86                         | 8    | 1,8           | 6,4   | 1,41 | 4,5  | 14,4 | 480                                | 1,00            |
|              | 2   |                         | 3,2                          |      |               | 2,38  | 1,1  | 3,6  |      |                                    |                 |
|              | 3   |                         | 4,5                          |      |               | 2,00  | 2,3  | 3,6  |      |                                    |                 |
| KTC-HT23-401 | 1   | 76                      | 1,86                         | 8    | 1,8           | 4,2   | 2,12 | 2,0  | 6,4  | 480                                | 1,00            |
|              | 2   |                         | 2,1                          |      |               | 4,24  | 0,5  | 1,6  |      |                                    |                 |
|              | 3   |                         | 3,0                          |      |               | 3,00  | 1,0  | 1,6  |      |                                    |                 |



BYmths

**ANEXO 3. CARACTERÍSTICAS GERAIS DA LIGA DE ALUMÍNIO ALZINTOK 75-T651**

| <b>e<br/>(mm)</b> |  | <b>Kg/u</b> | <b>T6</b> | <b>T651</b> |
|-------------------|---|-------------|-----------|-------------|
| 1,6               | 2000x1000   | 8,25        | ●         | ○           |
| 2,0               | 2000x1000   | 11,00       | ●         | ○           |
| 3,0               | 2000x1000   | 16,50       | ●         | ○           |
| 4,0               | 2000x1000   | 22,00       | ●         | ○           |
| 4,0               | 3020x1520   | 50,50       | ●         | ○           |
| 5,0               | 2000x1000   | 27,50       | ●         | ○           |
| 6,0               | 3000x1250   | 61,90       | ●         | ○           |
| 8,0               | 3020x1520   | 99,00       | ●         | ○           |
| 10,0              | 3020x1520   | 126,25      | ●         | ○           |
| 12,0              | 3020x1520   | 151,50      | ●         | ○           |
| 15,0              | 3020x1520   | 190,00      | ●         | ○           |
| 20,0              | 3020x1520   | 252,50      | ●         | ○           |
| 20,0              | 4000x2000   | 440,05      | ●         | ●           |
| 25,0              | 3020x1520   | 315,60      | ○         | ●           |
| 25,0              | 4000x2000   | 550,02      | ○         | ●           |
| 30,0              | 3020x1520   | 378,75      | ○         | ●           |
| 32,0              | 3020x1520   | 404,00      | ○         | ●           |
| 35,0              | 2020x1020   | 198,32      | ○         | ●           |
| 40,0              | 3020x1520   | 505,00      | ○         | ●           |
| 50,0              | 3020x1520   | 631,20      | ○         | ●           |
| 60,0              | 3020x1520   | 757,50      | ○         | ●           |
| 70,0              | 3000x2000   | 883,65      | ○         | ●           |
| 80,0              | 3020x1520   | 1.010,00    | ○         | ●           |
| 90,0              | 3020x1520   | 1.136,15    | ○         | ●           |
| 100,0             | 2020x1020   | 566,63      | ○         | ●           |
| 100,0             | 3020x1520   | 1.262,40    | ○         | ●           |
| 110,0             | 3020x1520   | 1.388,60    | ○         | ●           |
| 120,0             | 3020x1520   | 1.514,85    | ○         | ●           |
| 160,0             | 3020x1520   | 2.019,80    | ○         | ●           |
| 180,0             | 2500x1250   | 1.546,88    | ○         | ●           |
| 180,0             | 3020x1520   | 2.272,25    | ○         | ●           |
| 200,0             | 3020x1520   | 2.549,80    | ○         | ●           |

## ANEXO 4. SMC-B-PRO GUIA DE UTILIZAÇÃO

HobbyCNC Brasil

SMC-B-PRO GUIA DE UTILIZAÇÃO

### CONTROLADORA DE MOTOR DE PASSO BIPOLAR SMC-U-PRO

#### CARACTERÍSTICAS OPERACIONAIS

|                                  |  |
|----------------------------------|--|
| Tensão de Alimentação :          | 30 a 45 VCC ou 24 a 35 VAC                         |
| Entrada de Alimentação :         | Circuito Retificador Integrado                     |
| Tipo de Motores :                | Bipolar ( 4 ou 8 fios )                            |
| Tensão dos Motores :             | 2 a 24 Volts                                       |
| Corrente Por Fase :              | 1 a 5 Amperes Configurável                         |
| Potencia Máxima por Fase :       | 20 Watts ( corrente por fase * voltagem do motor ) |
| Motores Recomendados :           | 7,5 Kgf, 13 Kgf, 15 Kgf, 18 Kgf, 35 Kgf, 50 Kgf    |
| Gerenciamento de Corrente :      | PWM ( CHOPPER )                                    |
| Máxima Frequência de Operação :  | 20 Khz ( 20.000 hertz )                            |
| Frequência Útil de Operação :    | 12 Khz ( 12.000 hertz )                            |
| Redução de Corrente em Repouso : | Sim, ( 60% em 2 Segundos Inativo )                 |
| Resolução :                      | Passo Cheio e Meio Passo Configurável              |
| Forma de Configuração :          | Por Software ( Via porta serial )                  |
| Entradas de Sinais :             | Passo/Direção/Inibe ( step/dir/ena )               |
| Programas Compatíveis :          | TurboCNC, Kcam, Mach2, Mach3, EMC <sup>2</sup> ... |

#### DESCRIÇÃO

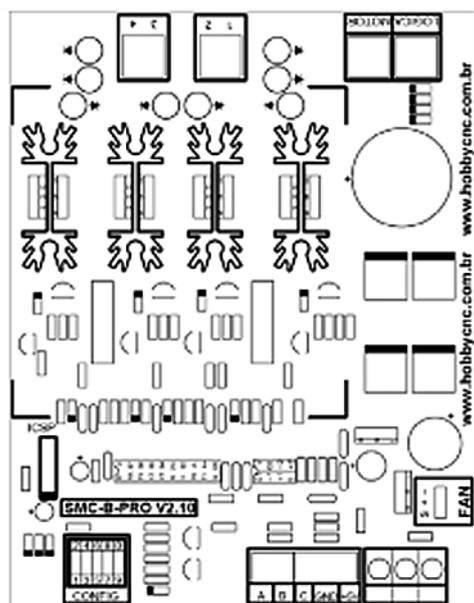
A interface SMC-B-PRO é um driver microcontrolado para motores de passo BIPOLARES, podendo trabalhar com correntes de até 5 amperes por fase limitando a potencia de 20 Watts por fase, o gerenciamento de corrente via PWM ( CHOPPER ) permite alimentação do motor com tensões maiores, o que permite alto desempenho com frequências até 20Khz modo livre e frequências superiores a 12Khz com carga no motor proporcionando velocidades acima de 2.000 RPMs ( Rotações por Minuto ), outro recurso muito importante é o sistema automático de detecção de inatividade, quando o motor fica parado por um período de aproximadamente 1 a 2 segundos a controladora reduz a corrente que esta fluindo para o motor, assim reduz o aquecimento do motor e da etapa de potência do motor sem nenhum comprometimento do desempenho do sistema. Esta controladora esta disponível em 2 modelos, o modelo CNC é aplicado em máquinas CNC, tem entrada de controle padrão STEP/DIR/ENA, a configuração de corrente e resolução de passos é manual via Software, nesta opção o firmware da interface trata os sinais de passo e direção vindos da porta paralela do computador ou CLPs ou outro gerador de pulsos qualquer, em sinais de acionamento das fases para o motor. É possível através do computador controlar velocidade, sentido de rotação e passo com grande precisão. Com aplicações em montagem de máquinas de CNC, automação de equipamentos, robôs cartesianos, manipuladores de objetos e máquinas de qualquer tipo com motores de passo. Compatível com grande quantidade de programas disponíveis na internet tais como TurboCNC, Kcam, Mach2, Mach3, EMC.

**A OPERAÇÃO COM CORRENTES ACIMA DE 3 AMPERE, É FORTEMENTE RECOMENDÁVEL A UTILIZAÇÃO DE RESFRIAMENTO FORÇADO ( MINI-VENTILADORES ) TANTO PARA AS PLACAS QUANTO PARA OS MOTORES.**

**CONTROLADORA DE MOTOR DE PASSO BIPOLAR SMC-B-PRO – MODO CNC**

Neste modo de operação a controladora pode ser ligada a um gerador de pulsos externo, um PC por exemplo, e assim pode receber sinais de comando no formato STEP/DIR, padrão da maioria dos programas de CNC. Para este modo de operação estão disponíveis as opções de configuração de tipo de passo, Passo completo / Meio passo, duas entrada de controle de fim de curso, entrada de sinais STEP ( passo ) e DIR ( direção ). Nas páginas seguintes veremos os detalhes de configuração.

**CONEXÕES**



| CODIGO | CONEC. | DESCRIÇÃO                |
|--------|--------|--------------------------|
| MOTOR  | BORNE  | ALIMENTAÇÃO 35 VCC / VAC |
| LOGICA | BORNE  | ALIMENTAÇÃO 12 VAC/VCC   |
| 1      | BORNE  | FASE 1 DO MOTOR          |
| 2      | BORNE  | FASE 2 DO MOTOR          |
| 3      | BORNE  | FASE 3 DO MOTOR          |
| 4      | BORNE  | FASE 4 DO MOTOR          |
| A      | BORNE  | ENTRADA SINAL STEP       |
| B      | BORNE  | ENTRADA SINAL DIR        |
| C      | BORNE  | ENTRADA SINAL ENA        |
| GND    | BORNE  | COMUM                    |
| +5V    | BORNE  | +5vcc P/ INTERFACE       |
| CONFIG | JUMPER | CONFIGURAÇÃO DE CORRENTE |
| FAN    | CNT    | CONEXÃO DE COOLER        |
| PWR    | LED    | INDICADOR DE ENERGIZADO  |
| STA    | LED    | INDICADOR DE STATUS      |

**NUNCA DE FORMA ALGUMA ...**

- 1 **NUNCA ALIMENTE A CONTROLADORA COM VOLTAGEM INFERIOR A 20VAC OU SUPERIOR A 35 VAC**
- 2 **NUNCA ALIMENTE A CONTROLADORA COM VOLTAGEM INFERIOR A 30VCC OU SUPERIOR A 45 VCC**
- 3 **NUNCA LIGUE AS FASES DO MOTOR DE FORMA INCORRETA**

**PROCEDIMENTOS DE CONFIGURAÇÃO**

- 1 Antes de ligar o equipamento, leia todo o manual
- 2 Conecte o motor como as instruções e pinagem correta
- 3 Conecte a alimentação nos bornes de alimentação
- 4 Conecte o cabo de configuração no conector config e na porta COM1 do PC
- 5 Execute o programa de configuração CONFIGBPRO.EXE para ajustar os parâmetros do motor

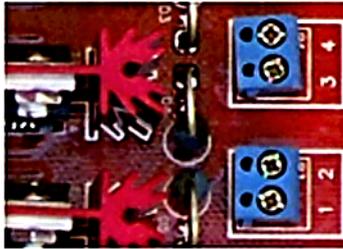
**PROCEDIMENTOS DE UTILIZAÇÃO**

- 1 Antes de ligar o equipamento, leia todo o manual
- 2 Configure a controladora para as características de operação desejadas
- 3 Conecte o motor como as instruções e pinagem adequada
- 4 Conecte o cabo de interligação na porta paralela e na controladora
- 5 Configure a porta paralela para SPP no BIOS do computador
- 6 Conecte a alimentação nos bornes de alimentação

**CONTROLADORA DE MOTOR DE PASSO SMC-B-PRO - MOTOR**

Para identificar a configuração de cores dos fios do motor de passo, a melhor opção é o manual do fabricante. Se o motor não dispõe de manual, com um multímetro você pode facilmente identificar as fases para motores de 4 fios, para motores de 6 fios, necessariamente deverá ter em mãos os dados de ligação do fabricante, pois a ligação em série ou em paralelo das bobinas invertidas causará mal funcionamento e danos no motor e controladora.

**CONEXÃO DO MOTOR DE PASSO**



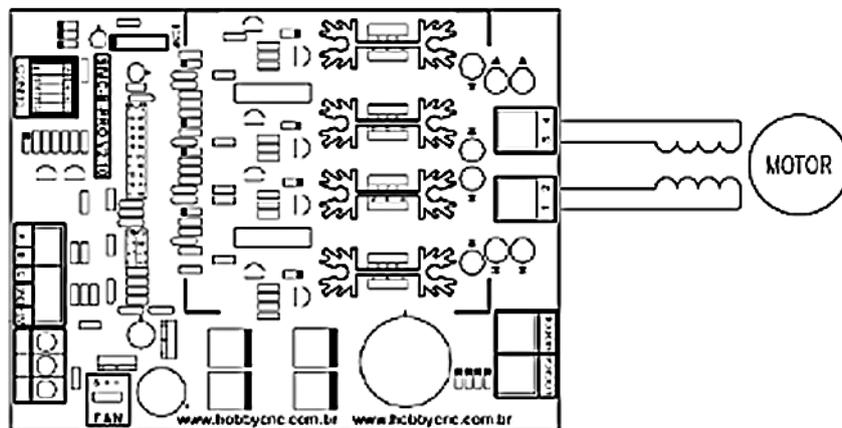
**ESPECIFICAÇÕES**

|   |                 |
|---|-----------------|
| 1 | FASE 1 DO MOTOR |
| 2 | FASE 2 DO MOTOR |
| 3 | FASE 3 DO MOTOR |
| 4 | FASE 4 DO MOTOR |

Após identificarmos a configuração de cores dos fios do motor de passo, conectamos os mesmos nos bornes 01 a 04. A imagem abaixo ilustra a conexão de um motor típico.

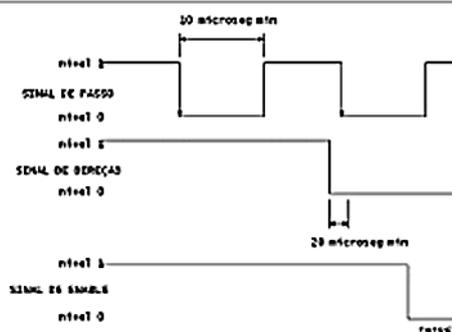
| CONTROLADORA | BORNE | FIOS | MOTOR |
|--------------|-------|------|-------|
|              | 04    | A    |       |
|              | 03    | C    |       |
|              | 02    | B    |       |
|              | 01    | D    |       |

**CONTROLADORA DE MOTOR DE PASSO BIPOLAR SMC-B-PRO - DIAGRAMA**



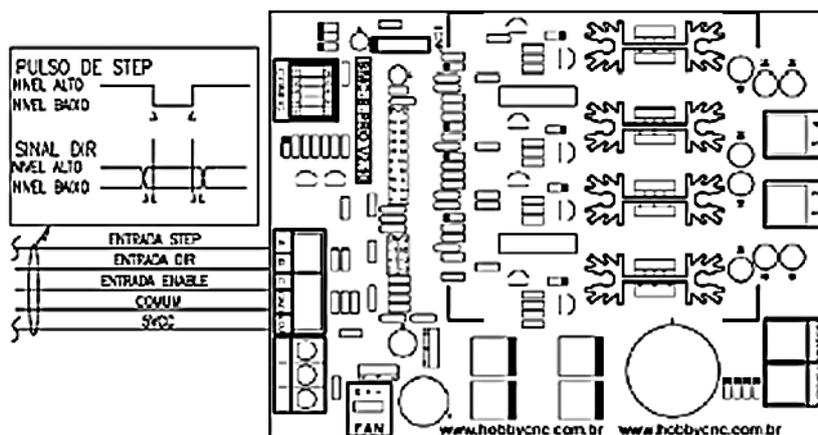
**CONTROLADORA DE MOTOR DE PASSO BIPOLAR SMC-B-PRO – SINAIS DE COMANDO**

Os sinais de comando são nível TTL compatíveis com a saída da porta paralela de PC, estas entradas tem proteção de sobretensão e subcorrente podendo ser ligadas diretamente em saídas de PLCs sem nenhuma interface adicional



|             |   |
|-------------|---|
| <b>STEP</b> | Este sinal é de entrada, e dispara os procedimentos internos do programa da controladora SMC-U para executar um passo no motor de passo, levando em conta os parâmetros de direção, 1/2 passo / passo completo e limitador de corrente. Este sinal deverá ser um pulso negativo com uma duração mínima de 10 microssegundos e um intervalo mínimo de 10 microssegundos. |
| <b>DIR</b>  | Este sinal é de entrada, e determina a direção na qual o motor irá executar o passo, este sinal deve estar estável pelo menos 20 microssegundos antes do pulso de passo ser gerado no conector STP. Na condição 0 o motor executará o passo no sentido horário, na condição 1 o motor executará o passo no sentido anti-horário.  |
| <b>ENA</b>  | Este sinal é de entrada, e inibe ou habilita o motor, sinal alto ( 1 ) habilita o motor e sinal baixo ( 0 ) inibe o motor   |
| <b>GND</b>  | Comum, deve ser conectado ao comum do gerador de pulsos   |
| <b>+5V</b>  | 5VCC para alimentação da interface  |

**CONTROLADORA DE MOTOR DE PASSO BIPOLAR SMC-B-PRO – DIAGRAMA GERAL**



## ANEXO 5. BPC-PRO GUIA DE UTILIZAÇÃO

HobbyCNC Brasil

BPC-PRO GUIA DE UTILIZAÇÃO

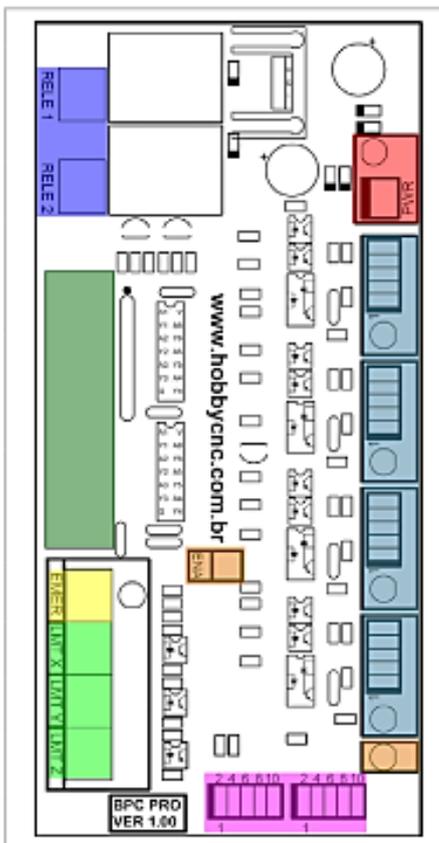
### INTERFACE BPC - STD

A interface BPC tem entradas digitais opto-isoladas para controle de motores de passo com os sinais STEP/DIR/ENA uma para controle de rele e uma para parada de emergencia.

### CARACTERÍSTICAS OPERACIONAIS

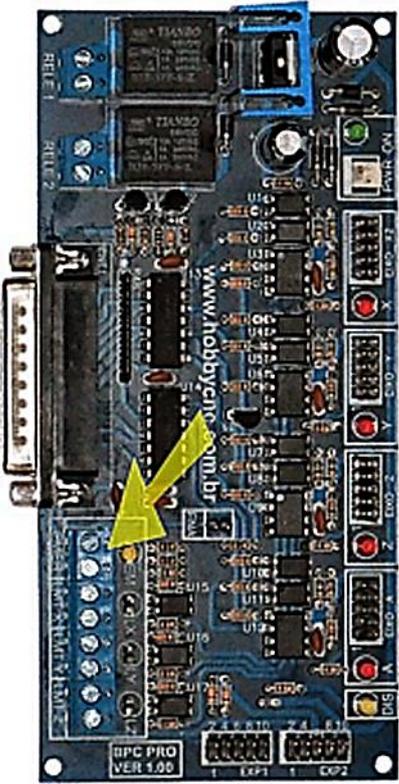
|                                      |  |
|--------------------------------------|--|
| Tensão de Alimentação :              | 9 a 12 VAC                                 |
| Isolamento Sinais PC / Drivers :     | Óptica                                     |
| Número de Drivers :                  | 4 ( QUATRO )                               |
| Entradas de Sinais Para Cada Motor : | Passo/Direção/Inibe ( step/dir/ena )       |
| Entradas de PARADA DE EMERGENCIA:    | Sim  |
| Entrada de Sinais de Rele :          | Sim ( 2 acionamentos de potencia até 10A ) |
| Entradas de FINS DE CURSO:           | Sim  |

### CONEXÕES

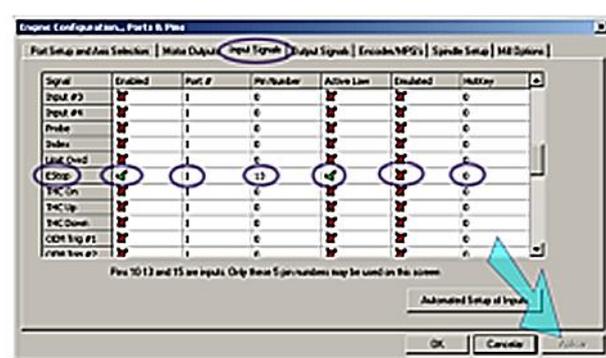


| CODIGO    | CONEC.   | DESCRIÇÃO                    |
|-----------|----------|------------------------------|
| VCC / VAC | CONECTOR | ALIMENTAÇÃO VCC / VAC        |
| RELE      | BORNE    | ACIONAMENTO DE CARGA ATÉ 10A |
| DB25      | DB25     | CONECTOR PORTA PARALELA      |
| PE        | BORNE    | PARADA DE EMERGENCIA         |
| LX        | BORNE    | LIMITE / HOME X              |
| LY        | BORNE    | LIMITE / HOME Y              |
| LZ        | BORNE    | LIMITE / HOME Z              |
| ENA       | JUMPER   | ATIVA HABILITAÇÃO            |
| X         | CONECTOR | CONECTOR P/ DRIVER EIXO X    |
| Y         | CONECTOR | CONECTOR P/ DRIVER EIXO Y    |
| Z         | CONECTOR | CONECTOR P/ DRIVER EIXO Z    |
| A         | CONECTOR | CONECTOR P/ DRIVER EIXO A    |
| EXP1      | CONECTOR | CONECTOR PARA EXPANSÃO       |
| EXP2      | CONECTOR | CONECTOR PARA EXPANSÃO       |

Estes sinais são ativados por chaves e determinam parada de emergência, a utilização destes controles é dependente do programa cnc que será utilizado



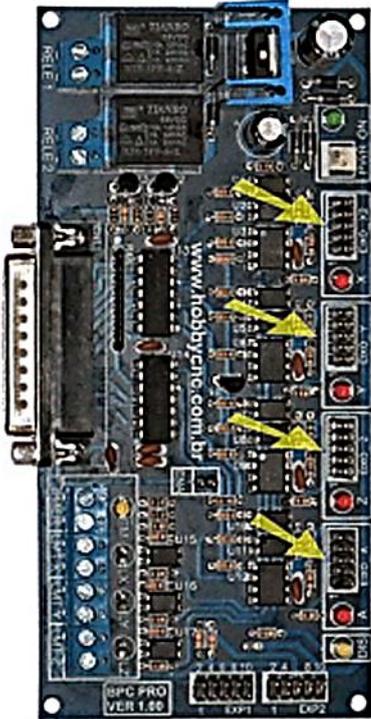
### CONFIGURAÇÃO MACH3



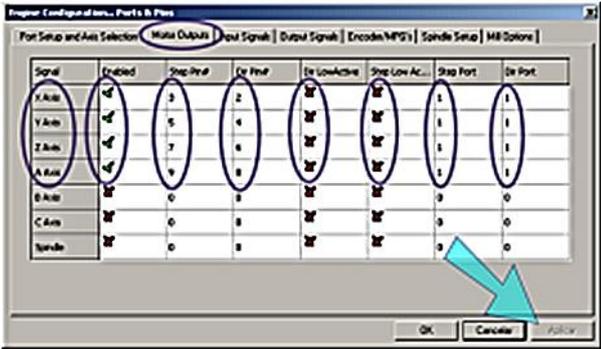
| ID | FUNCIONALIDADE       | SINAIS                   | PINOS DB25 |
|----|----------------------|--------------------------|------------|
| PE | PARADA DE EMERGENCIA | CHAVE NORMALMENTE ABERTA | 13         |

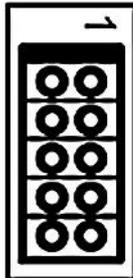
INTERFACE BPC – SINAIS DE CONTROLE

Estes sinais são enviados para driver onde são convertidos em movimento do motor



**CONFIGURAÇÃO MACH3**



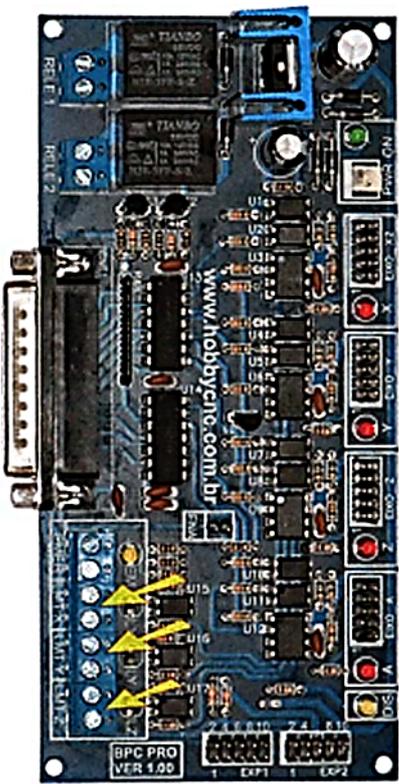


| SINAL            | DESCRIÇÃO  |
|------------------|--|
| <b>PASSO</b>     | Este sinal é de entrada, e dispara os procedimentos internos do programa do driver para executar um passo no motor, levando em conta os parâmetros de direção e resolução. Este sinal deverá ser um pulso positivo com uma duração mínima de 5 microssegundos e um intervalo mínimo de 5 microssegundos.                 |
| <b>DIREÇÃO</b>   | Este sinal é de entrada, e determina a direção na qual o motor irá executar o passo, este sinal deve estar estável pelo menos 1 microssegundo antes do pulso de passo ser gerado na entrada de PASSO. Na condição 0 o motor executará o passo em um sentido, na condição 1 o motor executará o passo no sentido Inverso. |
| <b>HABILITA</b>  | Este sinal é de entrada, e inibe ou habilita o motor, sinal alto ( 1 ) habilita o motor e sinal baixo ( 0 ) inibe o motor.   |
| <b>LED</b>       | Sinal de saída, indicador de status do driver  |
| <b>RESERVADO</b> | RESERVADO PARA FUTURAS EXPANSÕES   |
| <b>5VCC</b>      | Saída +5VCC para o BPC   |
| <b>GND</b>       | Comum, deve ser conectado ao comum do gerador de pulsos  |

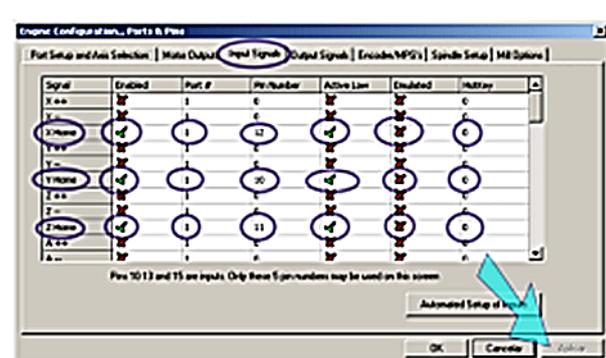
| ID       | FUNCIONALIDADE   | SINAL | PINO DB25 |
|----------|--|-------|-----------|
| <b>X</b> | Conector com sinais passo/direção/habilita para o eixo X | STEP  | 3         |
|          |  | DIR   | 2         |
| <b>Y</b> | Conector com sinais passo/direção/habilita para o eixo Y | STEP  | 5         |
|          |  | DIR   | 4         |
| <b>Z</b> | Conector com sinais passo/direção/habilita para o eixo Z | STEP  | 7         |
|          |  | DIR   | 6         |
| <b>A</b> | Conector com sinais passo/direção/habilita para o eixo A | STEP  | 9         |
|          |  | DIR   | 8         |

INTERFACE BPC – SINAIS DE CONTROLE

Estes sinais são de controle de referencia / limite utilizados para determinar o zero da maquina



**CONFIGURAÇÃO MACH3**



| ID             | FUNCIONALIDADE  | SINAL                           | PINO DB25 |
|----------------|---|---------------------------------|-----------|
| <b>LIMIT X</b> | RELE COM CONECTORES NORMALMENTE ABERTO PARA ACIONAMENTO DE CARGA ATÉ 10 AMPERES | CONTATO SECO NORMALMENTE ABERTO | 12        |
| <b>LIMIT Y</b> | RELE COM CONECTORES NORMALMENTE ABERTO PARA ACIONAMENTO DE CARGA ATÉ 10 AMPERES | CONTATO SECO NORMALMENTE ABERTO | 10        |
| <b>LIMIT Z</b> | RELE COM CONECTORES NORMALMENTE ABERTO PARA ACIONAMENTO DE CARGA ATÉ 10 AMPERES | CONTATO SECO NORMALMENTE ABERTO | 11        |

## ANEXO 6. CÓDIGO G CRIADO PELO ADAPTADOR A PARTIR DE UM PROGRAMA DE PEÇA EM STEP-NC

```

(..::: Header :::..)
(..::: Codigo G gerado
pelo Adaptador de STEP-NC a
Codigo G v1.0 R57 :::..)
(Arquivo: C:/mod/ran.p21)
N1 G17 G21 G54 G40 G90
N2 M05
N3 G90 G40 G55 G00 Z5.0000
N4 G01 F800.0000 X0.0000
Y0.0000
N5 G00 Z0.0000
N6 M06 T1
N7 G00 Z5.0000
N8 M03
N9 G54
N10 G40 G90 G01 F800.0
X0.0000 Y0.0000
N11 G90 G01 F400.0 Y50.0000
N12 G00 Z0.0000
N13 G01 Z-0.2500
N14 G90 G01 X101.5000
N15 Y51.5000
N16 X-1.5000
N17 Y48.5000
N18 X101.5000
N19 G41 D1
N20 G90 Y56.0000
N21 X-1.5000
N22 G90 Y44.0000
N23 X101.5000
N24 G90 Y59.0000
N25 X-1.5000
N26 G90 Y41.0000
N27 X101.5000
N28 Y60.0000
N29 X-1.5000
N30 Y40.0000
N31 X101.5000
N32 Z0.0000
N33 G40 G90 X0.0000
Y50.0000
N34 G91 G01 F400.0 Z-1.0000
N35 G90 G01 X101.5000
N36 Y51.5000
N37 X-1.5000
N38 Y48.5000
N39 X101.5000
N40 G41 D1
N41 G90 Y56.0000
N42 X-1.5000
N43 G90 Y44.0000
N44 X101.5000
N45 G90 Y59.0000
N46 X-1.5000
N47 G90 Y41.0000
N48 X101.5000
N49 Y59.2857
N50 X-1.5000
N51 Y40.7143
N52 X101.5000
N53 G40 G90 X0.0000
Y50.0000
N54 G91 G01 F400.0 Z-1.0000
N55 G90 G01 X101.5000
N56 Y51.5000
N57 X-1.5000
N58 Y48.5000
N59 X101.5000
N60 G41 D1
N61 G90 Y56.0000
N62 X-1.5000
N63 G90 Y44.0000
N64 X101.5000
N65 Y58.5714
N66 X-1.5000
N67 Y41.4286
N68 X101.5000
N69 G40 G90 X0.0000
Y50.0000
N70 G91 G01 F400.0 Z-1.0000
N71 G90 G01 X101.5000
N72 Y51.5000
N73 X-1.5000
N74 Y48.5000
N75 X101.5000
N76 G41 D1
N77 G90 Y56.0000
N78 X-1.5000
N79 G90 Y44.0000
N80 X101.5000
N81 Y57.8571
N82 X-1.5000
N83 Y42.1429
N84 X101.5000
N85 G40 G90 X0.0000
Y50.0000
N86 G91 G01 F400.0 Z-1.0000
N87 G90 G01 X101.5000
N88 Y51.5000
N89 X-1.5000
N90 Y48.5000
N91 X101.5000
N92 G41 D1
N93 G90 Y56.0000
N94 X-1.5000
N95 G90 Y44.0000
N96 X101.5000
N97 Y57.1429
N98 X-1.5000
N99 Y42.8571
N100 X101.5000
N101 G40 G90 X0.0000
Y50.0000
N102 G91 G01 F400.0 Z-
1.0000
N103 G90 G01 X101.5000
N104 Y51.5000
N105 X-1.5000
N106 Y48.5000
N107 X101.5000
N108 G41 D1
N109 G90 Y56.0000
N110 X-1.5000
N111 G90 Y44.0000
N112 X101.5000
N113 Y56.4286
N114 X-1.5000
N115 Y43.5714
N116 X101.5000
N117 G40 G90 X0.0000
Y50.0000
N118 G91 G01 F400.0 Z-
1.0000
N119 G90 G01 X101.5000
N120 Y51.5000
N121 X-1.5000
N122 Y48.5000
N123 X101.5000
N124 G41 D1
N125 Y55.7143
N126 X-1.5000
N127 Y44.2857
N128 X101.5000
N129 G40 G90 X0.0000
Y50.0000
N130 G91 G01 F400.0 Z-
0.1625
N131 G90 G01 X101.5000
N132 Y51.5000
N133 X-1.5000
N134 Y48.5000
N135 X101.5000
N136 G41 D1
N137 Y55.5982
N138 X-1.5000
N139 Y44.4018
N140 X101.5000
N141 M05
N142 G90 G40 G55 G00
Z5.0000
N143 G01 F800.0000 X0.0000
Y0.0000
N144 G00 Z0.0000
N145 M06 T2
N146 G00 Z5.0000
N147 M03
N148 G54
N149 G40 G90 G01 F800.0
X0.0000 Y0.0000
N150 G90 G01 F400.0
Y50.0000
N151 G00 Z0.0000
N152 G90 G00 Z-5.0000
N153 G91 G01 Z-2.0000
N154 G90 G01 X102.0000
N155 Y52.0000
N156 X-2.0000
N157 Y48.0000
N158 X102.0000
N159 G41 D2
N160 Y55.5982
N161 X-2.0000
N162 Y44.4018
N163 X102.0000
N164 G90 X104.0000
N165 G40 G90 G00 Z45.0000
N166 M05
N167 M30

```

## ANEXO 7. EQUAÇÕES USADAS NO MODELAMENTO MATEMÁTICO DA GEOMETRIA DE FEATURES

- **Equação reduzida da reta**

A equação de uma reta pode ser escrita de várias formas. Entre as mais uteis está a forma angular-intercepto, conhecida comumente como equação reduzida da reta. Esta equação define uma reta com coeficiente angular  $m$  e intercepto  $b$  [107]. A equação é dada por:

$$y = m * x + b \quad (\text{An7-1})$$

A equação An7-1 define uma reta para o plano XY. No entanto, neste trabalho é usada nos planos XZ e YZ descrevendo à largura e a profundidade, i.e. da equação An7-1(An7-1),  $x$  pode ser o plano X ou Y e  $y$  refere-se à profundidade ou ao plano Z.

O cálculo do coeficiente angular é dado pela equação An7-2, assim:

$$m = \frac{y2 - y1}{x2 - x1} \quad (\text{An7-2})$$

e o cálculo do intercepto resulta do isolamento da constante  $b$  da equação An7-1 ficando que a equação An7-3 seja:

$$b = y - m * x \quad (\text{An7-3})$$

Da mesma forma, é possível isolar a variável  $x$  da equação An7-1 ficando que a equação An7-4 seja:

$$x = \frac{y - b}{m} \quad (\text{An7-4})$$

- **Teorema de Pitágoras**

Este teorema baseia-se no cálculo sobre triângulos retângulos. Os triângulos retângulos consistem em dois catetos: um maior e outro menor, além da hipotenusa, que é o maior lado do triângulo. Chama-se de triângulo retângulo, um triângulo em que um dos seus ângulos é reto, i.e. mede  $90^\circ$  [108]. A Figura An7.1 mostra um triângulo retângulo com as suas características.

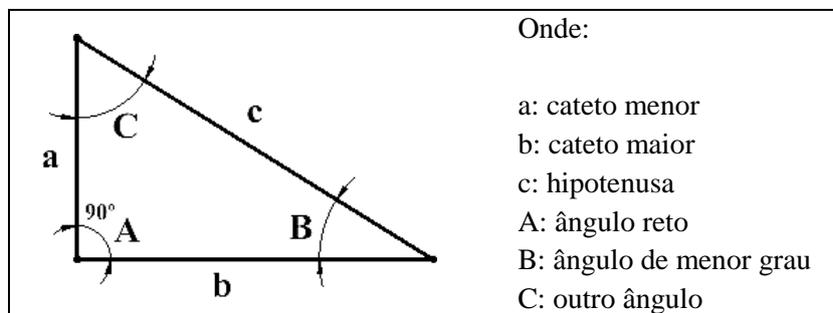


Figura An7.1: Triângulo retângulo

O teorema de Pitágoras expõe que a soma dos catetos elevados ao quadrado, é igual à hipotenusa elevada ao quadrado. Tomando como referência a Figura An7.1 para explicar este teorema, pode-se escrever a seguinte equação An7-5:

$$c^2 = a^2 + b^2 \quad (\text{An7-5})$$

Manipulando a equação An7-5, é possível isolar qualquer variável como mostrado nas equações An7-6 e An7-7.

$$a^2 = c^2 - b^2 \quad (\text{An7-6})$$

$$b^2 = c^2 - a^2 \quad (\text{An7-7})$$

- **Relações trigonométricas do triângulo retângulo**

No triângulo retângulo existem algumas importantes relações, e.g. o Teorema de Pitágoras, (explicado anteriormente), as relações trigonométricas entre outras. As relações trigonométricas existentes no triângulo retângulo admitem três casos: seno, cosseno e tangente. Tendo como referência a Figura An7.1, pode-se escrever as seguinte equações que expõem essas relações:

$$\sin B = \frac{a}{c}, \quad \cos B = \frac{b}{c}, \quad \tan B = \frac{c}{b} \quad (\text{An7-8})$$

$$\sin C = \frac{b}{a}, \quad \cos C = \frac{c}{a}, \quad \tan C = \frac{b}{c} \quad (\text{An7-9})$$

De cada uma destas equações (An7-8 e An7-9), é possível isolar o ângulo, se são conhecidas as variáveis independentes, e.g. com o  $\cos B$  (An7-8):

$$\cos B = \frac{b}{c}$$

Ficaria que o ângulo  $B$  é (equação An7-10):

$$B = \frac{1}{\cos} \left( \frac{b}{c} \right) \quad (\text{An7-10})$$

# ANEXO 8. MODELO E CERTIFICADO DE CALIBRAÇÃO CMM MITUTOYO



**LABORATÓRIO DE METROLOGIA**  
**REDE BRASILEIRA DE CALIBRAÇÃO**  
Laboratório de Calibração Acreditado pela CGCRE/INMETRO de acordo com a ABNT NBR ISO/IEC 17025, sob No. 031

Pag: 2 / 4

**CERTIFICADO DE CALIBRAÇÃO No. 7407/08**

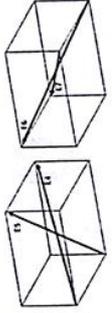
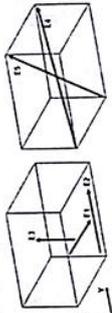
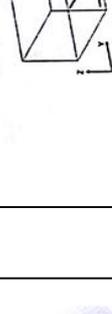
Cliente: Laboratório de Metrologia - LM - UNIR.

**IV. Resultado das Medições:**

4.1. Verificações iniciais do equipamento:

- a. Movimento da Máquina: OK
- b. Condições da área de Medição: OK
- c. Avaliação do Sist. Funcional: OK
- d. Limpeza das superfícies de ref.: OK

4.2. Verificação do equipamento:

4.2.1. Exatidão de medição Posição E1 - unidade:  $\mu\text{m}$

| Posição (L) | 1a   | 2a   | 3a   | erro médio |
|-------------|------|------|------|------------|
| 0           | 0,0  | 0,0  | 0,0  | 0,0        |
| 30          | -1,3 | -0,8 | -1,2 | -1,1       |
| 70          | -1,2 | -0,7 | -0,9 | -0,9       |
| 150         | -0,5 | -1,0 | 0,0  | -0,5       |
| 210         | 0,1  | 0,6  | 1,2  | 0,6        |
| 270         | 1,2  | 1,2  | 2,8  | 1,7        |

$U = 1,2 \pm 1,500 \mu\text{m}$

4.2.2. Exatidão de medição Posição E2 - unidade:  $\mu\text{m}$

| Posição (L) | 1a   | 2a   | 3a   | erro médio |
|-------------|------|------|------|------------|
| 0           | 0,0  | 0,0  | 0,0  | 0,0        |
| 30          | -0,9 | -2,0 | -2,1 | -1,7       |
| 70          | -1,2 | -2,2 | -2,8 | -2,1       |
| 150         | -1,8 | -1,4 | -3,1 | -2,1       |
| 210         | 0,4  | -0,2 | -0,7 | -0,2       |
| 270         | 1,6  | 0,7  | 0,6  | 1,0        |

$U = 1,2 \pm 1,200 \mu\text{m}$

onde: L = posição em "mm".

4. Rua Carlos Gomes, Av. João Cabral de Silva Borges, 1240 - Santa Amélia - CEP 04726-902 - São Paulo - SP - Tercos Chaves (11) 5443.0000 - Fax (11) 5443.3722  
 4. Rua Carlos Gomes, Av. João Cabral de Silva Borges, 1240 - Santa Amélia - CEP 04726-902 - São Paulo - SP - Tercos Chaves (11) 5443.0000 - Fax (11) 5443.3722  
 4. Rua Carlos Gomes, Av. João Cabral de Silva Borges, 1240 - Santa Amélia - CEP 04726-902 - São Paulo - SP - Tercos Chaves (11) 5443.0000 - Fax (11) 5443.3722  
 4. Rua Carlos Gomes, Av. João Cabral de Silva Borges, 1240 - Santa Amélia - CEP 04726-902 - São Paulo - SP - Tercos Chaves (11) 5443.0000 - Fax (11) 5443.3722



**LABORATÓRIO DE METROLOGIA**  
**REDE BRASILEIRA DE CALIBRAÇÃO**  
Laboratório de Calibração Acreditado pela CGCRE/INMETRO de acordo com a ABNT NBR ISO/IEC 17025, sob No. 031

Pag: 1 / 4

**CERTIFICADO DE CALIBRAÇÃO No. 7407/08**

Cliente: Laboratório de Metrologia - LM - UNIR.

Endereço: UNIR - Campus Universitário Darcy Ribeiro - Itaú Norte - Brasília, DF.  
Local de calibração: UNIR - Campus Universitário Darcy Ribeiro - Itaú Norte - Brasília, DF.

Ordem de Serviço: 024.012.001

**I. Identificação do Equipamento Calibrado:**

Máquina Tridimensional de Medição por Coordenadas  
Fabricante: Mitutoyo  
MMC Modelo: QM-353  
No de Série: 0072807 / Código: 188-314  
Identif. do Cliente: N/C  
Faixa Nominal: 300 x 300 x 300 mm  
Controlador Digital / Controlador  
Modelo: UDL-1YC  
No Série: N/C  
Resolução: 0,0005 mm  
Cabeçote  
Modelo: MH20  
No Série: SL7792  
Proble: T120-SF  
Modelo: T120-SF  
No Série: 710621  
Rack  
Modelo: -  
No Série: -  
Programa de Medição  
Versão: 13.0816  
Dongle No.: 10-102-RS19

**II. Notas:**

- A incerteza declarada é baseada em uma incerteza padrão combinada multiplicada por um fator de abrangência  $k=2,1$ , para um nível de confiança de aproximadamente 95%.
- Erro - Valor indicado - Valor Verdadeiro Convencional.
- Os sinais dos resultados tem o mesmo sentido das indicações positivas e negativas do sistema de coordenadas da máquina.
- A máquina foi verificada conforme Procedimento PMI-0030 na sua versão 12.
- Condição ambiental:  
Temperatura: 20,1  $\pm$  0,5°C.

**III. Padrões utilizados:**

Padrão Escalonado  
No Série: 921120  
Certificado No.: 282408  
Emissão em: 31/2008  
Validade: 10/2008

Padrão  
No Série: T1F7008  
Certificado No.: 2909/07  
Emissão em: 13/2007  
Validade: 04/2008

Data da Calibração: 8/8/2008

  
**Carlos Roberto de Azevedo**  
 Engenheiro de Metrologia  
 Emissão: 07/06/08

Este certificado tem validade de 12 meses a partir da data de emissão. O cliente deve manter o equipamento em boas condições de conservação e utilizar o mesmo para a mesma finalidade de uso. O cliente deve manter o equipamento em boas condições de conservação e utilizar o mesmo para a mesma finalidade de uso. O cliente deve manter o equipamento em boas condições de conservação e utilizar o mesmo para a mesma finalidade de uso. O cliente deve manter o equipamento em boas condições de conservação e utilizar o mesmo para a mesma finalidade de uso.

