

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

**DESENVOLVIMENTO DE UM SISTEMA PARA DIMENSIONAMENTO E CONTROLE DE  
POSICIONADORES HIDRÁULICOS**

**Dissertação submetida à**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**para a obtenção do grau de**

**MESTRE EM ENGENHARIA MECÂNICA**

**ALISSON DALSSASSO CORRÊA DE SOUZA**

**Florianópolis, Março de 2005.**

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

**DESENVOLVIMENTO DE UM SISTEMA PARA DIMENSIONAMENTO E CONTROLE DE  
POSICIONADORES HIDRÁULICOS**

**ALISSON DALSSASSO CORRÊA DE SOUZA**

**Esta dissertação foi julgada adequada para a obtenção do título de**

**MESTRE EM ENGENHARIA**

**ESPECIALIDADE ENGENHARIA MECÂNICA**

**sendo aprovada em sua forma final.**

\_\_\_\_\_  
**Prof. Victor Juliano De Negri, Dr. Eng. - Orientador**

\_\_\_\_\_  
**Prof. José Antônio Bellini da Cunha Neto, Dr. - Coordenador do Curso**

**BANCA EXAMINADORA**

\_\_\_\_\_  
**Prof. Jonny Carlos da Silva, Dr. Eng. - Presidente**

\_\_\_\_\_  
**Prof. Carlos Alberto Flesch, Dr. Eng.**

\_\_\_\_\_  
**Prof. Rômulo Silva de Oliveira, Dr. Eng.**

**“Se um homem vai começar com certezas, ele terminará em dúvidas; mas se ele estiver contente em começar com dúvidas, ele terminará em certezas.”**

**Francis Bacon**

**À minha família.**

## AGRADECIMENTOS

Agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

A Deus por fazer da minha vida uma vitória diária, proporcionando-me força e sabedoria para buscar a realização dos meus sonhos.

Aos meus pais Abel Corrêa de Souza e Jerusa Elena Dalsasso de Souza e meu irmão André Dalsasso Corrêa de Souza, que compartilham comigo as vitórias e os maus momentos sempre com um sorriso de confiança e palavras de incentivo.

A minha noiva Bartira Righetto Bussolo por todo o apoio, compreensão e amor dedicados a mim durante o desenvolvimento deste trabalho.

Ao Professor Victor Juliano De Negri pelo apoio, orientação e respeito demonstrados no decorrer do trabalho, através do qual foi possível a sua concretização.

Aos amigos e colegas da Universidade Federal de Santa Catarina, principalmente aos amigos do LASHIP pela amizade, incentivo, paciência e colaboração durante o desenvolvimento do trabalho.

À empresa BOSCH REXROTH Ltda. pela colaboração e financiamento prestados ao Projeto Plataforma Hidráulica Proporcional.

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

À instituição pela oportunidade de crescimento pessoal e profissional.

## SUMÁRIO

Lista de figuras.....	x
Lista de tabelas.....	xiii
Simbologia.....	xiv
Lista de Abreviaturas.....	xv
Resumo.....	xvi
Abstract.....	xvii
1. Introdução.....	1
1.1 Contextualização.....	1
1.1.1 Plataforma de Hidráulica Proporcional.....	1
1.1.1.1 Sistematização de Procedimentos para Dimensionamento e Análise Estática e Dinâmica de Circuitos Hidráulicos.....	6
1.2 Objetivos e metodologia.....	7
1.3 Justificativas.....	8
1.4 Estrutura deste documento.....	9
2. Sistemas Eletro-Hidráulicos de Controle de Posição.....	10
2.1 Apresentação.....	11
2.2 Controladores Automáticos.....	13
2.2.1 Controladores Eletrônicos.....	13
2.2.2 O Controlador PID.....	15
3. Sistemas de Aquisição de Dados e Controle.....	16
3.1 Sistema VXIbus.....	16
3.1.1 Apresentação.....	16
3.1.2 Maneiras de Controlar um Sistema VXI.....	17
3.1.3 Aquisição de Dados com VXI.....	19
3.2 Software LabVIEW.....	20
3.3 Acesso aos Instrumentos VXI no LabVIEW.....	22
3.4 Controle Digital.....	22
3.4.1 Problemas Ligados ao Controle de Sistemas Amostrados.....	23
3.4.1.1 Digitalização de um Sinal Analógico.....	23
3.4.1.1.1 Aliasing.....	24
3.4.1.2 Quantização e Codificação.....	25
3.4.1.3 Reconstituição do Sinal Analógico.....	25
3.4.1.4 Projeto de Controladores Discretos.....	26
3.4.1.4.1 Método de Projeto por Aproximações.....	26
3.4.1.4.2 Método de Projeto Direto.....	26
4. Desenvolvimento de Software Voltado ao LabVIEW.....	28
4.1 Modelos de Desenvolvimento.....	28

4.2 Modelos de Ciclo de Vida .....	28
4.2.1 Modelo Codificar e Corrigir .....	29
4.2.2 Modelo Cascata (Waterfall) .....	29
4.2.3 Prototipação .....	31
4.2.4 Modelo Espiral .....	32
4.3 Qualidade no Processo de Desenvolvimento .....	33
4.3.1 Requisitos de Qualidade .....	34
4.3.2 Gerência da Configuração .....	34
4.3.2.1 Recuperação de Versões Antigas de Arquivos no LabVIEW .....	35
4.3.2.2 Localização de Mudanças no LabVIEW .....	35
4.3.3 Diretrizes de Teste .....	35
4.3.3.1 Teste de Caixa Preta e Teste de Caixa Branca .....	36
4.3.3.2 Testes de Unidade, Integração e Sistema .....	36
4.3.3.2.1 Testes de Unidade .....	36
4.3.3.2.2 Testes de Integração .....	37
4.3.3.2.3 Testes de Sistema .....	38
4.3.4 Diretrizes de Estilo .....	39
4.3.5 Revisões de Projeto .....	40
4.3.6 Revisões de Código .....	40
4.3.7 Análise Pós-Projeto .....	41
4.4 Técnicas de Prototipagem e Projeto .....	41
4.4.1 Definição de Requisitos da Aplicação .....	41
4.4.2 Projeto Top-Down .....	43
4.4.3 Projeto Bottom-Up .....	43
4.4.4 Aspectos Fundamentais do Projeto .....	44
4.4.4.1 Abstração .....	44
4.4.4.2 Refinamento .....	44
4.4.4.3 Modularidade .....	44
4.4.4.4 Hierarquia de Controle .....	44
4.4.4.5 Estrutura dos Dados .....	45
4.4.4.6 Ocultação de Informação .....	45
4.4.5 Projeto com Múltiplos Desenvolvedores .....	45
4.4.6 Prototipagem do Painel Frontal .....	46
4.4.7 Testes de Desempenho .....	47
4.5 Documentação para Desenvolvimento em LabVIEW .....	47
4.5.1 Projeto e Desenvolvimento de Documentação .....	48
4.5.2 Desenvolvimento da Documentação do Usuário .....	48
4.5.2.1 Organização Sistemática da Documentação .....	48

4.5.2.2 Documentação de uma Aplicação .....	48
4.5.3 Criação de Arquivos de Ajuda .....	49
4.5.4 Descrição de VIs, Controles e Indicadores .....	49
4.5.4.1 Criação das Descrições do VI .....	49
4.5.4.2 Documentação dos Painéis Frontais .....	49
4.5.4.3 Criação das Descrições de Controles e Indicadores .....	50
5. Conceituação e Modelagem de Sistemas Automáticos .....	51
5.1 Conceituação de Sistemas Automáticos .....	51
5.2 Projeto de Sistemas Automáticos .....	52
5.3 Modelagem Genérica de Sistemas Automáticos .....	53
5.4 Modelo Detalhado de Sistema Automático .....	54
6. Modelagem do Aplicativo .....	56
6.1 Concretização do Sistema Automático .....	56
6.2 Modelagem Funcional/Estrutural do Aplicativo .....	60
6.2.1 Módulo “Interface com Usuário (Usuário 1)” .....	63
6.2.2 Módulo “Interface com Usuário (Usuário 2)” .....	65
6.2.3 Módulo “Gerenciador” .....	65
6.2.4 Módulo “Controlador” .....	68
7. Desenvolvimento do Aplicativo .....	71
7.1 Análise de Requisitos .....	71
7.2 Prototipação do Aplicativo .....	72
7.3 O Processo de Projeto .....	76
7.3.1 Projetos Preliminar e Detalhado .....	76
7.3.2 Características Desejadas no Projeto do Aplicativo .....	76
7.4 Hierarquia de Funções do Aplicativo .....	76
7.5 Documentação .....	78
7.5.1 Documentação ao Projetista/Programador .....	78
7.5.2 Documentação ao Usuário .....	78
7.6 Testes de Validação .....	79
7.7 Integração do Aplicativo com o Módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico” .....	81
7.8 Discussão .....	85
8. Conclusões .....	86
Referências bibliográficas .....	88
Apêndices .....	95
1. Guia de Estilo em LabVIEW .....	96
2. Tecnologia DataSocket .....	99
3. Legenda das Informações do Software .....	101



4. Especificações Gerais do Multímetro .....	109
5. Documentação do Software ao Projetista/Programador do Módulo “Interface Com Usuário (Usuário 1)” .....	113
6. Documentação do Software ao Usuário do Módulo “Interface Com Usuário (Usuário 1)” .....	119
7. Ajuda do “Passo 1” do Módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico” .....	122
8. Janelas dos Passos do Módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico” .....	125

## LISTA DE FIGURAS

Figura 1.1 – Diagrama da estrutura da Plataforma de Hidráulica Proporcional. ....	3
Figura 1.2 – a) DS3 Synergy; b) DS3 G. (BOSCH REXROTH GROUP, 2002a; 2002b). ....	3
Figura 1.3 – Ilustração do Módulo de Trabalho da Plataforma de Hidráulica Proporcional. ...	4
Figura 1.4 – Equipamento utilizado atualmente no LASHIP para o controle de posição de sistemas hidráulicos. ....	5
Figura 1.5 – Atual sistema de atuação hidráulico da Bancada de Trabalho. ....	6
Figura 1.6 – Tarefas do dimensionamento estático e dinâmico para a seleção da válvula e cilindro (FURST, 2001). ....	7
Figura 2.1 – Aviação: posicionamento de leme (EMBRAER, 2004). ....	10
Figura 2.2 – Colheitadeiras: controle de posição da plataforma (JOHN DEERE, 2005). ....	10
Figura 2.3 – Centrais hidrelétricas: posicionamento de pás. ....	11
Figura 2.4 – Ilustração de um sistema hidráulico de controle de posição. ....	12
Figura 3.1 – Hardware de um sistema de controle por computador (adaptado de COELHO <i>et al.</i> , 2001). ....	16
Figura 3.2 – Maneiras de controlar um sistema VXI (NATIONAL INSTRUMENTS CORPORATION, 2003). ....	18
Figura 3.3 – Desempenho versus custos dos sistemas VXI (AGILENT TECHNOLOGIES, 2001). ....	19
Figura 3.4 – Painel frontal de um programa. ....	21
Figura 3.5 – Diagrama de blocos de um programa. ....	21
Figura 3.6 – Acesso às VIs e funções de <i>Instrument I/O</i> e VXI. ....	22
Figura 3.7 – Multiplexador (PAGANO, 1999) ....	23
Figura 3.8 – Controle digital de processos contínuos (adaptado de TROFINO, 2002). ....	23
Figura 3.9 – Amostragem de sinais contínuos (PAGANO, 1999). ....	24
Figura 3.10 – Amostragem de um sinal senoidal com taxas próximas ao limite (ZURMELY, 2004). ....	25
Figura 3.11 – Sistema de controle discreto (PAGANO, 1999). ....	27
Figura 4.1 – Modelo do ciclo de vida cascata. ....	30
Figura 4.2 – Modelo de ciclo de vida espiral (adaptado de MAZZOLA; FARINES, 1998) ....	33
Figura 4.3 – Integração <i>top-down</i> (MAZZOLA; FARINES, 1998). ....	38
Figura 5.1 – Uma representação esquemática da mecatrônica (ADAMOWSKI; FURUKAWA, 2001). ....	52
Figura 5.2 – Representação de um sistema automático (DE NEGRI, 1996). ....	54
Figura 5.3 – Modelo funcional/estrutural refinado de um sistema automático (SOUTO, 2005). ....	55
Figura 6.1 – Modelo funcional/estrutural refinado proposto de um sistema automático. ....	57
Figura 6.2 – Modelo preliminar do Módulo de Trabalho. ....	58

Figura 6.3 – Modelo detalhado de Instrumentos VXI. ....	60
Figura 6.4 – Visão geral da relação entre os usuários e o software em LabVIEW. ....	61
Figura 6.5 – Visão refinada dos usuários da Bancada de Trabalho.....	61
Figura 6.6 – Modelagem preliminar do aplicativo.....	62
Figura 6.7 – Módulo “Interface com Usuário (Usuário 1)”. ....	64
Figura 6.8 – Primeiro detalhamento do módulo “Interface com Usuário (Usuário 1)”. ....	64
Figura 6.9 – Segundo detalhamento do módulo “Interface com Usuário (Usuário 1)”. ....	64
Figura 6.10 – Detalhamento da função “Envio de Dados”.....	65
Figura 6.11 – Detalhamento da função “Aquisição de Dados”.....	65
Figura 6.12 – Diagrama de escolha das válvulas da Bancada de Trabalho. ....	67
Figura 6.13 – Módulo “Gerenciador”.....	67
Figura 6.14 – Detalhamento do módulo “Gerenciador”. ....	67
Figura 6.15 – Detalhamento das funções “Aquisição de Dados” e “Envio de Dados”.....	68
Figura 6.16 – Módulo “Controlador”.....	69
Figura 6.17 – Detalhamento do módulo “Controlador”. ....	69
Figura 6.18 – Detalhamento das funções do “Controlador”.....	70
Figura 6.19 – Detalhamento da função “Aquisição de Dados (Usuário 1)”. ....	70
Figura 7.1 – Versão preliminar da “Interface com Usuário (Usuário 1)” do aplicativo.....	74
Figura 7.2 – Versão final da “Interface com Usuário (Usuário 1)” do aplicativo. ....	74
Figura 7.3 – Acesso aos controles da versão final da “Interface com Usuário (Usuário 1)” do aplicativo.....	75
Figura 7.4 – Acesso aos indicadores da versão final da “Interface com Usuário (Usuário 1)” do aplicativo.....	75
Figura 7.5 – Módulos principais do aplicativo.....	77
Figura 7.6 – Hierarquia do módulo “Interface com Usuário (Usuário 1)”.....	77
Figura 7.7 – Hierarquia do módulo “Interface com Usuário (Usuário 2)”.....	77
Figura 7.8 – Hierarquia do módulo “Gerenciador”.....	77
Figura 7.9 – Hierarquia do módulo “Controlador”.....	77
Figura 7.10 – Hierarquia do módulo “Aquisição de Dados (Usuário 1)”.....	78
Figura 7.11 – Relatório em HTML dos módulos de interface com os usuários. ....	79
Figura 7.12 – Gráfico gerado em Excel pela interface com usuário.....	80
Figura 7.13 – Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico (FURST, 2001). ....	82
Figura 7.14 – Passo 2 do módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”.....	83
Figura 7.15 – Passos 3 e 4 do módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”.....	83

Figura 7.16 - Hierarquia de funções do módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico” .....	84
Figura 7.17 – Hierarquia de funções do módulo “Interface com Usuário (Usuário 1)” .....	85

**LISTA DE TABELAS**

Tabela 4.1 – Análise de exposição de riscos. ....	33
Tabela 6.1 – Lista de informações para o módulo “Interface com Usuário (Usuário 1)” .....	63
Tabela 6.2 – Lista de informações para o módulo “Gerenciador” .....	66
Tabela 6.3 – Lista de informações para o módulo “Controlador” .....	68

**SIMBOLOGIA**

A1	– Cilindro hidráulico e carga
EE	– Energia elétrica
EH	– Energia hidráulica
EM	– Energia mecânica
f	– Frequência do sinal
$f_{am}$	– Frequência de amostragem
Inf	– Informações
M	– Massa
mat	– Material
$p_s$	– Pressão de suprimento
S1	– Sensor de posição
SA	– Sistema de atuação
SE	– Sinal elétrico
SM	– Sistema de medição
SP	– Sistema de pedido
SR	– Sistema de resposta
T	– Período de amostragem
U	– Tensão de comando
$U_r$	– Tensão de referência
V1	– Válvula direcional proporcional ou servoválvula
$x^{A1}$	– Posição real do êmbolo
Z1	– Controlador

**LISTA DE ABREVIATURAS**

A/D	– Analógico – Digital
C/A	– Canal/Agência
CLP	– Controlador Lógico Programável
D/A	– Digital – Analógico
DAQ	– <i>Data Acquisition</i>
E/S	– Entrada/Saída
FET	– <i>Field-effect transistor</i>
FireWire	– Padrão para conexão de periféricos
HTML	– <i>Hyper Text Markup Language</i>
I/O	– <i>Input/Output</i>
IEEE 1394	– Padrão para conexão de periféricos
IEEE-488	– Padrão de comunicação para instrumentos
LabVIEW	– <i>Laboratory Virtual Instrument Engineering Workbench</i>
LASHIP	– Laboratório de Sistemas Hidráulicos e Pneumáticos
PCI	– <i>Peripheral Component Interconnect</i>
PID	– Proporcional Integral Derivativo
Rede C/A	– Rede de Petri Canal/Agência
RS232	– Padrão de comunicação serial
RS485	– Padrão de comunicação serial
UPCH	– Unidade de potência e condicionamento hidráulico
VI	– <i>Virtual Instrument</i>
VISA	– <i>Virtual Instrument Software Architecture</i>
VME	– <i>Virtual Machine Environment</i>
VXI	– <i>VME Extensions for Instrumentation</i>

## RESUMO

As atividades de projeto de sistemas automáticos ocorrem em diferentes domínios tecnológicos e, dependendo da amplitude do sistema, pode haver o envolvimento de diversos especialistas, cada qual capaz de resolver aspectos técnicos em sua área do conhecimento. Por sua vez, entende-se que a integração de todos os especialistas é possível por meio de uma metodologia própria para sistemas automáticos que contemple, dentro do possível, o processo de projeto de sistemas técnicos e o desenvolvimento de software. Nesse contexto, este trabalho apresenta a elaboração de um aplicativo para um sistema de projeto e controle de posicionadores hidráulicos. Este software realiza o controle de posição de sistemas eletro-hidráulicos e também implementa um diagrama sistematizado de procedimentos para dimensionamento e análise estática e dinâmica de circuitos hidráulicos. Uma vez que esse software está incluso em um equipamento que se caracteriza por ser um sistema automático, foi utilizada neste trabalho uma metodologia própria para sistemas automáticos com o intuito de se alcançar as funções as quais o software desenvolvido deve contemplar. Por sua vez, a implementação destas funções foi realizada seguindo-se aspectos da engenharia de software. Demonstrou-se que esta metodologia empregada, a qual integra técnicas de descrição gráfica para sistemas automáticos e busca ser neutra quanto a possíveis implementações, mostrou-se uma ferramenta eficiente no projeto de sistemas automáticos.

*Palavras-chave:* projeto de sistemas automáticos, desenvolvimento de software, controle de posição de sistemas hidráulicos.



## ABSTRACT

The design activities for the mechatronic systems happen in different technological domains. The many specialists' involvement can happen depending on the complexity of the system, each one capable to solve technical aspects in your knowledge area. However, we understand that the integration of all these specialists is possible through an appropriate methodology for mechatronic systems that contemplates the process of technical systems design and software development as much as possible. In this context, this work presents the building of an application for a system of design and control of hydraulic positioning systems. For this, this software accomplishes the control of position of electro-hydraulic systems and also to implement a systematized diagram of procedures for the sizing and the analysis static and dynamic of hydraulic circuits. Because this application is included in an equipment that is characterized by being a mechatronic system, in this work is used an appropriate methodology for mechatronic systems with the intention of reaching the functions which the application to be developed should contemplate. Later, the implementation of these functions is accomplished following software engineering aspects. Like this, we can prove that this modelling methodology - which integrates graphic description techniques for mechatronic systems and tries to be neutral for possible implementations – is shown as an efficient tool in mechatronic systems design.

*Keywords:* mechatronic systems design, software development, hydraulic positioning systems.

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 Contextualização

Os mecanismos hidráulicos de controle de posição têm enorme aplicabilidade nos mais diversos campos da engenharia e o seu projeto adequado depende da correta aplicação de ferramentas de auxílio ao projeto estático e dinâmico.

Nesse contexto, este documento apresenta a elaboração de um aplicativo para um sistema de projeto e controle de posicionadores hidráulicos. Concorrentemente a esta ferramenta, um equipamento, denominado de Plataforma de Hidráulica Proporcional, visando intensificar a interação entre os conceitos teóricos e a aplicação prática destes está sendo desenvolvido no LASHIP – Laboratório de Sistemas Hidráulicos e Pneumáticos da UFSC.

#### 1.1.1 Plataforma de Hidráulica Proporcional

O objetivo do Projeto Plataforma de Hidráulica Proporcional é o desenvolvimento de um sistema didático de projeto e comprovação experimental que busca a sistematização do conhecimento de projeto de sistemas hidráulicos de controle de posição e a correlação entre modelagem matemática e verificação experimental do comportamento dinâmico de sistemas eletro-hidráulicos proporcionais.

Por isto, a Plataforma de Hidráulica Proporcional é formada por componentes usualmente utilizados no meio industrial e acompanhada da tecnologia característica dos sistemas de automação e controle que permite ao estudante a manipulação e a implementação de trabalhos técnicos e científicos.

De tal modo, o equipamento compõe-se de três módulos. O primeiro módulo (módulo 1) consiste de uma bancada de trabalho destinada à montagem e operação instrumentada de circuitos hidráulicos com ênfase em controle de posição, onde os componentes hidráulicos são controlados por um computador utilizando o software LabVIEW e um sistema VXI de aquisição de dados e controle, da National Instruments. O segundo módulo (módulo 2) consiste em um aplicativo de projeto do sistema hidráulico de controle de posição aplicável a circuitos compostos de válvula com controle contínuo direcional (VCCD) (servoválvula ou válvula proporcional), cilindro hidráulico simétrico ou assimétrico, sensor de posição e controlador. Empregando também o LabVIEW, ele deve implementar o método de projeto desenvolvido através de dissertação de mestrado concluída em 2001 no LASHIP (FURST, 2001). Finalmente, o terceiro módulo (módulo 3) consiste de um sistema hidráulico de geração de carga sobre o cilindro do circuito de controle de posição montado no módulo 1. Este sistema também será controlado através do sistema de aquisição de dados e controle citado

anteriormente e permitirá a geração de cargas de modo semelhante a sistemas reais, como em reguladores de velocidade de turbinas hidráulicas.

Especificamente, a Plataforma de Hidráulica Proporcional apresenta a estrutura geral mostrada na figura 1.1, compondo-se de um computador com software LabVIEW; um sistema de aquisição de dados e controle VXI; uma Unidade de Potência Hidráulica que inclui o circuito hidráulico propriamente dito; um CLP e um painel de interface com o usuário; e uma Bancada de Trabalho onde se realiza o posicionamento mecânico através do circuito de atuação hidráulico (válvula + cilindro). Assim sendo, como destacado na figura 1.1, designou-se como Módulo de Trabalho o agrupamento entre o Computador, o Sistema VXI e a Bancada de Trabalho. O Módulo de Trabalho está ilustrado na figura 1.3.

A aparência desta Bancada de Trabalho, a fim de que o leitor tenha uma idéia visual, é semelhante a DS3 Synergy ou DS3-G (mostradas na figura 1.2), da Bosch Rexroth. Como se pode observar, a Bancada de Trabalho possui dois lados, denominados “Lado 1” e “Lado 2”. Em cada um destes lados, pode-se montar um circuito de atuação hidráulico. No entanto, por questões de simplificação e de simetria do sistema como um todo, este trabalho referencia-se constantemente a apenas um circuito de atuação hidráulico relativo a um lado da bancada (Lado 1). Isto se deve ao fato de que o controle de posição do circuito de atuação hidráulico relacionado ao “Lado 2” é processado pelo computador do “Lado 1”, o qual está conectado ao sistema VXI, servindo o segundo computador apenas como um meio de interface com seu usuário<sup>1</sup>. Por isso, no decorrer deste trabalho referencia-se também somente ao computador do “Lado 1”, como se observa na figura 1.1.

De modo resumido, na Plataforma de Hidráulica Proporcional, um computador realiza o controle de posição do sistema de atuação hidráulico (válvula + cilindro) como ilustra a figura 1.3. Este sistema está localizado sobre a Bancada de Trabalho como indicado na figura 1.2. Para tal, um sistema VXI de aquisição de dados e controle desempenha a comunicação entre o computador e a Bancada de Trabalho como se pode ver na figura 1.1.

Para que possa realizar a aquisição de dados e controle, então, o sistema VXI possui os subseqüentes instrumentos:

- módulo de entradas FET multiplexado 16 canais – MOD. E1351 A – Hewlett Packard;
- módulo de saídas digitais 32 canais – MOD. 1463 A – Hewlett Packard;
- conversor D/A de 4 canais – MOD E1328 A – Hewlett Packard;
- multímetro 5 1/2 dígitos – MOD. E1411 B – Hewlett Packard.

---

<sup>1</sup> Os dados referentes ao controle de posição do circuito de atuação hidráulico relacionado ao “Lado 2” são trocados, por rede, entre os computadores com a utilização da tecnologia apresentada no apêndice 2.

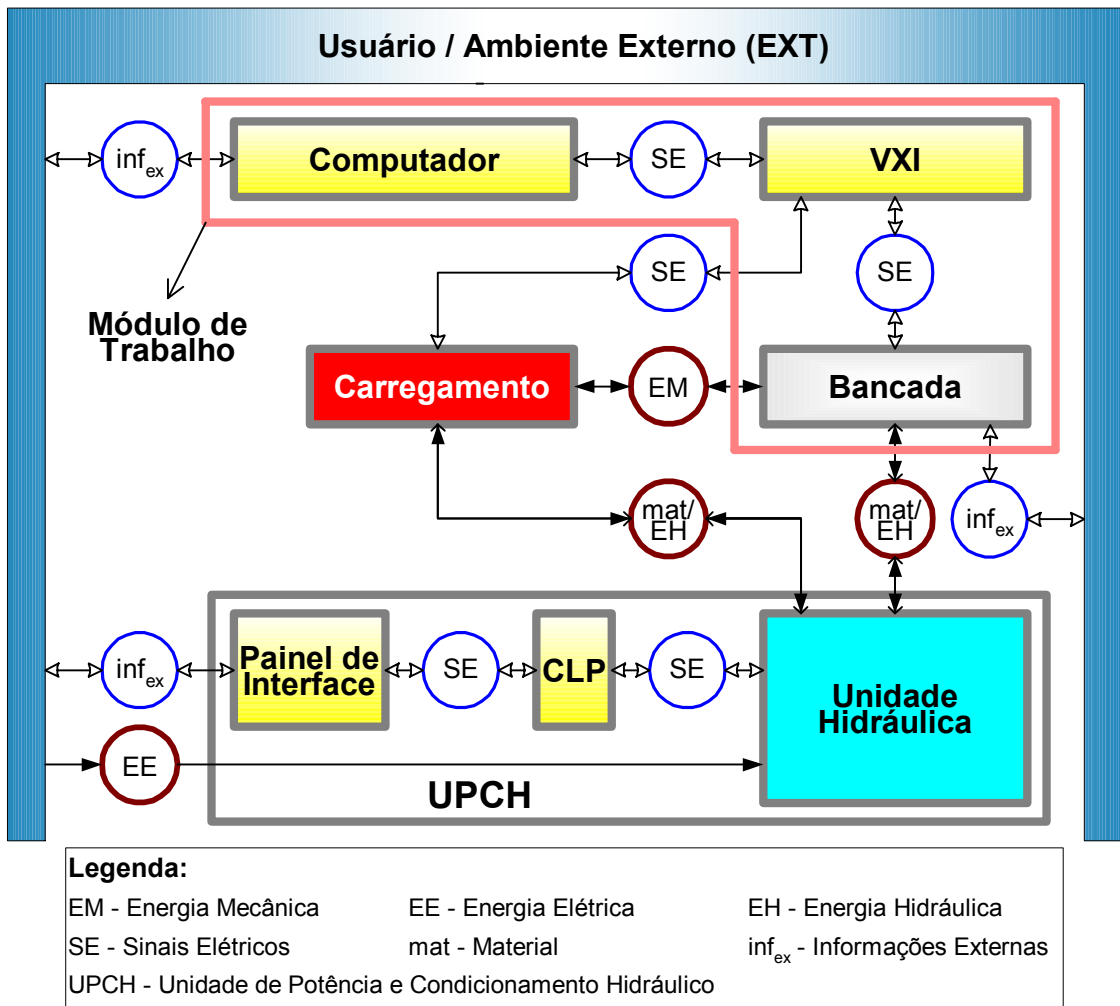


Figura 1.1 – Diagrama da estrutura da Plataforma de Hidráulica Proporcional.



Figura 1.2 – a) DS3 Synergy; b) DS3 G. (BOSCH REXROTH GROUP, 2002a; 2002b).

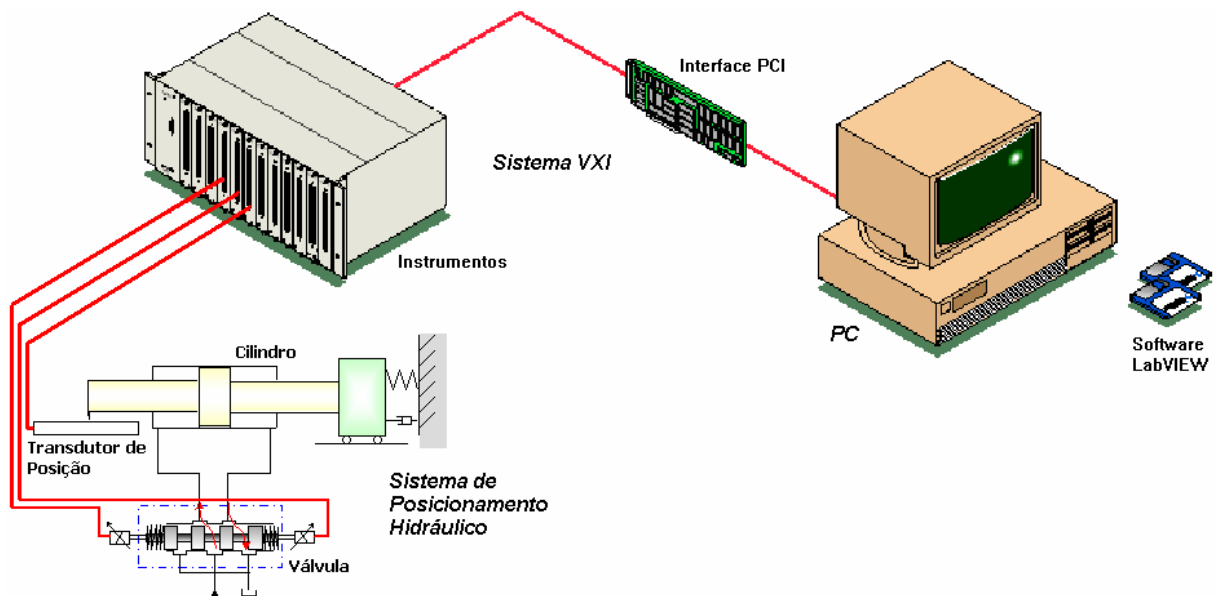


Figura 1.3 – Ilustração do Módulo de Trabalho da Plataforma de Hidráulica Proporcional.

Na figura 1.4 é mostrado o equipamento já construído no LASHIP para o controle de posição de sistemas hidráulicos e que é parte integrante do Módulo de Trabalho. Têm-se dois computadores, cada qual necessário para que os usuários possam realizar os experimentos da Bancada de Trabalho<sup>2</sup>, sobre um *Rack*. Por isso, definiu-se a notação de “Usuário 1” para o (s) usuário (s) do “Lado 1” da bancada e “Usuário 2” para o (s) usuário (s) do “Lado 2” da bancada.

Ainda, na figura 1.5 é detalhada a bancada que aparece no lado esquerdo da figura 1.4. Esta é um Módulo Demonstrativo para exposição em feiras, sendo que a Bancada de Trabalho para utilização em laboratório (semelhante às mostradas na figura 1.2) está sendo importada pela empresa Bosch Rexroth conforme o projeto da equipe do Projeto Plataforma de Hidráulica Proporcional (LASHIP).

---

<sup>2</sup> Por haver somente um sistema de atuação hidráulico, montado sobre uma bancada para uso exclusivo no Projeto Plataforma de Hidráulica Proporcional, apenas um “Usuário” pode controlá-lo de cada vez. No entanto, o “Usuário” que não o estiver controlando pode controlar, ao mesmo tempo, qualquer outro sistema disponível (dos vários existentes) no LASHIP.



Figura 1.4 – Equipamento utilizado atualmente no LASHIP para o controle de posição de sistemas hidráulicos.



Figura 1.5 – Atual sistema de atuação hidráulico da Bancada de Trabalho.

#### 1.1.1.1 Sistematização de Procedimentos para Dimensionamento e Análise Estática e Dinâmica de Circuitos Hidráulicos

O objetivo da sistematização de procedimentos para dimensionamento e análise estática e dinâmica de circuitos hidráulicos, desenvolvida através da dissertação de mestrado de FURST (2001), é a obtenção da definição da válvula proporcional ou servoválvula conforme catálogo industrial e o dimensionamento do cilindro como mostrado na figura 1.6.

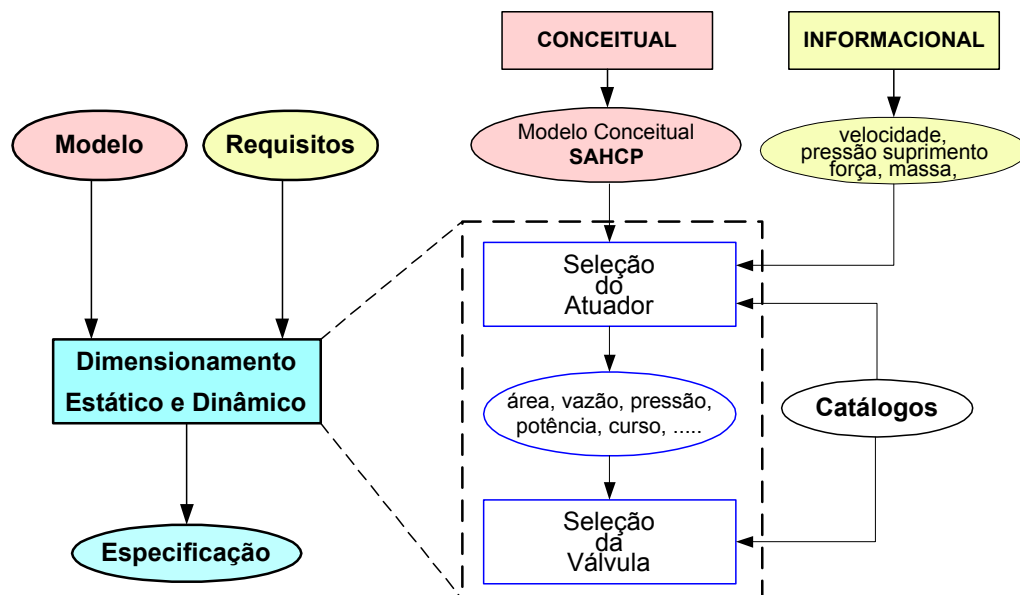


Figura 1.6 – Tarefas do dimensionamento estático e dinâmico para a seleção da válvula e cilindro (FURST, 2001).

Uma vez que cada atividade deste dimensionamento decompõe-se em diversos passos e tomadas de decisão, é notável a importância do desenvolvimento de um aplicativo para que o projetista, através do diagrama de FURST (2001) possa dimensionar um sistema de controle de posição hidráulico e, em seguida, comprovar experimentalmente, realizando o controle de posição na Bancada de Trabalho por este mesmo aplicativo, tal dimensionamento.

Portanto, ao implementar a sistematização de procedimentos (FURST, 2001) incorporando-a junto ao controle de posição de sistemas hidráulicos, o aplicativo se mostra como uma ferramenta completa para um sistema de projeto e controle de posicionadores hidráulicos.

Para tanto, o processo de projeto do aplicativo utiliza uma metodologia de modelagem de sistemas automáticos que busca, a partir da identificação das partes que compõem o Módulo de Trabalho, alcançar (através dos refinamentos de modelos) as funções necessárias para a construção ou implementação do aplicativo segundo as metodologias da engenharia de software delineadas no capítulo 4.

## 1.2 Objetivos e metodologia

O presente trabalho de dissertação visa, principalmente, a elaboração de um aplicativo que se enquadra nos objetivos do projeto da Plataforma de Hidráulica Proporcional, compreendendo um sistema para dimensionamento (referente ao módulo 2) e controle (referente ao módulo 1) de posicionadores hidráulicos. De tal modo, o aplicativo deve desempenhar as seguintes tarefas:



- realização do controle de posição de sistemas eletro-hidráulicos pertinente ao módulo 2;
- implementação do diagrama sistematizado de procedimentos para dimensionamento e análise estática e dinâmica de circuitos hidráulicos pertinente ao módulo 1.

Ciente de que o equipamento Plataforma de Hidráulica Proporcional caracteriza-se por ser um sistema automático, dentre os objetivos deste trabalho igualmente estão:

- a avaliação e/ou aprimoramento de princípios de modelagem e projeto de sistemas automáticos;
- a análise da possibilidade de associação de modelos de sistemas automáticos às soluções técnicas implementadas de acordo com abordagens em engenharia de software.

Desta forma, é apresentada neste documento a utilização de uma metodologia própria para sistemas automáticos (DE NEGRI, 1996) com o intuito de se alcançar as funções as quais o aplicativo deve contemplar. Posteriormente, a implementação destas funções é realizada seguindo-se aspectos da engenharia de software.

Assim, pode-se comprovar que essa metodologia de modelagem, a qual integra técnicas de descrição gráficas para sistemas automáticos e busca ser independente das possíveis implementações, mostra-se uma ferramenta eficiente no projeto de sistemas automáticos auxiliando na consolidação do aplicativo almejado.

Finalmente, visto a pretensão de sua fácil manutenção, além da entrega e de seu uso, segue-se atentamente aspectos relacionados ao desenvolvimento de software de qualidade para o emprego do aplicativo em linguagem LabVIEW.

### **1.3 Justificativas**

As atividades de projeto para os sistemas automáticos ocorrem em diferentes domínios tecnológicos e, dependendo da amplitude do sistema, pode haver o envolvimento de diversos especialistas, cada qual capaz de resolver aspectos técnicos em sua área do conhecimento.

No entanto, entende-se que a integração de todos os especialistas é possível por meio de uma metodologia própria para sistemas automáticos que contemple, dentro do possível, o processo de projeto de sistemas técnicos e o desenvolvimento de software. Por meio da utilização de tal metodologia no desenvolvimento deste trabalho, pode-se verificar a integração dos diferentes domínios tecnológicos que este envolve visto que o aplicativo insere-se em um equipamento (Plataforma de Hidráulica Proporcional) caracterizado por ser um sistema automático.

## 1.4 Estrutura deste documento

No capítulo 2 descreve-se a aplicabilidade e a configuração básica dos sistemas eletro-hidráulicos de controle de posição visto que o aplicativo desenvolvido deve, inclusive, realizar tal tipo de controle.

No capítulo 3, sistemas de aquisição de dados e controle são apresentados devido à sua necessidade para que se desempenhe o controle de posição dos sistemas eletro-hidráulicos. Assim, diferentes tecnologias de construção de controladores são exibidas a fim de que, posteriormente, se delinieie as vantagens e restrições da utilização do Sistema VXI aliada ao software LabVIEW.

No capítulo 4, assuntos como modelos de ciclo de vida, qualidade no processo de desenvolvimento, técnicas de prototipagem, documentação e estilo são abordados no que tangem o desenvolvimento de software voltado ao LabVIEW.

No capítulo 5, visto que o aplicativo a ser desenvolvido está incluso em um equipamento (Plataforma de Hidráulica Proporcional) que se caracteriza por ser um sistema automático, apresentam-se aspectos referentes à conceituação e modelagem de sistemas automáticos

No capítulo 6, a modelagem do Módulo de Trabalho, o qual é parte da Plataforma de Hidráulica Proporcional, é realizada. Segue-se, então, com a modelagem funcional/estrutural do aplicativo

No capítulo 7, explana-se como se deu o desenvolvimento do aplicativo após a modelagem desempenhada no capítulo 6. Assim, é mostrado como se realizaram a análise de requisitos, a prototipação, o processo de projeto, quais os aspectos fundamentais considerados no projeto, a hierarquia de funções do aplicativo, a documentação apresentada e os testes de validação executados. Ao final, realiza-se uma discussão sobre os resultados alcançados.

No capítulo 8, são apresentadas as conclusões e recomendações para trabalhos futuros.

## CAPÍTULO 2

### SISTEMAS ELETRO-HIDRÁULICOS DE CONTROLE DE POSIÇÃO

Os mecanismos de controle de posição têm enorme aplicabilidade nos mais diversos campos da engenharia. Como exemplos, a figura 2.1 e a figura 2.2 ilustram aplicações móveis enquanto a figura 2.3 ilustra uma aplicação estacionária.



Figura 2.1 – Aviação: posicionamento de leme (EMBRAER, 2004).



Figura 2.2 – Colheitadeiras: controle de posição da plataforma (JOHN DEERE, 2005).

Os mecanismos de natureza eletro-hidráulica são comandados por sinais de baixo consumo de energia e controlam com precisão os movimentos de cilindros e motores hidráulicos de mecânica pesada. Ainda, possuem enorme versatilidade devido a uma variedade

de de configurações, sobretudo no processamento de sinais que podem ser diferentemente projetados para a obtenção da atuação hidráulica desejada.

Independentemente do campo de aplicação, conforme SOUZA *et al.* (2004a), todos os sistemas de posicionamento hidráulicos possuem uma configuração básica padrão que utiliza servoválvulas ou válvulas direcionais proporcionais acionando cilindros hidráulicos, conforme abordado na seção seguinte.

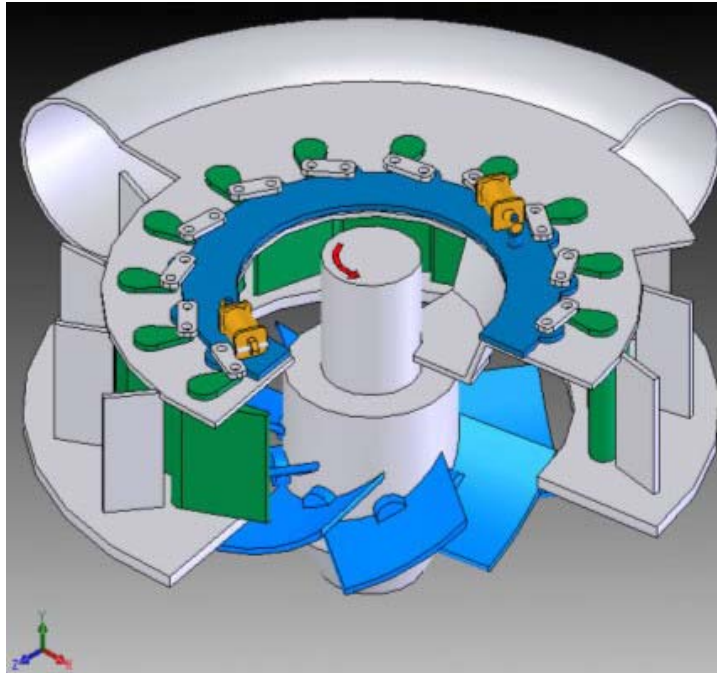


Figura 2.3 – Centrais hidrelétricas: posicionamento de pás.

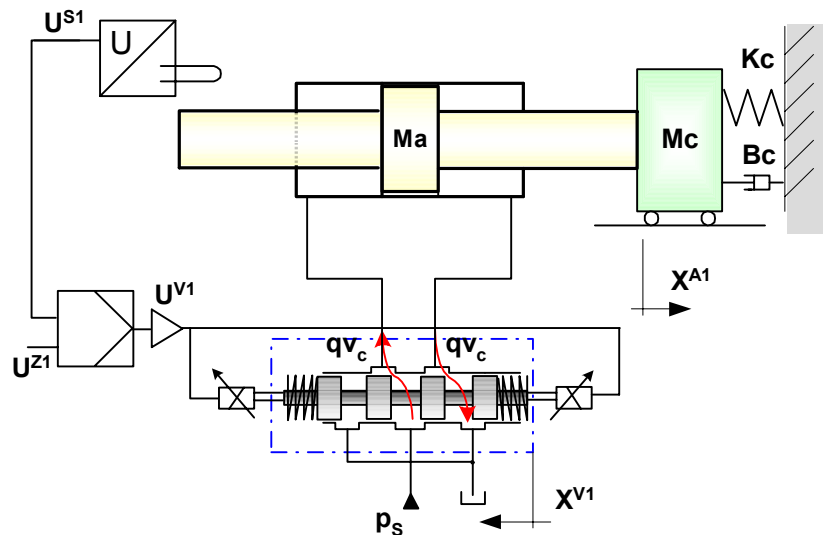
## 2.1 Apresentação

Na figura 2.4 é apresentado um posicionador eletro-hidráulico cuja finalidade é deslocar a massa ( $M$ ) a uma distância proporcional a um sinal em tensão de referência ( $U^{Z1}$ ). Para tanto, as partes que compõem este sistema de controle de posição são:

- válvula direcional proporcional ou servoválvula ( $V1$ );
- cilindro hidráulico e carga ( $A1$ );
- controlador ( $Z1$ ) e sensor de posição ( $S1$ ).

Neste sistema, de acordo com DE NEGRI (2001), a posição desejada para o cilindro é estabelecida pela tensão de referência ( $U_r$ ) que, através do controlador, gera uma tensão de comando ( $U^{V1}$ ) na válvula direcional continuamente variável (VDCV) (servoválvula ou válvula direcional proporcional), produzindo o deslocamento do elemento de controle principal (normalmente um carretel). Considerando que a pressão de suprimento da válvula seja mantida constante, o deslocamento de carretel principal da válvula provocará uma vazão no sentido da via de suprimento para uma das câmaras do cilindro enquanto que ocorrerá uma vazão da outra câmara para a linha de retorno da válvula. A vazão através da válvula pro-

move a variação da pressão nas câmaras do cilindro resultando na movimentação da massa ( $M$ ) que é medida através do sensor de posição ( $S1$ ) produzindo uma tensão  $U^{S1}$ . Esta tensão  $U^{S1}$ , de sinal contrário à tensão de referência  $U_r$ , produz a realimentação de posição. Em uma condição ótima, quando a posição desejada for atingida, a tensão de comando da válvula ( $U^{V1}$ ) estará anulada implicando que a posição real do êmbolo ( $x^{A1}$ ) corresponda exatamente à posição desejada.



**Legenda:**

$U^{S1}$ : tensão relacionada ao sensor de posição;  
 $U^{V1}$ : tensão de comando relacionada à válvula;  
 $U^{Z1} = U_r$ : tensão de referência;  
 $x^{A1}$ : posição real do êmbolo do cilindro;  
 $x^{V1}$ : posição do carretel da válvula;  
 $q_{v_c}$ : vazão de controle;  
 $p_s$ : pressão de suprimento;  
 $M_a$ : massa do cilindro;  
 $M_c$ : massa da carga;  
 $M = (M_a + M_c)$ : massa total;  
 $K_c$ : constante elástica da mola;  
 $B_c$ : coeficiente de atrito viscoso.

Figura 2.4 – Ilustração de um sistema hidráulico de controle de posição.

Estando adequadamente dimensionado, o sistema estabiliza rapidamente na posição desejada. Se o sinal de entrada variar com o tempo, o sistema atuará como um seguidor, de modo que a posição da haste estará variando no tempo, acompanhando o comando.

Por outro lado, a seleção inadequada do ganho de realimentação (influenciado pelo sensor S1 e pelo condicionador do sinal correspondente) e do ganho do amplificador pode também gerar instabilidade do sistema fazendo com que a massa oscile com intensidade crescente, afastando-se cada vez mais da posição desejada.

Dessa forma, a verificação da necessidade do dimensionamento adequado da válvula e do cilindro, para que se atendam às especificações de uma dada aplicação, reitera a

importância de se adotar uma sistematização de procedimentos como proposto por FURST (2001).

Ao mesmo tempo, é preciso que se tenha o conhecimento sobre os tipos de controladores automáticos bem como os parâmetros de controle a serem utilizados, para se alcançar o desempenho desejado tanto do ponto de vista estático quanto dinâmico, em um dado posicionador hidráulico. Por isso, o item seguinte contextualiza a utilização do controle por computador a fim de se reafirmar <sup>3</sup> o seu uso na Plataforma de Hidráulica Proporcional. Também, é justificado o emprego da estrutura de controle PID para o posicionamento hidráulico.

## 2.2 Controladores Automáticos

Um controlador automático tem como função produzir um sinal de controle que anule o erro (desvio), ou o reduza a um valor muito pequeno. O controlador compara o valor real da saída do processo com o valor desejado (*set-point*), determina o erro ou desvio, e produz o respectivo sinal de comando para o atuador.<sup>4</sup>

Na construção de posicionadores hidráulicos é possível encontrar soluções mecânico-hidráulicas e eletro-hidráulicas. As primeiras empregam válvulas de controle contínuo acionadas mecanicamente cujo sinal provém de mecanismos que capturam a posição do atuador e o sinal de referência. Por sua vez, os sistemas eletro-hidráulicos possuem a configuração mostrada na figura 2.4, sendo o controle implementado através de eletrônica analógica ou digital.

Ainda que hoje exista a construção de posicionadores mecânico-hidráulicos, a grande maioria das aplicações requer o emprego de processamento eletrônico e, em vista disto, detalha-se na próxima seção aspectos dos controladores eletrônicos.

### 2.2.1 Controladores Eletrônicos

Com o avanço da tecnologia eletrônica, em uma primeira fase analógica e mais recentemente digital, esta tecnologia ganhou tal dimensão que hoje em dia a grande maioria das aplicações recorre unicamente a controladores eletrônicos do tipo analógico e/ou digital (BAPTISTA, 2004).

De tal modo, os analógicos têm as seguintes características:

---

<sup>3</sup> A abordagem que segue apenas reafirma o uso do computador na Plataforma de Hidráulica Proporcional pelo fato de que este equipamento deve ser empregado, independentemente de como é realizado o controle, devido ao fato do presente trabalho visar a elaboração de um Aplicativo como apresentado na Introdução.

<sup>4</sup> O controle de posição do sistema hidráulico ilustrado na figura 2.4 exemplifica o funcionamento de um controlador.

- utilizam componentes como capacitores, resistores e amplificadores operacionais;
- são caros de construir;
- implementam algoritmos de controle limitados, como PID;
- seus componentes sofrem com o envelhecimento e a temperatura;
- difíceis de se atualizar, modificar e integrar.

De outra maneira, os controladores digitais possuem as características a seguir:

- empregam hardware computacional como processador, conversor A/D e conversor D/A;
- são de construção barata;
- implementam algoritmos de controle complexos como controle ótimo e controle adaptativo;
- seus componentes não sofrem com o envelhecimento e a temperatura;
- flexíveis, controláveis via software, de fácil modificação e atualização.

Para tanto, há muitas escolhas para o hardware computacional dos sistemas de controle digitais. Dentre estas, os **microcomputadores** apresentam as seguintes características:

- consistem basicamente de um processador, memória, portas e barramento;
- necessitam de dispositivos periféricos como conversores A/D e D/A, placa de E/S digital, entre outros;
- necessitam de dispositivos de armazenamento como HDD e FDD;<sup>5</sup>
- alto consumo de energia;
- normalmente satisfatório para aplicações de mesa (*desktop applications*).

No entanto, o controle por computador apresenta algumas vantagens, diante de outros controladores digitais, como (COELHO *et al.*, 2001):

- flexibilidade na implementação do controlador dinâmico na malha de realimentação (basta reprogramar o computador);
- facilidade para implementar controladores complexos (ótimo, preditivo, nebuloso, neural, adaptativo);
- baixo custo e alta confiabilidade na realização das várias malhas de um processo complexo;
- facilidade para incluir no computador as funções de alarme, de comando para partida e para desligamento do processo, bem como a função de supervisão global de processos complexos;
- elevada imunidade a ruídos.

Em contrapartida, de acordo com COELHO *et al.* (2001), algumas desvantagens são:

---

<sup>5</sup> HDD (*Hard Disk Drive*): drive de disco rígido; FDD (*Floppy Disk Drive*): drive de disquetes.

- custo elevado, especialmente das placas de aquisição de dados (conversores A/D e D/A, portas de E/S, interrupção), no caso do controle de pequenos sistemas SISO<sup>6</sup> (monovariáveis);
- análise e projeto complexos;
- perigos inerentes à engenharia de software em tempo real (tarefas concorrentes).

### 2.2.2 O Controlador PID

Conforme RICO (2000a), o controlador PID (Proporcional, Integral, Derivativo) é o tipo de controlador de estrutura fixa mais utilizado nas aplicações industriais, tanto no Brasil como no mundo. Trabalhos de pesquisa bem recentes, segundo RICO (2000a), mostram que mais de 95% das malhas de controle industriais são deste tipo e que as estruturas de controle distribuído mais complexas têm o PID como elemento básico.

Em geral, pode-se dizer que para processos com dinâmicas bem comportadas e especificações não muito exigentes, como o posicionamento hidráulico nas áreas de reguladores de velocidade, beneficiamento de madeira, máquinas agrícolas, siderurgia entre outras, o PID é uma solução que oferece um bom compromisso entre simplicidade e bom comportamento em malha fechada.

A ação integrativa (I) do PID, de acordo com RICO (2000a), garante que o sistema em malha fechada possa trabalhar sem erro em regime permanente para referências ou perturbações constantes. De outro modo, a ação derivativa (D) tem características preditivas, importantes para corrigir as dinâmicas transitórias.

---

<sup>6</sup> SISO: Sistemas com uma entrada e uma saída.



## CAPÍTULO 3

### SISTEMAS DE AQUISIÇÃO DE DADOS E CONTROLE

Para que se desempenhe o controle de posição dos sistemas eletro-hidráulicos por computador, conforme explanado no capítulo 2, faz-se necessária a utilização de sistemas de aquisição de dados e controle (SADC).

De tal modo, o diagrama de blocos simplificado de um sistema de controle por computador está mostrado na figura 3.1, onde o computador é o responsável pela manipulação dos dados provenientes do ambiente externo que se deseja atuar ou controlar.<sup>7</sup>

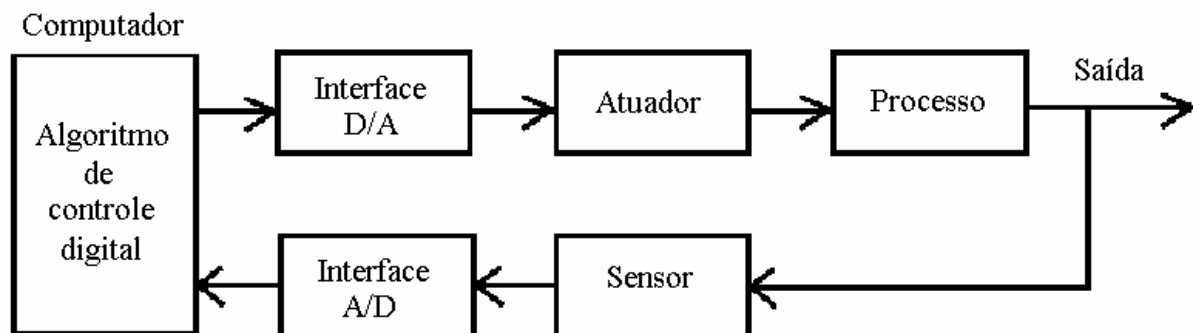


Figura 3.1 – Hardware de um sistema de controle por computador (adaptado de COELHO *et al.*, 2001).

Contudo, para que um dado possa ser manipulado pelo computador, esse dado deve ser primeiro colocado na forma digital. Para isso, existem diferentes hardwares para aquisição de dados disponíveis no mercado. Dentre esses, o sistema VXIbus aliado ao Software LabVIEW mostra-se uma ferramenta poderosa e muito utilizada na aquisição e controle de dados.

### 3.1 Sistema VXIbus

#### 3.1.1 Apresentação

O sistema VXIbus, elaborado pela National Instruments, consiste em uma plataforma para implementação de sistemas de instrumentação. Sua arquitetura é aberta, o que possibilita o aproveitamento das últimas tecnologias em termos de computação. Atualmente existe uma variada gama de produtos, em nível comercial, compatíveis com o padrão VXI.

De acordo com WOLFE (1996), devido a sua versatilidade, um único sistema VXI pode suportar diversos tipos de instrumentos, possibilitando seu uso em diversas aplica-

---

<sup>7</sup> Note, o leitor, que a função desempenhada pelo elemento *Controlador (Z1)* apresentado no item 2.1 é agora desempenhada através de um *computador*.

ções, desde a produção e testes de manufatura até o monitoramento de processos e controle. Isto é possível graças às diversas opções de condicionamento de sinais, que suportam um amplo conjunto de transdutores.

### 3.1.2 Maneiras de Controlar um Sistema VXI

Segundo NATIONAL INSTRUMENTS (1999), existem várias maneiras de se utilizar um Sistema VXI. Conforme se observa na figura 3.2, pode-se construir um sistema utilizando-se apenas instrumentos VXI ou se pode integrar um VXI a um sistema que utiliza instrumentos GPIB ou placas de aquisição. Cada uma das configurações tem os seus benefícios.

A primeira configuração, como apontada na figura 3.2, envolve um computador VXI embutido ao *mainframe*<sup>8</sup>, se conectando diretamente à *backplane*<sup>9</sup> do VXI. Essa configuração oferece o menor tamanho físico para um sistema VXI como também beneficia o desempenho devido à conexão direta à *backplane* do VXI.

A segunda configuração possui uma alta velocidade onde o MXIbus une um computador externo à *backplane* para controlar o VXI. O computador externo opera como se estivesse diretamente embutido dentro do *mainframe* do VXI. Esta configuração é funcionalmente equivalente ao método embutido, a não ser pela flexibilidade e pelo uso de uma vasta variedade de computadores e *workstations*<sup>10</sup>. Ressalta-se ainda, que esta é a configuração utilizada na Plataforma de Hidráulica Proporcional.

A terceira maneira utiliza barramento serial de baixo custo IEEE 1394 (ou FireWire<sup>11</sup>) para controlar um sistema VXI.

A quarta consiste em um *mainframe* VXI unido a um controlador externo pelo GPIB. O controlador comunica-se pelo GPIB com um módulo de interface GPIB-VXI instalado dentro do *mainframe* VXI. Por sua vez, a interface GPIB-VXI transparentemente desempenha a tradução entre o protocolo GPIB e o protocolo serial de palavra do VXI.

---

<sup>8</sup> *Mainframe*: Equipamento multiusuário, grande e rápido, projetado para gerenciar grandes quantidades de dados e tarefas complexas de processamento.

<sup>9</sup> *Backplane*: Uma placa ou estrutura de circuitos que suporta outras placas de circuitos, dispositivos e as interconexões entre os dispositivos, e fornece força e sinais de dados aos dispositivos suportados.

<sup>10</sup> *Workstation* (estação de trabalho): Um computador independente, com grande capacidade de processamento, do tipo usado em aplicações de projeto auxiliado por computador e outras que exijam máquinas de topo de linha, normalmente muito caras, com amplos recursos matemáticos ou gráficos.

<sup>11</sup> *FireWire*: O FireWire, ou IEEE 1394 Serial Bus, é um padrão para conexão de periféricos que tem o objetivo de ligar vários equipamentos, sem a necessidade de novas placas e tão pouco reconfigurar o PC.

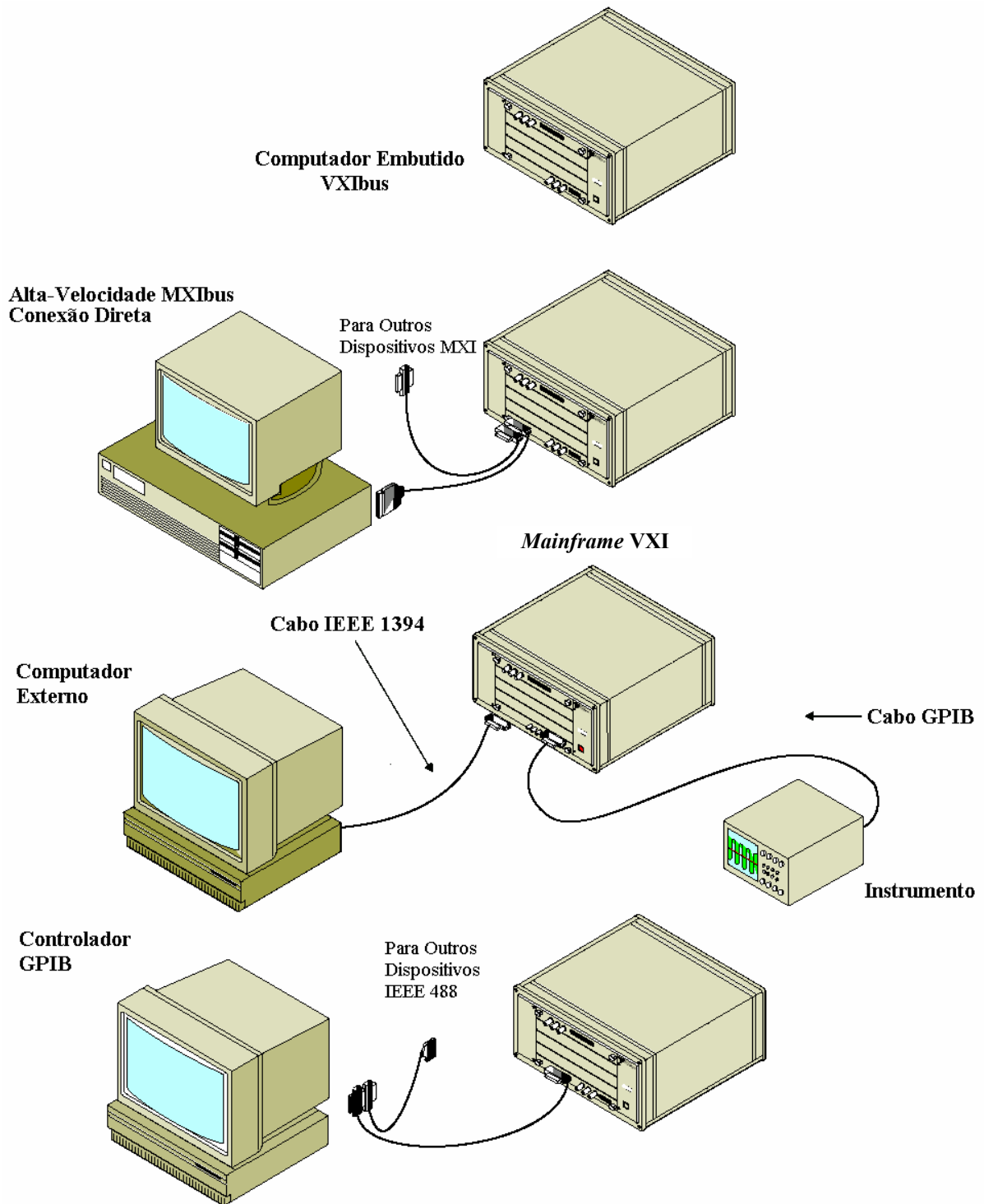


Figura 3.2 – Maneiras de controlar um sistema VXI (NATIONAL INSTRUMENTS CORPORATION, 2003).

Porém, conforme AGILENT TECHNOLOGIES (2001), a escolha<sup>12</sup> entre os sistemas mostrados acima depende do desempenho que se deseja para o sistema, o que conseqüen-

<sup>12</sup> Conforme especificações do Projeto Plataforma de Hidráulica Proporcional, a segunda configuração, na qual utiliza-se o cabo MXIbus, deve ser utilizada neste trabalho.

temente implica custos. Na figura 3.3, compara-se os diferentes sistemas em um gráfico de performance versus custos.

De acordo com NATIONAL INSTRUMENTS (2003a), a performance de um sistema inclui fatores como:

- tempo gasto por aplicações em computar, mostrar e executar E/S de disco;
- executar E/S sobre o VXIbus ou VMEbus;
- taxas de transferência de dados dos instrumentos;
- sincronização de operações entre os dispositivos que executam transações pelo MXIbus;
- capacidade de integração de produtos a um sistema controlado por MXI.



Figura 3.3 – Desempenho versus custos dos sistemas VXI (AGILENT TECHNOLOGIES, 2001).

### 3.1.3 Aquisição de Dados com VXI

Como assegura NATIONAL INSTRUMENTS (2000), pode-se empregar instrumentos para aquisição de dados VXI para várias aplicações, como aquisição de ondas, geração de ondas, interfaceamento digital, geração de pulso, medições de tensão estáticas e dinâmicas, análise transiente, *data logging*<sup>13</sup> e contador de frequência.

Devido à sua versatilidade e capacidades multifuncionais um único módulo de instrumento VXI pode substituir vários instrumentos em um sistema como entradas e saídas analógicas, entradas e saídas digitais, contadores e temporizadores. Pode-se usar estes produtos em muitas áreas de aplicação que variam de teste de produção e manufatura a processo de monitoração e controle e aquisição de dados, com várias opções para condi-

<sup>13</sup> *Data logging device*: Dispositivo de registro de dados.

cionar sinais de uma vasta gama de transdutores como termopares, termorresistores (RTDs)<sup>14</sup>, termistores e *strain gauges*<sup>15</sup>.

### 3.2 Software LabVIEW

O LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) foi desenvolvido pela National Instruments Corporation. Em NATIONAL INSTRUMENTS (2003b) afirma-se que este é uma ferramenta de programação para instrumentação e análise com enormes potencialidades, o qual pode ser executado sobre as mais populares plataformas (Microsoft Windows, Sun, Apple Macintosh, Workstations HP-UX etc.).

Este software parte da natureza seqüencial das linguagens de programação tradicionais e introduz um ambiente de programação gráfico, bem como todas as ferramentas necessárias para aquisição de dados, análise e apresentação. Com esta linguagem de programação gráfica, que foi designada “G”, a programação é desenvolvida dentro de um diagrama de blocos (notação normalmente usada por cientistas e engenheiros). Criado o diagrama de blocos, o LabVIEW irá proceder a compilação para código máquina.

De acordo com CRISTALDI *et al.* (1999), os programas no LabVIEW são chamados de *Virtual Instruments (VIs)* porque a aparência e as operações simulam instrumentos reais. Cada VI tem duas janelas separadas, mas relacionadas entre si, que são o Painel Frontal (*Front Panel*) e o Diagrama de Blocos (*Block Diagram*).

O Painel Frontal é uma interface interativa entre o usuário e o programa (aplicativo). É onde o usuário entra com os dados, usando o mouse ou o teclado, e então visualiza os resultados na tela do computador. Ou seja, o Painel Frontal é uma janela de execução de um programa como se pode observar na figura 3.4.

De outro modo, o Diagrama de Blocos, ilustrado na figura 3.5, é a representação de um programa ou algoritmo. É onde o programador cria o seu programa.

Por sua vez, há ainda o que se chama *subVI*. Um subVI, como o nome sugere, nada mais é que um subprograma ou um módulo de um VI de mais alto nível. É a utilização de um VI (subVI) dentro de outro VI.

---

<sup>14</sup> *RTD (Resistance Temperature Detector)*: A base do funcionamento é o conhecido fenômeno da variação da resistência elétrica dos metais com a temperatura. Os metais mais usados são platina, níquel, cobre, ferro, molibdênio e/ou ligas dos mesmos.

<sup>15</sup> *Strain gauges* (células extensométricas): Sendo as células extensométricas, transdutores normalmente conectados eletricamente a um circuito tipo *Ponte de Wheatstone*, a saída de sinal elétrico está associada à variação da resistência elétrica do *strain gauge* (FLESCH, 1997). Assim, através deste transdutor pode-se medir deslocamento, força, torque, aceleração, temperatura, vazão, pressão, etc.

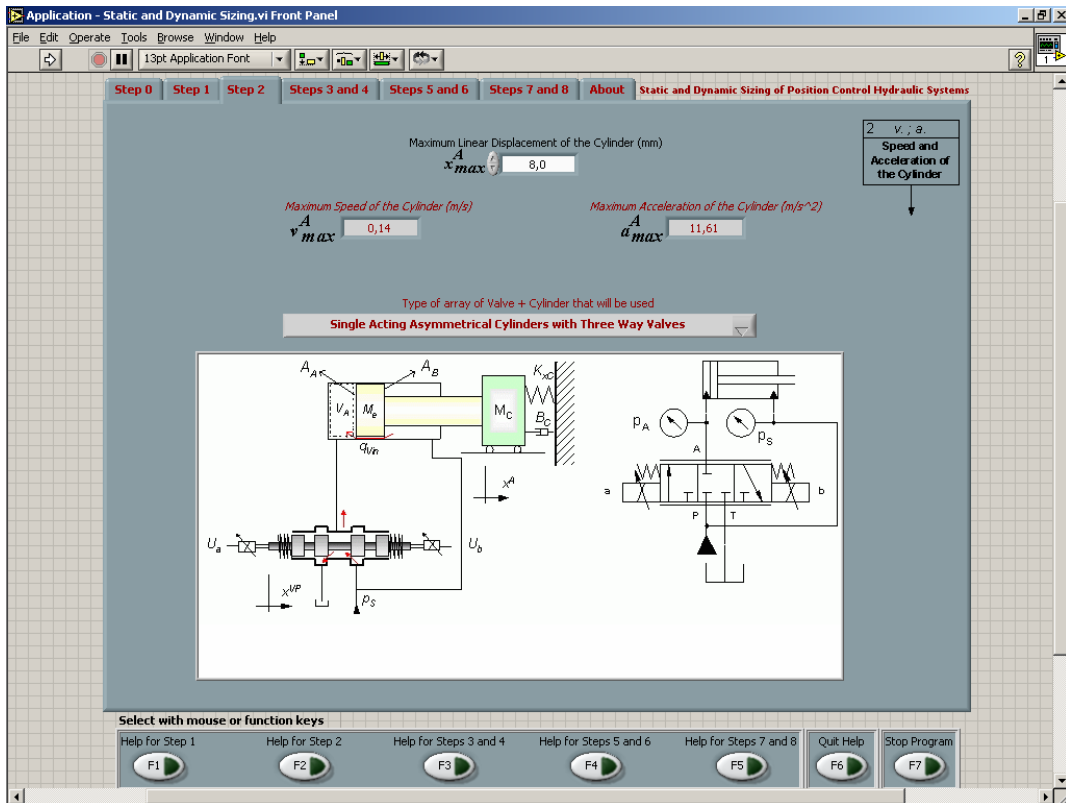


Figura 3.4 – Painel frontal de um programa.

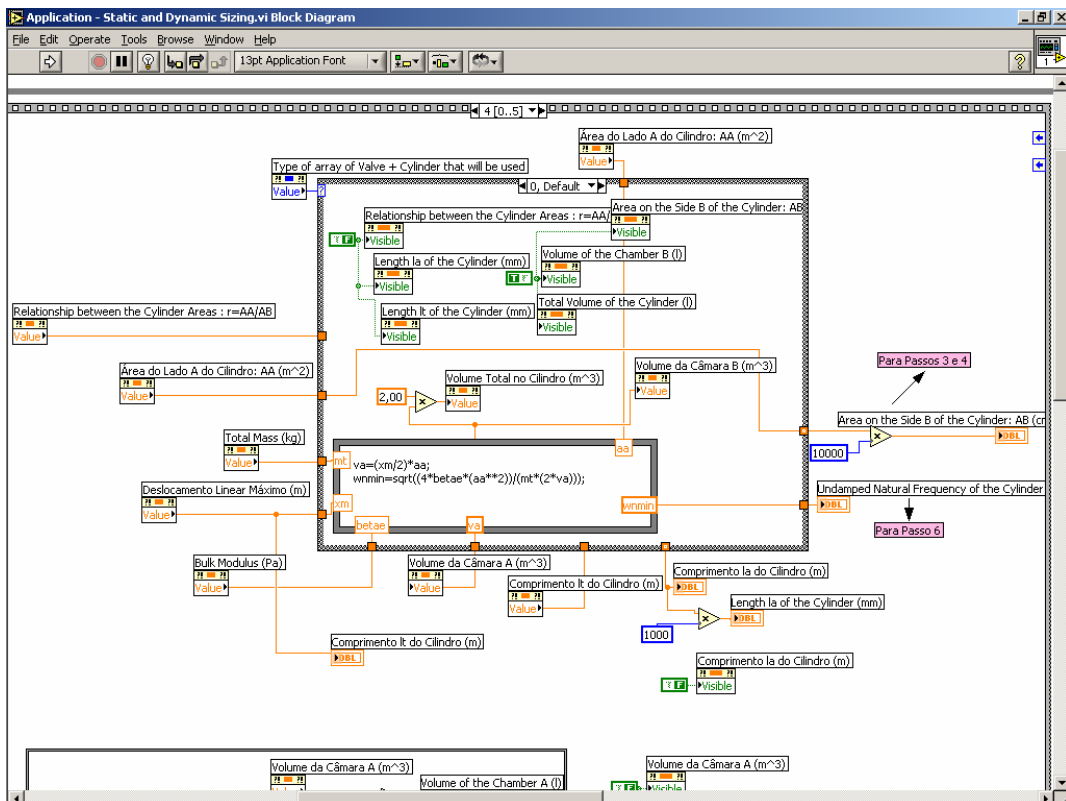


Figura 3.5 – Diagrama de blocos de um programa.

Ainda, o LabVIEW consegue integrar a aquisição de dados, a análise e a apresentação em um só sistema. Para a aquisição de dados, este suporta diversos padrões: RS-232/422, IEEE-488 (GPIB), e VXI, incluindo funções VISA (*Virtual Instrument Software Archi-*

*ecture*) bem como placas *plug-in* (de encaixe) para aquisição de dados DAQ (*Data Acquisition*).

Para a análise de dados, esse software inclui uma extensa biblioteca de análise, contendo funções para geração de sinais, processamento de sinal como filtros, janelas, estatística, regressão linear e aritmética matricial.

Dado que o LabVIEW é, por natureza, uma aplicação gráfica, está inerente a esse um pacote de apresentação de dados. Sendo assim, ele consegue gerar diagramas, bem como gráficos definidos pelo usuário.

### 3.3 Acesso aos Instrumentos VXI no LabVIEW

Uma biblioteca com centenas de instrumentos simplifica as aplicações de controle destes. Utiliza-se as VIs e funções de *Instrument I/O*, acessadas a partir da paleta de funções do painel frontal do LabVIEW (figura 3.6), para a interface com os dispositivos (instrumentos) VXI.

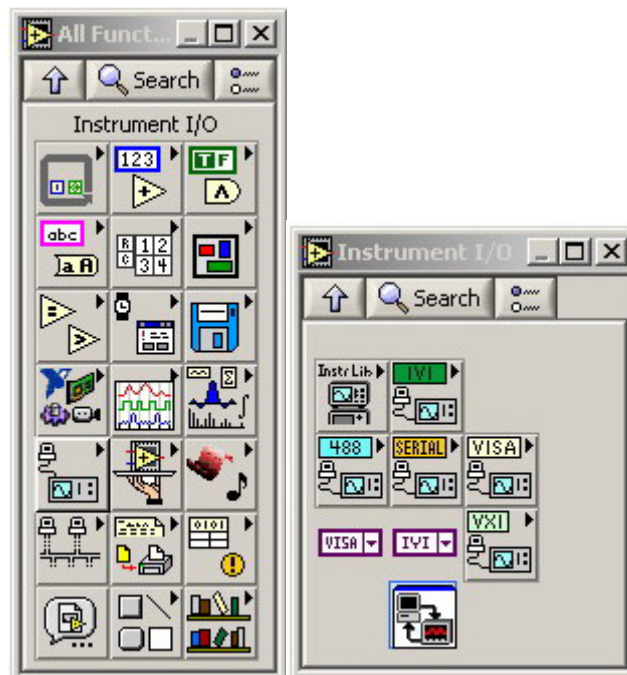


Figura 3.6 – Acesso às VIs e funções de *Instrument I/O* e VXI.

Assim sendo, o sistema VXI e o software LabVIEW se comunicam de modo a possibilitar o desenvolvimento de vários tipos de aplicações, dentre os quais se inclui o controle/monitoração de sistemas eletro-hidráulicos.

### 3.4 Controle Digital

Por diversas razões, de acordo com PAGANO (1999), alguns sistemas apresentam sinais disponíveis em determinados instantes de tempo discreto. Dentre as várias situações que originam o aparecimento de sinais discretos no tempo é interessante mencionar:

- repartição de um instrumento de alto custo (multiplexação), como mostra a figura 3.7;
- Controle de sistemas utilizando computadores digitais.

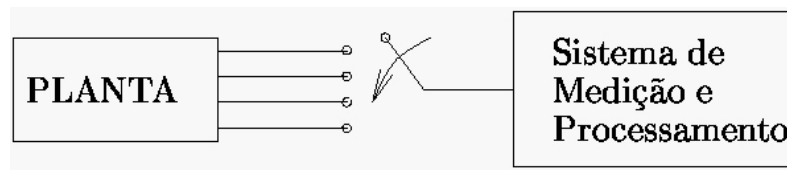


Figura 3.7 – Multiplexador (PAGANO, 1999)

Assim, na figura 3.8, mostra-se o caso mais comum de controle digital de processos contínuos.

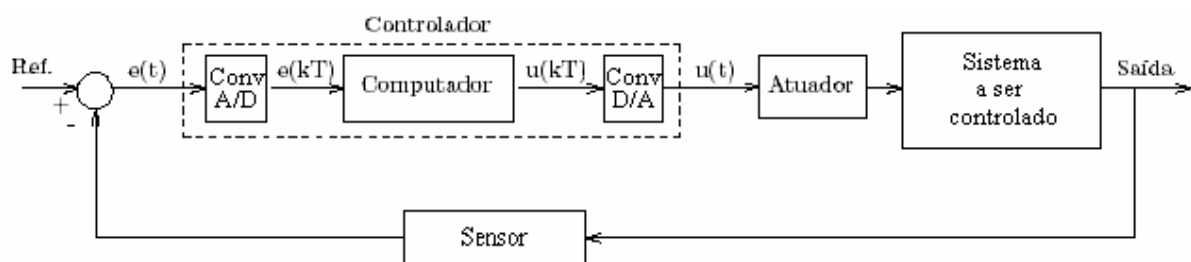


Figura 3.8 – Controle digital de processos contínuos (adaptado de TROFINO, 2002).

Neste caso, o controle é implementado de forma discreta, utilizando-se para tal, diferentes elementos (amostrador, conversores A/D e D/A, bloqueador)<sup>16</sup> que permitem processar os sinais analógicos do sistema contínuo. Sendo implementado de forma discreta, alguns problemas relacionados ao controle aparecem.

### 3.4.1 Problemas Ligados ao Controle de Sistemas Amostrados

#### 3.4.1.1 Digitalização de um Sinal Analógico

PAGANO (1999) declara que o primeiro problema apresentado é como realizar a amostragem dos sinais contínuos. Na figura 3.9 está representado o processo de amostragem de um sinal contínuo. Neste processo dois fatores são importantes:

- a escolha do período de amostragem  $T$  ou taxa de amostragem  $f = 1/T$ ;
- a representação matemática utilizada.

<sup>16</sup> Nesta figura, o conversor A/D é composto pelo amostrador e pela quantização do sinal. O conversor D/A, por sua vez, inclui o bloqueador.



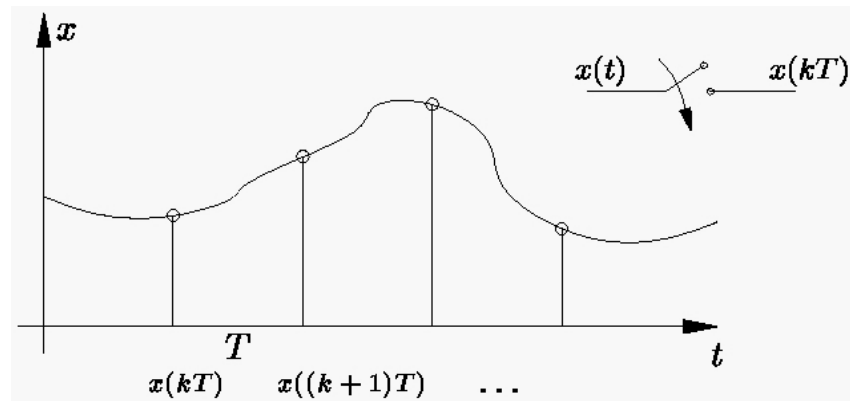


Figura 3.9 – Amostragem de sinais contínuos (PAGANO, 1999).

Em relação à escolha de  $T$ , não existiria uma perda significativa de informação caso este fosse suficientemente pequeno frente à velocidade de variação do fenômeno considerado. Isto implicaria, no entanto, em um custo elevado em termos de tempo de cálculo. Tem-se então, que quanto maior a frequência de amostragem, melhor a informação e mais alto o custo (IFEACHOR *et al.*, 1993; PROAKIS *et al.*, 1996). Portanto, existe um compromisso entre qualidade de informação e custo de cálculo.

No caso do controle por computador, o tempo que se leva para executar a malha de controle, chamado de tempo de ciclo do laço, pode ser afetado por razões como complexidade do aplicativo, memória disponível e velocidade de processamento do PC. Por sua vez, isto acaba afetando a frequência de amostragem na aquisição de sinais.

Não obstante, uma baixa frequência de amostragem pode ocasionar perda de informação como descrito a seguir.

#### 3.4.1.1.1 Aliasing

De acordo com o Teorema de Nyquist, a quantidade de amostras por unidade de tempo de um sinal, chamada taxa, ou frequência, de amostragem ( $f_{am}$ ), deve ser maior que o dobro da maior frequência contida no sinal a ser amostrado ( $f$ ), para que possa ser reproduzido integralmente sem erro de *aliasing* (ZURMELY, 2004)<sup>17</sup>. A metade da frequência de amostragem é chamada frequência de Nyquist e corresponde ao limite máximo de frequência do sinal que pode ser reproduzido. Como não é possível garantir que o sinal não contenha sinais acima deste limite (distorções, interferências, ruídos etc), IFEACHOR *et al.* (1993) declara que é necessário filtrar o sinal com um filtro passa-baixas com frequência de corte igual (ou menor) à frequência de Nyquist, ou filtro *anti-aliasing*.

A figura 3.10, segundo ZURMELY (2004), mostra um sinal senoidal sendo amostrado com taxas próximas ao limite. Na parte superior, a amostragem é praticada com frequência

<sup>17</sup> Idealmente, no entanto, em sistemas reais, um fator de três ou mais seria mais aconselhável (DE NEGRI *et al.*, 1998).

( $f_{am}$ ) maior que duas vezes a do sinal ( $f$ ): há amostras suficientes para que o sinal possa ser reproduzido sem erro de *aliasing*. No meio, a taxa de amostragem ( $f_{am}$ ) é igual a duas vezes a frequência do sinal ( $f$ ): não é possível a sua reprodução, pois, neste caso, não há defasagem entre os pontos de amostragem e o sinal. Por isso, a necessidade do "maior que o dobro" no Teorema de Nyquist. Na parte inferior, a frequência de amostragem ( $f_{am}$ ) é menor que o dobro da frequência do sinal ( $f$ ): a quantidade de amostras é insuficiente e o sinal reproduzido estará errado, em vermelho na figura 3.10. Este erro é causado pelo fenômeno de *aliasing*.

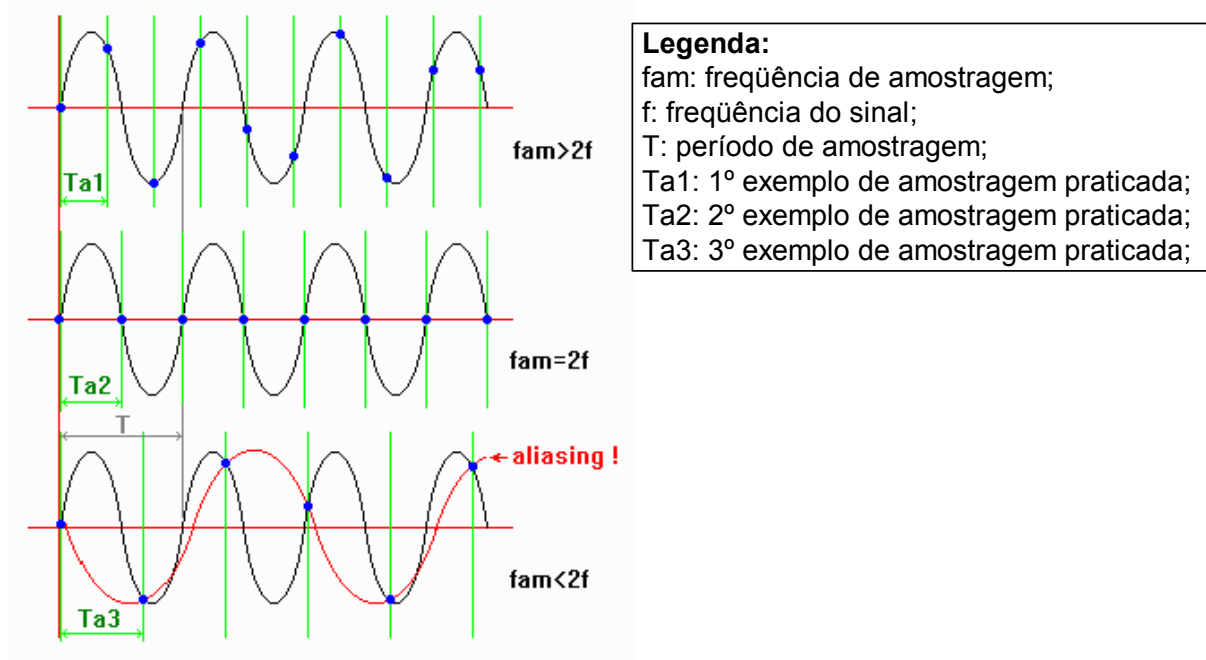


Figura 3.10 – Amostragem de um sinal senoidal com taxas próximas ao limite (ZURMELY, 2004).

#### 3.4.1.2 Quantização e Codificação

O segundo problema, conforme PAGANO (1999), é como fazer a conversão analógico/digital (A/D) e digital/análogo (D/A). PROAKIS *et al.* (1996) alega que este processo de conversão de sinais implica necessariamente quantificação dos sinais contínuos para poderem ser transformados em sinais discretos (codificação). Isto produz inevitavelmente erros cuja magnitude depende diretamente do número de bits dos conversores.

#### 3.4.1.3 Reconstituição do Sinal Analógico

Também, existe o problema de como recuperar o sinal analógico contínuo. Para resolver esta situação é necessário utilizar um elemento que permita interpolar o sinal discreto em amostras (IFEACHOR *et al.*, 1993; PROAKIS *et al.*, 1996). Esse processo denomina-se de bloqueamento ou interpolação e o elemento associado de bloqueador ou sustentador. A

interpolação pode ser realizada utilizando-se diferentes tipos de sinais, por exemplo, degraus e rampas.

#### 3.4.1.4 Projeto de Controladores Discretos

Finalmente, outra questão consiste em projetar o algoritmo de controle. Para solucionar esse problema são empregadas ferramentas de análise e projeto para tempo discreto.

Isso se deve ao fato de que, diferentemente da eletrônica analógica, os computadores digitais não podem executar funções de integração. Portanto, equações diferenciais que descrevem um controlador contínuo  $C(s)$  devem ser aproximadas utilizando-se equações a diferenças que envolvem somente termos de adição e multiplicação.

Duas técnicas de projeto de controle digitais são comentadas a seguir:

- por aproximação (ou emulação);
- projeto discreto ou direto.

##### 3.4.1.4.1 Método de Projeto por Aproximações

Conforme PAGANO (1999), esta técnica consiste em projetar um controlador contínuo  $C(s)$  para um determinado processo, utilizando ferramentas de domínio contínuo e, em um segundo passo, transladar o controlador do domínio contínuo ao discreto, mediante aproximações, obtendo assim o controlador discreto  $C(z)$ .

O controlador contínuo  $C(s)$  é aproximado mediante equações a diferenças obtidas através de diferentes métodos, como por exemplo: o método de Euler, de Tustin ou bilinear, aproximação zero-pólo etc.

##### 3.4.1.4.2 Método de Projeto Direto

A partir do fato de que é possível obter um modelo exato discreto que relacione as amostras da saída  $y(k)$  com as da entrada  $u(k)$ , pode-se projetar um controlador discreto para um determinado processo. Basicamente, de acordo com (PAGANO, 1999), os passos a seguir são:

- obter o modelo discreto  $G(z)$  da planta  $G(s)$  como mostrado na figura 3.11;
- utilizar  $G(z)$  para projetar o controlador  $C(z)$ .

Para obter o modelo discreto de um determinado processo ou sistema utilizam-se ferramentas computacionais. Na fase de análise e projeto podem-se aplicar técnicas conhecidas para os sistemas contínuos (projeto por lugar das raízes, métodos frequenciais, etc).

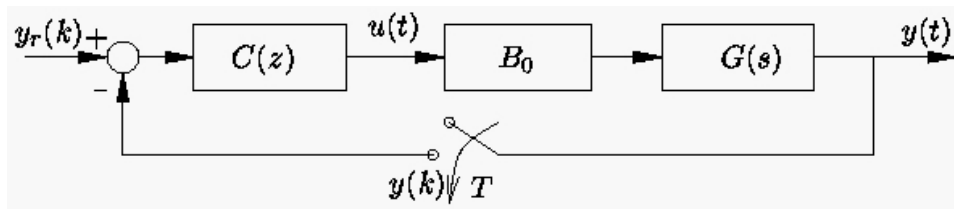


Figura 3.11 – Sistema de controle discreto (PAGANO, 1999).

## **CAPÍTULO 4**

### **DESENVOLVIMENTO DE SOFTWARE VOLTADO AO LABVIEW**

Tendo em vista a apresentação, no capítulo 3, do software LabVIEW aliado ao sistema VXIbus como uma ferramenta na aquisição e controle de dados para o desempenho do controle de posição por computador dos sistemas eletro-hidráulicos (delineado no capítulo 2), aponta-se neste capítulo os aspectos que devem ser considerados no desenvolvimento de um software com ênfase em LabVIEW.

#### **4.1 Modelos de Desenvolvimento**

O LabVIEW facilita o desenvolvimento de aplicações que utilizam componentes de aquisição de dados, testes e sistemas de controle. Devido às facilidades de criação de aplicações em LabVIEW, muitos programadores começam a desenvolver VIs com um planejamento relativamente pequeno. Para aplicações simples, como rápidos testes de laboratório ou aplicações de monitoramento, isto pode ser apropriado. Porém, de acordo com JALOTE (2003), para projetos de desenvolvimento de softwares mais complexos, um bom planejamento do projeto é vital.

Do mesmo modo, deve-se considerar alguns elementos fundamentais no desenvolvimento de um software de qualidade tais como:

- despende tempo suficiente no planejamento;
- assegurar-se de que a equipe inteira entendeu completamente os problemas que tem a resolver;
- ter uma estratégia de desenvolvimento flexível que minimize riscos e facilite mudanças.

#### **4.2 Modelos de Ciclo de Vida**

Para lidar com as complexidades de projetos que envolvam desenvolvimento de software, muitos profissionais aderem a um conjunto de princípios de desenvolvimento, os quais definem o campo de engenharia de software. O componente principal deste campo é o modelo do ciclo de vida, o qual descreve os passos a seguir no desenvolvimento do software – da fase inicial de conceito ao lançamento, manutenção e melhoramento subsequente do software (NATIONAL INSTRUMENTS, 2003c).

Atualmente existem muitos modelos de ciclo de vida diferentes, cada qual com vantagens e desvantagens em termos de tempo para o lançamento, qualidade e gerenciamento de risco.

Desse modo, esta seção descreve alguns dos modelos mais comuns usados em engenharia de software. No entanto, existem muitos modelos híbridos destes, de tal modo que se pode personalizar estes modelos a fim de se ajustarem às exigências de um projeto.

Embora esta seção seja teórica em sua discussão, na prática deve-se considerar todos os passos que estes modelos englobam, como decidir quais requisitos e especificações o projeto necessita e como lidar com suas mudanças.

Enfim, o modelo do ciclo de vida é fundamental para todo o processo de desenvolvimento. Boas decisões podem melhorar a qualidade do software a ser desenvolvido e diminuir seu tempo de desenvolvimento (NATIONAL INSTRUMENTS, 2003c).

#### 4.2.1 Modelo Codificar e Corrigir

O modelo codificar e corrigir provavelmente é a metodologia de desenvolvimento mais freqüentemente utilizada em engenharia de software. Começa com pequeno ou nenhum planejamento inicial. Imediatamente começa-se desenvolvendo, corrigindo problemas que ocorrem, até que o projeto esteja completo (MCCONNELL, 2003).

Esse modelo só é apropriado para pequenos projetos, os quais não se pretendem que sirvam como base para desenvolvimento futuro. Segundo SILVA (2002), as deficiências deste modelo levaram ao desenvolvimento de outros.

#### 4.2.2 Modelo Cascata (Waterfall)

O modelo cascata é o modelo clássico da engenharia de software (PRESSMAN, 1987), sendo este um dos mais antigos, e é extensamente usado em projetos de muitas das maiores empresas.

Por enfatizar o planejamento nas primeiras fases, ele captura falhas de projeto antes de eles serem desenvolvidos. Também, porque é documentado e planejado intensivamente, ele funciona bem para projetos nos quais o controle de qualidade é uma preocupação fundamental.

Conforme PRESSMAN (1987), o modelo do ciclo de vida cascata consiste em várias fases que não se sobrepõem, como mostrado na figura 4.1. O modelo começa com o estabelecimento dos requisitos de sistema e de software e continua com o projeto arquitetural, projeto detalhado, codificação, testes e manutenção. Também, serve como base para muitos outros modelos de ciclo de vida.

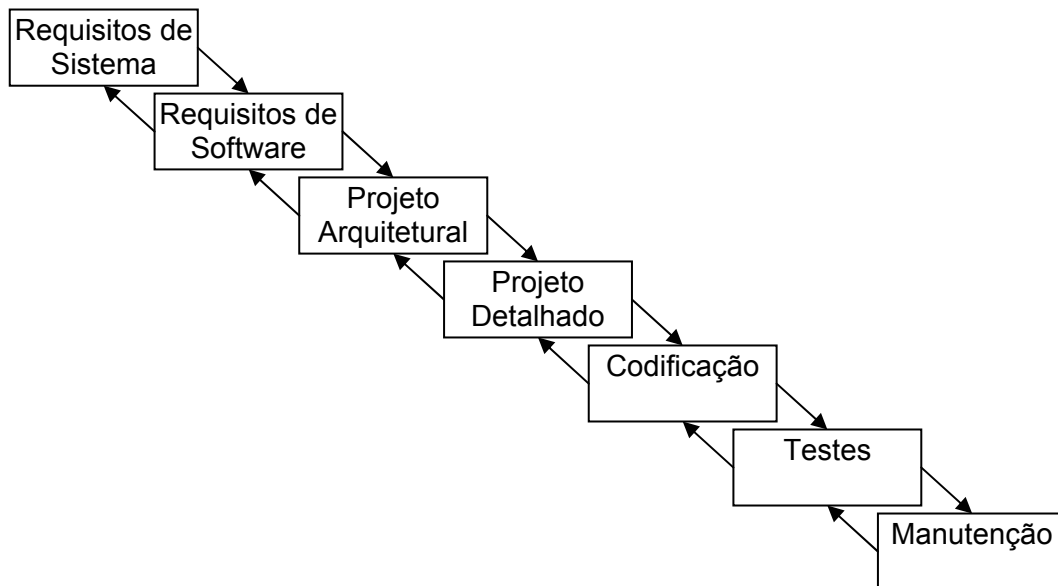


Figura 4.1 – Modelo do ciclo de vida cascata.

Sendo assim, as etapas deste modelo de desenvolvimento são (NATIONAL INSTRUMENTS, 2003c):

- Requisitos de sistema – Estabelecem os componentes para construir o sistema, inclusive os requisitos de hardware, ferramentas de software e outros componentes necessários. Exemplos incluem decisões em hardware, como placas *plug-in* (número de canais, velocidade de aquisição e assim por diante), e decisões externas do software, como bancos de dados ou bibliotecas.
- Requisitos de software – Estabelecem as expectativas da funcionalidade do software e identificam quais os requisitos de sistema afetam o software. A análise de requisitos inclui a determinação da interação necessária com outras aplicações e bancos de dados, requisitos de desempenho, requisitos de interface com o usuário, e assim por diante.
- Projeto arquitetural – Determina a estrutura do software de um sistema para satisfazer aos requisitos especificados. Este define os principais componentes e a interação desses componentes, mas não define a estrutura de cada componente. Também determina as interfaces externas e as ferramentas para utilizar no projeto.
- Projeto detalhado – Examina os componentes de software definidos na fase de projeto arquitetural. Produz uma especificação de como cada componente é implementado.
- Codificação – Implementa a especificação do projeto detalhado.
- Testes – Determina se o software satisfaz às exigências especificadas e encontra qualquer erro presente no código.
- Manutenção – Se refere aos problemas e pedidos de modificação posteriores ao lançamento do software.

Em algumas organizações, um painel de controle de mudanças mantém a qualidade do produto revendo cada mudança feita na fase de manutenção.

Em cada fase, cria-se documentos que explicam os objetivos e descrevem os requisitos para aquela fase. Ao término de cada fase, faz-se uma revisão para determinar se o projeto pode prosseguir à próxima. Também se pode incorporar prototipação em qualquer fase do projeto arquitetural e depois dele.

O método cascata não proíbe voltar a uma fase anterior, como por exemplo, da fase de projeto para a fase de requisitos. Porém, isto envolve um custo de retrabalho. Cada fase completada requer uma revisão formal e um extenso desenvolvimento de documentação. Assim, omissões cometidas na fase de requisitos custam caro para serem corrigidas posteriormente (TAVARES, 2002).

Por fim, as primeiras versões operacionais do software são obtidas nas etapas mais tardias do processo e, por isso, não se vê resultados por muito tempo. Esta demora pode ser desconcertante para a gerência e para os clientes. Muitos também acreditam que a quantidade de documentação é excessiva e inflexível (LEFFINGWELL *et al.*, 1999).

Muitos engenheiros recomendam versões modificadas do ciclo de vida cascata.

#### 4.2.3 Prototipação

Um dos principais problemas com o modelo cascata é que, freqüentemente, os requisitos não são completamente entendidos nas primeiras fases do desenvolvimento (JALOTE, 2003). Quando se alcançam as fases de projeto ou codificação é que se começa a ver como tudo trabalha junto e pode-se descobrir que se precisa ajustar os requisitos.

TAVARES (2002) assegura que a prototipação é uma ferramenta efetiva por demonstrar como um projeto satisfaz a um conjunto de requisitos. Pode-se construir um protótipo, ajustar os requisitos e revisar o protótipo várias vezes até que se tenha uma visão clara de todos os objetivos (WIKIMEDIA FOUNDATION, 2004). Além de elucidar os requisitos, um protótipo define também muitas áreas do projeto simultaneamente.

Porém, a prototipação tem suas desvantagens. Por parecer que se tem um sistema em funcionamento, os clientes podem esperar um sistema completo mais cedo do que é possível (NATIONAL INSTRUMENTS, 2003c; PRESSMAN, 1987). Na maioria dos casos são construídos protótipos que permitem que isto ocorra depressa, mas que impedem que o protótipo seja uma base efetiva para o desenvolvimento futuro. Portanto, precisa-se decidir cedo a pretensão de se usar o protótipo como uma base para desenvolvimento futuro, onde todas as partes precisam concordar com essa decisão antes do desenvolvimento começar.

Além disso, deve-se ter cuidado para que este modelo não se torne um disfarce de um ciclo de desenvolvimento codificar e corrigir. Antes de se começar a prototipação, deve-se juntar requisitos claros e criar um plano de projeto, além de limitar a quantidade de tempo (prazos) que será despendida nesta atividade. Depois que se terminar a prototipação, deve-



se retornar as fases previstas em um dos outros modelos de desenvolvimento (CRNKOVIC *et al.*, 2002), ou seja, a prototipação deve ser considerada como parte dos requisitos ou fases de projeto do modelo cascata.

Aspectos relacionados à prototipagem serão melhor delineados no item 4.4.

#### 4.2.4 Modelo Espiral

O modelo espiral é uma popular alternativa ao modelo cascata. Ele enfatiza o gerenciamento de risco de modo que se encontrem mais cedo os principais problemas no ciclo de desenvolvimento (TAVARES, 2002; LEFFINGWELL *et al.*, 1999). No modelo cascata, tem-se que completar o projeto antes de se começar a codificar. Com o modelo espiral, dissolve-se o projeto em um conjunto de riscos que precisam ser repartidos. Então, começam-se séries de repetições nas quais se analisa o risco mais importante, avaliam-se opções para solucionar o risco, lidar com o risco, avaliar os resultados e planejar a próxima iteração. A figura 4.2 ilustra o modelo de ciclo de vida espiral.

Risco é qualquer questão que não está definida claramente ou tem potencial para afetar o projeto adversamente (JALOTE, 2003). Para cada risco, deve-se considerar o seguinte:

- a chance de o risco ocorrer (probabilidade);
- a severidade do efeito do risco no projeto (perda).

Dessa maneira, segundo NATIONAL INSTRUMENTS (2003c), pode-se utilizar uma escala de 1 a 10 para cada um destes itens, com o número 1 representando a mais baixa probabilidade ou perda e 10 representando a mais alta. De tal modo, a tabela 4.1 ilustra como identificar os itens do projeto que representam maior risco (exposição ao risco)<sup>18</sup>.

Em geral, deve-se lidar primeiramente com os riscos de maior pontuação, sendo que cada iteração pode identificar novos riscos. Na fase final, avaliam-se os resultados com o cliente para que, baseado na sua ajuda, se possa reavaliar a situação, decidir sobre o próximo mais alto risco e recomeçar o ciclo. Este processo continua até que o software esteja acabado ou decida-se que os riscos são tão altos que finalizam o desenvolvimento, sendo possível que nenhuma das opções seja viável porque as opções sejam muito caras, demoradas ou não satisfaçam aos requisitos.

---

<sup>18</sup> Representam maior risco os itens com maior pontuação na coluna “Exposição ao Risco” da tabela 4.1. Por sua vez, a exposição ao risco é calculada multiplicando-se o valor da coluna “Probabilidade” pelo da coluna “Perda”.

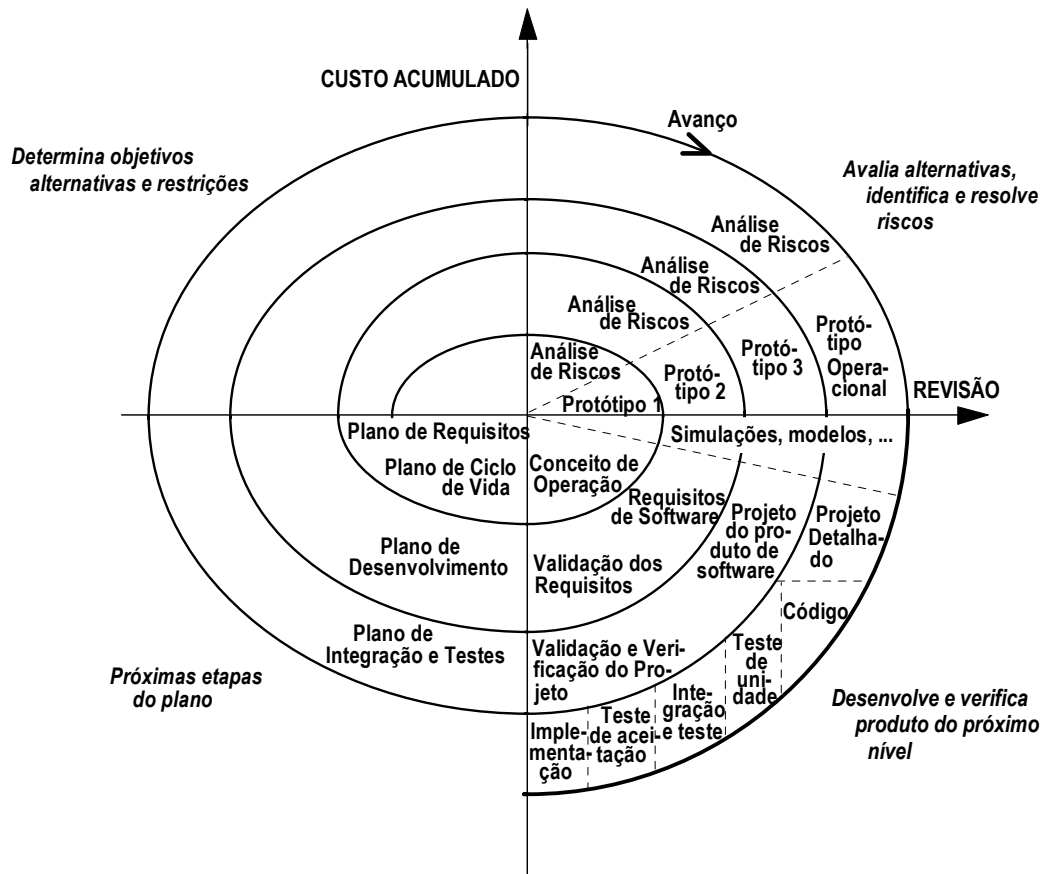


Figura 4.2 – Modelo de ciclo de vida espiral (adaptado de MAZZOLA; FARINES, 1998)

Tabela 4.1 – Análise de exposição de riscos.

ID	Risco	Probabilidade	Perda	Exposição ao Risco	Abordagem do Gerenciamento de Risco
1	Taxas de aquisição muito altas	5	9	45	Desenvolver um protótipo para demonstrar a viabilidade
2	Formato do arquivo pode não ser eficiente	5	3	15	Desenvolver medidas de desempenho para mostrar a velocidade da manipulação de dados
3	Interface com o usuário incerta	2	5	10	Envolver o cliente; desenvolver um protótipo

Assim sendo, a vantagem do modelo espiral sobre o cascata é que se pode avaliar mais rapidamente os riscos em cada ciclo, através de protótipos, podendo-se deparar com os principais obstáculos e selecionar alternativas nas primeiras fases onde os custos são menores.

### 4.3 Qualidade no Processo de Desenvolvimento

De acordo com SELNER (1999), os desenvolvedores têm que projetar um produto com qualidade desde o começo. Desta forma, deve-se considerar a qualidade durante todas

as fases de desenvolvimento – requisitos e especificações, projeto, codificação, testes, lançamento e manutenção (MCCONNELL, 2003).

Portanto, não se deve considerar os controles de qualidade como requisitos tediosos que impedem o desenvolvimento. A maioria dos controles de qualidade agiliza o desenvolvimento fazendo com que se descubram os problemas antes de eles estarem no software, quando as correções ainda são baratas.

#### 4.3.1 Requisitos de Qualidade

Conforme LEFFINGWELL *et al.* (1999), deve-se estabelecer as normas de qualidade para um produto durante a fase de requisitos e tratar o nível de qualidade desejado como um requisito. Para isso, analisam-se os benefícios e os custos das várias opções que se tem aplicando-se medidas de qualidade para o projeto. Alguns pontos a considerar incluem a facilidade de uso versus o poder e a complexidade, e a velocidade versus a robustez.

Para pequenos projetos não é preciso enfatizar a robustez. Entretanto, em projetos mais complicados que devem ser confiáveis, como aplicações de monitoramento e controle de um processo de fábrica, o software deve ser capaz de manipular entradas inválidas (NATIONAL INSTRUMENTS, 2003c). Por exemplo, se um operador selecionar uma tensão inválida ou uma corrente errada, a aplicação tem que manipular esses dados adequadamente. Logo, deve-se estabelecer tantas proteções quanto possíveis para prevenir problemas. Por isto, faz-se necessário selecionar um modelo de desenvolvimento de ciclo de vida que ajude a encontrar problemas assim que possível e que dê tempo para se fazerem revisões formais e testes completos.

#### 4.3.2 Gerência da Configuração

Conforme REIS (2003), este termo cobre um conjunto de políticas e atividades que têm como objetivo controlar mudanças assegurando-se que elas foram revisadas antes de serem feitas.

Mudanças ocorrem tardiamente no processo e precisam ser implementadas. Por exemplo, clientes introduzem freqüentemente novos requisitos na fase de projeto, ou problemas de desempenho que são descobertos durante o desenvolvimento levam a uma reavaliação do projeto. Pode-se, também, precisar reescrever uma seção de código para corrigir um problema encontrado nos testes.

Enfim, as mudanças podem afetar qualquer componente de um projeto desde requisitos e especificação até projeto, codificação e testes. Logo, se estas mudanças não são feitas cuidadosamente, pode-se introduzir novos problemas os quais podem retardar o desenvolvimento ou degradar a qualidade.

#### 4.3.2.1 Recuperação de Versões Antigas de Arquivos no LabVIEW

Há momentos em que é necessário restaurar uma versão antiga de um arquivo ou projeto. Um modo de acessar uma versão antiga de um arquivo ou projeto é fazer uma cópia auxiliar dos arquivos periodicamente. Porém, isso exige que se faça uma cópia do VI após toda mudança realizada para que se tenha acesso a todas as suas versões.

Para tanto, as ferramentas de controle do código fonte do LabVIEW provêm um modo de registrar versões novas de um arquivo e fazer uma cópia auxiliar da versão antiga. Além disso, dependendo como é configurado o sistema, essas ferramentas podem manter múltiplas cópias auxiliares de um arquivo (NATIONAL INSTRUMENTS, 2003b).

Até mesmo, por meio destas ferramentas de controle do código fonte, é possível rotular versões de arquivos com nomes descritivos (como *beta*, *v1.0*) e posteriormente restaurá-las à partir de um rótulo específico.

#### 4.3.2.2 Localização de Mudanças no LabVIEW

Ao se gerenciar um projeto de software, é importante monitorar as mudanças e progredir em direção dos objetivos especificados. Também, pode-se usar esta informação para determinar as áreas de problema de um projeto identificando os componentes que requerem muitas mudanças.

As ferramentas de controle do código fonte mantêm um registro de todas as mudanças feitas em arquivos e projetos. Deste modo, pode-se ver a informação da história de um arquivo ou sistema e gerar relatórios que contenham estes sumários de informações.

Além disso, o LabVIEW possui uma ferramenta a qual compara a mais recente versão de um projeto com outra versão a fim de se verificar as mudanças no projeto.

#### 4.3.3 Diretrizes de Teste

Primeiramente, deve-se decidir que nível de testes o projeto requer. Os engenheiros, sob a pressão do prazo final, freqüentemente prestam pouca atenção aos testes. No entanto, com um certo nível de testes garante-se ganho de tempo.

Para isso, os desenvolvedores têm que entender claramente o grau de testes que se espera. Assim como se desenvolvem as especificações de requisitos e de projeto, também se deve desenvolver um plano de testes que ajude a verificar o sistema e todos os seus componentes (LEFFINGWELL *et al.*, 1999). Os testes refletem as metas de qualidade que se quer alcançar.

Deste modo, deve-se considerar a fase de testes como parte das fases de projeto iniciais e empregar testes ao longo do desenvolvimento para encontrar e corrigir problemas o mais cedo possível.

Para tanto, há uma variedade de metodologias de teste que ajudam a aumentar a qualidade dos projetos de VI. As seções seguintes descrevem algumas destas metodologias de testes.

#### 4.3.3.1 Teste de Caixa Preta e Teste de Caixa Branca

O método de teste de caixa preta está baseado na funcionalidade esperada do software, sem o conhecimento de como este trabalha (WIKIMEDIA FOUNDATION, 2004). É chamado de teste de caixa preta porque não se está interessado no funcionamento interno.

Conforme MAZZOLA e FARINES (1998), o teste de caixa preta refere-se a todo teste que implica verificação do funcionamento do software através de suas interfaces, o que, na maioria das vezes, permite verificar a operacionalidade de todas as suas funções.

Sendo assim, pode-se executar testes de caixa preta na interface de um subVI a fim de se avaliar os resultados obtidos para vários valores de entradas. Caso a robustez seja uma meta de qualidade, pode-se incluir dados de entrada errôneos para ver se o subVI os trata de modo bem sucedido. Por exemplo, para entradas numéricas, ver como o subVI trata valores infinitos e outros valores fora da faixa esperada.

De outro modo, um teste de caixa branca está relacionado a um exame minucioso de sua estrutura interna e detalhes procedimentais (MAZZOLA; FARINES, 1998; WIKIMEDIA FOUNDATION, 2004). Os caminhos lógicos definidos no software são exaustivamente testados, pondo a prova conjuntos bem definidos de condições ou laços. Durante o teste, o estado do programa pode ser analisado diversas vezes para eventual comparação com condições de estado esperadas para aquela situação.

No entanto, o teste de caixa branca não é muito prático porque testar todos os possíveis caminhos é muito difícil e, por isso, pode-se escolher testar os caminhos mais importantes ou complexos para programas de grande porte.

#### 4.3.3.2 Testes de Unidade, Integração e Sistema

Deve-se utilizar os testes de caixa preta e de caixa branca de forma isolada ou combinada para testar qualquer componente de software, seja em um VI individual ou em uma aplicação completa.

Os testes de unidade, integração e sistema são fases do projeto no qual se pode aplicar testes de caixa preta e testes de caixa branca.

##### 4.3.3.2.1 Testes de Unidade

O teste de unidade almeja a verificação de erros existentes nas unidades (módulos) de projeto do mesmo (PRESSMAN, 1987), onde é importante valer-se das informações contidas no documento de projeto detalhado do software as quais servirão de guia para sua aplicação.

Pode-se usar o teste de unidade ao testar componentes individuais do software (WIKIMEDIA FOUNDATION, 2004). Alguns exemplos de testes de unidade, de problemas comuns, poderiam considerar o seguinte (NATIONAL INSTRUMENTS, 2003c):

- condições limite para cada entrada, como “arrays” e “strings” vazias, ou valor 0 para um tamanho de entrada;
- valores inválidos para cada entrada, como -3 para um tamanho de uma entrada;
- combinações estranhas de entradas;
- arquivos perdidos e nomes de caminhos errados;
- o que acontece quando o usuário clica com o mouse ou aperta o botão de cancelamento em uma caixa de diálogo do arquivo;
- o que acontece se o usuário aborta o VI.

#### 4.3.3.2.2 Testes de Integração

Os testes de unidade normalmente encontram a maioria dos erros, mas os de integração podem revelar problemas imprevistos. Isso porque os módulos podem não trabalhar conjuntamente como o esperado, interagindo de modo imprevisto por causa do modo que manipulam dados compartilhados.

Desse modo, realiza-se os testes de integração em uma combinação de unidades em busca de erros surgidos quando da integração das diferentes unidades componentes do software (WIKIMEDIA FOUNDATION, 2004).

No entanto, de acordo com JALOTE (2003), não se deve realizar os testes de integração no qual se combinam todos os componentes e se tenta testar o programa de mais alto nível (*top-level*). Este método pode ser trabalhoso porque é difícil de se determinar a fonte específica de problemas dentro de um grande conjunto de VIs. Ao invés disso, deve-se realizar testes incrementalmente, com uma abordagem *top-down* ou *bottom-up* (PRESSMAN, 1987).

A integração *top-down* é uma estratégia onde os módulos são integrados segundo um movimento de cima para baixo (JALOTE, 2003). Este processo inicia-se no módulo de controle principal M1 seguindo um caminho de descida como representado na figura 4.3.

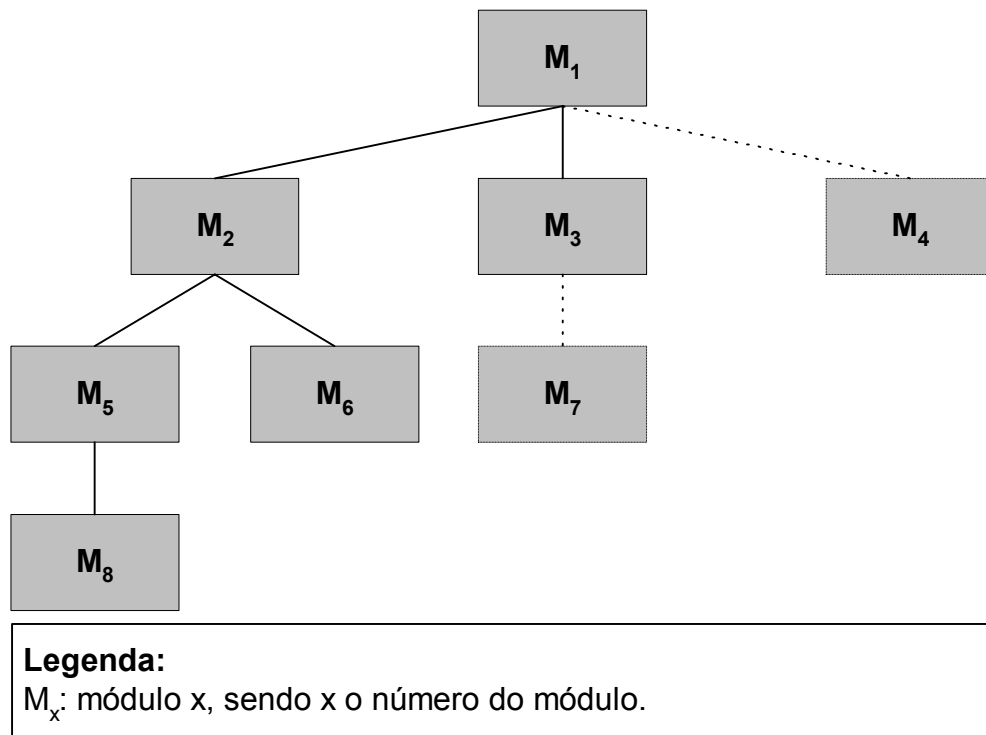


Figura 4.3 – Integração *top-down* (MAZZOLA; FARINES, 1998).

De outro modo, com uma abordagem *bottom-up* primeiramente testa-se os módulos de mais baixo nível e gradualmente vai se combinando os módulos de mais alto nível (JALOTE, 2003). Assim, começa-se testando um número pequeno de componentes combinado em um sistema simples como um teste piloto. Depois que se combinou um conjunto de módulos e verificou-se que eles trabalham juntos, adicionam-se componentes testando-os com o subsistema já depurado.

Ou seja, a abordagem *bottom-up* consiste em testes que gradualmente aumentam em extensão enquanto a abordagem *top-down* consiste em testes que são gradualmente refinados à medida que novos componentes são adicionados.

Juntamente a estas abordagens, deve-se executar testes de regressão a cada passo para verificar se as características previamente testadas ainda funcionam. O teste de regressão consiste em repetir alguns ou todos os testes realizados anteriormente.

#### 4.3.3.2.3 Testes de Sistema

O teste de sistema acontece após o teste de integração para determinar se o produto satisfaz às expectativas do cliente e se ter certeza de que o software trabalha como o esperado dentro do sistema de hardware (MAZZOLA; FARINES, 1998; CRNKOVIC *et al.*, 2002; WIKIMEDIA FOUNDATION, 2004).

Destarte, tendo em vista que a maioria das aplicações de LabVIEW executam algum tipo de E/S e também comunicação com outras aplicações, o teste de sistema deve considerar perguntas como:

- Os requisitos de desempenho são satisfeitos?
- Se a aplicação comunica-se com outra aplicação, falhas inesperadas daquela aplicação são bem tratadas?

Ainda, pode-se completar estes testes com testes alfa e beta. Os testes alfa e beta servem para encontrar casos de teste que os desenvolvedores não consideraram ou concluíram. Com o teste alfa, realizado nas instalações do desenvolvedor, observa-se o comportamento do cliente e vai se registrando as anomalias detectadas durante a utilização do software (MAZZOLA; FARINES, 1998). Com o teste beta, quando o teste alfa estiver completo, os clientes testam o produto em campo. Neste caso, os erros encontrados normalmente são registrados pelo próprio usuário e encaminhados ao desenvolvedor na forma de um relatório escrito.

Em NATIONAL INSTRUMENTS (2003c), é relatado que os testes alfa e beta são os únicos mecanismos de testes de algumas empresas. Infelizmente, os testes alfa e beta podem, na verdade, ser inexatos e não ser um substituto para outras formas de testar rigorosamente cada componente. Assim, quando este tipo de teste é feito tardiamente no processo de desenvolvimento, fica difícil e caro incorporar as mudanças necessárias.

#### 4.3.4 Diretrizes de Estilo

Abordagens incompatíveis de desenvolvimento e de interfaces com o usuário podem ser um problema quando diversos desenvolvedores trabalham em um projeto. Cada desenvolvedor tem o seu próprio estilo de desenvolvimento, preferências de cor, técnicas de exibição, práticas de documentação e metodologias de diagrama de blocos de tal modo que um desenvolvedor pode usar variáveis globais e estruturas de seqüenciamento extensivamente enquanto outro pode preferir usar fluxo de dados.

Por conseguinte, conforme a NATIONAL INSTRUMENTS (2003c), técnicas de estilo incompatíveis criam um software que parece ser ruim. Se VIs tiverem comportamentos diferentes para um mesmo botão, usuários podem se confundir e achar que a interface com o usuário dos VIs é de difícil utilização. Além do mais, estilos incompatíveis também fazem com que o software seja de difícil manutenção.

Assim sendo, deve-se estabelecer um conjunto inicial de diretrizes para a equipe de desenvolvimento de VI e incluir regras à medida que o projeto progride. Além disso, pode-se utilizar estas diretrizes em projetos futuros.

Finalmente, criar um padrão de estilo de desenvolvimento em qualquer linguagem de programação é muito difícil porque cada desenvolvedor tem seu próprio estilo de desenvol-



vimento. Por isso, é preciso selecionar um conjunto de diretrizes compatível com todos os membros da equipe a fim de assegurar que todos sigam essas diretrizes.<sup>19</sup>

#### 4.3.5 Revisões de Projeto

As revisões de projeto servem para identificar e corrigir problemas durante o desenvolvimento. Destarte, deve-se discutir metas de qualidade e considerar as seguintes perguntas (NATIONAL INSTRUMENTS, 2003c):

- O projeto incorpora testes?
- Existe tratamento de erros?
- Há alguma suposição no sistema que poderia ser inválida?

Ao mesmo tempo, é necessário verificar no projeto características que são essenciais ao invés de características que são suplementares, documentando sempre os resultados da revisão de projeto e qualquer mudança recomendada.

#### 4.3.6 Revisões de Código

Uma revisão de código, de acordo com NATIONAL INSTRUMENTS (2003c), é semelhante a uma revisão de projeto a não ser por analisar o código em vez do projeto. Durante uma revisão de código, deve-se perguntar e responder algumas das seguintes perguntas:

- O que acontece se um VI ou função específica retorna um erro? Os erros são tratados e/ou relatados corretamente?
- Há alguma condição de competitividade? Um exemplo de uma condição de competitividade é um diagrama de blocos que lê e escreve uma variável global. Deste modo, há o potencial de que um diagrama de blocos paralelo tente manipular simultaneamente esta mesma variável global, resultando em perda de dados.
- O diagrama de blocos é bem implementado? Os algoritmos são eficientes em termos de velocidade e/ou memória em uso?
- O diagrama de blocos é de fácil manutenção? O desenvolvedor faz bom uso de hierarquia, ou ele está colocando muita funcionalidade em um único VI? O desenvolvedor segue as diretrizes estabelecidas?

Ainda, há várias outras características que se pode procurar em uma revisão de código.

---

<sup>19</sup> O apêndice 1 apresenta um resumo de um Guia de Estilo em LabVIEW proposto por NATIONAL INSTRUMENTS (2003c).

### 4.3.7 Análise Pós-Projeto

Ao término de cada fase no processo de desenvolvimento, deve-se considerar uma análise pós-projeto para discutir o que foi bem feito e o que não foi feito.

Cada desenvolvedor precisa avaliar o projeto honestamente discutindo os obstáculos que reduziram o nível de qualidade do projeto. Para tanto, faz-se necessário considerar perguntas como as seguintes durante uma reunião de análise pós-projeto (NATIONAL INSTRUMENTS, 2003c):

- O que foi feito corretamente? O que funcionou bem?
- O que foi feito errado? O que se pode melhorar?
- Há áreas específicas do projeto ou código que precisam ser revistas?
- Uma revisão de projeto ou revisão de código são necessárias?
- Os sistemas de qualidade estão funcionando? Pode-se encontrar mais problemas se os requisitos de qualidade forem modificados? Há melhores maneiras de se conseguir os mesmos resultados?

Além disso, reuniões de análise pós-projeto podem ajudar a corrigir problemas de cronograma.

## 4.4 Técnicas de Prototipagem e Projeto

Quando se inicia um projeto de programação, a decisão de como começar pode ser assustadora. Muitos desenvolvedores em LabVIEW, segundo a NATIONAL INSTRUMENTS (2003c), imediatamente iniciam construindo alguns dos VIs que pensam ser necessários com um processo de desenvolvimento codificar e corrigir e, depois, percebem que na verdade precisam de algo diferente do que já construíram. Por conseguinte, desenvolvem, refazem ou descartam código desnecessariamente.

Em consequência deste fato, este item aborda pontos de planejamento de projeto, incluindo abordagens de programação, prototipagem e testes de desempenho.

### 4.4.1 Definição de Requisitos da Aplicação

A etapa de análise de requisitos é um conjunto de atividades de essencial importância para o processo de desenvolvimento de um software e deve ser desempenhada, portanto, independentemente da abordagem e técnica utilizadas. O completo entendimento dos requisitos de software é um ponto essencial para o sucesso de um projeto de software.

Deste modo, a análise de requisitos é uma tarefa que envolve um trabalho de descoberta, refinamento, modelagem e especificação das necessidades e desejos relativos ao aplicativo/software que deverá ser desenvolvido. Nesta tarefa, tanto o cliente como o desenvolvedor desempenham um papel de grande importância, já que cabe ao primeiro a formula-

ção (de modo concreto) das necessidades em termos de funções e desempenho, enquanto o segundo atua como indagador, consultor e solucionador de problemas.

MAZZOLA e FARINES (1998) afirmam que esta etapa é de suma importância no processo de desenvolvimento de um software, principalmente porque ela estabelece o elo entre a alocação do software em nível de sistema (realizada na etapa de Engenharia de Sistema) e o projeto do software. Desta forma, ela permite que o engenheiro de sistemas especifique as necessidades do software em termos de funções e de desempenho, estabeleça as interfaces do software com os demais elementos do sistema e especifique as restrições de projeto. Ao engenheiro de software (ou analista), a análise de requisitos permite uma alocação mais precisa do software no sistema e a construção de modelos do processo, dos dados e dos aspectos comportamentais que serão tratados pelo software. Ao projetista, esta etapa proporciona a obtenção de uma representação da informação e das funções que poderá ser traduzida em projeto arquitetural. Além disso, é possível definir os critérios de avaliação da qualidade do software a serem verificados uma vez que o software esteja concluído.<sup>20</sup>

Por isso, antes de se desenvolver um projeto detalhado de um sistema, deve-se definir claramente as metas, fazendo-se uma lista de requisitos. Alguns requisitos são específicos, como os tipos de E/S, taxas de amostragem ou a necessidade de análise em tempo real. Assim, deve-se fazer alguma pesquisa nesta primeira fase para assegurar que se possam satisfazer as especificações. Outros requisitos dependem das preferências do usuário, como formatos de arquivo ou estilos de gráfico (LEFFINGWELL *et al.*, 1999).

De tal modo, é preciso distinguir-se os requisitos absolutos dos desejáveis. Isto é bom para se ter uma idéia sobre quais características podem ser sacrificadas no caso de o cronograma estar apertado.

Ainda, deve-se ter o cuidado para que os requisitos não sejam tão detalhados que acabem por restringir o projeto. Por exemplo, quando se projetar um sistema de E/S, o cliente tem, provavelmente, certos requisitos de taxa de amostragem e de exatidão de medição. Porém, o custo também é uma restrição. Estas questões devem estar inclusas nos requisitos. Caso se possa evitar a especificação do hardware, pode-se ajustar o projeto depois de se começar a prototipação e os testes de desempenho de vários componentes NATIONAL INSTRUMENTS (2003c). Contanto que os custos estejam dentro do orçamento especificado e sejam conhecidas as questões de amostragem e de exatidão de medição, o cliente pode não se preocupar se o sistema utiliza um tipo particular de placa *plug-in* ou outro hardware.

---

<sup>20</sup> No caso deste trabalho, o presente aluno atuou como engenheiro de sistema, engenheiro de software e projetista enquanto o orientador do trabalho, principalmente, atuou como o cliente.

#### 4.4.2 Projeto Top-Down

A programação por diagrama de blocos a qual o LabVIEW utiliza foi projetada para ser de fácil entendimento. A maioria dos engenheiros já utiliza diagramas de blocos para descrever sistemas. O diagrama de blocos do LabVIEW faz com que seja mais fácil a conversão dos diagramas de blocos dos sistemas que se cria em código executável (NATIONAL INSTRUMENTS, 2003c).

O conceito básico é dividir a tarefa em partes manejáveis em lugares lógicos. Para isso, deve-se começar com um diagrama de blocos de alto nível que descreva os principais componentes do VI. Por exemplo, pode-se ter um diagrama de blocos que consiste em um bloco para configuração, um bloco para aquisição, um bloco para análise dos dados adquiridos, um bloco para exibição dos resultados, um bloco para salvar os dados no disco e um bloco para limpeza ao fim da execução do VI.

Depois que se determina os blocos de alto nível, deve-se criar um diagrama de blocos que utilize esses blocos. Cada bloco representará um subVI. Deste modo, verifica-se a funcionalidade dos subVIs para o diagrama de blocos de mais alto nível que eles (diagrama o qual chamará os subVIs).

Após a construção desses subVIs, determina-se a função de cada bloco e como este trabalha. Verifica-se se um determinado bloco gera informações necessárias aos VIs subsequentes. Para isso, deve-se ter certeza de que o VI de mais alto nível é capaz de passar os dados entre os VIs. Por fim, faz-se a documentação de cada subVI, a qual é facilitada no LabVIEW, a fim de se documentar a funcionalidade do VI e suas entradas e saídas.

O refinamento do projeto dá-se pela divisão de cada um dos blocos componentes em outros mais detalhados (WIKIMEDIA FOUNDATION, 2004). Por exemplo, pode-se dividir um subVI em outros subVIs de mais baixo nível cada qual contendo uma das ações principais que o subVI de mais alto nível tem que executar no diagrama de blocos do LabVIEW.

#### 4.4.3 Projeto Bottom-Up

O projeto de sistema *bottom-up* pode ser útil como um método de desenvolvimento quando usado junto com o projeto *top-down*. Nesse tipo de projeto, constroem-se os componentes de mais baixo nível progredindo-se, então, para a hierarquia acima, a fim de se reunir os pedaços gradualmente até que se tenha o sistema acabado (WIKIMEDIA FOUNDATION, 2004; JALOTE, 2003).

O problema com o projeto *bottom-up* é que, como não se começa com uma idéia clara do sistema completo, pode-se construir pedaços que não se combinam corretamente.

Por sua vez, há casos específicos nos quais a utilização do projeto *bottom-up* é apropriada. Por exemplo, se o projeto está limitado por alguma funcionalidade de baixo nível,

pode-se construir primeiramente esta funcionalidade a fim de se adquirir uma idéia de como será utilizada (NATIONAL INSTRUMENTS, 2003c).

Na maioria dos casos, é bom que se utilize uma estratégia de projeto *top-down*, e se necessário, usar alguns componentes do projeto *bottom-up*.

#### 4.4.4 Aspectos Fundamentais do Projeto

Conforme MAZOLLA e FARINES (1998), alguns conceitos podem ser levados em conta durante a etapa de projeto. Desta forma, estes são discutidos a seguir.

##### 4.4.4.1 Abstração

De acordo com PRESSMAN (1987), à medida que se vai "descendo" nos níveis de abstração, a linguagem de representação vai se aproximando de questões de implementação, até que, no nível mais baixo, a solução é representada de modo a que possa ser derivada diretamente para uma linguagem de implementação.

Durante o projeto, partindo do projeto preliminar para o projeto detalhado, múltiplos níveis de abstração vão sendo definidos, aproximando-se cada vez mais da Implementação, que é o nível mais baixo de abstração.

##### 4.4.4.2 Refinamento

Esta técnica sugere como ponto de partida a definição da arquitetura do aplicativo a ser desenvolvido, sendo que esta vai sendo refinada sucessivamente até atingir níveis de detalhes procedimentais (PRESSMAN, 1987). Este processo dá origem a uma hierarquia de representações, onde uma descrição macroscópica de cada função vai sendo decomposta passo-a-passo até se obter representações bastante próximas de uma linguagem de implementação (realizado no item 6.2).

##### 4.4.4.3 Modularidade

A utilização do conceito de modularidade oferece resultados em curto prazo, uma vez que, ao dividir-se um grande problema em problemas menores, as soluções são encontradas com esforço relativamente menor.

##### 4.4.4.4 Hierarquia de Controle

A hierarquia de controle é a representação, usualmente sob a forma hierarquizada, da estrutura do aplicativo no que diz respeito aos seus componentes. O objetivo não é apresentar detalhes procedimentais ou de seqüenciamento entre processos, mas de estabelecer as relações entre os diferentes componentes do aplicativo, explicitando os níveis de abstração (refinamento) aos quais eles pertencem.

#### 4.4.4.5 Estrutura dos Dados

A estrutura dos dados representa os relacionamentos lógicos entre os diferentes elementos de dados individuais. À medida que o projeto se aproxima da implementação, esta representação assume fundamental importância, já que a estrutura da informação vai exercer um impacto significativo no projeto detalhado.

#### 4.4.4.6 Ocultação de Informação

De acordo com este princípio, os módulos devem ser decompostos para que as informações (os procedimentos e dados) contidas em cada módulo sejam inacessíveis aos módulos que não tenham necessidade destas informações. Ao realizar a decomposição conforme este princípio, o projetista proporciona um grau relativamente alto de independência entre os módulos, o que é altamente desejável em um projeto.

#### 4.4.5 Projeto com Múltiplos Desenvolvedores

Um dos principais desafios na fase de planejamento é estabelecer as áreas de projeto de cada desenvolvedor. Por exemplo, um complexo sistema de monitoração de dados pode ter um conjunto de VIs para exibir e manipular dados e outro conjunto de VIs para adquirir e salvar informações. Estes dois módulos são substanciais, não se sobrepõem e podem ser nomeados a diferentes desenvolvedores.

Inevitavelmente, há alguma interação entre os módulos. Um dos principais objetivos do trabalho das primeiras fases de desenvolvimento é projetar como esses módulos interagem entre si. O sistema de exibição de dados tem que ter acesso aos dados que precisam ser exibidos. O componente de aquisição precisa prover esta informação para o outro módulo. Em uma primeira fase do desenvolvimento, deve-se projetar os conectores dos VIs necessários para transferir informações entre os dois módulos. Igualmente, se houver estruturas de dados globais que devem ser compartilhadas, deve-se analisá-las e defini-las na primeira fase do projeto arquitetural, antes que os desenvolvedores comecem a trabalhar individualmente nos componentes.

Nas primeiras fases, cada desenvolvedor pode criar subVIs preliminares com a interface do conector definida para um módulo compartilhado. Este subVI preliminar pode não fazer nada, ou, se for um VI que devolve informação, pode gerar dados aleatórios. Esta flexibilidade permite cada membro da equipe de desenvolvimento continuar o desenvolvimento sem ter que esperar pela conclusão de outros módulos, além do que também torna fácil que os indivíduos executem módulos de testes de unidade como descrito no item 4.3.3.2.1.

Quando os componentes estiverem próximos da conclusão, pode-se integrar os módulos substituindo estes subVIs preliminares pelas reais contrapartes (NATIONAL

INSTRUMENTS, 2003c). Posteriormente, então, se pode executar testes de integração para verificar o trabalho do sistema como um todo.

#### 4.4.6 Prototipagem do Painel Frontal

De acordo com NATIONAL INSTRUMENTS (2003c), há vários modos para prototipar um sistema no LabVIEW. Em sistemas com requisitos de E/S difíceis de se satisfazer, pode-se desenvolver um protótipo para testar o controle e as taxas e laços de aquisição. Também, em protótipos de E/S, dados aleatórios podem simular dados adquiridos no sistema real.

Sistemas com muitos requisitos de interface com usuário são perfeitos para prototipagem. No papel, é difícil determinar-se o método a utilizar para exibir dados ou colocar um “prompt” de comando para o usuário. Então, em vez disso, deve-se avaliar os controles e os indicadores dos painéis frontais do VI necessários ao projeto. Além disso, é conveniente deixar o diagrama de blocos vazio e entender como os controles trabalham e como várias ações requerem outros painéis frontais.

Assim, deve-se evitar a implementação de diagramas de blocos nas primeiras fases de criação de protótipos, devendo-se apenas criar os painéis frontais com botões, caixas de lista etc., pensando-se em o que precisa acontecer quando o usuário realiza seleções. Para isso, pode-se fazer perguntas como:

- O botão deveria conduzir a outro painel frontal?
- Alguns controles no painel frontal deveriam ser ocultados e substituídos por outros?

Caso sejam apresentadas novas opções, deve-se seguir essas idéias criando-se novos painéis frontais para ilustrar os resultados. Este tipo de prototipagem ajuda a definir os requisitos para um projeto e dá uma idéia melhor de seu escopo.

Porém, a prototipagem não pode resolver todos os problemas de desenvolvimento. Tem que se ter cuidado em como se apresenta o protótipo aos clientes e não dar um senso errôneo de que se está tendo um rápido progresso no projeto. Por isso, tem-se que deixar claro ao cliente, seja ele um cliente externo ou outros membros de sua companhia, que este protótipo é estritamente para propósitos de projeto (NATIONAL INSTRUMENTS, 2003c).

Claro que a prototipagem do painel frontal só trata de componentes da interface com o usuário e não de restrições de E/S, tipos de dados ou questões do algoritmo no projeto. Identificar questões do painel frontal pode ajudar a melhor definir algumas destas áreas porque dá uma idéia das principais estruturas de dados que se precisa manter, no entanto não trata de todas estas questões. Para essas questões, precisa-se utilizar um dos outros métodos descritos nesta seção, como testes de desempenho ou projeto *top-down*.

#### 4.4.7 Testes de Desempenho

Para sistemas de E/S com requisitos de muitos pontos de dados ou de altas taxas de transferência, deve-se testar antecipadamente os componentes relacionados ao desempenho visto que o teste pode demonstrar que suposições de projeto estejam incorretas (NATIONAL INSTRUMENTS, 2003c).

Por exemplo, caso se planeje o emprego de um instrumento em um sistema de aquisição de dados, deve-se construir alguns testes simples os quais executem o tipo de E/S que se planeja utilizar. Enquanto as especificações podem indicar que o instrumento é capaz de controlar a aplicação que se está criando, pode-se achar que o *triggering* (disparo), por exemplo, leva muito mais tempo do que se espera. Pode-se achar também que o *switching* (chaveamento) entre canais com ganhos diferentes não podem ser feitos à taxa necessária sem se reduzir a exatidão de medição da amostragem, ou que embora o instrumento possa controlar as taxas, não se tem tempo o bastante por parte do software para a execução da análise desejada.

Um protótipo simples das partes de tempo crítico da aplicação pode revelar estes tipos de problemas. No LabVIEW também se pode averiguar os tempos médios de execução de um VI ou subVI, onde as causas de variações de cronometragem incluem interrupções do sistema, atualizações de tela, interação com o usuário e alocação de memória inicial.

### 4.5 Documentação para Desenvolvimento em LabVIEW

Ao se documentar VIs cuidadosamente, criam-se VIs utilizáveis e manuteníveis. A completa documentação reduz confusões e faz com que modificações futuras sejam feitas de modo mais fácil. Assim, o software que se desenvolve requer vários documentos, onde as duas principais categorias para esta documentação, conforme a NATIONAL INSTRUMENTS (2003c), são as seguintes:

- Documentação relacionada ao projeto – Inclui requisitos, especificações, planos de projeto detalhados, planos de teste e documentos do histórico das modificações.
- Documentação ao Usuário – Explica como usar o software.

O estilo de cada um destes documentos é diferente. O público da documentação relacionada ao projeto geralmente tem extenso conhecimento das ferramentas que se está documentando. O público da documentação do usuário tem menor grau de entendimento e experiência com o software.

O tamanho e o estilo de cada documento podem variar dependendo do projeto. Para ferramentas simples que são utilizadas *in-house* (onde foi desenvolvido), provavelmente não se precisa de muita documentação. Caso o produto esteja destinado a clientes externos, tem-se que despende uma quantia significativa de tempo para desenvolver detalhadamente



a documentação orientada ao usuário a qual dá suporte ao produto. Ainda, para produtos que têm que passar por um processo de certificação de qualidade tem-se que assegurar que a documentação relacionada ao projeto satisfaz todos os requisitos.

#### 4.5.1 Projeto e Desenvolvimento de Documentação

O formato e o nível de detalhe da documentação desenvolvida é determinado pelas metas de qualidade do projeto. Se o projeto tem que atingir uma norma de qualidade como a ISO 9000, o formato e o nível de detalhe da documentação é diferente do formato e do nível de detalhe de um projeto *in-house*.

Para tanto, o LabVIEW inclui características que simplificam o processo de criação de documentação para os VIs que se projeta:

- Janela para registrar mudanças feitas em um VI.
- Caixa de diálogo de impressão a qual serve para criar cópias impressas do painel frontal, diagrama de blocos, conectores e descrição de um VI. Também se pode utilizar esta para imprimir os nomes e descrições de controles e indicadores de um VI e os nomes e caminhos de quaisquer subVIs. Pode-se imprimir esta informação, gerar arquivos de texto, arquivos HTML ou RTF para criar arquivos de ajuda compilados.

#### 4.5.2 Desenvolvimento da Documentação do Usuário

Organizar a documentação sistematicamente ajuda os usuários a aprender sobre seu produto, VI ou aplicação. Diferentes usuários têm diferentes necessidades de documentação. Esta seção propõe técnicas para criar e organizar a documentação que ajuda estas diferentes classes de usuários. O formato da documentação do usuário depende do tipo de produto que se cria.

##### 4.5.2.1 Organização Sistemática da Documentação

Para se fazer uma documentação mais útil ao usuário, deve-se organizar a documentação de um modo sistemático. Pode-se dividir a informação de ajuda em três categorias: conceitos, procedimentos e material de referência. De tal modo, NATIONAL INSTRUMENTS (2003c) afirma que a documentação criada deve refletir estas três categorias.

##### 4.5.2.2 Documentação de uma Aplicação

Caso se esteja desenvolvendo uma aplicação para usuários pouco experientes com o LabVIEW, a documentação requer um material mais introdutório. Pode-se criar um documento que provê requisitos de sistema, instruções básicas de instalação e uma visão geral

sobre como os pacotes funcionam. Se o pacote usar E/S, deve-se incluir requisitos de hardware e alguma instrução de configuração que o usuário tem que realizar antes de utilizar a aplicação.

Para cada painel frontal com que o usuário interage, deve-se prover uma visão do painel frontal e uma descrição dos principais controles e indicadores.

#### 4.5.3 Criação de Arquivos de Ajuda

Pode-se criar ajuda on-line ou documentos de referência caso se tenham as ferramentas de desenvolvimento certas. Documentos de ajuda on-line são documentos de texto formatados que se pode criar utilizando um programa de processamento de texto, como o Microsoft Word e as ferramentas que compilam ajuda. Também se pode criar documentos de ajuda on-line em HTML com um compilador de ajuda HTML.

Depois de criar os documentos fontes, utiliza-se um compilador de ajuda para criar o documento de ajuda. Após criar e compilar os arquivos de ajuda, acrescenta-os ao menu de Ajuda do LabVIEW ou à aplicação colocando-os no seu diretório de ajuda (NATIONAL INSTRUMENTS, 2003c).

#### 4.5.4 Descrição de VIs, Controles e Indicadores

De acordo com NATIONAL INSTRUMENTS (2003b), pode-se agregar informação ao usuário em cada VI que se cria utilizando-se a característica de descrição do VI, colocando-se instruções no painel frontal e incluindo-se descrições de cada controle e indicador.

##### 4.5.4.1 Criação das Descrições do VI

Deve-se incluir os seguintes itens em uma descrição de VI:

- uma visão geral do VI;
- instruções sobre como usar o VI;
- descrições das entradas e saídas.

##### 4.5.4.2 Documentação dos Painéis Frontais

Um modo de prover importantes instruções, de acordo com NATIONAL INSTRUMENTS (2003b), é colocar um bloco de texto proeminentemente no painel frontal. Uma lista de passos importantes é valiosa. Caso um bloco de texto tome muito espaço no painel frontal, ao invés disso pode-se utilizar um botão de ajuda no painel frontal.

#### 4.5.4.3 Criação das Descrições de Controles e Indicadores

Em NATIONAL INSTRUMENTS (2003c) é apresentado que se deve incluir uma descrição para cada controle e indicador, os quais precisam de uma descrição que inclui as seguintes informações:

- funcionalidade;
- tipo de dados;
- faixa de valores válida (para entradas);
- valor padrão (para entradas).

Ao mesmo tempo se pode listar o valor padrão em parênteses como parte do nome do controle ou do indicador.

## CAPÍTULO 5

### CONCEITUAÇÃO E MODELAGEM DE SISTEMAS AUTOMÁTICOS

Visto que o aplicativo desenvolvido está incluso em um equipamento (Plataforma de Hidráulica Proporcional) que se caracteriza por ser um sistema automático, é apresentado neste documento o emprego de uma metodologia própria para sistemas automáticos com o intuito de se alcançar as funções as quais o aplicativo deve contemplar. Esta metodologia é exibida neste capítulo.

#### 5.1 Conceituação de Sistemas Automáticos

O termo sistema automático ou mecatrônico tem sido adotado para designar uma aplicação que envolva automação e/ou controle (DE NEGRI, 2004). Ou seja, pode-se observar o problema segundo uma visão lógica (automação) ou então de maneira mais aprofundada, avaliando, ao longo do tempo, a resposta da posição, força, velocidade, vazão ou qualquer outra variável (controle).

Existem vários artigos que discutem a definição de Mecatrônica, porém se verifica que o ponto comum a maioria das abordagens é, mais que a simples soma, a integração de diferentes tecnologias.

Atualmente a Mecatrônica é entendida como uma filosofia relacionada à aplicação combinada de conhecimentos de áreas tradicionais como a Engenharia Mecânica, Eletrônica e Computação de forma integrada e concorrente, conforme mostra a figura 5.1. Como afirmam ADAMOWSKI e FURUKAWA (2001), uma combinação para ser concorrente deve extrair o que há de mais adequado em cada uma das áreas, de tal forma que o resultado final é, mais do que a simples soma de tais especialidades, uma sinergia entre elas.

Assim, o conceito de Mecatrônica representa a combinação adequada dessas áreas, possibilitando as seguintes características no projeto:

- simplificação do sistema mecânico;
- redução de tempo e de custo de desenvolvimento;
- facilidade de se introduzir modificações ou novas capacidades;
- flexibilidade para receber futuras modificações ou novas funcionalidades.

Segundo COSTE *et. al* (2001), o projeto de um sistema complexo pode requerer o uso de diferentes linguagens. O sistema automático é um típico domínio deste tipo de projeto, onde suas atividades ocorrem em diferentes domínios tecnológicos e, dependendo da amplitude do sistema, haverá o envolvimento de diversos especialistas, cada qual capaz de resolver aspectos técnicos em sua área do conhecimento (DE NEGRI, 1997).

Para cada um dos subsistemas que compõem um sistema automático, modelos formais que suportam modelagem, análise ou programação têm sido desenvolvidos. LIU (2000)

afirma que a mixagem de modelos heterogêneos para projetar sistemas complexos está recebendo cada vez mais atenção do mundo acadêmico e industrial, e STAUNER (2001) afirma que técnicas de descrição gráfica provêm um meio muito útil dar suporte à comunicação entre estes modelos.

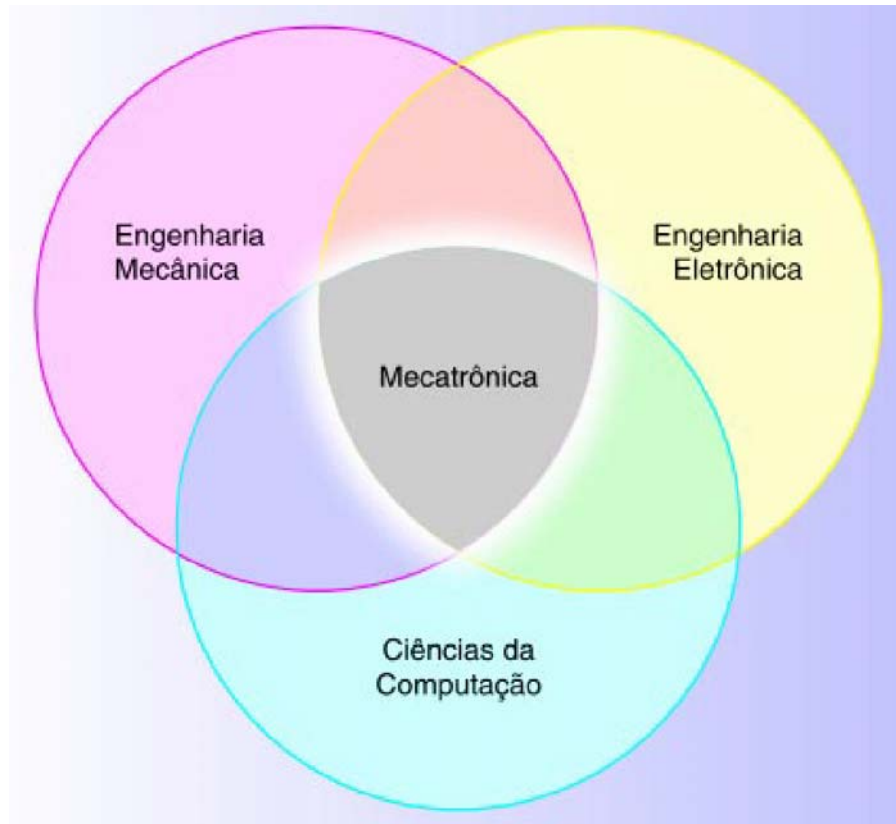


Figura 5.1 – Uma representação esquemática da mecatrônica (ADAMOWSKI; FURUKAWA, 2001).

Conforme CORTÉS *et. al* (1999), um modelo precisa capturar as características do sistema e descrever sua funcionalidade. Esta descrição, de acordo com CARRO (2003), deve ser neutra em relação a possíveis implementações das funções em software ou em hardware.

Como DE NEGRI (1996), entende-se que a integração de todos os especialistas é possível por meio de uma metodologia própria para sistemas automáticos que contemple, dentro do possível, o processo de projeto de produtos e o projeto de software. De tal modo, o presente trabalho emprega a modelagem exposta em seguida, a qual integra técnicas de descrição gráfica para sistemas automáticos e busca ser neutra quanto a possíveis implementações.

## 5.2 Projeto de Sistemas Automáticos

A modelagem de sistemas automáticos deve contemplar três perspectivas principais: função, comportamento e estrutura (DE NEGRI, 1996).

A perspectiva funcional, por meio da rede Canal/Agência, é empregada para caracterizar genericamente os sistemas automáticos enquanto os modelos comportamentais e estruturais mais concretos, como diagramas de blocos, desenhos esquemáticos e diagramas de circuitos, possibilitam o detalhamento dos sistemas de medição e de atuação.

Os modelos citados acima, juntamente com outros aplicáveis às diversas áreas técnicas que compõe um sistema automático, são ferramentas importantes para projeto, análise ou para uma mera apresentação de um sistema. O uso desses modelos torna-se evidente através do processo de projeto, visto como um conjunto de atividades que busca, a partir da identificação das necessidades, alcançar as especificações detalhadas para a construção ou implementação de um sistema técnico.

No item 6.2 ocorre a aplicação de uma metodologia própria para sistemas automáticos, possibilitando alcançar as funções que o aplicativo deverá contemplar. Subseqüentemente, a implementação dessas funções é realizada seguindo-se aspectos da engenharia de software, como descritos no capítulo 4.

### **5.3 Modelagem Genérica de Sistemas Automáticos**

Para que se possa projetar, instalar ou desempenhar a manutenção de um sistema automático, isto é, de um sistema de automação e/ou controle, é importante que se compreenda claramente quais suas partes principais e como são interligadas.

Independentemente do domínio da aplicação e da complexidade deste, um sistema automático pode ser decomposto em dois subsistemas: um sistema de informação e um sistema energético/material, segundo representado na figura 5.2.

O sistema de informação engloba os equipamentos que processam sinais e dados, tais como computadores, controladores lógicos programáveis, controladores analógicos e digitais, válvulas de processamento de sinais, entre outros.

Por sua vez, o sistema energético/material (processo físico ou químico) abrange máquinas, dispositivos e equipamentos que transformam ou processam energia e/ou matéria.

Assim, conforme indicado na figura 5.2, o sistema de informação deve ser capaz de extrair informações do sistema energético/material, processá-las e, posteriormente, utilizá-las para alterar o funcionamento deste. Além da troca de informações entre esses dois subsistemas, há também o recebimento e fornecimento de energia (ene), matéria (mat) e informação (inf) em relação ao ambiente externo.

O modelo geral de um sistema automático ou mecatrônico apresentado na figura 5.2 emprega a notação de rede de Petri Canal/Agência (Rede C/A), detalhada em DE NEGRI (1996) e HEUSER (1990), a qual utiliza dois elementos básicos:

- os canais, representados por círculos, que indicam os elementos passivos (recursos energéticos, materiais e de informação);

- as agências, representadas por retângulos, que indicam os elementos ativos (processos) que consomem e transformam os recursos.

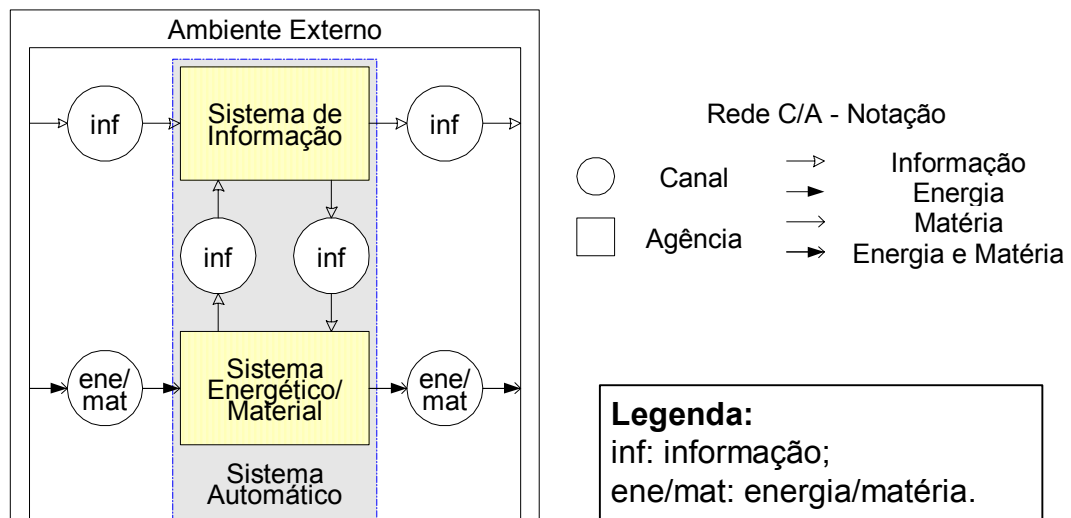


Figura 5.2 – Representação de um sistema automático (DE NEGRI, 1996).

Esses elementos são conectados através de arcos direcionados que determinam o sentido do fluxo dos recursos (DE NEGRI, 1996).

#### 5.4 Modelo Detalhado de Sistema Automático

A figura 5.3 consiste em um detalhamento da figura 5.2 onde são evidenciados alguns subsistemas, que compõem os sistemas energético/material e de informação, dando destaque aos relacionados diretamente com os canais de informação existentes dentro do sistema automático, ou seja:

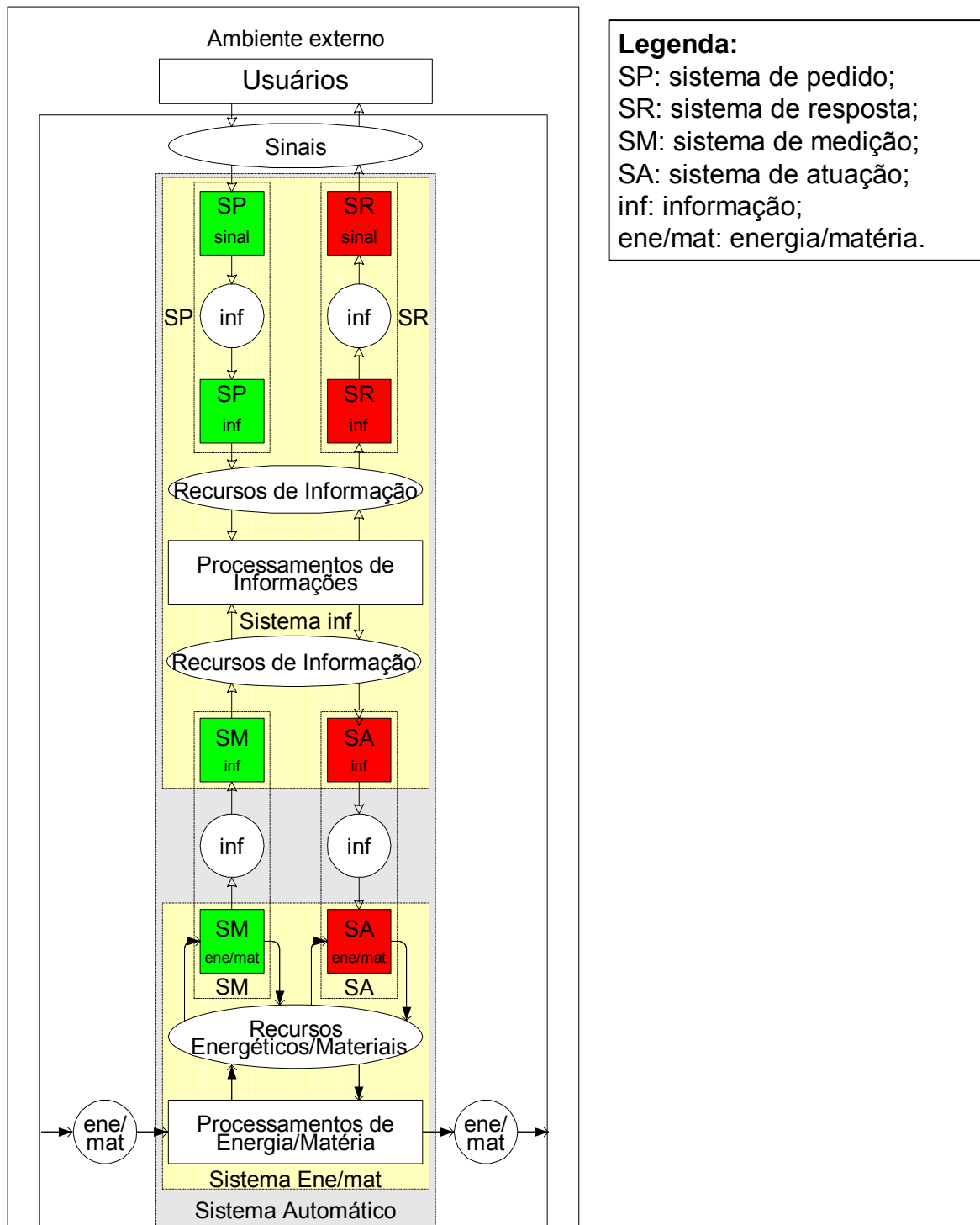
O *sistema de medição (SM)* que sofre a ação do fluxo de energia, extraindo uma informação relativa ao valor de uma variável física;

O *sistema de atuação (SA)* que cumpre a tarefa oposta, ou seja, recebe uma informação quanto ao valor da variável física a ser ajustada, processa-a e por fim atua sobre o fluxo energético ou material.

O *sistema de pedido (SP)* que sofre a ação de sinais, extraindo uma informação relativa aos desejos dos usuários.

O *sistema de resposta (SR)* que, de modo oposto, recebe uma informação relativa aos desejos dos usuários e o explicita sob forma de sinais.

Portanto, a figura 5.3 estabelece um modelo funcional e também fixa a estrutura geral de um sistema automático, servindo como ponto de partida para o processo de projeto evoluir através do refinamento desta rede C/A segundo regras simples e bem definidas (DE NEGRI, 1996).



**Legenda:**  
 SP: sistema de pedido;  
 SR: sistema de resposta;  
 SM: sistema de medição;  
 SA: sistema de atuação;  
 inf: informação;  
 ene/mat: energia/matéria.

Figura 5.3 – Modelo funcional/estrutural refinado de um sistema automático (SOUTO, 2005).



## CAPÍTULO 6

### MODELAGEM DO APLICATIVO

Como comentado anteriormente no item 1.2, o presente trabalho de dissertação compreende a elaboração de um aplicativo que se enquadra nos objetivos do projeto da Plataforma de Hidráulica Proporcional, realizando o controle de posição de sistemas eletro-hidráulicos e também implementando o diagrama sistematizado de procedimentos para dimensionamento e análise estática e dinâmica de circuitos hidráulicos, diagrama o qual foi desenvolvido por FURST (2001).

Por isto, neste capítulo, é delineado o processo de modelagem do Módulo de Trabalho<sup>21</sup>. Posteriormente, a modelagem do aplicativo exibida.

#### 6.1 Concretização do Sistema Automático

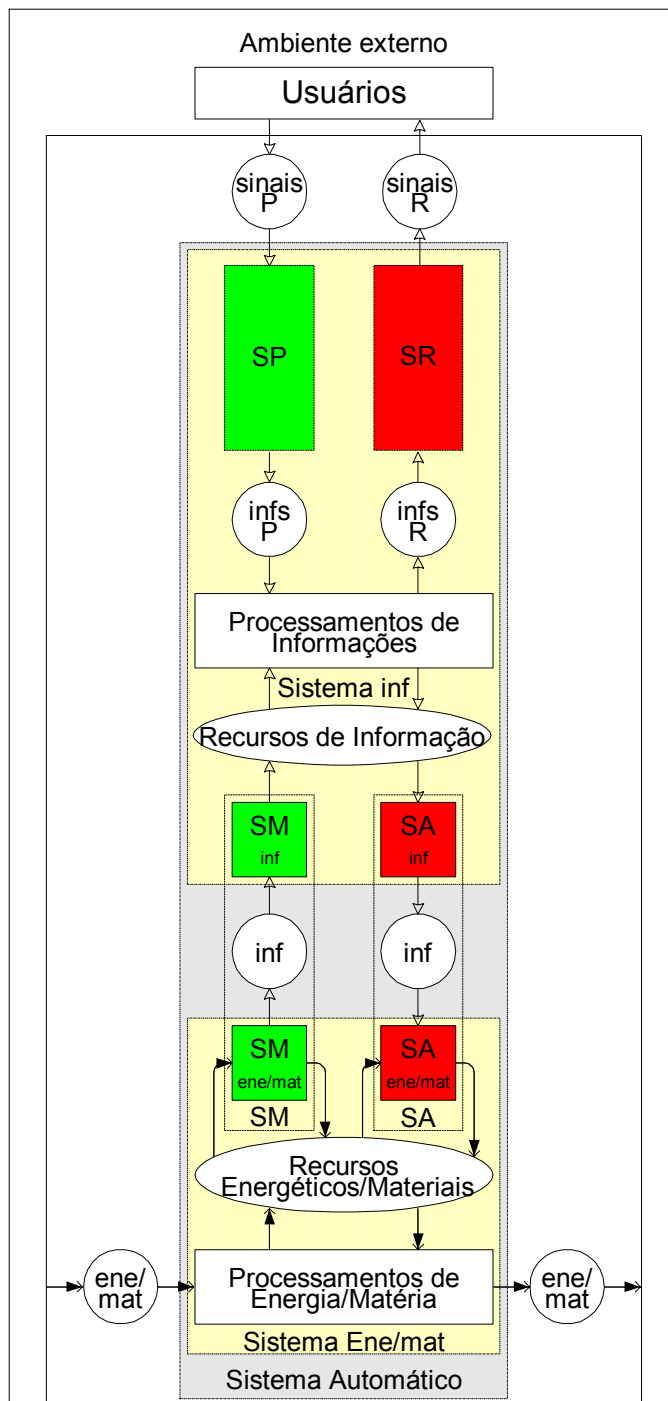
A partir do modelo da figura 5.3 chegou-se ao modelo apresentado figura 6.1 a fim de, através do agrupamento de agências e divisão de canais, se obter um modelo mais apropriado, visto que no desenvolvimento deste aplicativo o que importa é o que necessita-se como informação, sendo o sinal apenas um reflexo desta informação processada por computador. Isto porque, nos conjuntos de agências “SP sinal” com “SP inf” e “SR sinal” com “SR inf” que estão sendo condensados, as agências “SP inf” e “SR inf” significam *drivers* de computador, sendo que o programador/desenvolvedor não precisa trabalhar nesse nível, como será mostrado no item 6.2.

Correlacionando o modelo funcional/estrutural refinado de um sistema automático (figura 6.1) com a ilustração da figura 1.3 e o diagrama da figura 1.1, concretizou-se o modelo apontado na figura 6.2.<sup>22</sup>

---

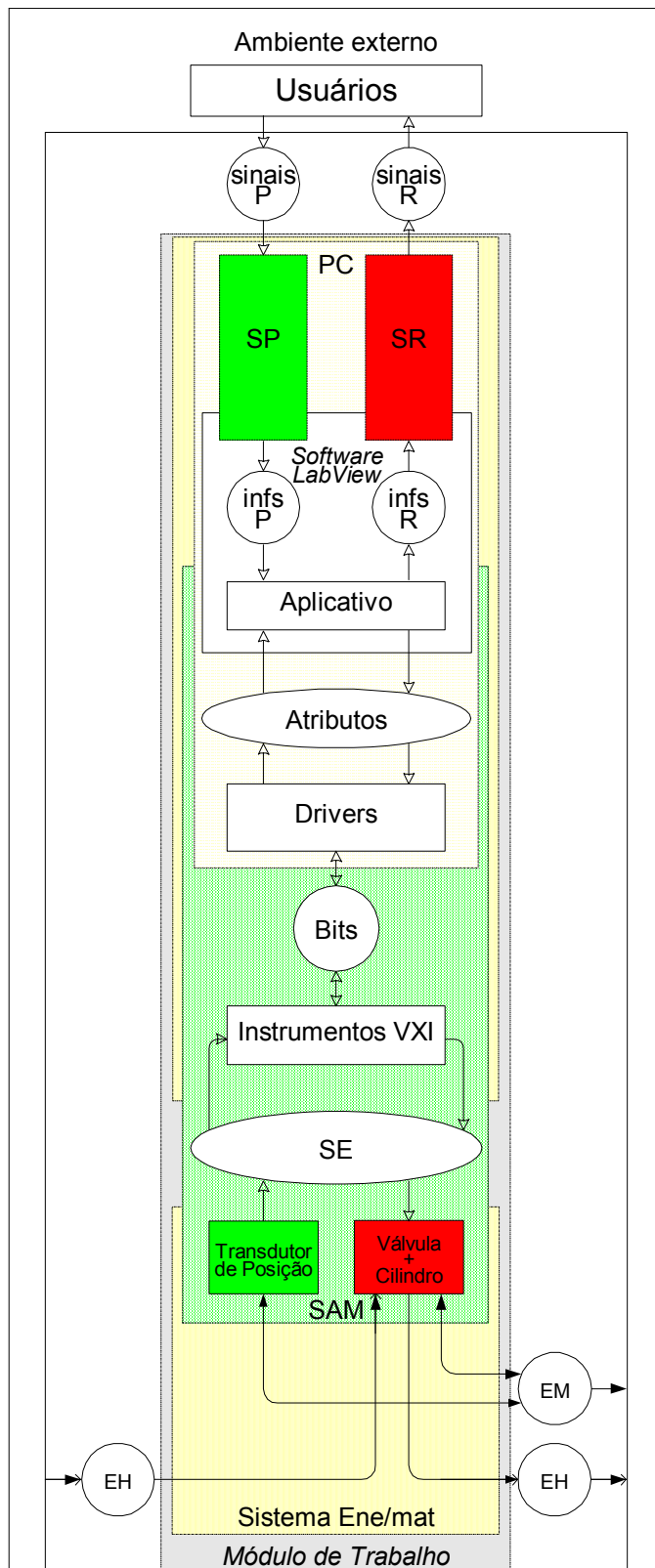
<sup>21</sup> A modelagem do Módulo de Trabalho serve exclusivamente para a implementação do controle de posição de sistemas eletro-hidráulicos. A implementação do diagrama sistematizado de procedimentos para dimensionamento e análise estática e dinâmica de circuitos hidráulicos será mostrada somente no item 7.7.

<sup>22</sup> Poderia este modelo ter um maior ou menor grau de abstração. Se não o teve, no entanto, foi devido ao julgamento de que este modelo é adequado para que a modelagem funcional/estrutural do aplicativo (apresentada no item 6.2) prospere.



**Legenda:**  
 sinais P: sinais de pedido;  
 sinais R: sinais de resposta;  
 infs P: informações de pedido;  
 infs R: informações de resposta;  
 SP: sistema de pedido;  
 SR: sistema de resposta;  
 SM: sistema de medição;  
 SA: sistema de atuação;  
 inf: informação;  
 ene/mat: energia/matéria.

Figura 6.1 – Modelo funcional/estrutural refinado proposto de um sistema automático.



**Legenda:**  
 sinais P: sinais de pedido;  
 sinais R: sinais de resposta;  
 infs P: informações de pedido;  
 infs R: informações de resposta;  
 SE: Sinais Elétricos;  
 EM: Energia Mecânica;  
 EH: Energia Hidráulica;  
 SAM: Sistema de Atuação e Medição;  
 PC: Computador Pessoal.

Figura 6.2 – Modelo preliminar do Módulo de Trabalho.<sup>23</sup>

Através da análise da figura 6.2, denota-se que:

<sup>23</sup> Percebe-se que o “SAM” deste modelo tem uma forma equivalente ao sistema de controle por computador apresentado na figura 3.1.

- “Usuários” – Corresponde aos dois usuários que podem operar o Módulo de Trabalho.
- “infs P” – Corresponde às informações relacionadas aos sinais de pedido dos usuários.
- “infs R” – Corresponde às informações relacionadas aos sinais de resposta aos usuários.
- “Aplicativo” – Corresponde ao aplicativo a ser desenvolvido, em LabVIEW.
- “Atributos” – Corresponde às informações trocadas entre o aplicativo e a placa PCI.
- “Drivers” – Corresponde a um conjunto de rotinas o qual permite que os atributos do aplicativo sejam levados em forma de bits aos instrumentos VXI e, de modo oposto, que os bits sejam levados ao aplicativo em forma de atributos.
- “Bits” – Corresponde às informações trocadas entre a placa PCI e os instrumentos VXI.
- “Instrumentos VXI” – Corresponde aos instrumentos: conversor D/A, multímetro, multiplexador e chaves digitais.
- “SE” – Corresponde aos sinais elétricos trocados entre instrumentos VXI e o transdutor de posição e a válvula + cilindro.
- “Válvula + Cilindro” – Corresponde ao sistema de atuação hidráulico do Módulo de Trabalho. Apesar da figura 6.2 apresentar apenas um conjunto de Válvula + Cilindro, estes são em um número de dois conjuntos. Sendo, assim, um conjunto para cada lado da bancada.
- “Transdutor de Posição” – Corresponde ao sistema de medição do Módulo de Trabalho, relativo à posição do cilindro hidráulico. Novamente, apesar da figura 6.2 apresentar apenas um transdutor de posição, há um total de dois transdutores. Sendo, deste modo, um para cada lado da bancada.
- “EH” – Corresponde aos recursos de energia hidráulica referente ao sistema de atuação hidráulico (válvula + cilindro).
- “EM” – Corresponde aos recursos de energia mecânica gerados com deslocamento (translação) do sistema de atuação hidráulico (válvula + cilindro). Este deslocamento é medido pelo “Transdutor de Posição”.

Por conseguinte, um detalhamento do canal “Instrumentos VXI” exibido na figura 6.2 está indicado na figura 6.3.

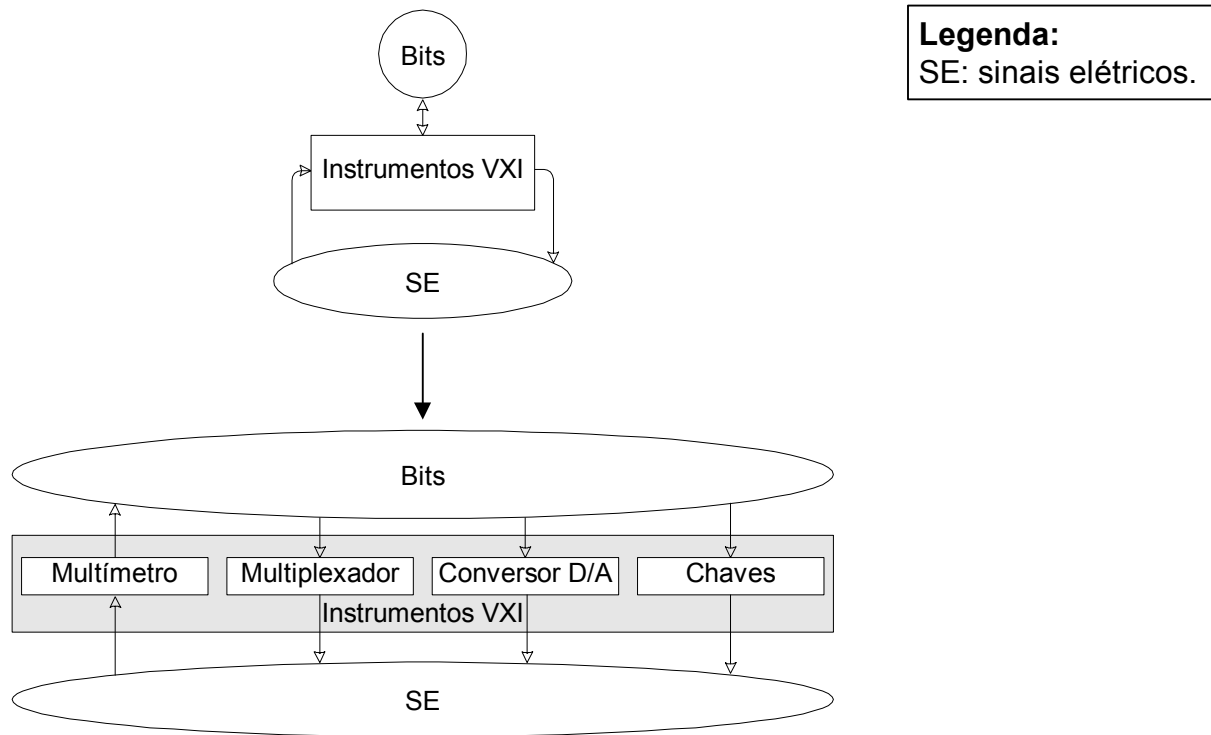


Figura 6.3 – Modelo detalhado de Instrumentos VXI. <sup>24</sup>

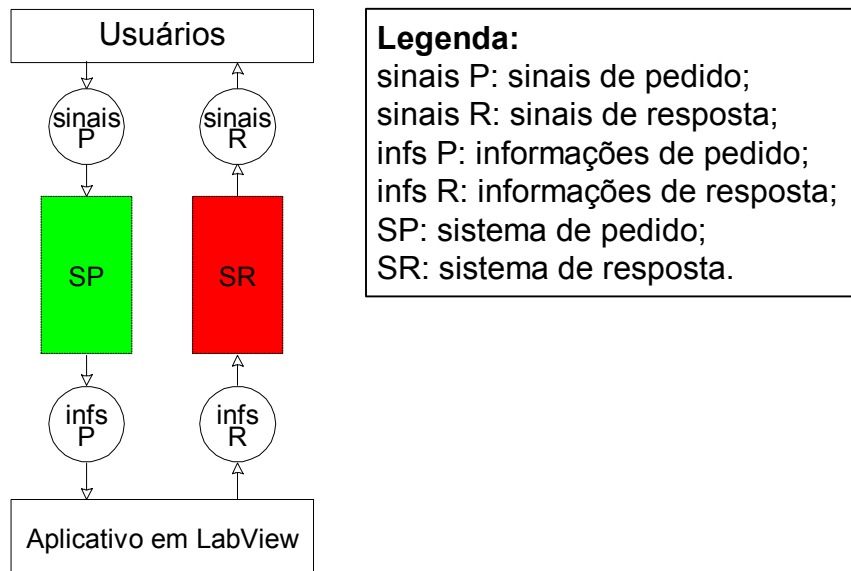
## 6.2 Modelagem Funcional/Estrutural do Aplicativo

A figura 6.4, extraída da figura 6.2, é uma visão geral da relação entre os usuários e o aplicativo em LabVIEW sendo que, na Plataforma de Hidráulica Proporcional, têm-se dois usuários que podem realizar experimentos na Bancada de Trabalho, cada qual a partir de seu computador<sup>25</sup>.

De tal forma, pode-se refiná-la para se obter a figura 6.5. Assim, informações de pedido (infs P) são feitas pelo *Usuário 1* e pelo *Usuário 2* ao aplicativo e informações de resposta (infs R) são dadas pelo aplicativo para o *Usuário 1* e para o *Usuário 2*.

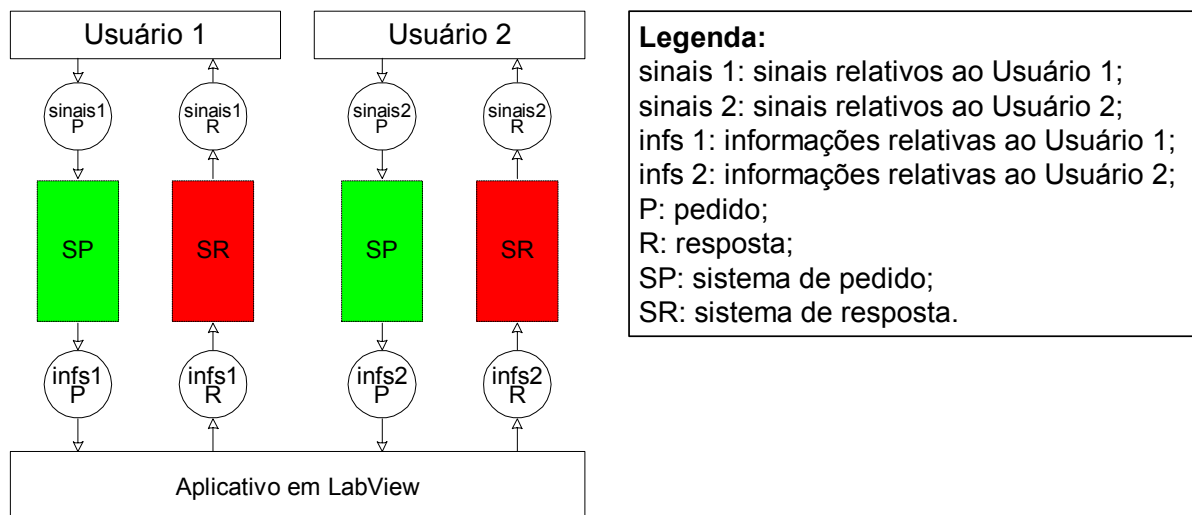
<sup>24</sup> O elemento denominado “Chaves” indica as saídas digitais do sistema VXI (E1463A).

<sup>25</sup> Como se percebe na figura 6.2, apenas um computador (PC) está presente na modelagem do sistema. Isto se deve ao fato de que o importante na modelagem são os Usuários, sendo o computador (PC) apenas um meio destes usuários executarem as funções desejadas. Se os módulos, descritos a seguir, estão implementados em um ou dois computadores, pouco importa. Estes são aspectos de implementação do software e não de sua modelagem. Referir-se ao apêndice 2 para explicações.



**Legenda:**  
 sinais P: sinais de pedido;  
 sinais R: sinais de resposta;  
 infs P: informações de pedido;  
 infs R: informações de resposta;  
 SP: sistema de pedido;  
 SR: sistema de resposta.

Figura 6.4 – Visão geral da relação entre os usuários e o software em LabVIEW.



**Legenda:**  
 sinais 1: sinais relativos ao Usuário 1;  
 sinais 2: sinais relativos ao Usuário 2;  
 infs 1: informações relativas ao Usuário 1;  
 infs 2: informações relativas ao Usuário 2;  
 P: pedido;  
 R: resposta;  
 SP: sistema de pedido;  
 SR: sistema de resposta.

Figura 6.5 – Visão refinada dos usuários da Bancada de Trabalho.

A partir das informações de pedido e de respostas relacionadas aos usuários, como indicado na figura 6.6, pode-se obter o modelo preliminar do aplicativo<sup>26</sup>, o qual é composto por quatro módulos:

- “Interface com Usuário (Usuário 1)”: responsável por interagir com as informações que o *Usuário 1* envia e recebe por meio do computador do “Lado 1”. Estas informações estão listadas no item 6.2.1.
- “Interface com Usuário (Usuário 2)”: responsável por interagir com as informações que o *Usuário 2* envia e recebe por meio do computador do “Lado 2”.

<sup>26</sup> O apêndice 3 apresenta detalhadamente a lista completa das informações relativas ao software.

Essas informações estão listadas no item 6.2.1 e são idênticas, a não pelo fato de serem referentes ao *Usuário 2*, às do *Usuário 1*.

- “Gerenciador”: responsável por gerenciar as informações sobre as escolhas dos usuários (*Usuário 1* e *Usuário 2*) referentes às válvulas direcionais proporcionais e às válvulas reguladoras de pressão utilizadas na Bancada de Trabalho. Este gerenciamento determina quais as saídas do conversor D/A e quais as chaves digitais do sistema VXI serão utilizadas no experimento de acordo com as escolhas dos usuários. Para tanto, este módulo está melhor explicado no item 6.2.3.
- “Controlador”: responsável por controlar a posição do cilindro hidráulico, este módulo realiza o controle de posição do cilindro hidráulico através da informação do sinal de referência e demais informações providas do (s) módulo (s) de interface com o usuário (*Usuário 1* e/ou *Usuário 2*). Este módulo está detalhado no item 6.2.4.

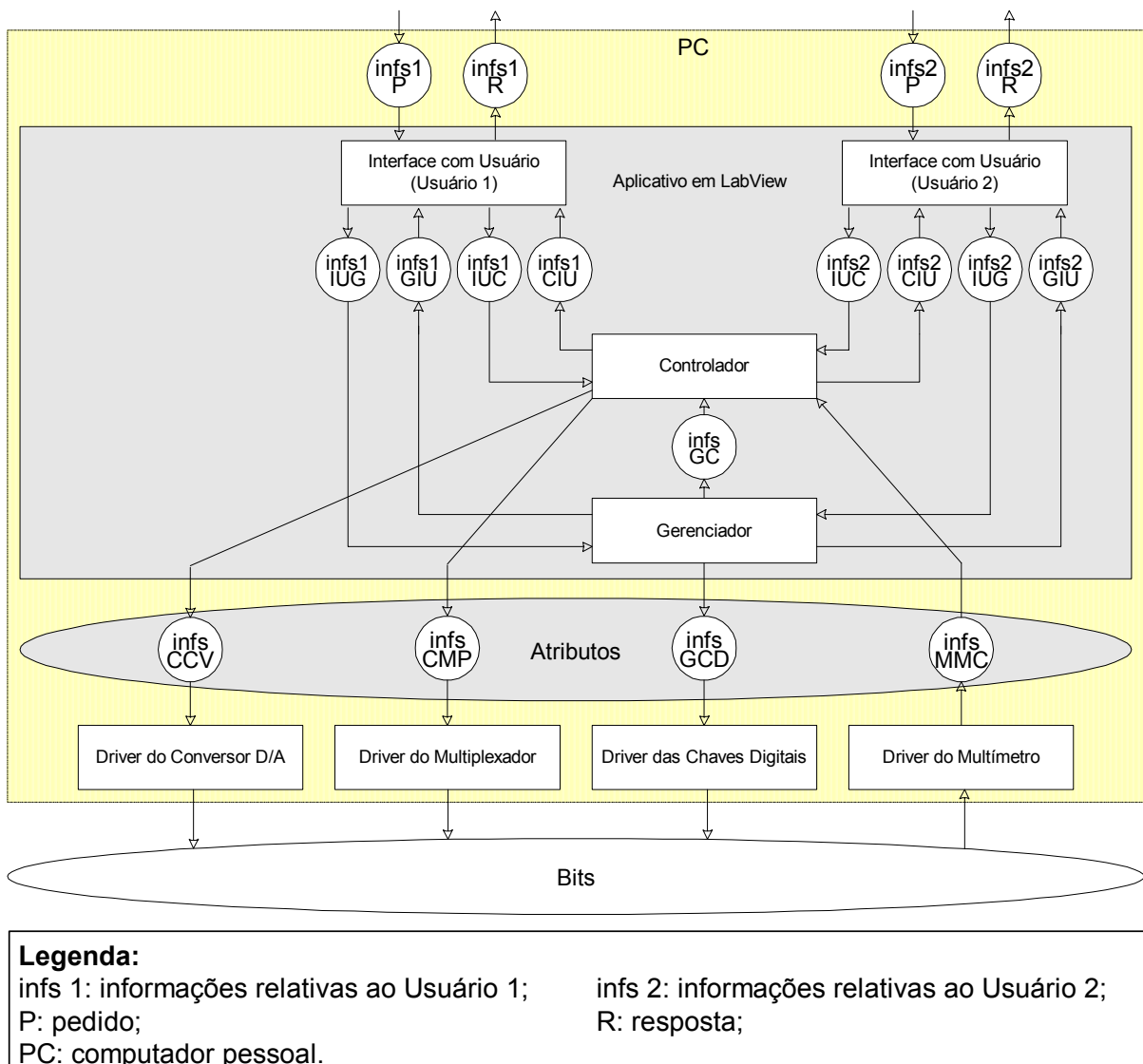


Figura 6.6 – Modelagem preliminar do aplicativo.

## 6.2.1 Módulo “Interface com Usuário (Usuário 1)”

A tabela 6.1 lista, de forma resumida, as informações relativas a este módulo. Por isso, maiores informações devem ser pesquisadas no apêndice 3.

Tabela 6.1 – Lista de informações para o módulo “Interface com Usuário (Usuário 1)”

“infs1 IUG”	Informações do módulo "Interface com Usuário (Usuário 1)" para o módulo "Gerenciador".
“infs1 GIU”	Informações do módulo "Gerenciador" para o módulo "Interface com Usuário (Usuário 1)".
“infs1 IUC”	Informações do módulo "Interface com Usuário (Usuário 1)" para o módulo "Controlador".
“infs1 CIU”	Informações do módulo "Controlador" para o módulo "Interface com Usuário (Usuário 1)".
“infs1 P”	Informações de Pedido do <i>Usuário 1</i> .
“infs1 R”	Informações de Resposta do <i>Usuário 1</i> .

O módulo “Interface com Usuário (Usuário 1)”, mostrado na figura 6.7, é responsável por enviar as informações “infs1 IUG” e “infs1 IUC” aos módulos “Gerenciador” e “Controlador” respectivamente a partir das informações “infs1 P”. Também, é responsável por adquirir as informações “infs1 GIU” e “infs1 CIU” dos módulos “Gerenciador” e “Controlador” respectivamente e subseqüentemente apontar as informações “infs1 R”.

Desta maneira, as informações de pedido do *Usuário 1* (“infs1 P”), como detalhadas no apêndice 3, compreendem:

- “Válvula da Bancada” e “Válvula Reguladora”<sup>27</sup> a serem utilizados pelo *Usuário 1* no experimento da bancada. Estas escolhas implicam na seleção, realizada automaticamente no módulo “Gerenciador”, das saídas digitais (chaves digitais). De tal modo, a seleção das saídas digitais são detalhadas no item 6.2.3.
- “Frequência (Hz)”, “amplitude”, “fase”, “offset” e “tipo do sinal” que se deseja para o sistema de atuação hidráulico.
- “Intervalo de amostragem (ms)”, “Ganho P” e “Ti” que o *Usuário 1* deseja utilizar para controlar o sistema de atuação hidráulico

---

<sup>27</sup> O termo “Válvula da Bancada” refere-se às válvulas direcionais proporcionais enquanto o termo “Válvula Reguladora” refere-se às válvulas reguladoras de pressão da Bancada de Trabalho.



- “Canal do Multiplexador” e “Canal do Conversor D/A”<sup>28</sup> que o *Usuário 1* deseje utilizar. Isto devido ao multiplexador e ao conversor D/A serem instrumentos compartilhados entre o *Usuário 1* e *Usuário 2*.

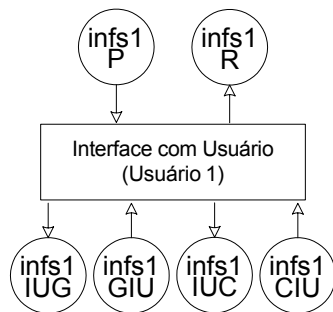


Figura 6.7 – Módulo “Interface com Usuário (Usuário 1)”.

Em um primeiro detalhamento do módulo “Interface com Usuário (Usuário 1)”, pode-se evidenciar as funções “Envio de Dados” e “Aquisição de Dados” como mostrado na figura 6.8.

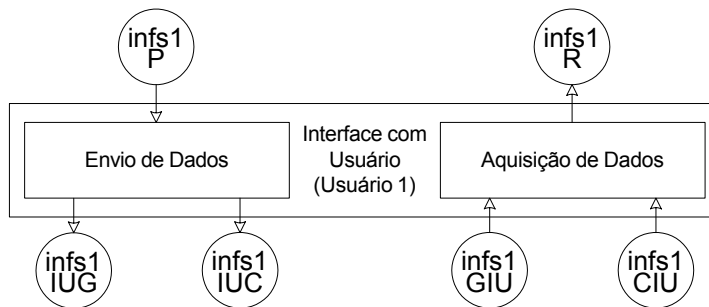


Figura 6.8 – Primeiro detalhamento do módulo “Interface com Usuário (Usuário 1)”.

Por sua vez, em um segundo detalhamento do módulo “Interface com Usuário (Usuário 1)”, pode-se evidenciar que a função “Envio de Dados” pode ser decomposta em “Envio de Dados ao Controlador” e “Envio de Dados ao Gerenciador” enquanto que a função “Aquisição de Dados” pode ser decomposta em “Aquisição de Dados do Controlador” e “Aquisição de Dados do Gerenciador” como mostrado na figura 6.9.

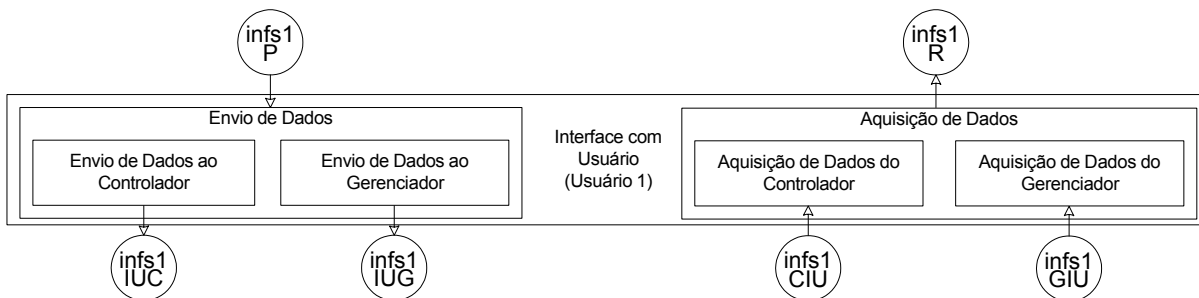


Figura 6.9 – Segundo detalhamento do módulo “Interface com Usuário (Usuário 1)”.

<sup>28</sup> O “Canal do Conversor D/A” é escolhido pelo *Usuário 1* somente no modo manual. Por padrão, este parâmetro é escolhido de modo automático, como será explicado no item 6.2.3.

Além disso, a função “Envio de Dados ao Controlador” pode ser decomposta em “Envio do Sinal de Referência” e “Envio dos Parâmetros de Escolha” enquanto a função “Envio de Dados ao Gerenciador” pode ser detalhada como “Envio de Dados de Escolha de Válvulas” (figura 6.10).

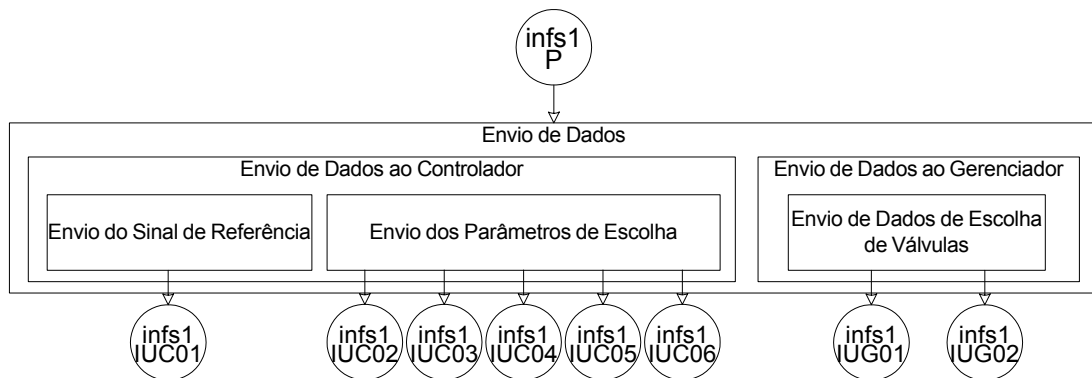


Figura 6.10 – Detalhamento da função “Envio de Dados”.

Finalmente, a função “Aquisição de Dados do Controlador” pode ser detalhada como “Aquisição de Dados para Mostrar no Gráfico” enquanto a função “Aquisição de Dados do Gerenciador” pode ser detalhada como “Aquisição de Dados de Desabilitação de Escolha de Válvulas” (figura 6.11).

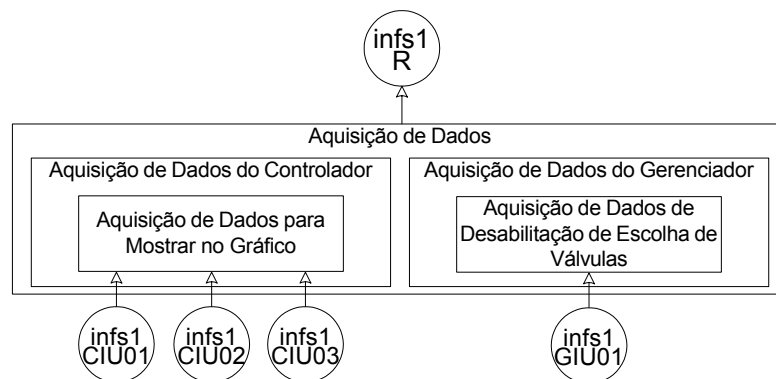


Figura 6.11 – Detalhamento da função “Aquisição de Dados”.

### 6.2.2 Módulo “Interface com Usuário (Usuário 2)”

Por este módulo ser similar ao módulo “Interface com Usuário (Usuário 1)” apresentado no item 6.2.1, a não ser pelo fato de se referenciar ao Usuário 2, este não será aqui delineado.

### 6.2.3 Módulo “Gerenciador”

A tabela 6.2 lista, de forma resumida, as informações relativas a este módulo. Por isso, maiores informações devem ser pesquisadas no apêndice 3.

Tabela 6.2 – Lista de informações para o módulo “Gerenciador”

“ifs1 GIU”	Informações do módulo "Gerenciador" para o módulo "Interface com Usuário (Usuário 1)".
“ifs2 GIU”	Informações do módulo "Gerenciador" para o módulo "Interface com Usuário (Usuário 2)".
“ifs1 IUG”	Informações do módulo "Interface com Usuário (Usuário 1)" para o módulo "Gerenciador".
“ifs2 IUG”	Informações do módulo "Interface com Usuário (Usuário 2)" para o módulo "Gerenciador".
“ifs GCD”	Informações do módulo "Gerenciador" para o "Driver das Chaves Digitais".
“ifs GC”	Informações do módulo "Gerenciador" para o módulo "Controlador".

A necessidade deste módulo pode ser visualizada no diagrama da figura 6.12. Os usuários (*Usuário 1* e *Usuário 2*) possuem seis válvulas direcionais proporcionais (denominadas como informação “Válvula da Bancada”) e duas válvulas reguladoras de pressão (denominadas como informação “Válvula Reguladora”) à disposição para realizarem os experimentos na Bancada de Trabalho.

Para isto, utilizam-se quatro saídas analógicas (Conversor D/A) e quatorze saídas digitais (Chaves Digitais) do Sistema VXI para que sejam escolhidas as válvulas que os usuários desejam e assim, sejam realizados os controles de posição dos sistemas de atuação hidráulicos (do *Usuário 1* e do *Usuário 2*) simultaneamente. Desta forma, a seleção destas saídas analógicas (canais do conversor D/A) combinadas às saídas digitais (canais das chaves digitais) são feitas de maneira automática no módulo gerenciador.

Assim sendo, associada à estrutura da figura 6.12 há uma lógica, a qual deve ser implementada em LabVIEW, que define seu comportamento. Por exemplo, se as duas primeiras válvulas proporcionais apresentadas nesta estrutura são escolhidas, sendo uma pelo *Usuário 1* e outra pelo *Usuário 2*, o módulo “Gerenciador” (partindo-se do pré-suposto que o *Usuário 1* realizou a escolha anteriormente ao *Usuário 2*) selecionará a saída analógica 01 em conjunto com a saída digital 01 para o controle de posição do atuador hidráulico do *Usuário 1*, enquanto selecionará a saída analógica 02 em conjunto com a saída digital 10 para o controle de posição do atuador hidráulico do *Usuário 2*.

Destarte, o módulo “Gerenciador” é responsável por adquirir informações dos módulos “Interface com Usuário (Usuário 1)” e “Interface com Usuário (Usuário 2)”, processá-las e posteriormente enviar informações a estes módulos e também ao “Driver das Chaves Digitais” e ao “Controlador” conforme está indicado na figura 6.13.

Deste modo, detalhando-se o módulo “Gerenciador” podem ser evidenciadas as funções “Aquisição de Dados” e “Envio de Dados” (figura 6.14).

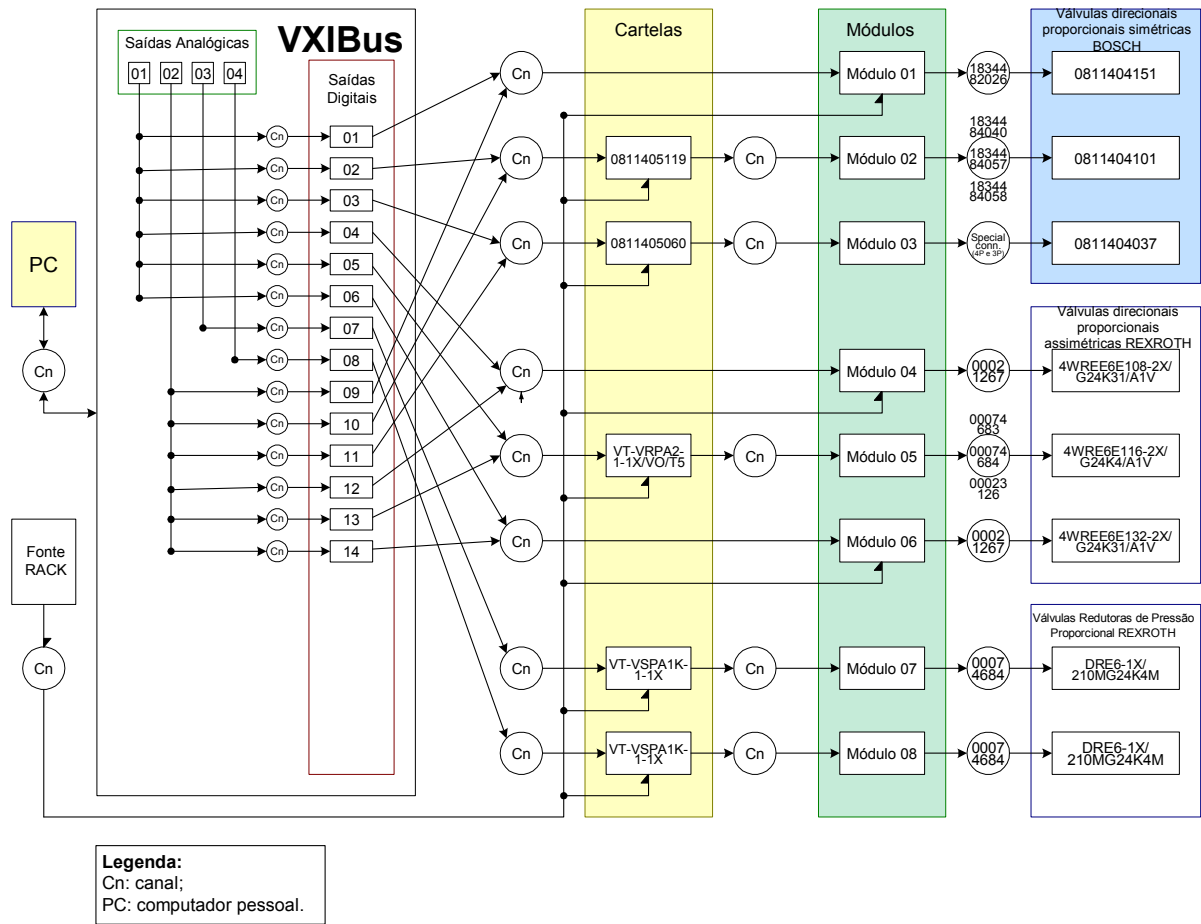


Figura 6.12 – Diagrama de escolha das válvulas da Bancada de Trabalho.

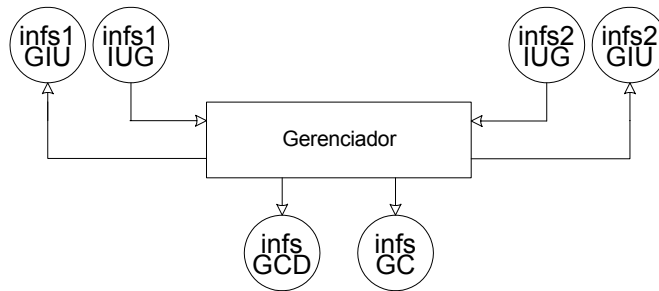


Figura 6.13 – Módulo "Gerenciador".

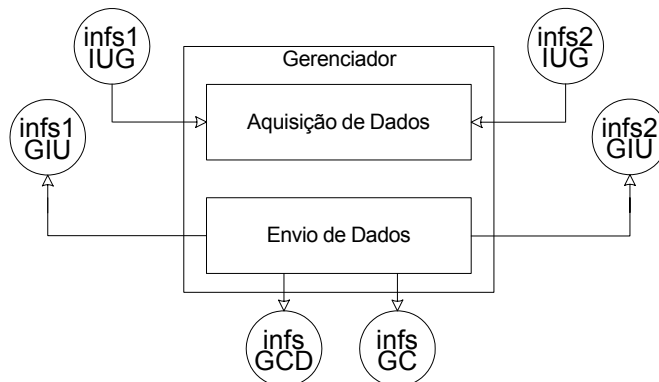


Figura 6.14 – Detalhamento do módulo "Gerenciador".

Por sua vez, podem-se detalhar as funções “Aquisição de Dados” e “Envio de Dados” como indicado na figura 6.15.

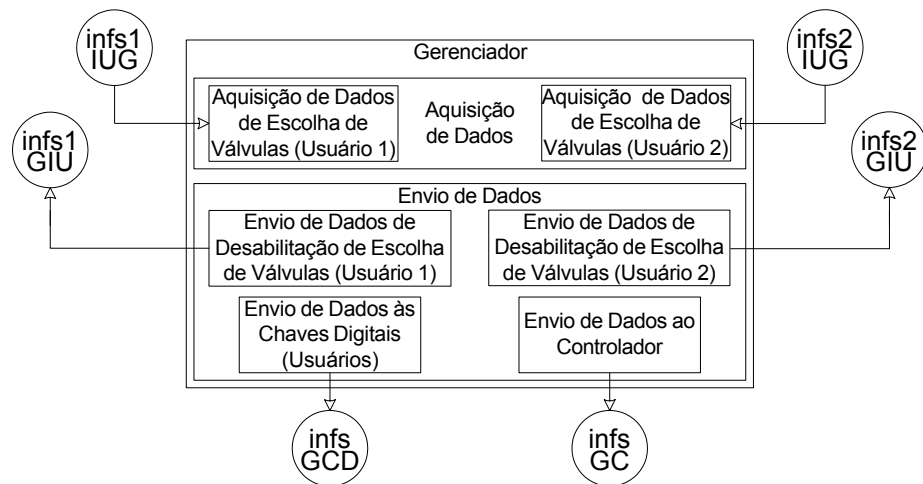


Figura 6.15 – Detalhamento das funções “Aquisição de Dados” e “Envio de Dados”.

A função “Aquisição de Dados” é composta pelas funções “Aquisição de Dados de Escolha de Válvulas (Usuário 1)” e “Aquisição de Dados de Escolha de Válvulas (Usuário 2)”.

Ao mesmo tempo, a função “Envio de Dados” é composta pelas funções “Envio de Dados de Desabilitação de Escolha de Válvulas (Usuário 1)”, “Envio de Dados de Desabilitação de Escolha de Válvulas (Usuário 2)”, “Envio de Dados às Chaves Digitais (Usuários)” e “Envio de Dados ao Controlador”, onde as duas primeiras funções servem para que os usuários não escolham a mesma válvula.

### 6.2.4 Módulo “Controlador”

A tabela 6.3 lista, de forma resumida, as informações relativas a este módulo. Por isso, maiores informações devem ser pesquisadas no apêndice 3.

Tabela 6.3 – Lista de informações para o módulo “Controlador”

“infs1 CIU”	Informações do módulo "Controlador" para o módulo "Interface com Usuário (Usuário 1)".
“infs2 CIU”	Informações do módulo "Controlador" para o módulo "Interface com Usuário (Usuário 2)".
“infs1 IUC”	Informações do módulo "Interface com Usuário (Usuário 1)" para o módulo "Controlador".
“infs2 IUC”	Informações do módulo "Interface com Usuário (Usuário 2)" para o módulo "Controlador".
“infs CCV”	Informações do módulo "Controlador" para o "Conversor D/A".
“infs CMP”	Informações do módulo "Controlador" para o "Driver do Multiplexador".

"infs MMC"	Informações do "Driver do Multímetro" para o módulo "Controlador".
"infs GC"	Informações do módulo "Gerenciador" para o módulo "Controlador".

O módulo "Controlador" é responsável por adquirir e enviar informações com os módulos "Interface com Usuário (Usuário 1)" e "Interface com Usuário (Usuário 2)". Ainda, é responsável por enviar informações ao "Driver do Conversor D/A" e "Driver do Multiplexador" e também por receber informações do "Driver do Multímetro" e do módulo "Gerenciador". Isso pode ser visto na figura 6.16.

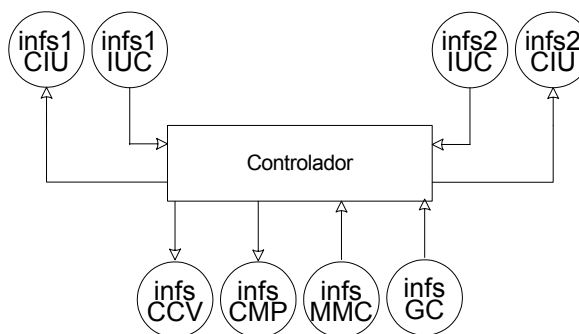


Figura 6.16 – Módulo "Controlador".

Detalhando-se o módulo "Controlador" são evidenciadas as funções "Aquisição de Dados", "Envio de Dados" e "Malha de Controle" (figura 6.17).

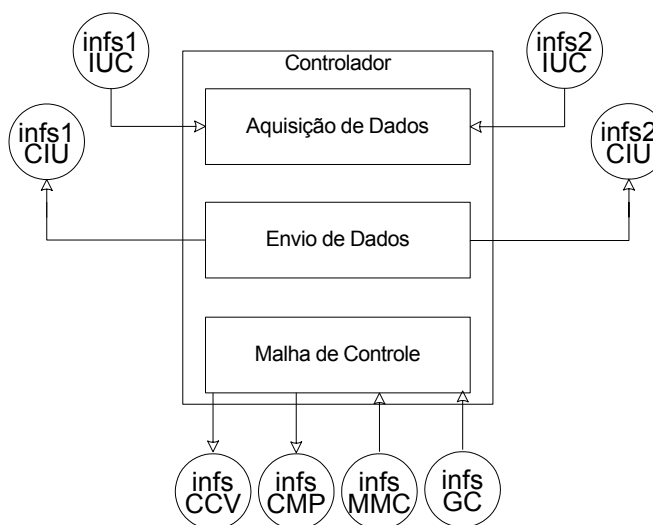


Figura 6.17 – Detalhamento do módulo "Controlador".

Por sua vez, como mostrado na figura 6.18, a função "Aquisição de Dados" é composta pelas funções "Aquisição de Dados (Usuário 1)" e "Aquisição de Dados (Usuário 2)". Nesta mesma figura pode-se ver também que a função "Envio de Dados" é composta pelas funções "Envio de Dados para Mostrar no Gráfico (Usuário 1)" e "Envio de Dados para Mostrar no Gráfico (Usuário 2)".

Além disso, a função "Malha de Controle" é composta pelas funções "Malha de Controle (Usuário 1)" e "Malha de Controle (Usuário 2)". A "Malha de Controle", como se nota

pelas informações que maneja, é responsável por adquirir a posição do cilindro através da leitura do multímetro e também, por enviar o sinal através do conversor D/A à válvula escolhida. Para tanto, é ela que também envia a informação do canal a ser utilizado pelo multiplexador, provinda do módulo “Gerenciador”, para que se realize a leitura do multímetro. Isto se deve ao fato de que se dispõe de apenas um multímetro para dois usuários. Desta forma, o multímetro é compartilhado (multiplexado) entre o *Usuário 1* e o *Usuário 2*.

Finalmente, a função “Aquisição de Dados (Usuário 1)” é composta pelas funções “Aquisição do Sinal de Referência” e “Aquisição dos Parâmetros de Escolha” como se vê na figura 6.19. A “Aquisição de Dados (Usuário 2)” é composta pelas mesmas funções.

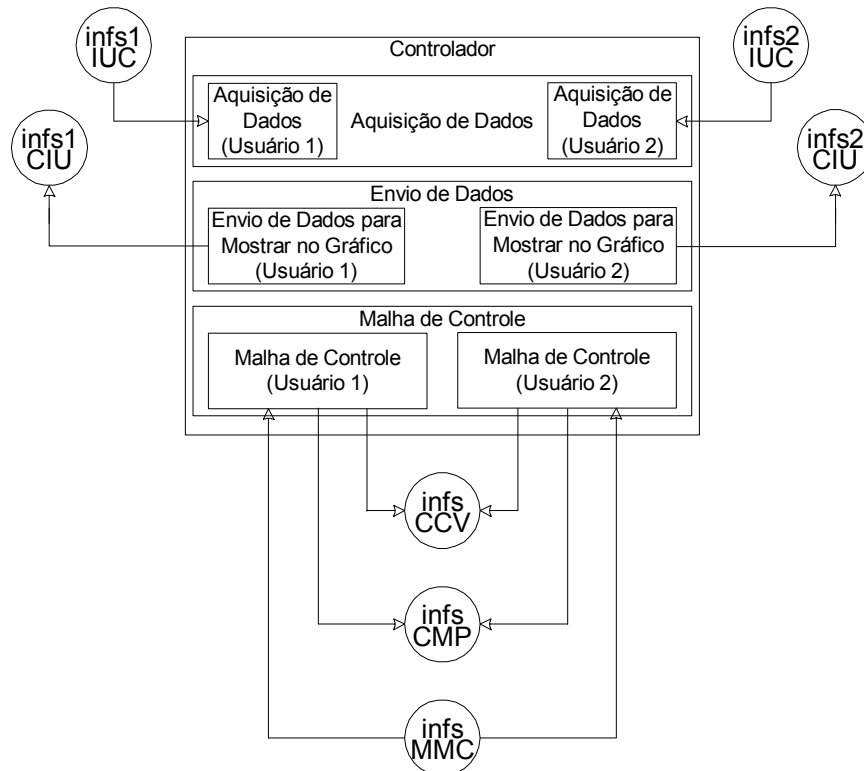


Figura 6.18 – Detalhamento das funções do “Controlador”.

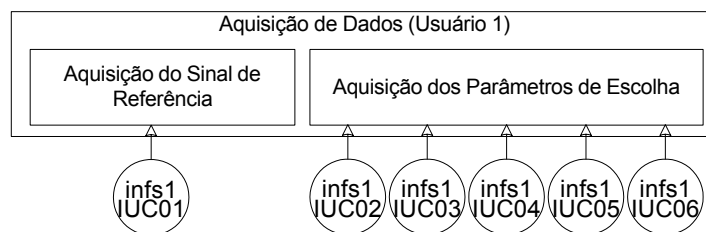


Figura 6.19 – Detalhamento da função “Aquisição de Dados (Usuário 1)”.

## CAPÍTULO 7

### DESENVOLVIMENTO DO APLICATIVO

Este capítulo descreve como se deu o desenvolvimento e, por conseguinte, a implementação do aplicativo, após a modelagem exibida no item 6.2.

Os conceitos apresentados no capítulo 4 norteiam de forma geral o desenvolvimento documentado a seguir. Especificamente, os aspectos abordados no item 4.3 são empregados com vistas a alcançar qualidade neste processo de desenvolvimento. Enfatizando-se um aplicativo robusto, estabeleceu-se, por sua vez, tantas proteções quanto possíveis para prevenir problemas. Também, de acordo com o apêndice 1, um conjunto de diretrizes foi selecionado para facilitar a manutenção deste e a gerência do projeto do aplicativo, auxiliada pelas ferramentas de controle do código fonte do LabVIEW, foi importante para monitorar as mudanças e progredir em direção dos objetivos especificados. Por sua vez, revisões de projeto e de código foram constantemente utilizadas para identificar e corrigir problemas durante o desenvolvimento. Enfim, ao término de cada fase no processo de desenvolvimento, considerou-se uma análise pós-projeto para discutir o que foi e o que não foi bem feito.

#### 7.1 Análise de Requisitos

No capítulo 6, utilizou-se os requisitos do Módulo de Trabalho como um todo para que se alcançasse as funções que o aplicativo deve contemplar. Contudo, neste capítulo são apresentados os principais requisitos específicos para a elaboração do aplicativo, sendo eles:

1. Tipos de E/S: teclado, mouse, monitor, drivers, leitura de medição de multímetro (A/D), conversão D/A;
2. Arquivos de saída: escrever relatório em formato HTML e gerar tabela em Excel com dados do gráfico;
3. Necessidade de comunicação com outras aplicações: Microsoft Excel / Internet Explorer;
4. Faixa de taxas de amostragem (ciclo de leitura e atuação) <sup>29</sup> para o controle do sistema hidráulico: (10 a 100) ms <sup>30</sup>;

---

<sup>29</sup> Devido ao controle do sistema de posição ser realizado por computador, deve-se ter ciência de que esta faixa de taxas de amostragem está sujeita à capacidade de processamento e memória do mesmo, bem como a velocidade de transmissão de dados via rede (no caso do computador que não está diretamente conectado ao Sistema VXI).

<sup>30</sup> Esta faixa corresponde a trabalhar com sistemas cuja frequência natural seja da ordem de (5 a 50) Hz. Considerando-se o *aliasing*, divide-se a frequência da faixa por dois:  $(1/10 \text{ ms})/2 = 5 \text{ Hz}$ .



5. Faixa de atuação da tensão de saída (atuação sobre a válvula realizada pelo conversor D/A): (0 a 10) V;
6. Faixa de medição da tensão de entrada (leitura da posição do cilindro realizada com o multímetro): (0 a 10) V;
7. Resolução do multímetro (para a leitura de posição do atuador hidráulico):  $10^{-3} V$  <sup>31</sup>;
8. Número de canais nas chaves digitais (saídas digitais para chaveamento decorrente da escolha das válvulas entre os usuários): 14 canais <sup>32</sup>;
9. Número de canais no multiplexador (para leitura no multímetro): 2 canais;
10. Número de canais no conversor D/A (para atuação do sistema hidráulico): 4 canais;
11. Necessidade de análise em tempo real: não;
12. Necessidade de transmissão de dados em rede (para comunicação entre os módulos do aplicativo): utilização de *Datasocket* <sup>33</sup> do LabVIEW;
13. Necessidade de execução em outros sistemas operacionais: primordialmente, não;
14. Custos de hardware: visto que já se possui praticamente todo o hardware de antemão, inclui-se apenas os gastos com um multiplexador FET e com um PC;
15. Estilos de gráficos: limpo;
16. Integração com o módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”.

## 7.2 Prototipação do Aplicativo

A partir da coleta de requisitos, um modelo do aplicativo (um protótipo) pode ser construído de modo a permitir uma melhor avaliação por parte do desenvolvedor e do cliente.

O paradigma da prototipação tem como ponto de partida uma solicitação de proposta encaminhada pelo cliente. De posse desta solicitação, conforme estabelecido por MAZZOLA e FARINES (1998), os seguintes passos foram realizados:

- análise da solicitação para identificar a necessidade da prototipação;
- especificação resumida dos requisitos, realizada pelo analista, de modo a representar o domínio da informação e os domínios funcionais e comportamen-

---

<sup>31</sup> Conforme mostra o apêndice 4, este requisito é cumprido.

<sup>32</sup> Como pode ser visto na figura 6.12, são necessários 14 canais de chaves digitais.

<sup>33</sup> *Datasocket*: Uma tecnologia que simplifica a troca de dados entre diferentes aplicativos que estejam em um mesmo computador ou em computadores conectados através de uma rede.

tais do aplicativo, para que a atividade de particionamento do problema possa ser efetuada;

- revisada a especificação resumida dos requisitos, realizou-se um projeto do protótipo, concentrando-se principalmente nos aspectos arquiteturais e de dados e deixando de lado os aspectos procedimentais;
- desenvolvimento do protótipo, quando possível a partir da utilização de blocos de construção de aplicativos preexistentes;
- apresentação do protótipo (testado) ao cliente para que este pudesse realizar sua avaliação; a partir desta avaliação, o cliente pôde sugerir extensões ou reformular alguns requisitos para que o aplicativo pudesse melhor corresponder às reais necessidades;
- repetição iterativa dos dois passos anteriores até que todos os requisitos tivessem sido formalizados.

Desta forma, exibe-se a seguir um exemplo de evolução do aplicativo devido ao seguimento deste passos no processo de prototipagem. As figuras seguintes mostram a evolução “visual” do aplicativo após testes realizados pelos clientes (orientador deste trabalho e pessoal do LASHIP), a qual se deu devido a requisitos como facilidade de uso do aplicativo e compreensão do sistema como um todo (sistema ilustrado na figura 1.3).

Nota-se que a versão mostrada na figura 7.1, apesar de desempenhar as funções básicas especificadas no item 6.2, não é de fácil utilização devido ao fato que se tem todos os controles e indicadores em uma mesma tela (o que pode confundir o usuário) e também não permite a compreensão do sistema (ilustrado na figura 1.3) como um todo sem que se leia primeiramente um arquivo de ajuda ou se tenha alguma explicação prévia do mesmo.

A versão final apresentada na figura 7.2, no entanto, supre as deficiências anteriormente comentadas de sua versão precedente. Nota-se que visualmente o usuário se intera rapidamente tanto da utilização do aplicativo quanto da compreensão do sistema, devendo-se ao fato de que um desenho esquemático do sistema automático ‘Módulo de Trabalho da Plataforma de Hidráulica Proporcional’, ilustrado na figura 1.3, foi adicionado nesta versão do aplicativo.

De tal modo, nesta versão os controles e os indicadores de interesse podem ser acessados facilmente com um clique do mouse sobre as partes do sistema correspondentes a estes, como mostrados na figura 7.3 e na figura 7.4.

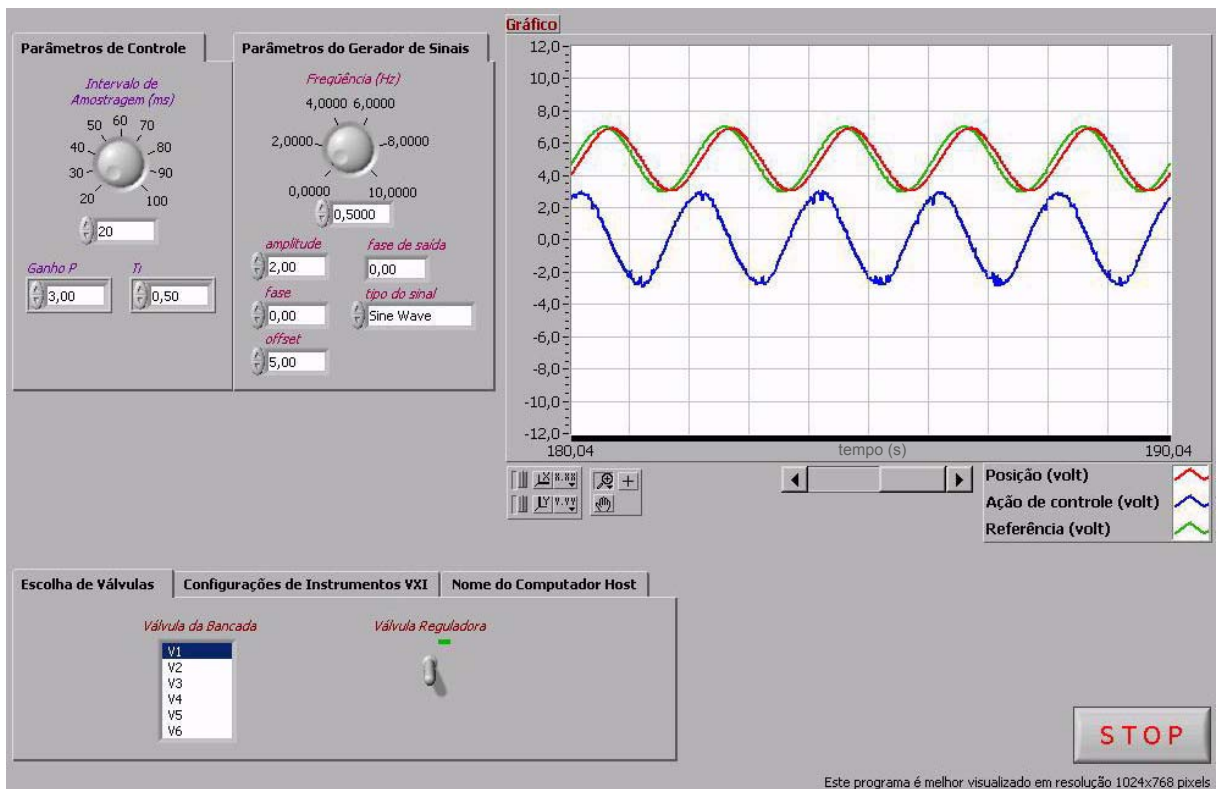


Figura 7.1 – Versão preliminar da “Interface com Usuário (Usuário 1)” do aplicativo.

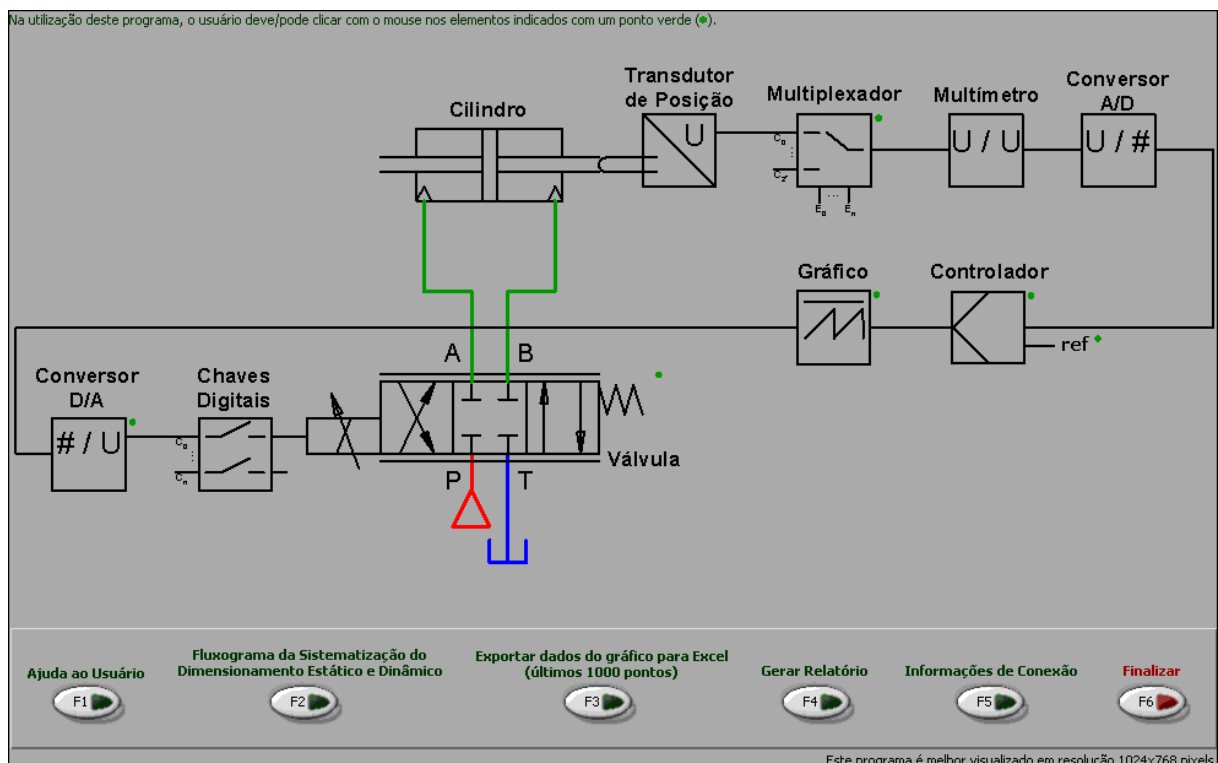


Figura 7.2 – Versão final da “Interface com Usuário (Usuário 1)” do aplicativo.

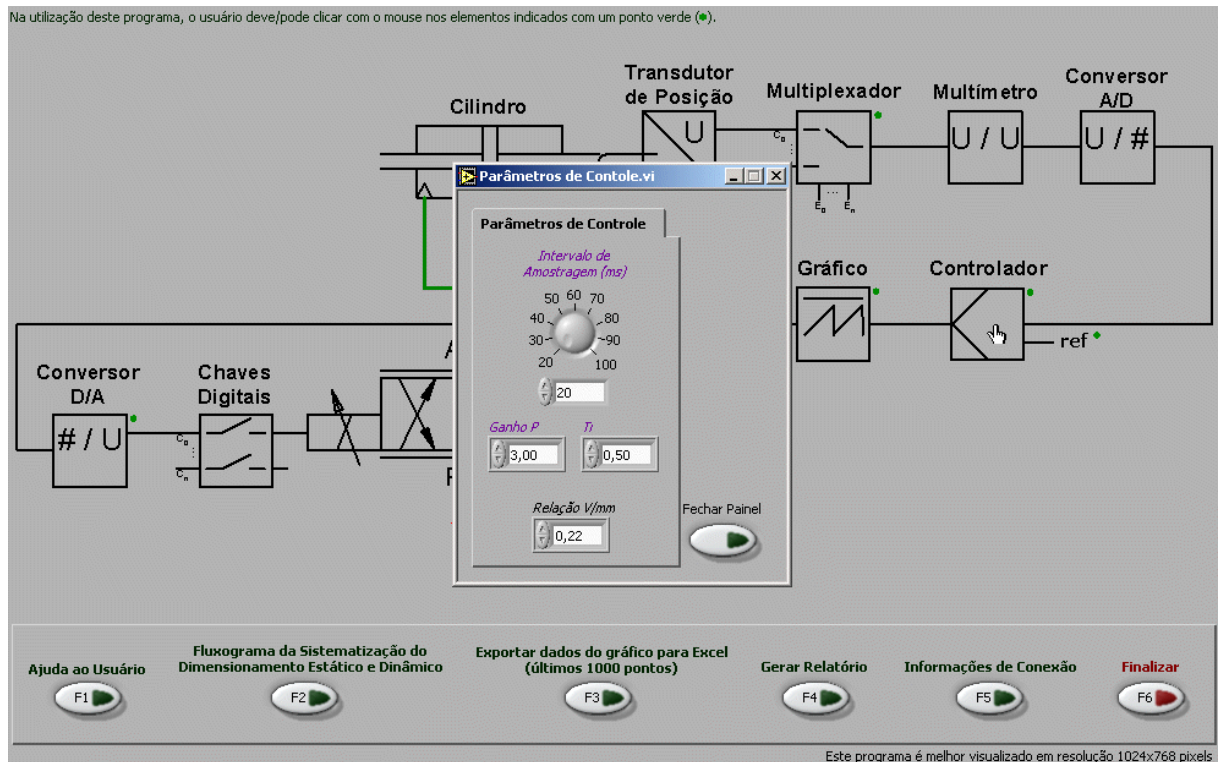


Figura 7.3 – Acesso aos controles da versão final da “Interface com Usuário (Usuário 1)” do aplicativo.

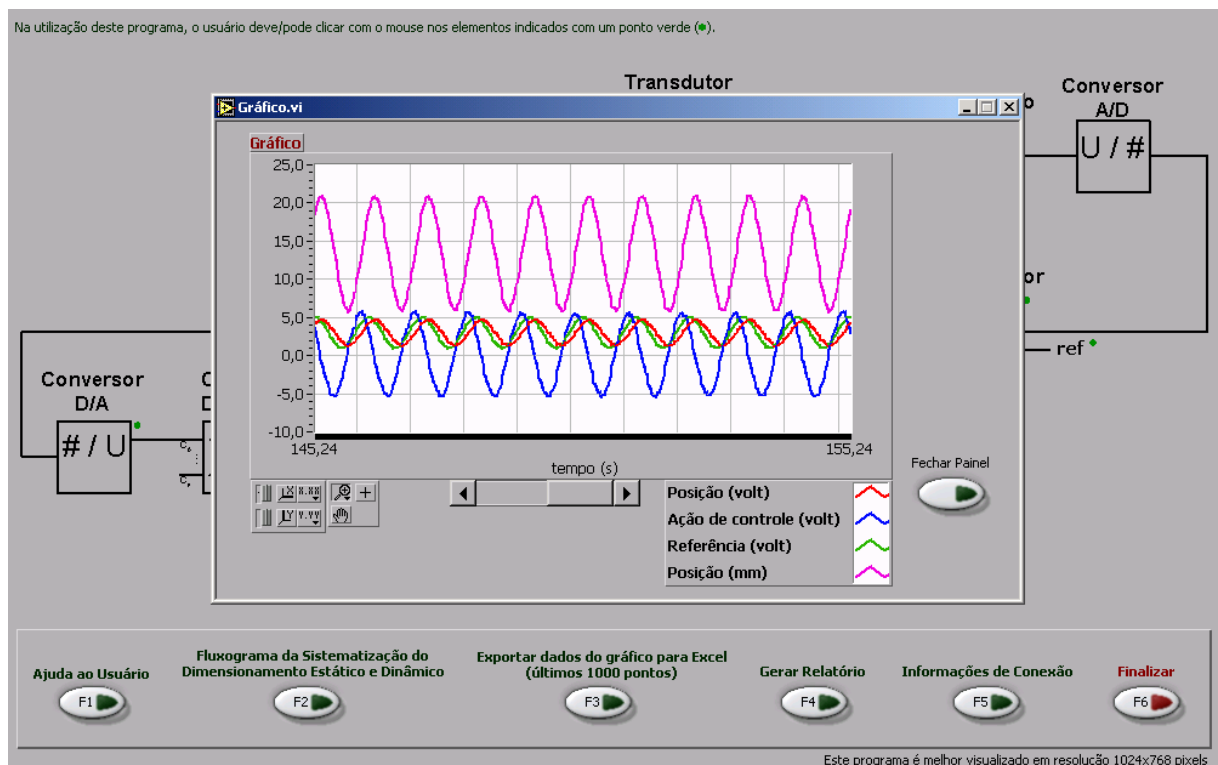


Figura 7.4 – Acesso aos indicadores da versão final da “Interface com Usuário (Usuário 1)” do aplicativo.

### 7.3 O Processo de Projeto

#### 7.3.1 Projetos Preliminar e Detalhado

A etapa de projeto é caracterizada pelo conjunto de atividades que permite traduzir os requisitos definidos na etapa análise de requisitos em uma representação do aplicativo a ser construído. Na sua forma mais clássica, o primeiro resultado obtido no projeto é uma visão da estrutura do aplicativo em termos de componentes, sendo que, a partir de procedimentos de refinamento, chega-se a um nível de especificação bastante próxima da codificação do programa.

Do ponto de vista do gerenciamento do processo de desenvolvimento, a etapa de projeto foi conduzida basicamente em dois principais estágios:

- o projeto preliminar, o qual permitiu estabelecer, a partir dos requisitos, a arquitetura do aplicativo e da informação relacionada;
- o projeto detalhado, o qual permitiu aperfeiçoar a estrutura do aplicativo e definir representações algorítmicas de seus componentes.

#### 7.3.2 Características Desejadas no Projeto do Aplicativo

A obtenção de um aplicativo/software de boa qualidade está relacionada ao sucesso de cada uma das etapas do seu desenvolvimento. Desta maneira, em conformidade com o item 4.4.4, detectou-se algumas características desejáveis no que diz respeito à etapa de projeto:

- deve apresentar uma organização hierárquica, definida de modo a utilizar racionalmente todos os elementos do aplicativo;
- deve basear-se em princípios de modularidade, ou seja, propor um particionamento do aplicativo em elementos que implementem funções ou subfunções do aplicativo;
- deve conduzir a uma definição em módulos com alto grau de independência, como forma de facilitar as tarefas de manutenção ao longo da vida do aplicativo;
- deve ser fiel à especificação dos requisitos definida na etapa anterior.

### 7.4 Hierarquia de Funções do Aplicativo

Conforme a modelagem do aplicativo realizada no item 6.2 e considerando-se os aspectos fundamentais do projeto delineados em 4.4.4, chegou-se à hierarquia de funções conforme está descrita na presente seção.

De acordo com a figura 6.6, o aplicativo é composto pelos quatro módulos principais indicados na figura 7.5.

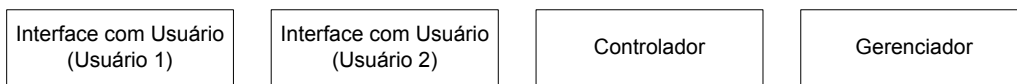


Figura 7.5 – Módulos principais do aplicativo.

Por sua vez, com base na figura 6.10 e na figura 6.11, têm-se que o módulo “Interface com Usuário (Usuário 1)” contempla as funções como mostrado na figura 7.6.

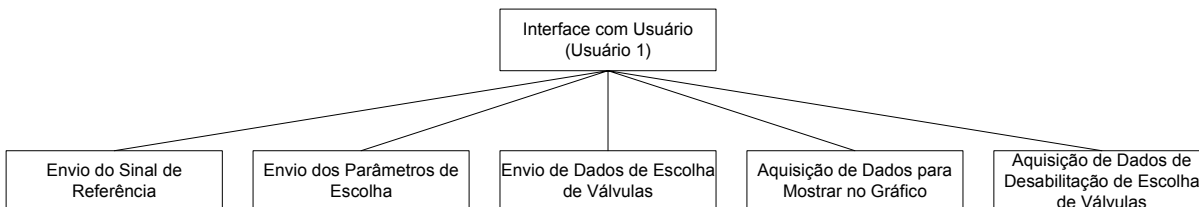


Figura 7.6 – Hierarquia do módulo “Interface com Usuário (Usuário 1)”.

Do mesmo modo, o módulo “Interface com Usuário (Usuário 2)” contempla as funções como mostrado na figura 7.7.

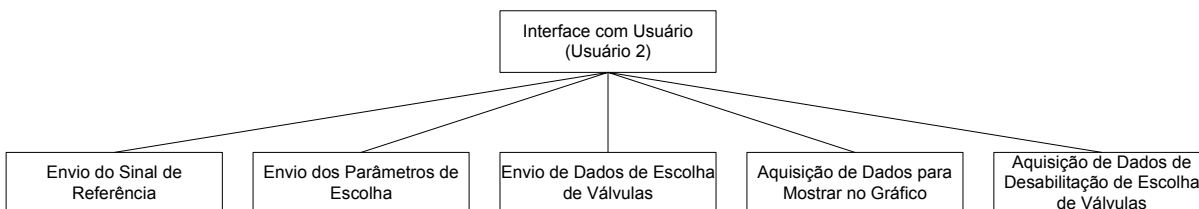


Figura 7.7 – Hierarquia do módulo “Interface com Usuário (Usuário 2)”.

Ao mesmo tempo, a hierarquia do módulo “Gerenciador”, apontada na figura 7.8, a-póia-se na figura 6.15.

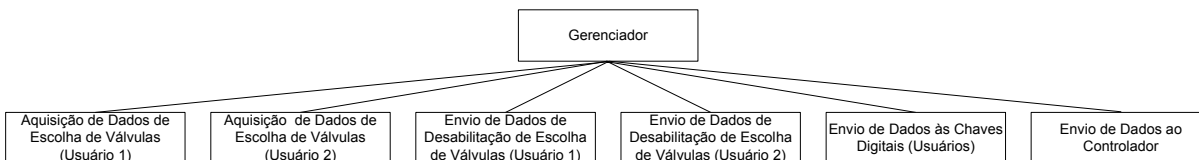


Figura 7.8 – Hierarquia do módulo “Gerenciador”.

Também, partindo-se da figura 6.18 e da figura 6.19 tem-se a hierarquia do módulo “Controlador” como explicitada na figura 7.9.

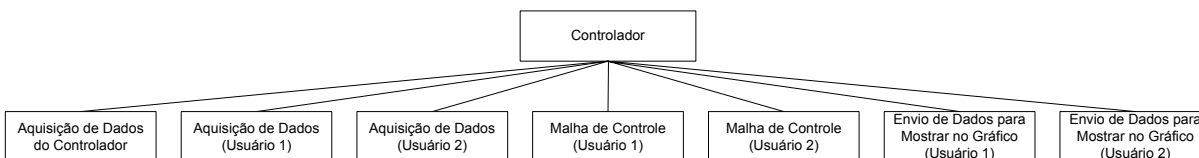


Figura 7.9 – Hierarquia do módulo “Controlador”.

Por fim, o módulo “Aquisição de Dados (Usuário1)”, bem como o referente ao *Usuário 2*, constituem-se das funções indicadas na figura 7.10.

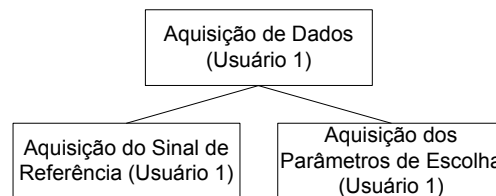


Figura 7.10 – Hierarquia do módulo “Aquisição de Dados (Usuário 1)”.

De tal modo, definida a hierarquia de funções conforme mostrada neste item, pode-se implementar estas funções em qualquer que seja a linguagem desejada. Assim, a implementação destas funções se deu através do software LabVIEW conforme apresentado no capítulo 4 e documentado de acordo com o item 7.5.

## 7.5 Documentação

Dois tipos de documentação do aplicativo foram gerados em HTML:

- A documentação ao Projetista/Programador – Voltada ao público que tem extenso conhecimento do projeto e das ferramentas que se está documentando. Inclui, para os VIs e subVIs: descrições, painéis conectores (com as variáveis de entrada e saída), painéis frontais, lista com a descrição dos controladores e indicadores, diagramas de blocos, lista dos subVIs e posição na hierarquia.
- A documentação ao Usuário – Voltada ao público que tem menor grau de entendimento e experiência com o aplicativo, explica como usar o aplicativo.

### 7.5.1 Documentação ao Projetista/Programador

Devido à extensão da documentação gerada apenas uma pequena parte desta documentação é mostrada neste documento. No caso, é mostrada apêndice 5 uma parte do módulo de “Interface com Usuário (Usuário 1)”.

Contudo, pode-se visualizar que esta documentação compreende tanto os aspectos do modelo estrutural do aplicativo quanto o modelo funcional do aplicativo referente à implementação em LabVIEW.

### 7.5.2 Documentação ao Usuário

A documentação ao usuário, diferentemente da documentação ao Projetista/Programador, é relativamente simples. São apenas documentados os módulos “Interface com Usuário (Usuário 1)” e “Interface com Usuário (Usuário 2)” visto que os usuários têm contato apenas com estes módulos conforme pode ser visto figura 6.6. No entanto, apenas a documentação do módulo “Interface com Usuário (Usuário 1)” é exibida no apêndice 6 já que a documentação do módulo “Interface com Usuário (Usuário 2)” é similar a primeira. Durante a utilização do aplicativo, esta documentação pode ser acessada clicando-se em um botão da “Interface com Usuário” e é simples e curta.

Ainda, conforme indicado no apêndice 6, pode-se gerar um relatório em HTML<sup>34</sup> com alguns dados relevantes para o usuário. Este relatório, por sua vez, é exibido na figura 7.11.

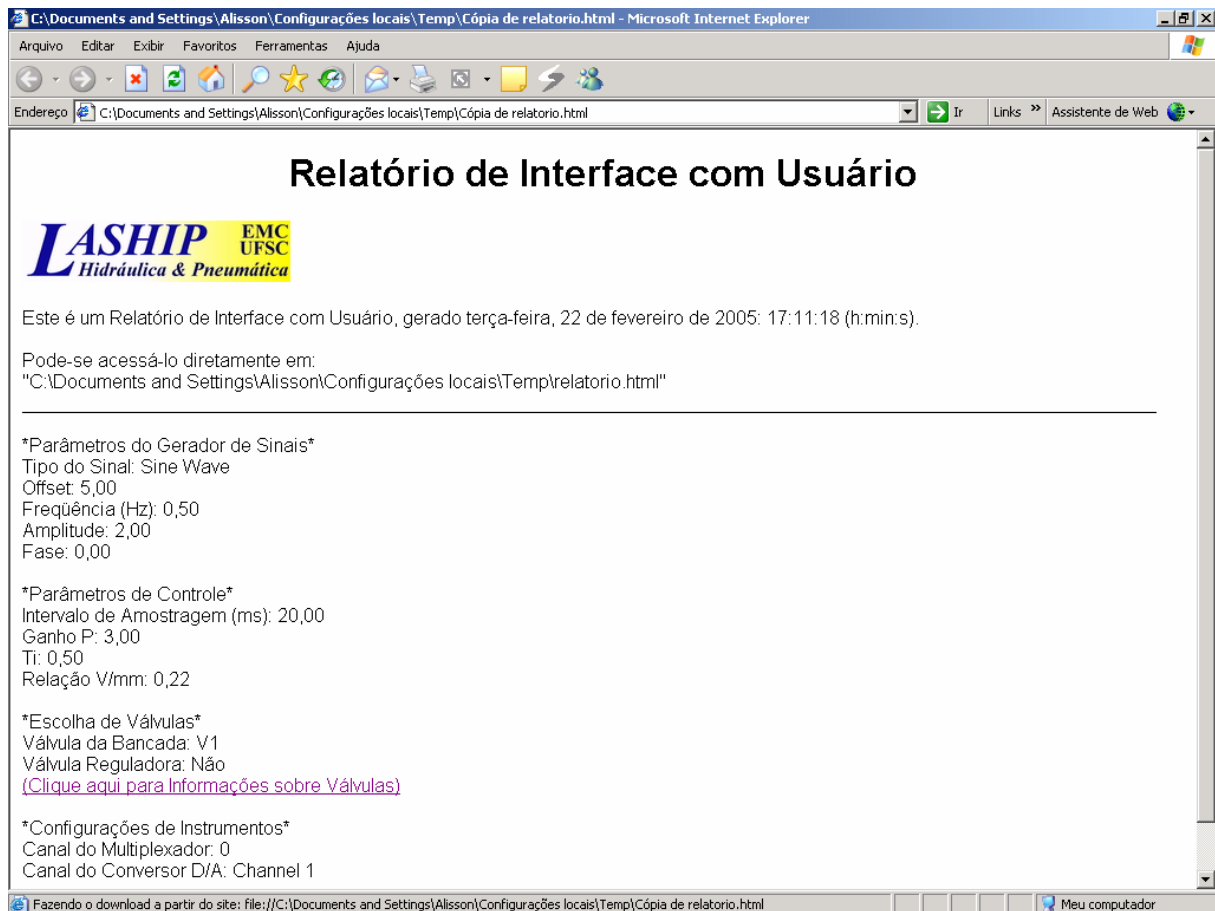


Figura 7.11 – Relatório em HTML dos módulos de interface com os usuários.

## 7.6 Testes de Avaliação

Os testes de avaliação foram realizados com o objetivo determinar se o aplicativo, mesmo que funcionando corretamente (em consequência das etapas de teste anteriores, como de unidade e de integração) apresenta as propriedades e as funcionalidades definidas na etapa de especificação dos requisitos.

Apesar do esforço na avaliação/validação de um aplicativo/software, na maioria das vezes é extremamente difícil para o desenvolvedor ou programador prever as diferentes maneiras como este será utilizado.

Principalmente no caso de aplicativos interativos (como o aplicativo em questão), os usuários muitas vezes experimentam comandos e entradas de dados os mais diversos, o que pode ser um verdadeiro atentado à robustez de um sistema. Da mesma forma, dados

<sup>34</sup> A necessidade de se gerar este relatório também foi constatada devido ao emprego da prototipação, cujos benefícios de sua utilização foram comentados no item 7.2.



de saída ou formatos de apresentação de resultados podem parecer ótimos para o analista, mas completamente inadequados para o usuário.

Desta forma, testes alfa foram realizados no presente aplicativo com algumas pessoas do LASHIP<sup>35</sup>. Devido à realização desses testes, falhas de programação e deficiências do aplicativo foram sanados.

Como exemplo de falha de programação, cita-se a não concordância que existia entre o gráfico gerado pelo aplicativo, em Excel, com os dados de simulação que o usuário deseja guardar (exemplificado na figura 7.12) e o gráfico apresentado na interface com usuário (exemplificado na figura 7.4).

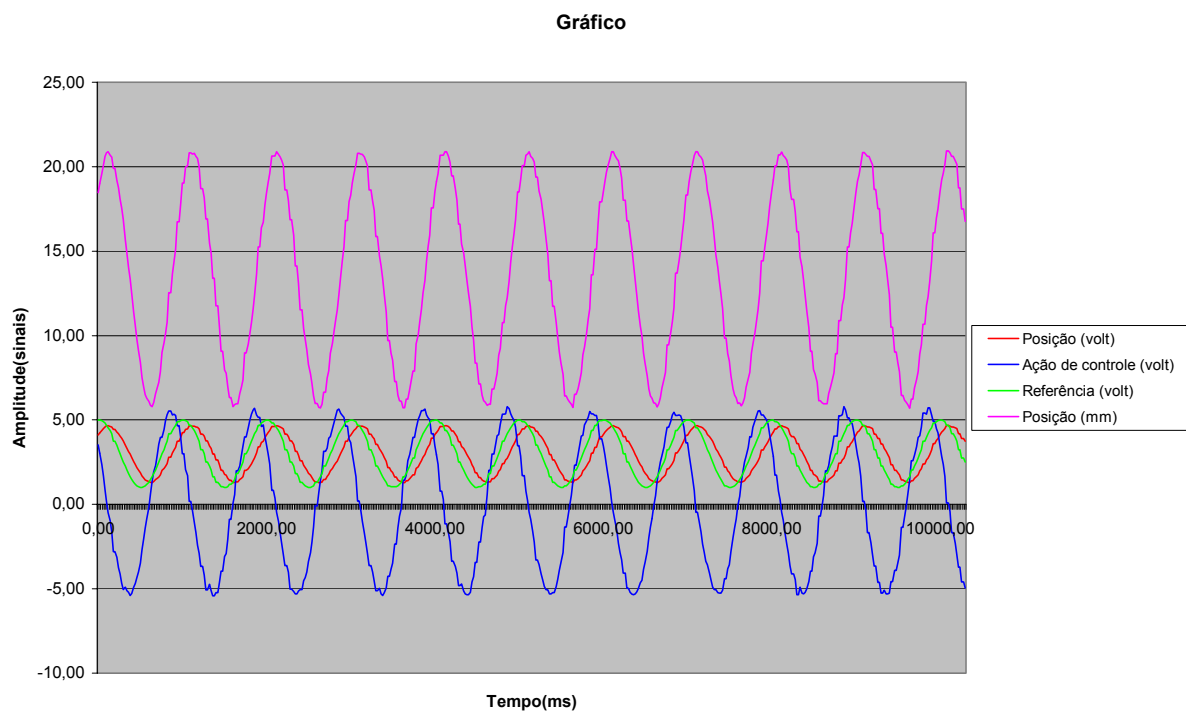


Figura 7.12 – Gráfico gerado em Excel pela interface com usuário.

O usuário dispõe da opção de selecionar, com o mouse, a faixa de dados que deseja visualizar no gráfico do módulo de interface com usuário (figura 7.4), onde o padrão para o eixo das abscissas é de dez (10) segundos. Para este valor padrão, o gráfico (figura 7.12) era gerado corretamente. Porém, caso o usuário modificasse este valor e posteriormente requisitasse que fosse gerado o gráfico em Excel, o módulo (VI) do aplicativo responsável por desempenhar tal tarefa não atualizava este valor. Desta forma, a base de tempo de simulação deste gráfico (figura 7.12) não condizia com a realidade e, conseqüentemente, com o gráfico da interface com usuário (figura 7.4). Somente após a utilização do aplicativo por

---

<sup>35</sup> Testes beta, onde os clientes testam o produto em campo, não foram realizados neste trabalho pelo fato de que as pessoas consideradas nos testes alfa, apesar de serem do LASHIP, não estavam familiarizados com o Projeto Plataforma de Hidráulica Proporcional. Por conseqüência, não tinham conhecimento prévio sobre o aplicativo.

alunos de Pós-Graduação da Engenharia Mecânica este problema, dentre outros, foi concertado.

### **7.7 Integração do Aplicativo com o Módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”**

Finalmente, cabe comentar sobre o módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”<sup>36</sup>, o qual pode ser acessado através da “Interface com Usuário” como exibido no apêndice 6.

Nesse, igualmente, se empregou o processo de projeto (7.3), os aspectos fundamentais do projeto (4.4.4), a documentação (7.5) e os testes de validação (7.6) como propostos anteriormente.

O fluxograma está mostrado na figura 7.13. Para tanto, esse módulo apresenta-se como exibido na figura 7.14<sup>37</sup>. Para cada passo do fluxograma<sup>38</sup>, existe um botão de ajuda ao usuário como está exemplificado no apêndice 7.

Observa-se que o fluxograma apresentado na figura 7.13 é de difícil utilização caso o usuário não faça antes um intenso estudo sobre o seu funcionamento e equações que este envolve. Assim, a prototipação desse módulo supriu essa deficiência de modo a tirar a carga de conhecimento do usuário sobre a utilização desse fluxograma.

Como se pode observar na figura 7.14, este módulo também procurou ser o mais amigável possível, onde ilustrações (ver figura do sistema de atuação hidráulico) e controles (ver botões) buscam, respectivamente, facilitar as escolhas de opções e os valores de parâmetros que o usuário deve efetuar. Ao mesmo tempo, na figura 7.15, nota-se a busca pela elucidação (ver seta) sobre os valores encontrados automaticamente pelo aplicativo, visto que o usuário já não precisa mais conhecer profundamente as equações envolvidas e os caminhos a serem tomados no fluxograma (figura 7.13).

---

<sup>36</sup> Isto se deveu ao fato de que este módulo é um ambiente independente no âmbito do Módulo de Trabalho da Plataforma de Hidráulica Proporcional.

<sup>37</sup> Há também uma versão em inglês deste módulo conforme requerido por parte da empresa BOSCH REXROTH (ver figura 3.4).

<sup>38</sup> As telas de cada passo do aplicativo são mostradas no apêndice 8.

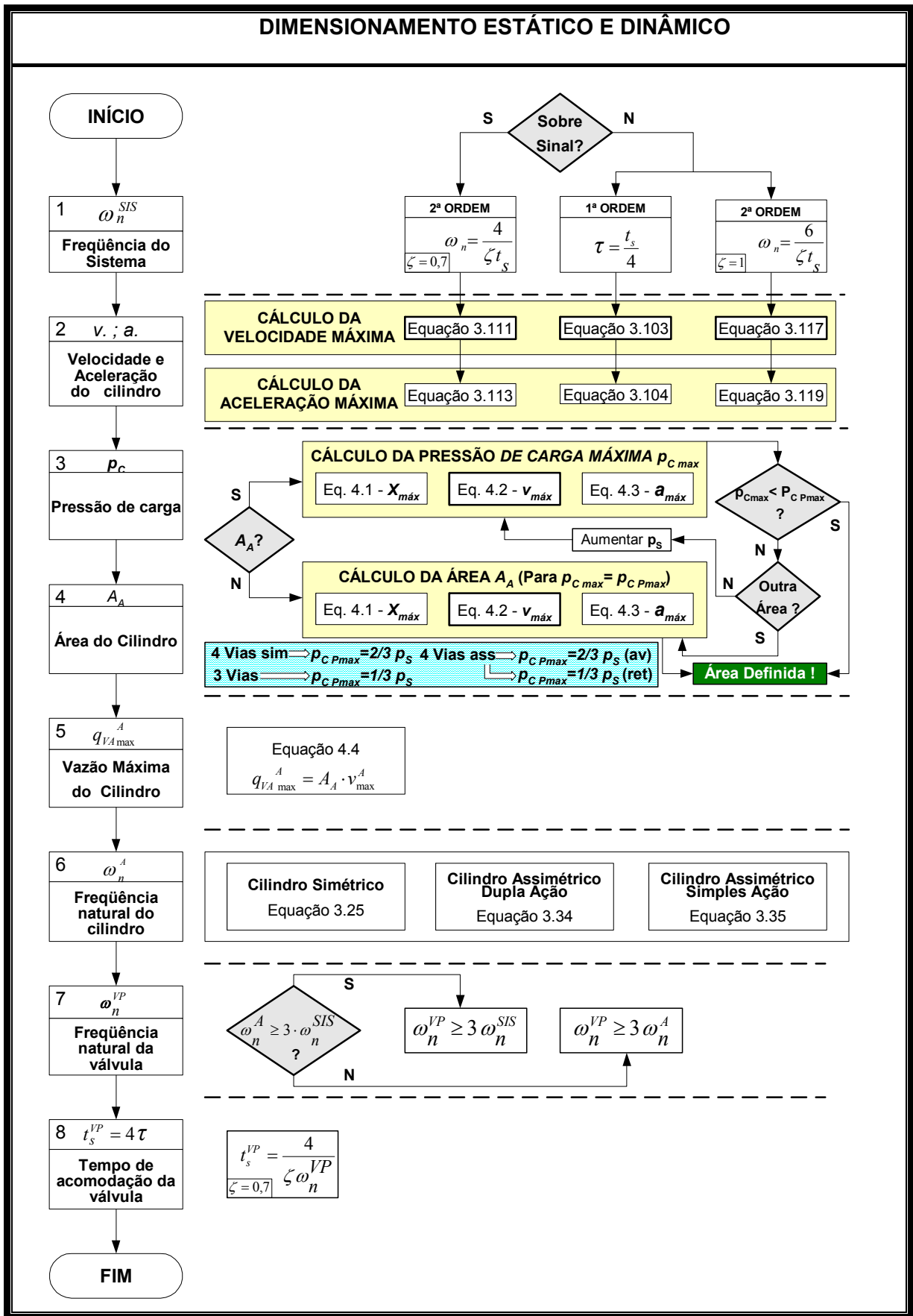


Figura 7.13 – Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico (FURST, 2001).

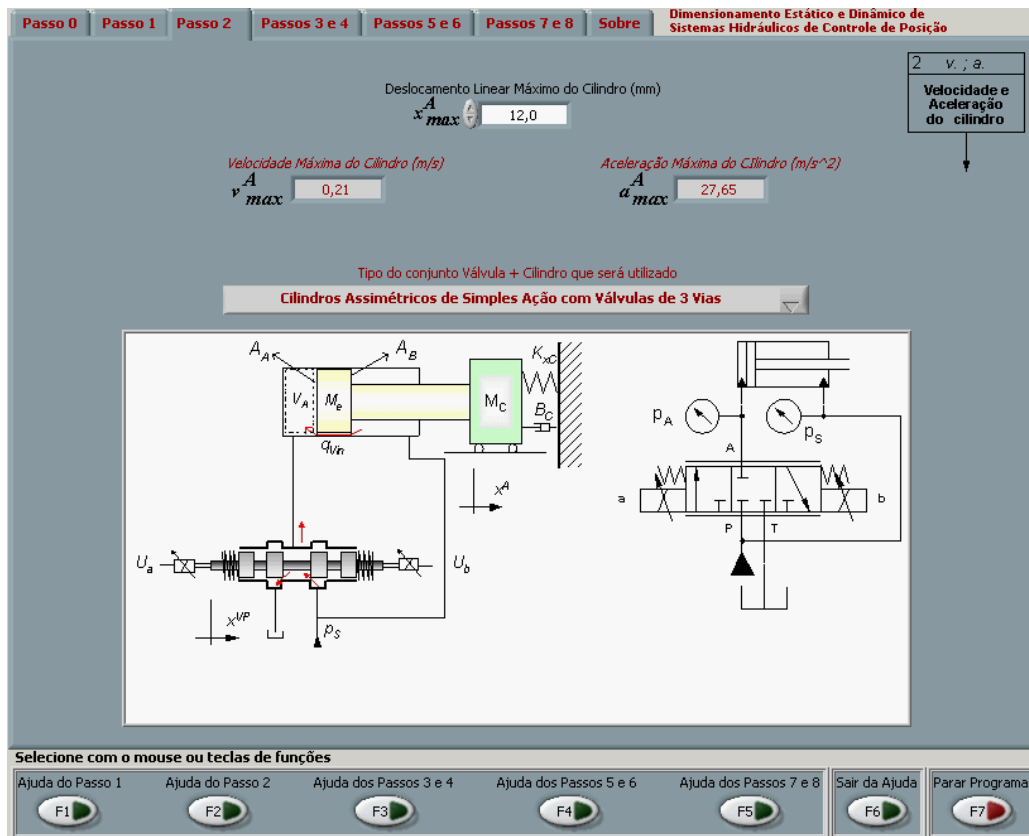


Figura 7.14 – Passo 2 do módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”.

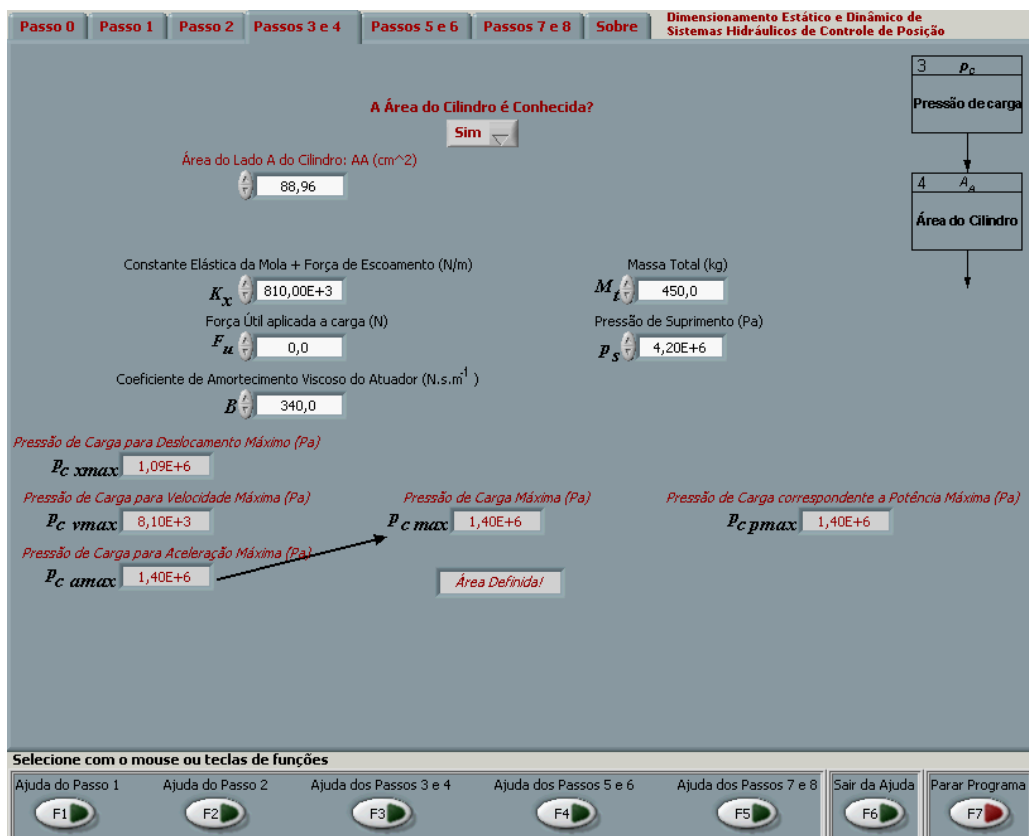


Figura 7.15 – Passos 3 e 4 do módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”.

Diferentemente da parte do aplicativo que desempenha o controle de posição de sistemas eletro-hidráulicos, este módulo não possui muitos subVI's para desempenhar suas funções, como se observa na figura 7.16. Isso se deve ao fato de que apesar da quantidade elevada de variáveis envolvidas no fluxograma, os cálculos (funções) desse módulo são relativamente simples de se implementar.

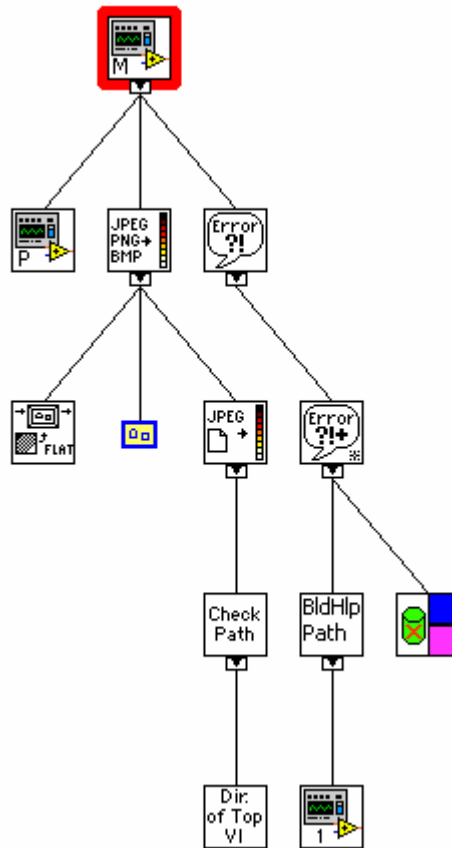


Figura 7.16 - Hierarquia de funções do módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”.

Frente ao que foi comentado nesse item 7.7, este módulo do aplicativo foi inclusive utilizado para auxiliar no levantamento das especificações técnicas dos componentes válvula proporcional e cilindro do Projeto Plataforma de Hidráulica Proporcional devido à simplicidade que oferece. O procedimento para o dimensionamento dos componentes foi executado no sentido de se determinarem campos de aplicação com os pares válvulas/cilindros para atingir uma ampla faixa de especificações dinâmicas conforme os requisitos preliminares do Projeto Plataforma de Hidráulica Proporcional. Com o intuito de atender à pressão de suprimento ( $p_s$ ) da unidade de potência hidráulica projetada, de não ultrapassar a velocidade dos cilindros e ainda de se obter a faixa de respostas dinâmicas requisitadas, foram realizadas simulações através deste módulo para diferentes conjuntos de válvulas/cilindros.

Por fim, é importante salientar que, além dos testes alfa realizados no LASHIP, este módulo foi, após ter sido apresentado, utilizado pelo pessoal da empresa Bosch Rexroth na Feira Internacional de Mecânica 2004 (São Paulo – SP). Constatou-se assim, a aprovação quanto à sua funcionalidade e facilidade de uso.

## 7.8 Discussão

O processo de projeto referente ao desempenho do controle de posição eletro-hidráulico utilizou uma metodologia de modelagem de sistemas automáticos que, a partir da identificação das partes que compõem o sistema Módulo de Trabalho da Plataforma de Hidráulica Proporcional, alcançou (através dos refinamentos de modelos) as funções necessárias para a construção ou implementação do aplicativo segundo as metodologias da engenharia de software exibidas no início deste capítulo.

A figura 7.17 mostra a hierarquia de funções do Módulo “Interface com Usuário (Usuário 1)” retirada da documentação do aplicativo apresentada no apêndice 6.

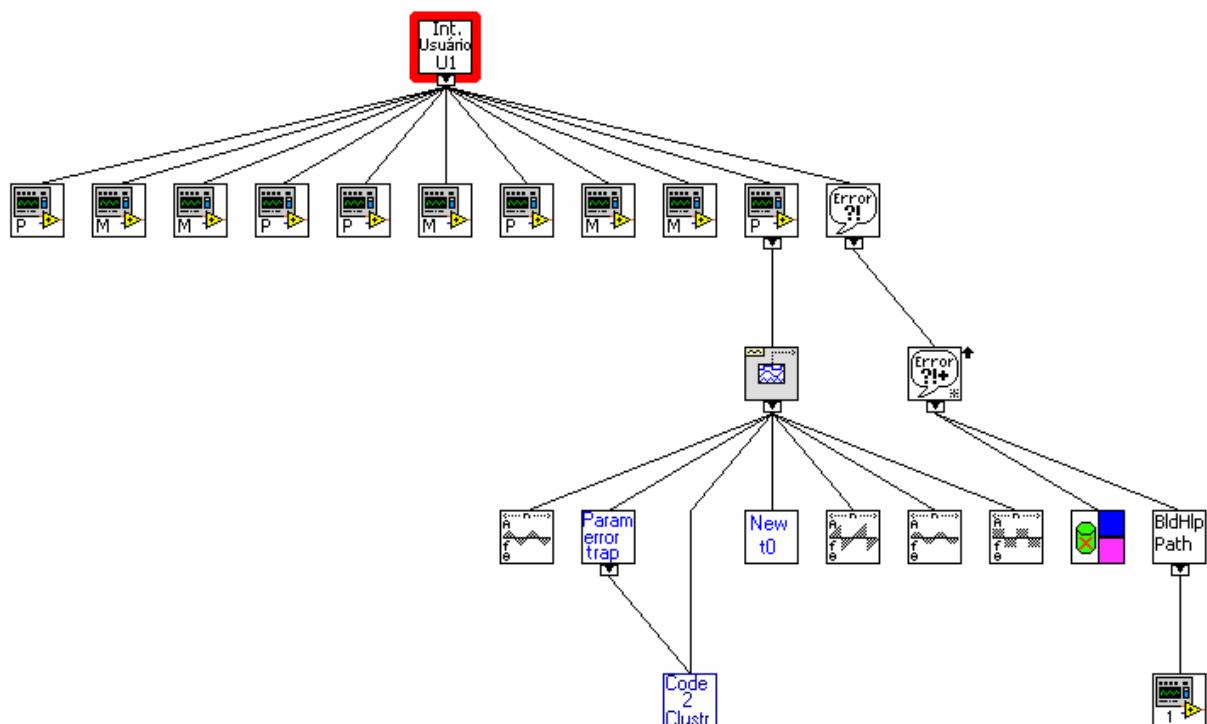


Figura 7.17 – Hierarquia de funções do módulo “Interface com Usuário (Usuário 1)”.

Conforme a lista de subVIs exibidas na documentação do apêndice 6, pode-se verificar, então, que a hierarquia de funções alcançada no item 7.4 realmente corresponde aos módulos do aplicativo implementados. Isto comprova que a modelagem de sistemas automáticos proposta para se realizar o desenvolvimento do aplicativo se mostrou uma ferramenta eficiente.

## **CAPÍTULO 8**

### **CONCLUSÕES**

A motivação inicial para a construção do aplicativo do presente trabalho de dissertação levou à identificação de um problema mais geral no que diz respeito à forma corrente de modelagem de sistemas automáticos.

Ciente de que o equipamento Plataforma de Hidráulica Proporcional caracteriza-se por ser um sistema automático, realizou-se neste trabalho uma avaliação e um aprimoramento de princípios de modelagem e projeto de sistemas automáticos. Também, analisou-se a possibilidade de associação de modelos de sistemas automáticos às soluções técnicas implementadas de acordo com abordagens em engenharia de software.

Para tanto, fez-se uso de modelos gráficos capazes de portar um grande número de informações de forma concisa e plenamente adequados para o projeto. Por sua vez, estes modelos possibilitaram um tratamento sistêmico igual para o aplicativo/software e para a parte física do sistema como um todo, ou seja, do Módulo de Trabalho da Plataforma de Hidráulica Proporcional. Portanto, por meio desta metodologia própria para sistemas automáticos, contemplou-se, dentro do possível, o processo de projeto de sistemas técnicos e o desenvolvimento de software.

A integração de modelos e de tecnologias foi especialmente alcançada com a Rede de Petri Canal/Agência (Rede C/A), uma notação multidisciplinar baseada em regras simples de construção e interpretação que permitiram descrever requisitos funcionais e especificações estruturais.

Através do mecanismo de refinamentos e condensações, a rede C/A, mostrada no início da apresentação do sistema, sofreu transformações correspondendo à evolução do projeto, possibilitando a associação de modelos referentes às soluções técnicas escolhidas (engenharia de software mostrada no capítulo 4).

Desta forma, pôde-se comprovar que essa metodologia de modelagem, a qual integra técnicas de descrição gráficas para sistemas automáticos e busca ser independente das possíveis implementações, mostrou-se uma ferramenta eficiente no projeto de sistemas automáticos de modo que auxiliou na consolidação do aplicativo almejado.

A obtenção de um aplicativo de fácil manutenção, com pequeno tempo de desenvolvimento e facilmente utilizável decorreu do cumprimento de diretrizes relacionadas ao desenvolvimento de software de qualidade, características estas já bem conhecidas no ramo da engenharia de software e, neste caso, utilizadas com a linguagem LabVIEW.

Subseqüentemente à modelagem do aplicativo, sua implementação se deu através de prototipação, a qual se mostrou como uma técnica de desenvolvimento perfeita para sistemas com muitos requisitos de interface com o usuário.

Frente aos seus objetivos, alcançou-se o desempenho de suas tarefas, realizando o controle de posição de sistemas eletro-hidráulicos e implementando o diagrama sistematizado de procedimentos para dimensionamento e análise estática e dinâmica de circuitos hidráulicos. Com isso, o aplicativo enquadrou-se nos objetivos do projeto da Plataforma de Hidráulica Proporcional, compreendendo um sistema para dimensionamento (referente ao módulo 2) e controle (referente ao módulo 1) de posicionadores hidráulicos.

Observa-se que o aplicativo desenvolvido se apresenta como uma ferramenta importante no aprendizado de alunos de graduação e pós-graduação, diferenciada, por sua abordagem e didática, das que existem na indústria.

Finalmente, entende-se que outros trabalhos poderão decorrer desta dissertação, principalmente no sentido de avaliar e/ou aprimorar os princípios de modelagem e projeto através da sua aplicação em campos distintos como equipamentos mecatrônicos com acionamentos elétricos ou pneumáticos.



## REFERÊNCIAS BIBLIOGRÁFICAS

ADAMOWSKI, J. C.; FURUKAWA, C. M. **Mecatrônica uma abordagem voltada à automação industrial**. Revista Mecatrônica Atual. São Paulo SP: Editora Saber Ltda. Ano-1. Nº1, Outubro-Novembro de 2001.

AGILENT TECHNOLOGIES. **16-Channel FET Multiplexer: Agilent E1351A Data Sheet**. United States: 2004. 4 p.

AGILENT TECHNOLOGIES. **Feeling Comfortable with VXIbus**. United States: apr. 2001. 28 p.

BACK, N. **Metodologia de projeto de produtos industriais**. Rio de Janeiro: Guanabara Dois, 1983. 389 p.

BAPTISTA, L. F. **Controlo de Sistemas I**. Apostila – Escola Náutica Infante D. Henrique, Paço de Arcos, Portugal, 2004.

BHATTACHARYYA, S., et al. **Heterogeneous Concurrent Modeling and Design in Java**. Memorandum UCB/ERL M02/23, University of California, Berkeley, CA USA 94720, Aug. 2002.

BOLLMANN, A. **Fundamentos da Automação Industrial Pneutrônica: Projetos de Comandos Binários Eletropneumáticos**. São Paulo: ABHP, 1996. 278 p.

BORGES, A. P. **Instrumentação Virtual Aplicada A Um Laboratório Com Acesso Pela Internet**. 128 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2002.

BOSCH REXROTH GROUP. **4/2 and 4/3 proportional directional valves direct operated, with electrical position feedback - Types 4WRE and 4WREE**. Bosch Rexroth: Sep. 2000.

BOSCH REXROTH GROUP. **DS3 Synergy**. Version 3.1. Apr. 2002. 108 p.

BOSCH REXROTH GROUP. **DS3-G**. Version 1.0. Apr. 2002. 4 p.

BRUSSEL, H. M. J. V. **Mechatronics – A Powerful Concurrent Engineering Framework**. IEEE/ASME Transactions on Mechatronics. vol. 1, no.2, March 1996. p.127-136.

BUSTEC PRODUCTION LTD. **VXI Data Acquisition Instrumentation by Bustec Production Ltd**. Disponível em: <www.bustec.ie>. Acesso em: novembro de 2003.

CARRO, L.; WAGNER, F. R. **Sistemas Computacionais Embarcados (cap. 2, em Jornadas de Atualização em Informática)**. XXII Congresso da SBC, Campinas, SP, 2003, p. 45-94.

CAVACO, M. A. M. **Parte II – Metrologia**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, 2002.

CHALMERS UNIVERSITY OF TECHNOLOGY. **Thermo and Fluid Dynamics Home Page**. <<http://www.tfd.chalmers.se/~hani/phdproject/proright.html>>. Acesso em julho de 2004.

COELHO, A. A. R.; ALMEIDA, O. M.; SANTOS, J. E. S.; SUMAR, R. R.; BITTENCOURT, F. **Da Teoria à Prática: Projeto Motor Taco-Gerador**. Cobenge 2001. 8 p.

CORTÉS, L. A.; ELES, P.; PENG, Z. **A Survey on Hardware/Software Codesign Representation Models**. SAVE Project Report, Dept. of Computer and Information Science, Linköping University, Linköping, Jun. 1999. 10 p.

COSTE, P., et al. **Multilanguage Design of Heterogeneous Systems**. 7th International Workshop on Hardware/Software Co-Design (CODES'99). 5 p.

CRISTALDI, L.; FERRERO, A.; PIURI, V. **Programmable instruments, virtual instruments, and distributed measurement systems: what is really useful, innovative and technically sound?**. IEEE Instrumentation & Measurement Magazine, vol. 2, Sep. 1999. p. 20-27.

CRNKOVIC, I.; LARSSON, M. **Building Reliable Component-Based Software Systems**. Norwood: Artech House, Jul, 2002. 454 p. ISBN 1-580-53327-2.

DAMASEVICIUS, R.; STUIKYS, V. **Application of UML for Hardware Design Based on Design Process Model**. Asia South Pacific Design Automation Conference (ASP-DAC 2004), Yokohama, Japan, Jan. 27-30, 2004. p. 244-249.

DE NEGRI, V. J. **Estruturação da modelagem de sistemas automáticos e sua aplicação a um banco de testes para sistemas hidráulicos**. 180 f. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 1996.

DE NEGRI, V. J. **Integração da Tecnologia Hidráulica e Pneumática com CLP's**. Apostila – Universidade Federal de Santa Catarina, Florianópolis, Jul. 1999. 52 p.

DE NEGRI, V. J. **Introdução aos Sistemas para Automação e Controle Industrial**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, Fev. 2004.

DE NEGRI, V. J. **Sistemas Automáticos: Conceitos, Modelos e Projeto**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, Mar. 1997. 41 p.

DE NEGRI, V. J. **Sistemas Hidráulicos e Pneumáticos para Controle e Automação: Partes I, II e III**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, Mar. 2001.

DE NEGRI, V. J., ATTÍE, S. S., TOLEDO, L. B. **Controle de posição utilizando servoválvulas e válvulas proporcionais eletro-hidráulicas (parte I)**. Revista ABHP, São Paulo, n. 106, p. 15-18, set/out 1997.

DE NEGRI, V. J., ATTÍE, S. S., TOLEDO, L. B. **Controle de posição utilizando servoválvulas e válvulas proporcionais eletro-hidráulicas (parte II)**. Revista ABHP, São Paulo, n. 107, p. 15-18, nov/dez 1997.

DE NEGRI, V. J.; KINCELER, R.; SILVEIRA, J. **Automação e controle experimental em hidráulica e pneumática**. Apostila – Universidade Federal de Santa Catarina, Florianópolis, Nov. 1998.

DE NEGRI, V. J.; VIEIRA, A, D. **Integração de tecnologias para a automação industrial com sistemas hidráulicos e pneumáticos**. Anais do V SEMINÁRIO NACIONAL DE HIDRÁULICA E NEUMÁTICA, Florianópolis, 1997. p. 81-101.

EMBRAER. **Embraer Home Page**. Disponível em: <<http://www.embraer.com.br>>. Acesso em: dezembro de 2004.

FLESCHE, C. A. **Circuitos Eletrônicos Básicos na Medição e Automação**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, 1997.

- FLESCH, C. A. **Sistemas de medição**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, 1998.
- FORCELLINI, F. A. **Projeto Conceitual**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, 2002. 161 p.
- FURST, F. L.; DE NEGRI, V. J. **Projeto de sistemas hidráulicos de controle de posição: Projeto PADCT/REIVAX, Capacitação industrial para construção de sistemas hidráulicos de controle de turbinas**. LASHIP – UFSC, Florianópolis, 2002, 125 p.
- FURST, L. F. **Sistematização do projeto preliminar de circuitos hidráulicos com controle de posição**. 132 f. Dissertação (Mestrado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 2001.
- GOMES, A. E. **Métricas e Estimativas de Software - O início de um rally de regularidade**. Disponível em: <[www.linhadecodigo.com.br](http://www.linhadecodigo.com.br)>. Acesso em: novembro de 2004.
- GONÇALVES JR, A. A. **Metrologia – Parte I**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, 2002.
- HEUSER, C. A. **Modelagem Conceitual de sistemas**. In: V EBAI, 1990. 150p.
- HEWLETT PACKARD. **HP E1326B/E1411B 5 1/2 Digit Multimeter User's Manual**. 4 ed. United States: 1996. 279 p.
- HEWLETT PACKARD. **HP E1328A D/A Converter Module User's Manual**. 5 ed. United States: 1996. 85 p.
- HEWLETT PACKARD. **HP E1343A/44A/45A/47A Relay Multiplexer User's Manual**. 5 ed. United States: 1997. 87 p.
- IFEACHOR, E. C.; JERVIS, B. W. **Digital Signal Processing: A Practical Approach**. Addison Wesley, 1993. ISBN: 0-201-54413-X.
- INMETRO – INSTITUTO NACIONAL DE METROLOGIA, NORMALIZAÇÃO E QUALIDADE INDUSTRIAL. **Vocabulário Internacional de Termos Fundamentais e Gerais de Metrologia** – Versão Brasileira do “Vocabulary of basic and general terms in metrology” publicado em 1993 pela ISSO/IEC/OIML/BIPM. Rio de Janeiro: INMETRO, mar. 1995.
- ISERMANN, R. **Modeling and design methodology for mechatronic systems**. IEEE/ASME Trans. Mechatronics. vol. 1, Mar. 1996. p. 16-28.
- ISO. **ISO 10770-1 – Hydraulic fluid power – Electrically modulated hydraulic control valves – Part 1: Test methods for four-way directional flow control valves**. ISO, Switzerland: 1998. 40 p.
- IVANTYSNOVA, M. **Fluid power education – demands of the 21st century**. PROCEEDINGS OF 1ST BRATISLAVIAN FLUID POWER SYMPOSIUM, 1998. Slovakia: Slovak University of Technology in Bratislava, 1998. p. 05-12.
- JALOTE, P. **An Integrated Approach to Software Engineering**. 2. ed. New York: Springer-Verlag, 1997. 497 p. ISBN: 0-387-94899-6.
- JOHN DEERE. **The John Deere Home Page**. <<http://www.deere.com>>. Acesso em: março de 2005.

JR., D. C. da S.; JR., C. J. N. C.; FERNANDES, A. O. **Desenvolvimento de Sistemas Integrados de Hardware e Software**. VI ERI-Sul – Escola Regional de Informática, Pelotas-Blumenau-Curitiba, Maio 1998. 10 p.

KILIAN, C. T. **Modern Control Technology: Components and Systems**. 2 ed. Delmar Thomson Learning, Dec. 2000. 636 p. ISBN: 0-766-82358-X.

LEFFINGWELL, D.; WIDRIG, D. **Managing Software Requirements**. 1. ed. Addison Wesley, Oct. 1999. 528 p. ISBN: 0-201-61593-2.

LINSINGEN, I. von. **Fundamentos de Sistemas Hidráulicos**. 1. ed. Florianópolis: Editora da UFSC, 2001. 399 p.

LIU, J.; LEE, E. A. **Component-based Hierarchical Modeling of Systems with Continuous and Discrete Dynamics**. Proc. of the 2000 IEEE International Conference on Control Applications and IEEE Symposium on Computer-Aided Control System Design (CCA/CACSD'00), Anchorage, AK, Sep. 25-27, 2000. p. 95-100.

LIU, X.; LIU, J.; EKER, J.; LEE, E. A. **Heterogeneous Modeling and Design of Control Systems**. Software-Enabled Control: Information Technology for Dynamical Systems, T. Samad and G. Balas (eds.), New York City: IEEE Press, 2003.

MARCON, C. A. M., et al. **Modelagem e Descrição de Sistemas Computacionais – Um Estudo de Caso de Comparação das Linguagens VHDL e SDL**. IBERCHIP, 2002. 10 p.

MAZZOLA, V. B.; FARINES, J-M. **Metodologias de Concepção de Software e de Sistemas**. Apostila – Universidade Federal de Santa Catarina, Florianópolis. 1998.

MCCONNELL, S. **Professional Software Development: Shorter Schedules, Higher Quality Products, More Successful Projects, Enhanced Careers**. 1. ed. Addison Wesley, Jun. 2003. 272 p. ISBN: 0-321-19367-9.

MCMILLAN, G. K.; CONSIDINE, D. M. **Process/Industrial Instruments and Controls Handbook**. 5 ed. McGraw-Hill, 1999. 1332 p. ISBN: 0-07-012582-1.

MROZEK, Z. **Computer Aided Design of Mechatronic Systems**. Int. J. Appl. Math. Comput. Sci., 2003, vol. 13, No. 2, p. 255-267.

NATIONAL INSTRUMENTS. **Instrupedia™, Your Interactive Encyclopedia of Measurement and Automation**. Estados Unidos: 2003. v. 8, n. 1. 1 disco compacto. 350484F-01.

NATIONAL INSTRUMENTS. **LabVIEW 7, User Manual**. Apr. 2003.

NATIONAL INSTRUMENTS. **LabVIEW Development Guidelines**. United States: apr. 2003. 167 p.

NATIONAL INSTRUMENTS. **National Instruments – Test and Measurement**. Disponível em: <www.ni.com>. Acesso em: janeiro de 2004.

NATIONAL INSTRUMENTS. **VXI Solutions Product Guide**. United States: 1999. p. 777-788.

NATIONAL INSTRUMENTS. **VXI VI Reference Manual**. United States: 2000. 167 p.

NI DAQ Designer™ 2003, Measurement and Automation Configuration Advisor. **National Instruments Corporation**. Estados Unidos: 2003. 1 disco compacto. 350401H-01.

- OGATA, K. **Engenharia de Controle Moderno**. 2. ed. Rio de Janeiro: Prentice Hall do Brasil, 1993.
- OTT, H. W. **Noise Reduction Techniques in Electronic Systems**. 2. ed. John Wiley & Sons, 1988. 488 p. ISBN 0-471-85068-3.
- PAGANO, D. J. **Teoria de Sistemas Amostrados e Controle Digital**. Apostila – Universidade Federal de Santa Catarina, Florianópolis, 1999. 70 p.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach**. New York: McGraw-Hill, 1987. 567 p. ISBN: 0-070-50783-X.
- PROAKIS, J. G.; MANOLAKIS, D. G. **Digital Signal Processing**. 3. ed. Prentice-Hall. 1996.
- PUTTEN, A. F. P. Van. **Electronic Measurement Systems. Theory & Practice**. 2. ed. Bristol: Institute of Physics Publishing, 1996.
- QUEIROZ, M. H.; CURY, J. E. R. **Modular supervisory control of large discrete-event systems**. Discrete event systems: analysis and control. Proc. WODES 2000, Ghent, Belgium: Kluwer Academic Publishers, 2000. p. 103-110.
- QUEIROZ, M. H.; SANTOS, E. A. P.; CURY, J. E. R. **Síntese modular do controle supervísório em diagrama escada para uma célula de manufatura**. Simpósio Brasileiro de Automação Inteligente, 5º, 2001, Canela – RS. Anais do V simpósio brasileiro de automação inteligente, Canela - RS, 2001.
- REIS, C. R. **Caracterização de um Processo de Software para Projetos de Software Livre**. 195 f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos, São Paulo. 2003.
- RICO, J. E. N. **Projeto de Controladores PID Industriais**. Apostila– Universidade Federal de Santa Catarina, Florianópolis, 2000. 39 p.
- RICO, J. E. N. **Sistemas Realimentados**. Apostila – Universidade Federal de Santa Catarina, Florianópolis, 2000. 182 p.
- SANGIOVANNI-VINCENTELLI, A.; MARTIN, G. **Platform-Based Design and Software Design Methodology for Embedded Systems**. IEEE Design & Test of Computers, vol. 18, issue 6, Nov. 2001, p. 23-33.
- SANTOS, E. A. P. **Contribuições ao projeto conceitual de sistemas de manipulação e montagem automatizados**. 194 f. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Santa Catarina, Florianópolis, 2003.
- SELNER, C. **Análise de Requisitos para Sistemas de Informações, Utilizando as Ferramentas da Qualidade e Processos de Software**. 228 f. Dissertação (Mestrado em Engenharia de Produção e Sistemas) – Universidade Federal de Santa Catarina, Florianópolis, 1999.
- SILVA, J. C. **Sistemas Especialistas**. Apostila (POSMEC) – Universidade Federal de Santa Catarina, Florianópolis, 2002. 69 p.
- SIMÕES, C. A. **Sistemática de Métricas, Qualidade e Produtividade**. Developers' Magazine, set. 1999.

SOUTO, R. B. **Projeto de Sistemas Automáticos com Modelagem e Controle da Comunicação com o Ambiente Externo**. Dissertação (Mestrado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 2005.

SOUZA, A. D. C.; DE NEGRI, V. J.; FURST, F. L. **Metodologia de Projeto de Posicionadores Hidráulicos: Parte - 1**. Revista Mecatrônica Atual. São Paulo SP: Editora Saber Ltda. Ano-3. Nº19. Dezembro de 2004-Janeiro de 2005.

SOUZA, A. D. C.; DE NEGRI, V. J.; FURST, F. L. **Metodologia de Projeto de Posicionadores Hidráulicos: Parte - 2**. Revista Mecatrônica Atual. São Paulo SP: Editora Saber Ltda. Ano-3. Nº20. Fevereiro-Março de 2005.

SOUZA, A. D. C.; DE NEGRI, V. J.; FURST, F. L. **Posicionadores Hidráulicos: Construções Típicas e Funcionamento**. Revista Mecatrônica Atual. São Paulo SP: Editora Saber Ltda. Ano-3. Nº18. Outubro-Novembro de 2004.

STAUNER, T.; PRETSCHNER, A.; PÉTER, I. **Approaching a hybrid UML-RT: Tool support and Formalization**. Technical report, TU München, 2001. 16 p.

TAVARES, T. A. **Análise e Projeto Estruturados de Sistemas**. Departamento de Informática e Matemática Aplicada – Universidade Federal do Rio Grande do Norte. 2002. 50 p.

THE MATHWORKS. **The MathWorks - MATLAB and Simulink for Technical Computing Home Page**. <<http://www.mathworks.com>>. Acesso em: julho de 2004.

TROFINO, A. **Sistemas Lineares**. Apostila (DAS) – Universidade Federal de Santa Catarina, Florianópolis, 2002. 171 p.

TSUKUMO, A. N., et al. **Modelos de Processo de Software: Visão Global e Análise Comparativa**. Anais da VII Conferência Internacional de Tecnologia de Software - Qualidade de Software, Curitiba, 1996. p. 219-234.

VAN ROMPAEY, K., et al. **CoWare – A design environment for heterogeneous hardware/software systems**. The Morgan Kaufmann Systems On Silicon Series: Readings in hardware/software co-design, 2001. p. 412-426.

VAROTO, A. C. **Visões em arquitetura de software**. 108 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Matemática de Estatística da Universidade de São Paulo, São Paulo, 2002.

VDC. **Welcome to The VXI Data Center**. Disponível em: <[www.vxidatcenter.com](http://www.vxidatcenter.com)>. Acesso em: novembro de 2003.

VOROS, N. S., et al. **Hardware/Software Co-Design of Complex Embedded Systems: An Approach Using Efficient Process Models, Multiple Formalism Specification and Validation via Co-Simulation**. Journal Design Automation for Embedded Systems, vol. 8, no. 1, 2003. 45 p.

VXI PLUG&PLAY ALLIANCE. **VXIPNP**. Disponível em: <[www.vxipnp.org](http://www.vxipnp.org)>. Acesso em: novembro de 2003.

VXI SOLUTIONS, INTERACTIVE CD. **National Instruments Corporation**. Estados Unidos: 2003. 1 disco compacto. 350753A-00.

VXI TECHNOLOGY, INC. **VXI TECHNOLOGY, INC**. Disponível em: <[www.vxitech.com](http://www.vxitech.com)>. Acesso em: novembro de 2003.

VXIBUS CONSORTIUM. **Welcome to the VXibus Consortium!**. Disponível em: <[www.vxibus.org](http://www.vxibus.org)>. Acesso em: novembro de 2003.

WIKIMEDIA FOUNDATION, INC. **Wikipedia, the free encyclopedia**. Disponível em: <<http://en.wikipedia.org>>. Acesso em: novembro de 2004.

WOLFE, R. **Short Tutorial on VXI/MXI**. National Instruments Corporation, Application Note 030, April 1996. 14 p.

YOUCEF-TOUMI, K. **Modeling, Design and Control Integration: A Necessary Step in Mechatronics**. IEEE/ASME Transactions on Mechatronics. vol. 1, no.1, Mar. 1996. p. 29-38.

ZURMELY, R. M. **Digitalização de um Sinal Analógico**. Disponível em: <<http://planeta.terra.com.br/lazer/py4zbz/teoria/digitaliz.htm>>. Acesso em: novembro de 2004.

## APÊNDICES



## APÊNDICE 1

### GUIA DE ESTILO EM LABVIEW

Esta seção descreve práticas indicadas para boa técnica de programação e estilo conforme NATIONAL INSTRUMENTS (2003c). No entanto, deve-se lembrar que estas são apenas recomendações e não regras rígidas.

O fato mais importante não é o estilo específico que se usa, mas o uso consistente daquele estilo. Desta maneira, são apresentadas a seguir listas de verificação de estilo as quais ajudam na manutenção da qualidade no desenvolvimento de VIs. Para se ganhar tempo, a revisão das listas antes e durante o desenvolvimento é necessária.

Ainda, segundo NATIONAL INSTRUMENTS (2003c), para se criar um VI com êxito, deve-se considerar quem precisa usar o VI e por quais razões. Assim, deve-se lembrar do seguinte:

- Usuários precisam de um painel frontal limpo.
- Desenvolvedores precisam de um diagrama de blocos de fácil leitura.
- Usuários e desenvolvedores precisam de documentação completa.

Portanto, pode-se utilizar esta seção e as listas de verificação de estilo como um ponto de partida para criar um guia de estilo próprio, personalizando este conforme as preferências de desenvolvimento e o conhecimento das especificações do projeto em questão.

#### ***Lista de Verificação de Estilo***

A utilização das listas de verificação seguintes serve para ajudar a manter o estilo consistente e a qualidade. Por consequência, pode-se copiar estas listas de verificação para usar em todos os projetos em LabVIEW, adicionando, removendo ou modificando-se quaisquer das seguintes diretrizes a fim de personalizar estas listas de verificação conforme as especificações do projeto em questão.

#### ***Lista de Verificação para VI***

- Organizar os VIs em um diretório hierárquico, onde os VIs de mais alto nível são facilmente acessados em seus diretórios e os subVIs em seus subdiretórios.
- Dar nomes significativos para os VIs sem uso de caracteres especiais.
- Distinguir entre VIs de exemplo, VIs de mais alto nível, subVIs, controles e variáveis globais os salvando em subdiretórios ou bibliotecas separadas no mesmo diretório, ou lhes dando nomes descritivos como X Principal.vi, X Exemplo.vi, X Global.vi, e X TipoDef.ctl.
- Escrever uma descrição do VI.

- Incluir o nome do programador e/ou da companhia e a data na Descrição do VI.
- Criar ícones significativos para os VIs.
- Evitar o uso de painéis de conectores com mais de 16 terminais.
- Ajustar as opções de impressão de modo que a exibição esteja no formato mais útil possível.
- Fazer testes com os VIs para conferir erros de condições, valores inválidos e botões de cancelamento.
- Salvar VIs de teste em um diretório separado para que se possa os usar novamente.
- Carregar e testar os VIs em várias plataformas para assegurar-se de que tamanho e posição de janela(s) estão corretos.
- Evitar a utilização de VIs em uma plataforma específica caso os VIs precisarem executar em várias plataformas.

#### ***Lista de Verificação de Painel Frontal***

- Dar nomes significativos aos controles, bem como utilizar adequadamente letras maiúsculas e minúsculas.
- Conferir a colocação consistente de nomes de controle.
- Colocar valores padrão entre parênteses nas legendas dos controles dos painéis frontais dos subVIs.
- Incluir informações de unidade em nomes. Por exemplo, se possível utilizar: Tempo Limite (10 Segundos).
- Escrever descrições para controles e indicadores.
- Organizar os controles logicamente. Para a interface com o usuário dos VIs, colocar os controles mais importantes nas posições mais proeminentes. Para os subVIs, pôr as entradas na esquerda e as saídas na direita. O painel de conectores deve seguir o mesmo esquema.
- Organizar os controles de modo atrativo, utilizando-se para isto as ferramentas para alinhamento e distribuição de objetos.
- Não sobrepor controles.
- Utilizar cores moderadamente.
- Prover um “Botão de Parada” se necessário para o VI.
- Utilizar controles enumerados para permitir futuras expansões de opções.
- Rotular os controles personalizados com o mesmo nome do arquivo.
- Ter certeza de que todos os controles no painel frontal são do mesmo estilo. Por exemplo, não usar controles do tipo clássico e 3D ao mesmo tempo no painel frontal.

***Lista de Verificação do Diagrama de Blocos***

- Evitar a criação de diagramas de blocos extremamente grandes. Limitar o tamanho horizontal e/ou verticalmente necessário para se ver o diagrama de blocos inteiro no monitor.
- Rotular controles, funções importantes, constantes, nodos de propriedade, variáveis locais, variáveis globais e estruturas.
- Dados de fluxo devem partir da esquerda para a direita.
- Alinhar e distribuir funções, terminais e constantes.
- Etiquetar longas linhas de fluxo de dados com pequenos rótulos de fundo branco.
- Evitar a colocação de objetos de diagramas de blocos, como subVIs ou estruturas, em cima de linhas de fluxo de dados.
- Fazer bom uso da reutilização, com subVIs testadas.
- Assegurar-se de que o programa pode lidar com as condições de erro e valores inválidos.
- Revisar a eficiência e precisão do VI.
- Caso se abram referências para um objeto do LabVIEW, como uma aplicação, controle ou VI, fechá-las convenientemente para que não permaneçam utilizando memória desnecessariamente.
- Utilizar nomes curtos para melhor legibilidade dos diagramas de blocos.
- Colocar qualquer controle do painel conector de um subVI no lado esquerdo do diagrama de blocos.

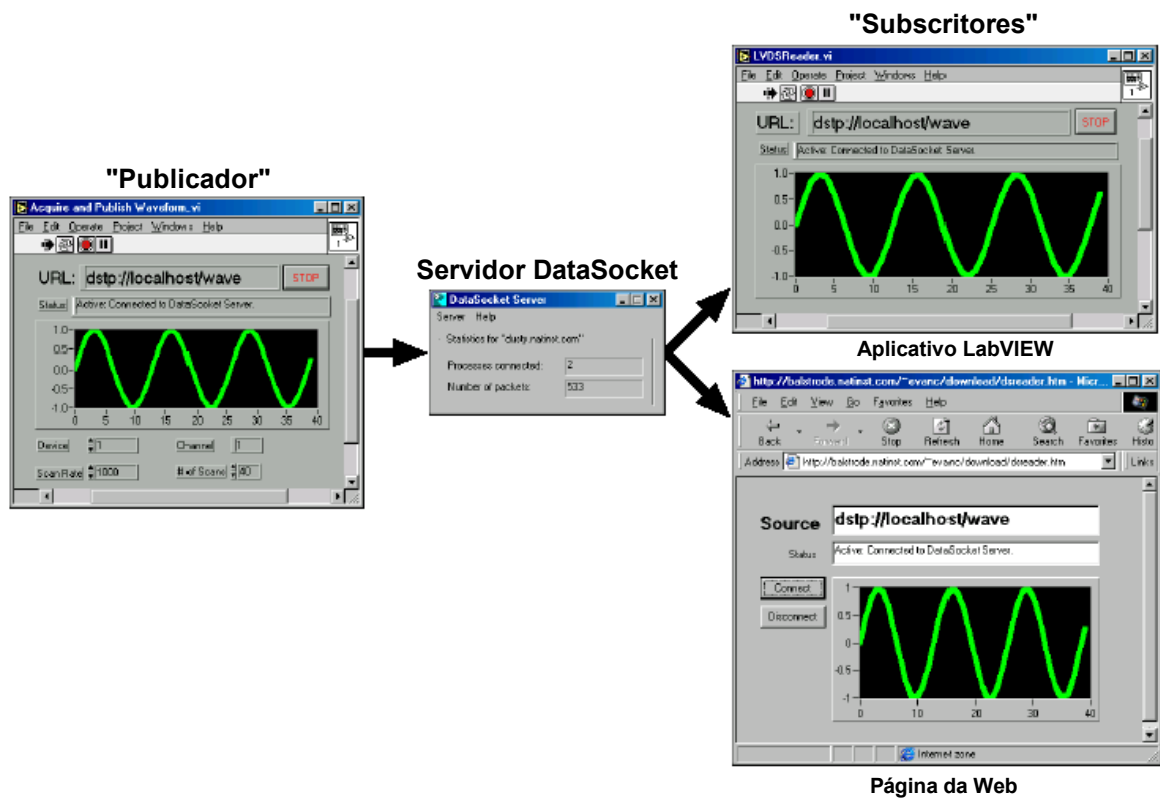
## APÊNDICE 2

### TECNOLOGIA DATASOCKET

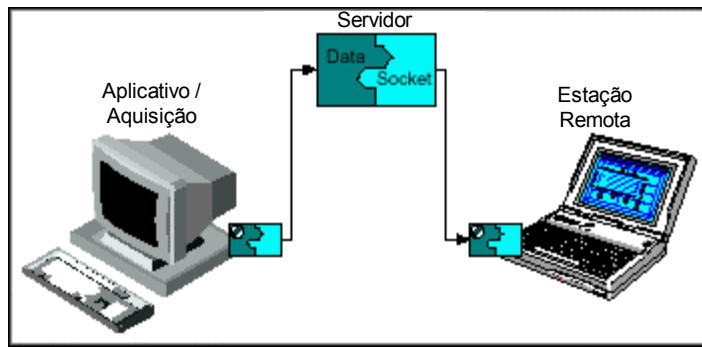
O *DataSocket* é uma coleção de ferramentas desenvolvida pela National Instruments que facilita a troca de dados e informações entre aplicativos e/ou diferentes fontes de dados e destinatários em rede.

Para tanto, requerem-se três elementos: um aplicativo que publique os dados (*publisher*), o *DataSocket Server* (Servidor DataSocket), e um aplicativo que subscreva os dados (*subscriber*).

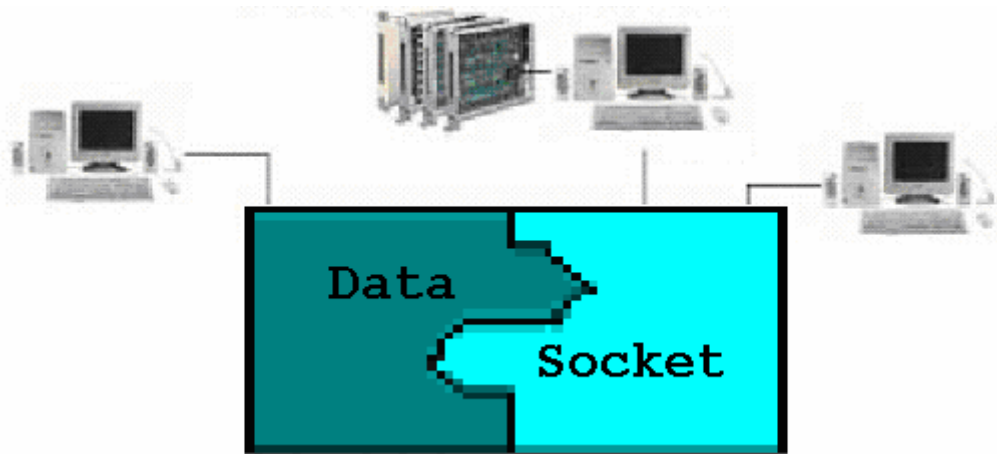
Como exibe a figura abaixo, o aplicativo “publicador” escreve os dados no servidor *DataSocket*. O aplicativo “subscritor” lê os dados do servidor. Sendo assim, tanto o “publicador” quanto o “subscritor” são clientes do *DataSocket Server*.



Como é ilustrado a seguir, os três elementos podem residir na mesma máquina, no entanto o mais comum é estarem em diferentes.



Então, com esta tecnologia pode-se executar aplicativos (ou módulos) sejam em uma mesma máquina ou em várias. Ainda, como é apresentado na figura seguinte, pode-se ter um computador conectado a um sistema de instrumentação e controle para que outros, além deste, acessem estes instrumentos através da leitura e escrita proporcionada pelo *DataSocket Server*.



Finalmente, como propósito deste apêndice mostra-se que os módulos (subVIs) que compõem um aplicativo em LabVIEW (VI) podem estar distribuídos em um computador ou mais. A maneira como os dados são trocados entre estes, é uma questão de implementação. Reafirma-se, com isto, que a modelagem do aplicativo do presente trabalho é independente do número de computadores que se possui no Módulo de Trabalho.

### APÊNDICE 3

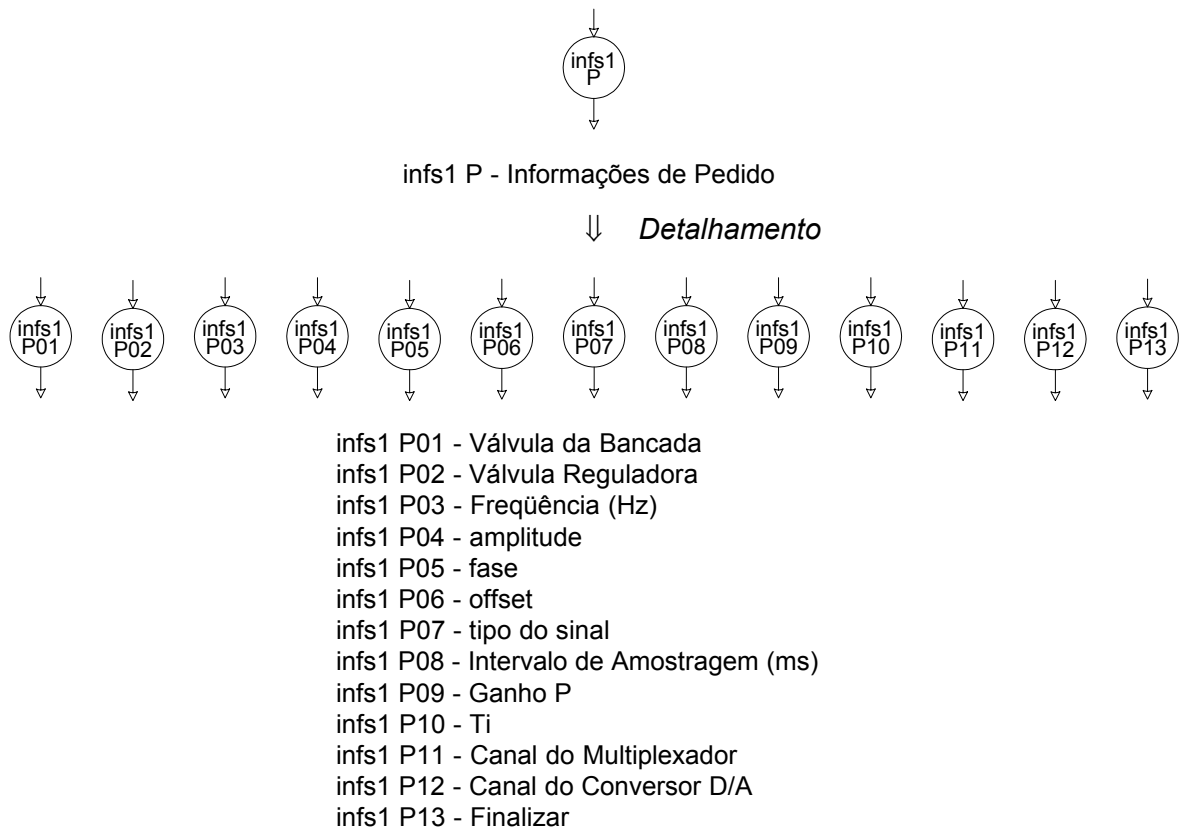
#### LEGENDA DAS INFORMAÇÕES DO SOFTWARE

Este apêndice lista as informações (canais) utilizadas na modelagem do software. Desta forma, as informações são apresentadas de forma genérica e posteriormente de forma detalha.

Para tanto, deve-se observar que:

- infs1: informações relacionadas ao Usuário 1;
- infs2: informações relacionadas ao Usuário 2.

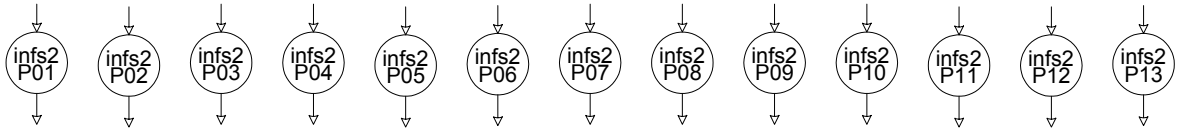
#### ***infs P:***





infs2 P - Informações de Pedido

↓ *Detalhamento*



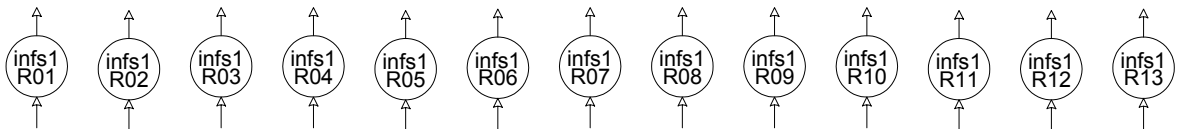
- infs2 P01 - Válvula da Bancada
- infs2 P02 - Válvula Reguladora
- infs2 P03 - Freqüência (Hz)
- infs2 P04 - amplitude
- infs2 P05 - fase
- infs2 P06 - offset
- infs2 P07 - tipo do sinal
- infs2 P08 - Intervalo de Amostragem (ms)
- infs2 P09 - Ganho P
- infs2 P10 - Ti
- infs2 P11 - Canal do Multiplexador
- infs2 P12 - Canal do Conversor D/A
- infs2 P13 - Finalizar

**infs R:**

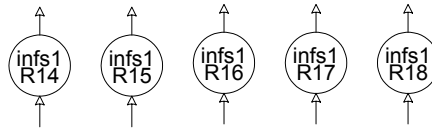


infs1 R - Informações de Resposta

↓ *Detalhamento*



- infs1 R01 - Válvula da Bancada
- infs1 R02 - Válvula Reguladora
- infs1 R03 - Freqüência (Hz)
- infs1 R04 - amplitude
- infs1 R05 - fase
- infs1 R06 - offset
- infs1 R07 - tipo do sinal
- infs1 R08 - Intervalo de Amostragem (ms)
- infs1 R09 - Ganho P
- infs1 R10 - Ti
- infs1 R11 - Canal do Multiplexador
- infs1 R12 - Canal do Conversor D/A
- infs1 R13 - Finalizar

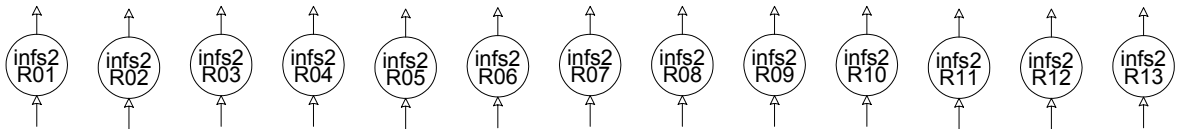


- infs1 R14 - Sinal de Referência Y
- infs1 R15 - Posição do Cilindro
- infs1 R16 - Ação de Controle
- infs1 R17 - Fase de Saída
- infs1 R18 - Desabilitação de Escolha de Válvulas

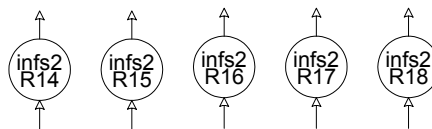
—



infs2 R - Informações de Resposta



- infs2 R01 - Válvula da Bancada
- infs2 R02 - Válvula Reguladora
- infs2 R03 - Frequência (Hz)
- infs2 R04 - amplitude
- infs2 R05 - fase
- infs2 R06 - offset
- infs2 R07 - tipo do sinal
- infs2 R08 - Intervalo de Amostragem (ms)
- infs2 R09 - Ganho P
- infs2 R10 - Ti
- infs2 R11 - Canal do Multiplexador
- infs2 R12 - Canal do Conversor D/A
- infs2 R13 - Finalizar



- infs2 R14 - Sinal de Referência Y
- infs2 R15 - Posição do Cilindro
- infs2 R16 - Ação de Controle
- infs2 R17 - Fase de Saída
- infs2 R18 - Desabilitação de Escolha de Válvulas

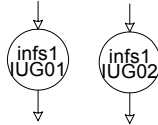


**infs IUG:**



infs1 IUG - Informações da "Interface com Usuário" para o "Gerenciador"

⇓ *Detalhamento*



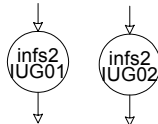
infs1 IUG01 - Válvula da Bancada  
infs1 IUG02 - Válvula Reguladora

—



infs2 IUG - Informações da "Interface com Usuário" para o "Gerenciador"

⇓ *Detalhamento*



infs2 IUG01 - Válvula da Bancada  
infs2 IUG02 - Válvula Reguladora

**infs GIU:**



infs1 GIU - Informações do "Gerenciador" para a "Interface com Usuário"

⇓ *Detalhamento*



infs1 GIU01 - Desabilitação de Escolha de Válvulas

—



infs2 GIU - Informações do "Gerenciador" para a "Interface com Usuário"

⇓ *Detalhamento*



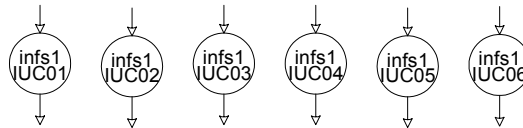
infs2 GIU01 - Desabilitação de Escolha de Válvulas

**infs IUC:**



infs1 IUC - Informações da "Interface com Usuário" para o "Controlador"

⇓ *Detalhamento*



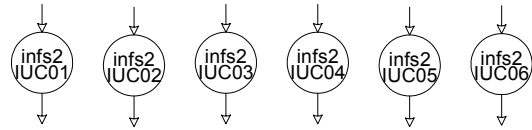
- infs1 IUC01 - Sinal de Referência Y
- infs1 IUC02 - Intervalo de Amostragem (ms)
- infs1 IUC03 - Ganho P
- infs1 IUC04 - Ti
- infs1 IUC05 - Canal do Multiplexador
- infs1 IUC06 - Canal do Conversor D/A

—



infs2 IUC - Informações da "Interface com Usuário" para o "Controlador"

⇓ *Detalhamento*



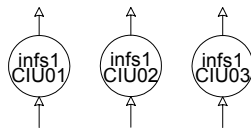
infs2 IUC01 - Sinal de Referência Y  
 infs2 IUC02 - Intervalo de Amostragem (ms)  
 infs2 IUC03 - Ganho P  
 infs2 IUC04 - Ti  
 infs2 IUC05 - Canal do Multiplexador  
 infs2 IUC06 - Canal do Conversor D/A

**infs CIU:**



infs1 CIU - Informações do "Controlador" para a "Interface com Usuário"

⇓ *Detalhamento*



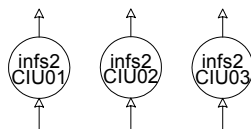
infs1 CIU01 - Sinal de Referência Y  
 infs1 CIU02 - Posição do Cilindro  
 infs1 CIU03 - Ação de Controle

—

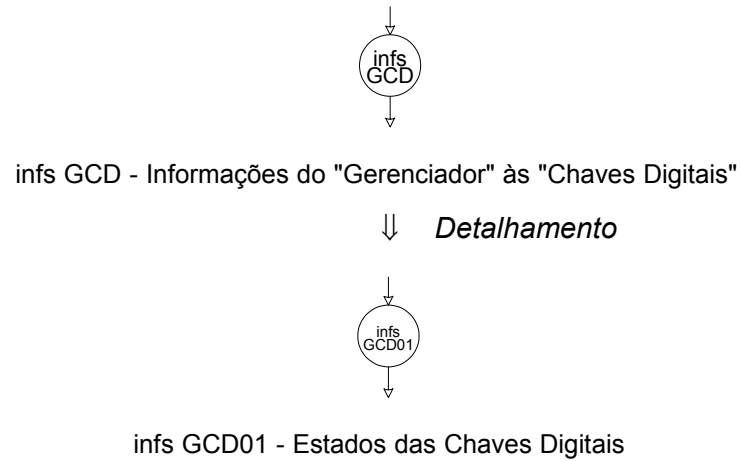
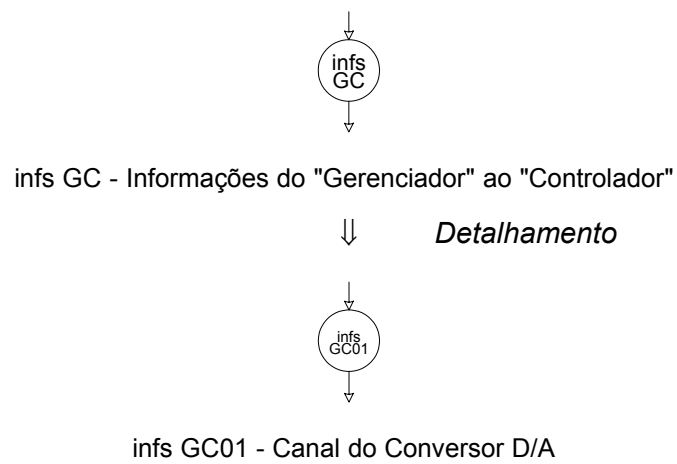
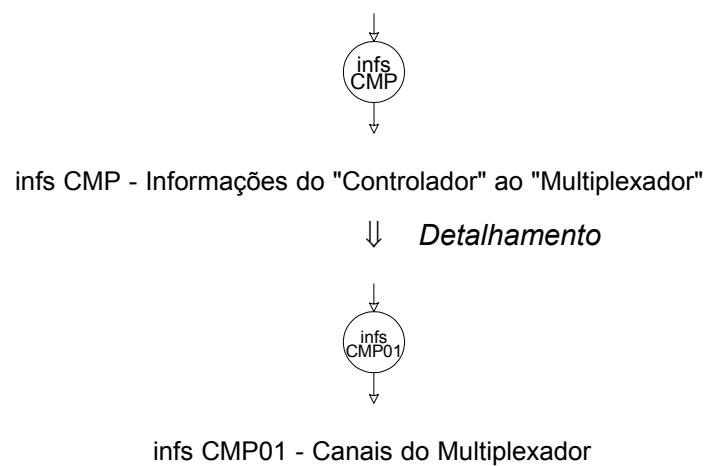


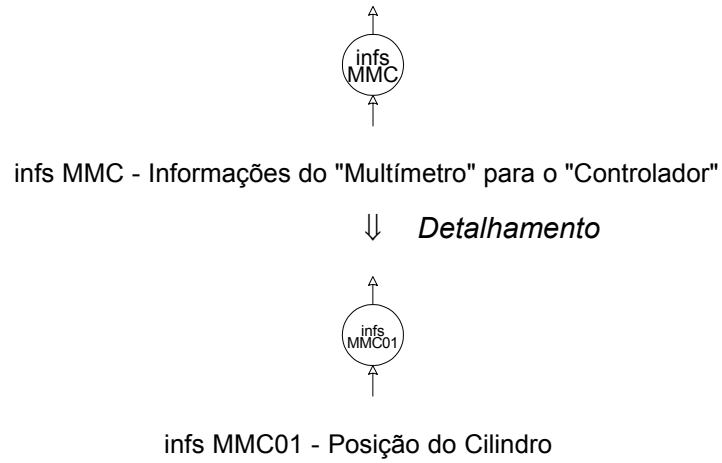
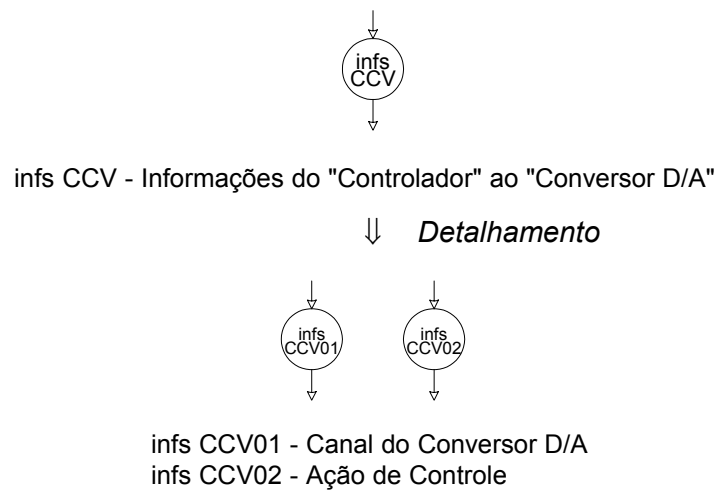
infs2 CIU - Informações do "Controlador" para a "Interface com Usuário"

⇓ *Detalhamento*



infs2 CIU01 - Sinal de Referência Y  
 infs2 CIU02 - Posição do Cilindro  
 infs2 CIU03 - Ação de Controle

**infs GCD:****infs GC:****infs CMP:**

**infs MMC:****infs CCV:**

APÊNDICE 4

ESPECIFICAÇÕES GERAIS DO MULTÍMETRO

Appendix A  
**HP E1326B/E1411B Multimeter Specifications**

**General Specifications**

**Reading Rate Conditions:** Autozero off, fixed range, default trigger delay, offset comp off, Sample Source 'TIMER' for rates >15 readings/second.

**Noise Rejection Conditions:** CMR measured with 1kΩ in both HIGH and LOW leads with a 10% imbalance, LOW connected to COMMON at source, measured with respect to earth ground. NMR is for specified frequencies ±0.1%.

	320 ms	267 ms	Aperture		2.5 ms	100 μs	10 μs
			20 ms	16.7 ms			
<b>Typical Reading Rates (rdgs/sec)</b>							
DC voltage	3	3.5	49	59	365	3125	13000
Four-wire resistance	3	3.5	49	59	365	3125	13000
AC voltage	1.3	1.4	1.9	1.9	1.9	1.9	1.9
<b>Resolution (bits/digits)</b>							
Binary bits	±22	±22	±20	±20	±18	±15	±14
Decimal digits	6.5	6.5	6	6	5.5	4.5	4
<b>Noise Rejection (dB)</b>							
<b>DC Voltage &amp; Resistance</b>							
DC Common mode rejection	150 dB	150 dB	150 dB	150 dB	150 dB	150 dB	150 dB
50Hz Power line cycles (NPLCs)	16	--	1	--	--	--	--
Normal mode (50 Hz) rejection	84 dB	0 dB	60 dB	0 dB	0 dB	0 dB	0 dB
60Hz Power line cycles (NPLCs)	--	16	--	1	--	--	--
Normal mode (60 Hz) rejection	0 dB	84 dB	0 dB	60 dB	0 dB	0 dB	0 dB
400Hz Power line cycles (NPLCs)	128	--	8	--	1	--	--
Normal mode (400 Hz) rejection	84 dB	0 dB	84 dB	0 dB	60 dB	0 dB	0dB
<b>AC Voltage: DC to 400 Hz</b>							
Common mode rejection	110 dB	110 dB	110 dB	110 dB	110 dB	110 dB	110 dB

**Input Characteristics:** Maximum Nondestructive Input (volts)

Terminals	DC	AC RMS	AC Peak
Hi to lo	300	300	450
Hi to common	300	300	450
Hi to chassis	300	300	450
Lo to common	15	10	15
Lo to chassis	300	300	450
Common to chassis	300	300	450

Input amplifier bandwidth: 80 kHz

**Autorange settling time:**

The multimeter autoranges and settles faster than its minimum (fastest) sample rate.

**DC Voltage**

Accuracy conditions for table below: Autozero on, one hour warmup. Temperature within  $\pm 5^{\circ}\text{C}$  of calibration temperature (module calibrated at  $18^{\circ}\text{C}$  to  $28^{\circ}\text{C}$ ).

Resolution vs. Aperture (volts)						
Range	Input Resistance	267/320 ms	16.7/20 ms	2.5 ms	100 $\mu\text{s}$	10 $\mu\text{s}$
125 mV	>100 M $\Omega$	.03 $\mu\text{V}$	.12 $\mu\text{V}$	0.5 $\mu\text{V}$	4.0 $\mu\text{V}$	7.6 $\mu\text{V}$
1 V	>100 M $\Omega$	.24 $\mu\text{V}$	1.0 $\mu\text{V}$	4.0 $\mu\text{V}$	30 $\mu\text{V}$	61 $\mu\text{V}$
8 V	>100 M $\Omega$	2.0 $\mu\text{V}$	7.6 $\mu\text{V}$	30 $\mu\text{V}$	250 $\mu\text{V}$	488 $\mu\text{V}$
64 V	10 M $\Omega$ $\pm 5\%$	15 $\mu\text{V}$	61 $\mu\text{V}$	250 $\mu\text{V}$	2.0 mV	3.9 mV
300 V	10 M $\Omega$ $\pm 5\%$	122 $\mu\text{V}$	488 $\mu\text{V}$	2.0 mV	16 mV	31 mV

Accuracy conditions for table below: Autozero on, one hour warmup. Within 24 hours and  $\pm 1^{\circ}\text{C}$  of calibration temperature (module calibrated at  $18^{\circ}\text{C}$  to  $28^{\circ}\text{C}$ ).

24-Hour Accuracy vs. Aperture $\pm$ (% of reading + volts)					
Range	267/320 ms	16.7/20 ms	2.5 ms	100 $\mu\text{s}$	10 $\mu\text{s}$
125 mV	.008% + 5.0 $\mu\text{V}$	.008% + 5.0 $\mu\text{V}$	.008% + 10 $\mu\text{V}$	.05% + 30 $\mu\text{V}$	.05% + 60 $\mu\text{V}$
1 V	.008% + 10 $\mu\text{V}$	.008% + 15 $\mu\text{V}$	.008% + 15 $\mu\text{V}$	.05% + 100 $\mu\text{V}$	.05% + 200 $\mu\text{V}$
8 V	.005% + 50 $\mu\text{V}$	.005% + 50 $\mu\text{V}$	.005% + 80 $\mu\text{V}$	.05% + 750 $\mu\text{V}$	.05% + 1.5 mV
64 V	.01% + 1.0 mV	.01% + 1.0 mV	.01% + 1.0 mV	.05% + 5.0 mV	.05% + 20 mV
300 V	.01% + 5.0 mV	.01% + 5.0 mV	.01% + 5.0 mV	.05% + 30 mV	.05% + 80 mV

Accuracy conditions for table below: Autozero on, one hour warmup. Within 90 days and  $\pm 5^{\circ}\text{C}$  of calibration temperature (module calibrated at  $18^{\circ}\text{C}$  to  $28^{\circ}\text{C}$ ).

90-Day Accuracy vs. Aperture $\pm$ (% of reading + volts)					
Range	267/320 ms	16.7/20 ms	2.5 ms	100 $\mu\text{s}$	10 $\mu\text{s}$
125 mV	.023% + 5.0 $\mu\text{V}$	.023% + 5.0 $\mu\text{V}$	.023% + 10 $\mu\text{V}$	.065% + 30 $\mu\text{V}$	.115% + 60 $\mu\text{V}$
1 V	.013% + 10 $\mu\text{V}$	.013% + 15 $\mu\text{V}$	.013% + 15 $\mu\text{V}$	.055% + 100 $\mu\text{V}$	.1% + 200 $\mu\text{V}$
8 V	.010% + 50 $\mu\text{V}$	.01% + 50 $\mu\text{V}$	.01% + 80 $\mu\text{V}$	.055% + 750 $\mu\text{V}$	.1% + 1.5 mV
64 V	.015% + 1.0 mV	.015% + 1.0 mV	.015% + 1.0 mV	.055% + 5.0 mV	.1% + 20 mV
300 V	.015% + 5.0 mV	.015% + 5.0 mV	.015% + 5.0 mV	.055% + 30 mV	.1% + 80 mV

Accuracy conditions for table below: Autozero on, one hour warmup. Within 1 year and  $\pm 5^{\circ}\text{C}$  of calibration temperature(module calibrated at  $18^{\circ}\text{C}$  to  $28^{\circ}\text{C}$ ).

1-Year Accuracy vs. Aperture $\pm$ (% of reading + volts)					
Range	267/320 ms	16.7/20 ms	2.5 ms	100 $\mu\text{s}$	10 $\mu\text{s}$
125 mV	.033% + 5.0 $\mu\text{V}$	.033% + 5.0 $\mu\text{V}$	.033% + 10 $\mu\text{V}$	.075% + 30 $\mu\text{V}$	.125% + 60 $\mu\text{V}$
1 V	.023% + 10 $\mu\text{V}$	.023% + 15 $\mu\text{V}$	.023% + 15 $\mu\text{V}$	.065% + 100 $\mu\text{V}$	.110% + 200 $\mu\text{V}$
8 V	.020% + 50 $\mu\text{V}$	.020% + 50 $\mu\text{V}$	.020% + 80 $\mu\text{V}$	.065% + 750 $\mu\text{V}$	.110% + 1.5 mV
64 V	.025% + 1.0 mV	.025% + 1 mV	.025% + 1.0 mV	.065% + 5.0 mV	.110% + 20 mV
300 V	.025% + 5.0 mV	.025% + 5 mV	.025% + 5.0 mV	.065% + 30 mV	.110% + 80 mV

Temperature Coefficient $\pm$ (% of reading)/ $^{\circ}\text{C}$		
Range	Temperature Coefficient	10 $\mu\text{s}$ Aperture
125 mV	0.003	0.013
1 V	0.001	0.01
8 V	0.001	0.01
64 V	0.001	0.01
Conditions: $0^{\circ}\text{C}$ to (cal temp $- 5^{\circ}\text{C}$ ), (cal temp $+ 5^{\circ}\text{C}$ ) to $55^{\circ}\text{C}$ .		

Autozero Off Offset Error (volts)	
Range	Additional Offset Error
125 mV	10 $\mu\text{V}$
1 V	10 $\mu\text{V}$
8 V	10 $\mu\text{V}$
64 V	1 mV
Conditions: Stable environment, 24 hours, $\pm 1^{\circ}\text{C}$ .	



**Timer/Pacer**

Timer Range: 76 µsec to 65.5 msec  
 Resolution: 2 µs  
 Accuracy: 0.01%

**Programmable Delay**

Delay Range: 40 µsec to 16 sec  
 Resolution: 2 µsec  
 Accuracy: 0.01%

**External Trigger**

Trigger Condition: negative edge  
 Minimum Pulse Width: 100 nsec  
 Maximum Trigger Rate: 5 kHz  
 (fixed range, 10 µsec aperture)

**Typical Reading Storage**

<b>HP 75000 Mainframe</b>	<b>Number of Readings</b>
Series B with standard memory	50,000
Series B with 512 kB memory (HP E1300/01A Opt. 010)	100,000
Series B with 1 MB memory (HP E1300/01A Opt. 011)	200,000
Series C with Command Module (HP E1406A)	100,000
Series C plus E1406A with Opt. 010	200,000

**Module Size/Device Type:**

B, register-based (E1326B)  
 C, register-based (E1411B)

**Connectors Used:** P1

**No. Slots:** 1 (standard terminal panel takes 2 slots)

**VXIbus Interface Capability:**

Slave, interrupter, A16, D16

**Interrupt Level:** 1-7, selectable

**Power Requirements:**

	<b>±5</b>	<b>±12</b>
Peak module current, IPM (A):	0.20	0.55
Dyanmic module current, IDM (A):	0.01	0.01

**Watts/Slot:** 8.5 (E1411B) 4.2 (E1326B)

**Cooling/Slot:**

0.14 mm H<sub>2</sub>O @ 0.71 liter/sec (E1411B)  
 0.07 mm H<sub>2</sub>O@ 0.35 liter/sec (E1326B)

**Humidity:** 65% 0° to 40°C

**Operating Temperature:** 0° to 55°C

**Storage Temperature:** -40° to 75°C

**EMC, RFI, Safety:**

meets FTZ 1046/1984, CSA 556B.  
 IEC 348, UL 1244

**APÊNDICE 5****DOCUMENTAÇÃO DO SOFTWARE AO PROJETISTA/PROGRAMADOR DO MÓDULO  
“INTERFACE COM USUÁRIO (USUÁRIO 1)”****Interface com Usuário (Usuário 1).vi**

Este VI, chamado "Interface com Usuário (Usuário 1)", faz a interface com o Usuário 1.

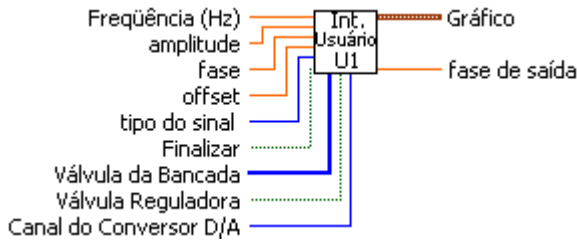
Para tanto, realiza a aquisição das seguintes informações do Usuário 1:

- Válvula da Bancada;
- Válvula Reguladora;
- Frequência (Hz);
- amplitude;
- fase;
- offset;
- tipo do sinal;
- Intervalo de Amostragem (ms);
- Ganho P;
- $T_i$  ;
- Canal do Multiplexador;
- Canal do Conversor D/A;

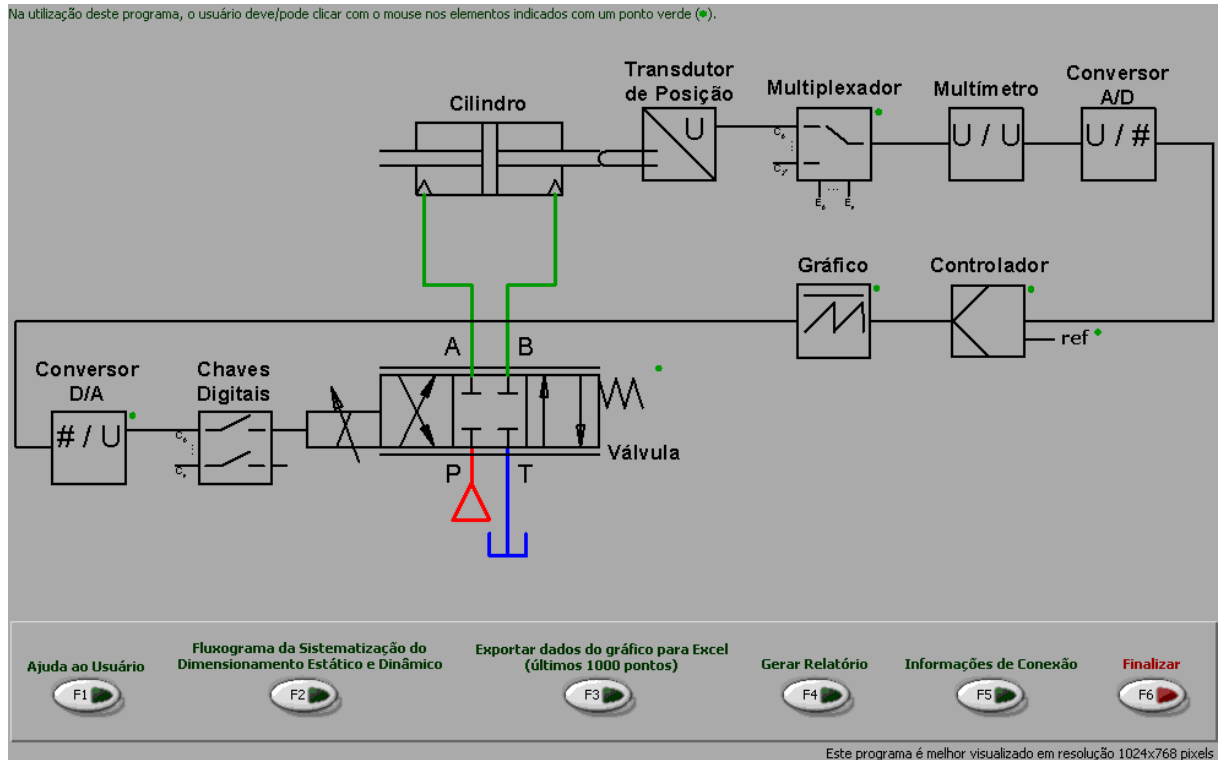
E retorna as seguintes informações ao Usuário 1, providas dos módulos "Controlador" e "Gerenciador":

- Sinal de Referência;
  - Posição do Cilindro;
  - Ação de Controle;
  - Fase de Saída;
  - Desabilitação de Escolha de Válvulas;
-

**Connector Pane**



**Front Panel**



**Controls and Indicators**

**[I32] Válvula da Bancada**

É a válvula que o usuário deseja utilizar na bancada.

**[DBL] Freqüência (Hz)**

É a freqüência do sinal a ser gerado.

**[DBL] amplitude**

É a amplitude do sinal a ser gerado. A amplitude é também o valor de pico de tensão.

**[DBL] fase**

É a fase inicial, em graus, do sinal a ser gerado.

**DBL offset**

É o DC offset do sinal a ser gerado.

**DBL tipo do sinal**

É o tipo de sinal a ser gerado.

**TF Finalizar**

É o botão que finaliza a execução de "Interface com Usuário".

**TF Válvula Reguladora**

Especifica se o usuário deseja utilizar a válvula reguladora da bancada.

Para o Usuário 1, isto implica na escolha da válvula reguladora V7.

Para o Usuário 2, isto implica na escolha da válvula reguladora V8.

**I32 Canal do Conversor D/A**

É o canal a ser utilizado do conversor D/A (HPE1328A) do sistema VXI.

Para o Usuário 1, isto implica na escolha do canal do conversor D/A a ser utilizado pelo Usuário 1 no controle da Válvula da Bancada.

Para o Usuário 2, isto implica na escolha do canal do conversor D/A a ser utilizado pelo Usuário 2 no controle da Válvula da Bancada.

Valor padrão (valores ajustados conforme a Documentação da Bancada Proporcional):

Para o Usuário 1, Channel 1.

Para o Usuário 2, Channel 2.

**DBL fase de saída**

É a fase de saída do sinal gerado em graus.

**DBL Gráfico**

Mostra os dados para o usuário.

- Posição do Cilindro (volt);

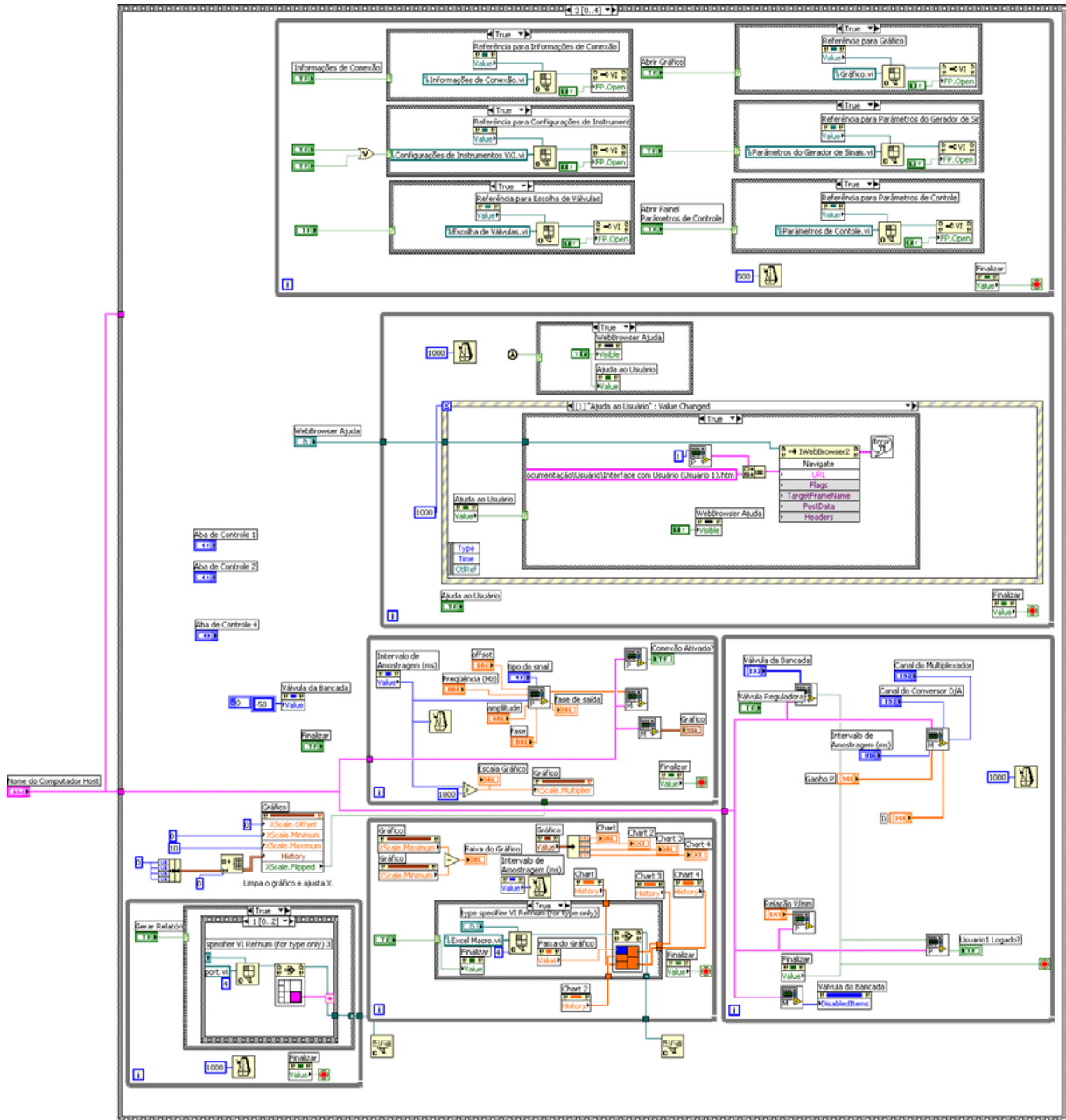
- Ação de Controle (volt);

- Referência (volt);

- Posição do Cilindro (mm);

---

**Block Diagram**



**List of SubVIs**



**Envio dos Parâmetros de Escolha - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Envio dos Parâmetros de Escolha - IU1.vi



**Aquisição de Dados para Mostrar no Gráfico - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Aquisição de Dados para Mostrar no Gráfico - IU1.vi

**Envio do Sinal de Log do Usuario (Usuário 1) - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Envio do Sinal de Log do Usuario (Usuário 1) - IU1.vi

**Gerador de Sinais - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Gerador de Sinais - IU1.vi

**Envio de Dados de Escolha de Válvulas - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Envio de Dados de Escolha de Válvulas - IU1.vi

**Aquisição de Dados de Desabilitação de Válvulas - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Aquisição de Dados de Desabilitação de Válvulas - IU1.vi

**Envio do Sinal de Referência - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Envio do Sinal de Referência - IU1.vi

**Simple Error Handler.vi**

C:\Arquivos de programas\National Instruments\LabVIEW 6.1\vi.lib\Utility\error.llb\Simple Error Handler.vi

**Aquisição do Path para String - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Aquisição do Path para String - IU1.vi

**Aquisição de Dados de Conexão Estabelecida - IU1.vi**

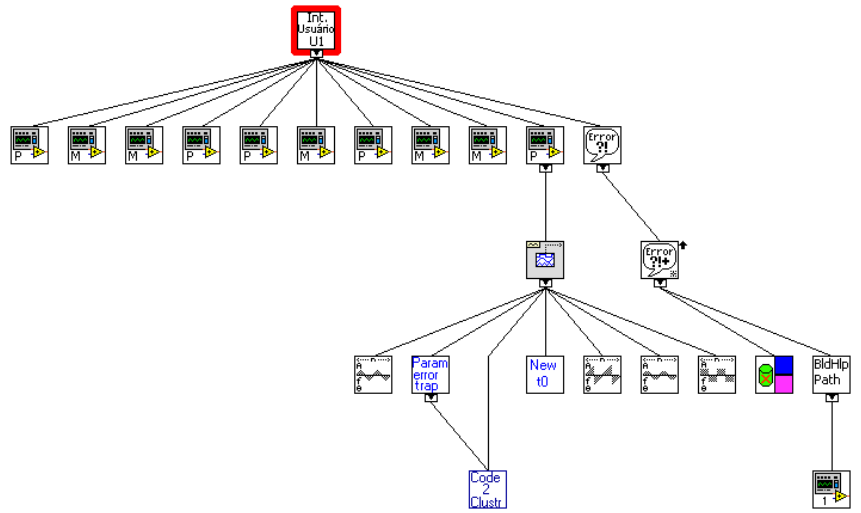
D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Aquisição de Dados de Conexão Estabelecida - IU1.vi

**Envio da Relação V-mm - IU1.vi**

D:\Users\alisson\Projeto Bancada\Programas em LabView\Bancada Hidráulica Proporcional\Computador1\Interface com Usuário (Usuário 1)\Envio da Relação V-mm - IU1.vi

---

### Position in Hierarchy



**APÊNDICE 6****DOCUMENTAÇÃO DO SOFTWARE AO USUÁRIO DO MÓDULO “INTERFACE COM USUÁRIO (USUÁRIO 1)”****Ajuda ao Usuário**

A "Interface com Usuário (Usuário 1)", como o nome sugere, faz a interface com o Usuário 1.

Para tanto, realiza a aquisição das seguintes informações do Usuário 1:

- Válvula da Bancada;
- Válvula Reguladora;
- Frequência (Hz);
- amplitude;
- fase;
- offset;
- tipo do sinal;
- Intervalo de Amostragem (ms);
- Ganho P;
- $T_i$  ;
- Canal do Multiplexador;
- Canal do Conversor D/A;

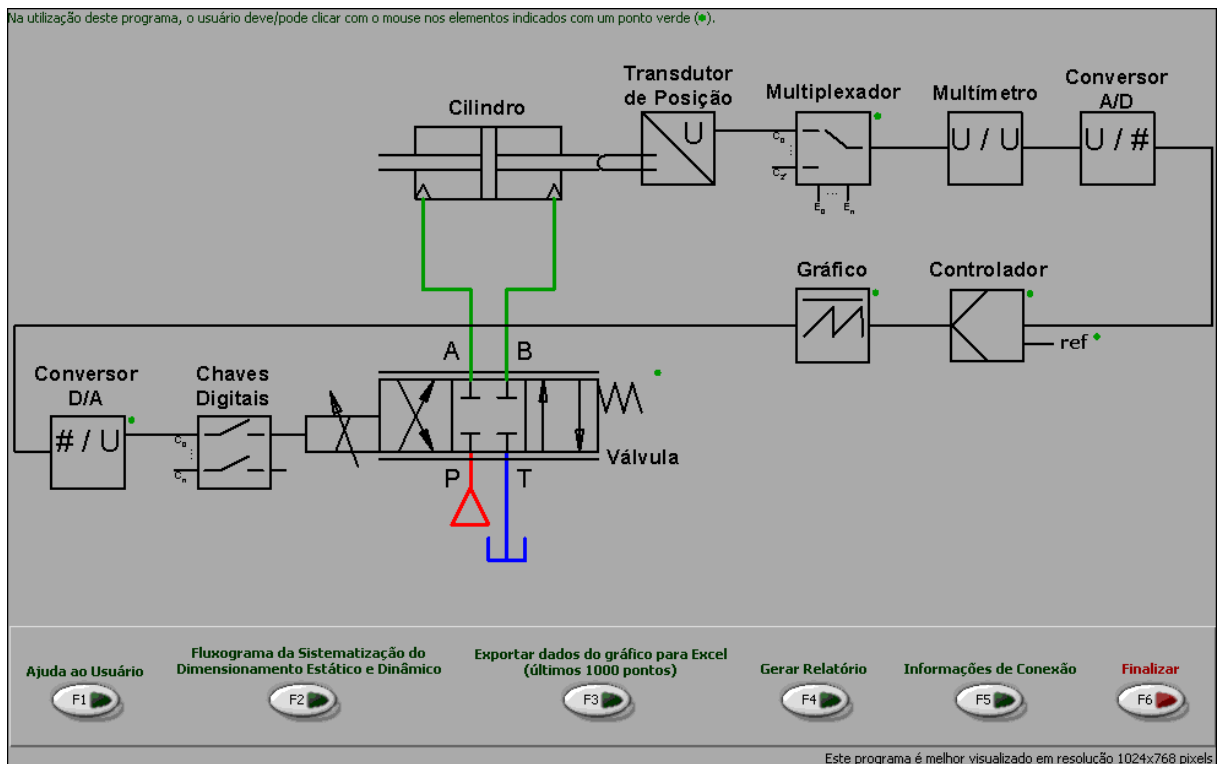
E retorna as seguintes informações ao Usuário 1:

- Sinal de Referência;
- Posição do Cilindro;
- Ação de Controle;
- Fase de Saída;
- Desabilitação de Escolha de Válvulas;

---

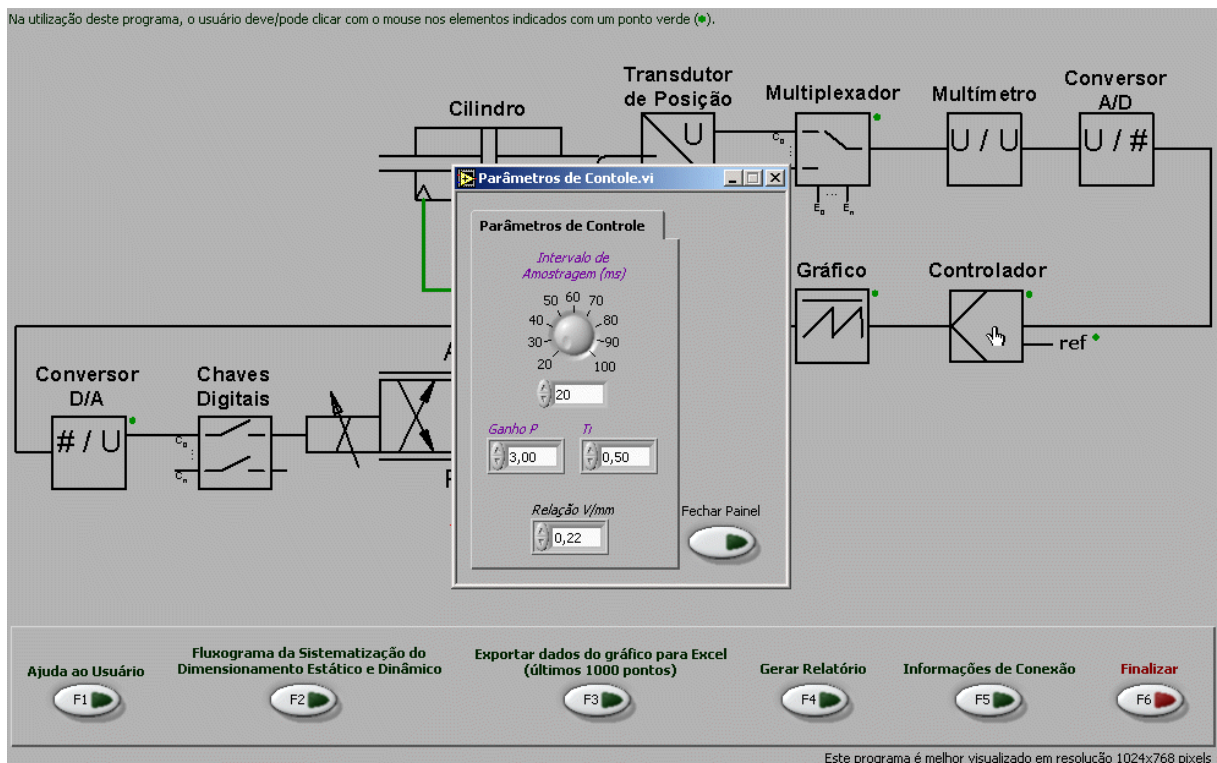
Para isso, esta interface possui um painel frontal onde se deve/pode clicar com o mouse nos elementos indicados com um ponto verde ( ● ).





Ao clicar-se em um destes elementos, uma janela referente ao elemento clicado é aberta para que o usuário possa realizar escolhas através de controles e/ou visualizar indicações através de indicadores.

Por exemplo, a janela "Parâmetros de Controle" é aberta quando se clica sobre o elemento "Controlador" como mostrado abaixo:



Ainda, pode-se abrir mais de uma janela ao mesmo tempo. Para isto, basta clicar nos elementos que se deseja abrir sucessivamente. Para maior comodidade, estas janelas podem ser arrastadas de modo à que melhor se adequem à preferência do usuário.

Para se fechar as referidas janelas basta clicar-se no botão "Fechar Painel" de cada janela aberta.

---

Pressionando-se a tecla de função F2, chama-se o módulo "Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico", utilizado para se fazer o dimensionamento estático e dinâmico de Sistemas Hidráulicos de Controle de Posição.

As informações referentes ao manuseio deste programa são obtidas no mesmo.

---

Pressionando-se a tecla de função F3, gera-se um arquivo em excel com os dados mostrados no gráfico.

Após um tempo de processamento, que varia com o PC utilizado, um gráfico é gerado no arquivo "dados.xls" quando ocorrer a finalização da passagem dos dados do software LabView à tabela do arquivo "dados.xls" do software excel.

\*Esta ação finaliza o programa "Interface com Usuário (Usuário 1)".

---

Pressionando-se a tecla de função F4, gera-se um arquivo em html com alguns dados relevantes para o usuário.

Este é chamado de "Relatório de Interface com Usuário".

Após um tempo de processamento, que varia com o PC utilizado, o usuário é informado sobre a localização do arquivo foi gravado.

\*Esta ação finaliza o programa "Interface com Usuário (Usuário 1)".

---

Pressionando-se a tecla de função F5, obtém-se informações sobre a comunicação DataSocket do LabView.

Se indicar verdade, os módulos do programa estão se comunicando corretamente.

Se indicar falso, deve-se executar o módulo de "Inicialização", localizado no computador host (Computador1).

---

Pressionando-se a tecla de função F6, finaliza-se o programa "Interface com Usuário (Usuário 1)".

## APÊNDICE 7

### AJUDA DO “PASSO 1” DO MÓDULO “FLUXOGRAMA DA SISTEMATIZAÇÃO DO DIMENSIONAMENTO ESTÁTICO E DINÂMICO”

Como se pode observar a seguir, na ajuda do Passo 1 é apresentado o fluxograma geral da sistematização de procedimentos para dimensionamento e análise estática e dinâmica de circuitos hidráulicos<sup>39</sup> desenvolvido através da dissertação de mestrado de FURST (2001).

Apesar de não serem mostradas neste documento, há de se saber que os outros passos também possuem uma ajuda semelhante a esta.

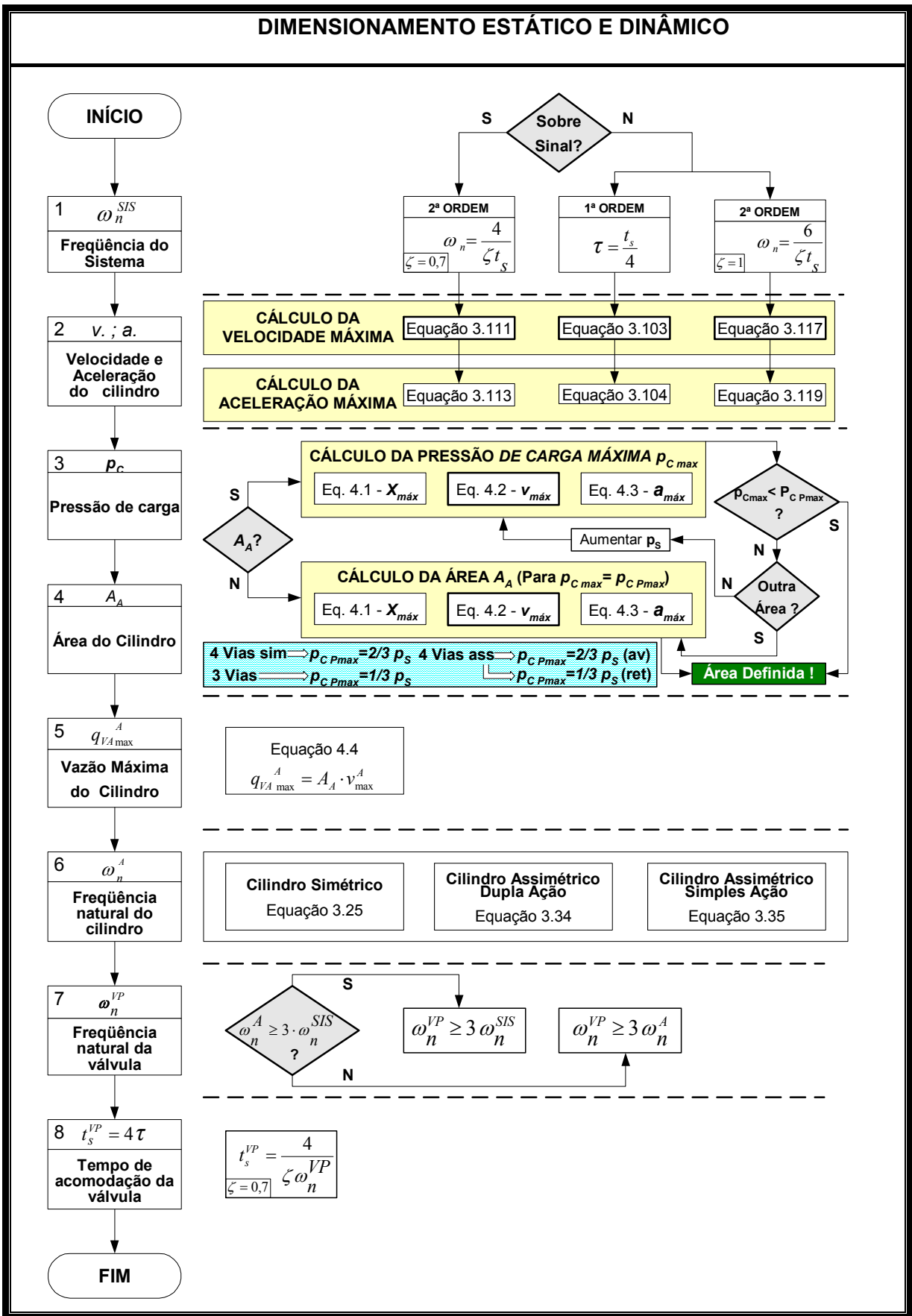
#### ***Ajuda do “Passo 1”***

##### **Início**

Conforme o fluxograma apresentado na figura a seguir, inicia-se o procedimento avaliando se o sistema pode ou não ter sobre sinal, ou seja, se o sistema permite um valor de pico na curva de resposta no tempo  $y(t)$ .

---

<sup>39</sup> Para informações sobre as equações envolvidas neste fluxograma, referenciar-se ao trabalho de FURST (2001).



### Passo 1

Em casos em que o sobre sinal é permitido, adota-se como modelo preliminar para o posicionador uma representação de 2ª ORDEM com razão de amortecimento  $\zeta = 0,7$  e, de posse do tempo de acomodação  $t_s$  desejado, calcula-se a frequência  $\omega_n$ . Se o sobre sinal não é permitido o projetista terá a opção de escolher um modelo de 1ª ORDEM ou 2ª ORDEM com razão de amortecimento  $\zeta = 1$ .

A curva da resposta ao degrau unitário para um sistema de 2ª ORDEM está representada na figura abaixo para valores de razão de amortecimento  $\zeta$  variando de 0 a 2. Percebe-se que para sistemas subamortecidos ( $\zeta < 1$ ) com  $\zeta$  variando entre 0,4 e 0,8, o tempo de acomodação  $t_s$  do sistema é maior do que para o sistema com amortecimento crítico ( $\zeta = 1$ ) ou sobreamortecido ( $\zeta > 1$ ).

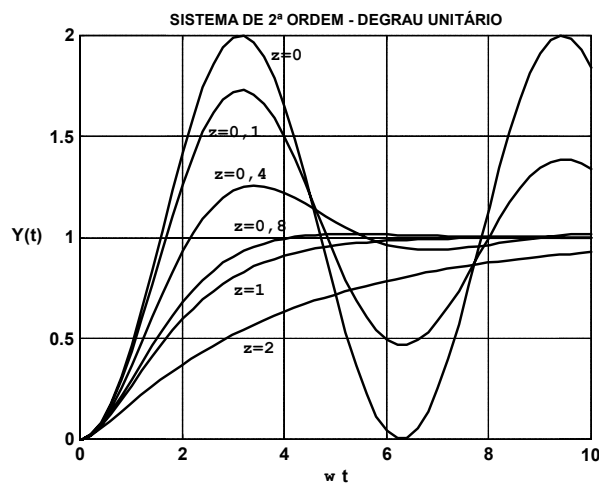


Figura – Resposta ao degrau unitário de um sistema de 2ª ORDEM

Para sistemas de 2ª ORDEM, tem-se que a constante de tempo é dada por  $\tau = \frac{1}{\zeta \cdot \omega_n}$ , ou seja, para uma dada frequência natural ( $\omega_n$ ) a constante de tempo é função do razão de amortecimento ( $\zeta$ ).

Utilizando-se o critério de faixa de tolerância de 2% para a resposta em regime permanente, tem-se que o tempo de acomodação  $t_s$  é dado por:

$$t_s = 4 \cdot \tau = \frac{4}{\zeta \cdot \omega_n}; \text{ para } 0 < \zeta < 0,9 \quad (01)$$

$$t_s = 6 \cdot \tau = \frac{6}{\zeta \cdot \omega_n}; \text{ para } \zeta = 1 \quad (02)$$

**APÊNDICE 8**

**JANELAS DOS PASSOS DO MÓDULO “FLUXOGRAMA DA SISTEMATIZAÇÃO DO DIMENSIONAMENTO ESTÁTICO E DINÂMICO”**

Este apêndice mostra as janelas do módulo “Fluxograma da Sistematização do Dimensionamento Estático e Dinâmico”.

No Passo 0, uma figura mostra as principais tarefas do projeto preliminar de um sistema de controle de posição.

A primeira etapa consiste no dimensionamento estático e dinâmico do circuito hidráulico escolhido. Para isso, os requisitos de projeto definidos na fase de projeto informacional são também considerados, incluindo forças a serem vencidas, tempo de resposta desejado e posição máxima do cilindro.

Assim sendo, utiliza-se as janelas dos passos subseqüentes para a execução de tal etapa que resulta em um conjunto de especificações, a qual deve ser seguida por uma próxima etapa de pré-seleção da válvula comercial através dos catálogos de fabricantes.

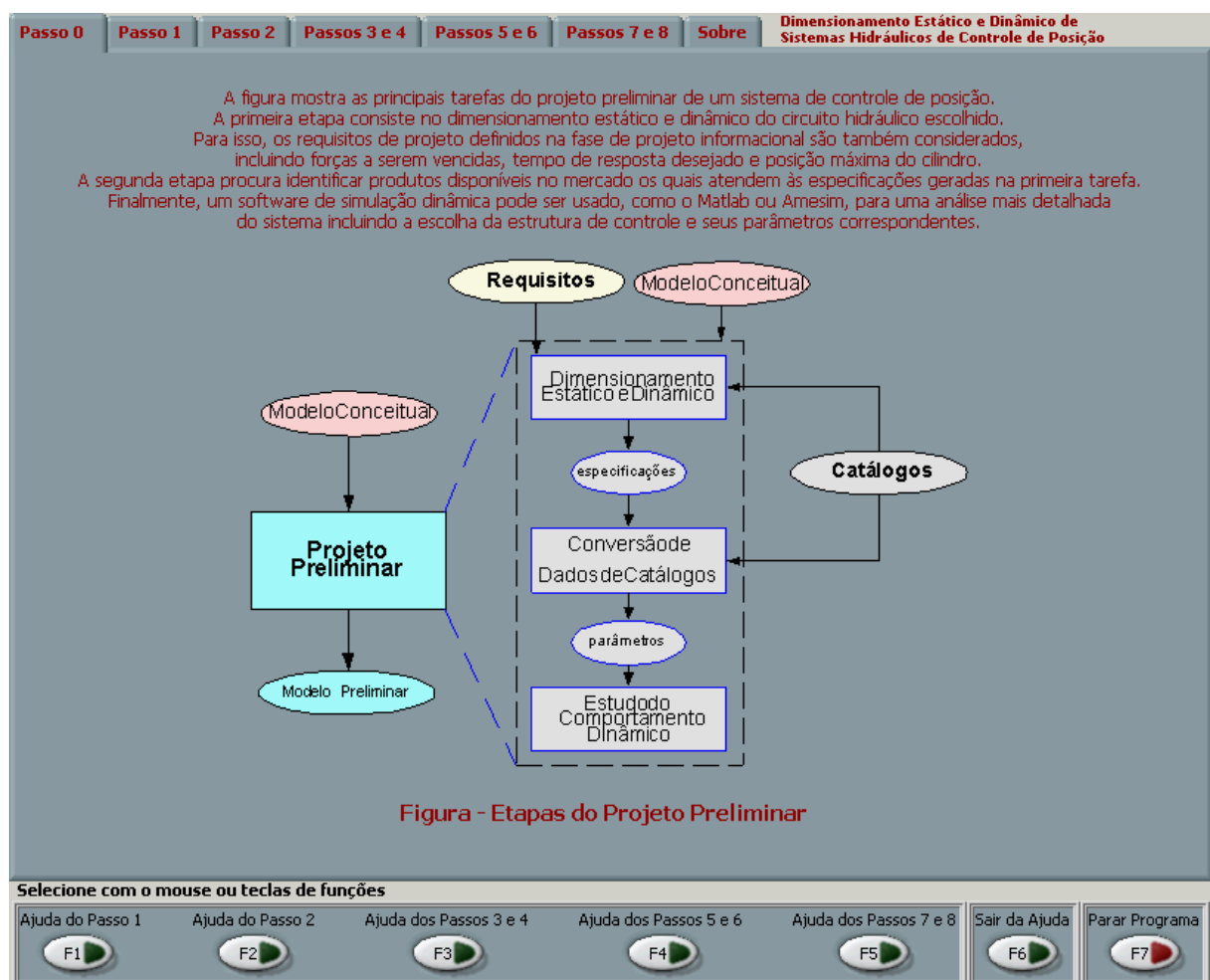


Figura – Janela do Passo 0

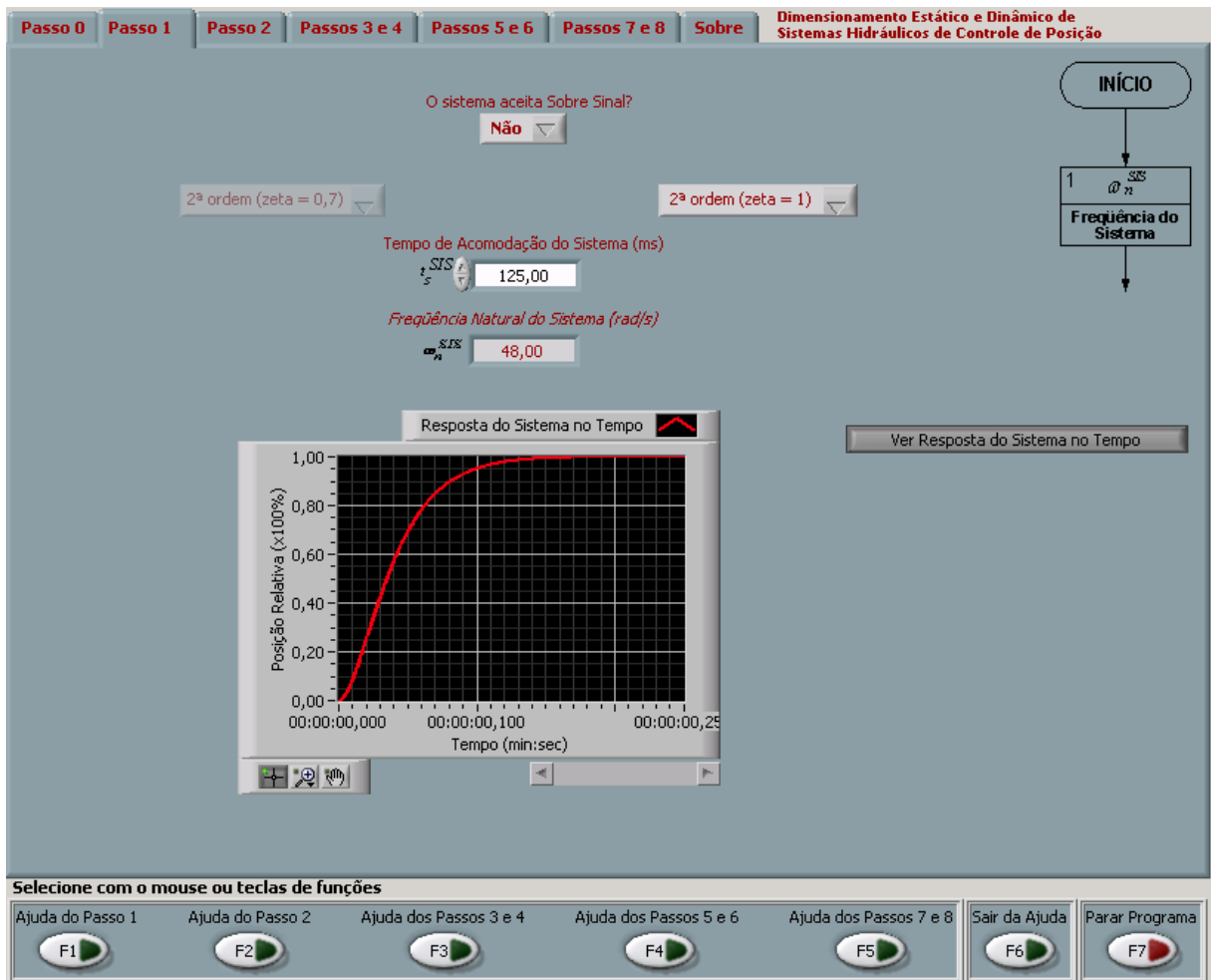


Figura – Janela do Passo 1

Passo 0 Passo 1 **Passo 2** Passos 3 e 4 Passos 5 e 6 Passos 7 e 8 Sobre Dimensionamento Estático e Dinâmico de Sistemas Hidráulicos de Controle de Posição

2 v. ; a.  
**Velocidade e Aceleração do cilindro**

Deslocamento Linear Máximo do Cilindro (mm)  
 $x_{max}^A$  12,0

Velocidade Máxima do Cilindro (m/s)  
 $v_{max}^A$  0,21

Aceleração Máxima do Cilindro (m/s<sup>2</sup>)  
 $a_{max}^A$  27,65

Tipo do conjunto Válvula + Cilindro que será utilizado  
**Cilindros Assimétricos de Simples Ação com Válvulas de 3 Vias**

Selecione com o mouse ou teclas de funções

Ajuda do Passo 1 (F1) Ajuda do Passo 2 (F2) Ajuda dos Passos 3 e 4 (F3) Ajuda dos Passos 5 e 6 (F4) Ajuda dos Passos 7 e 8 (F5) Sair da Ajuda (F6) Parar Programa (F7)

Figura – Janela do Passo 2



**Dimensionamento Estático e Dinâmico de Sistemas Hidráulicos de Controle de Posição**

**A Área do Cilindro é Conhecida?**

**Sim**

Área do Lado A do Cilindro: AA (cm<sup>2</sup>)

88,96

Constante Elástica da Mola + Força de Escoamento (N/m)

$K_x$  810,00E+3

Força Útil aplicada a carga (N)

$F_u$  0,0

Coefficiente de Amortecimento Viscoso do Atuador (N.s.m<sup>-1</sup>)

$B$  340,0

Massa Total (kg)

$M_t$  450,0

Pressão de Suprimento (Pa)

$p_s$  4,20E+6

3  $p_c$

Pressão de carga

↓

4  $A_a$

Área do Cilindro

↓

Pressão de Carga para Deslocamento Máximo (Pa)

$p_{c\ xmax}$  1,09E+6

Pressão de Carga para Velocidade Máxima (Pa)

$p_{c\ vmax}$  8,10E+3

Pressão de Carga para Aceleração Máxima (Pa)

$p_{c\ amax}$  1,40E+6

Pressão de Carga Máxima (Pa)

$p_{c\ max}$  1,40E+6

Pressão de Carga correspondente a Potência Máxima (Pa)

$p_{c\ pmax}$  1,40E+6

**Área Definida!**

**Selecione com o mouse ou teclas de funções**

Ajuda do Passo 1	Ajuda do Passo 2	Ajuda dos Passos 3 e 4	Ajuda dos Passos 5 e 6	Ajuda dos Passos 7 e 8	Sair da Ajuda	Parar Programa
F1	F2	F3	F4	F5	F6	F7

Figura – Janela dos Passos 3 e 4

Passo 0 Passo 1 Passo 2 Passos 3 e 4 Passos 5 e 6 Passos 7 e 8 Sobre Dimensionamento Estático e Dinâmico de Sistemas Hidráulicos de Controle de Posição

5  $q_{va}^A$   
Vazão Máxima do Cilindro

↓

6  $\omega_n^A$   
Frequência natural do cilindro

↓

Vazão Máxima do Cilindro ( $m^3/s$ )

$q_{va}^A$

Vazão Máxima do Cilindro (l/min)

$q_{va}^A$

Módulo de Compressibilidade (Pa)

$\beta_e$

Volume da Câmara A (l)

Frequência Natural do Cilindro (rad/s)

$\omega_n^A$

Tempo de Acomodação  $t_s^A$  do Cilindro (ms)

$t_s^A$

Selecione com o mouse ou teclas de funções

Ajuda do Passo 1	Ajuda do Passo 2	Ajuda dos Passos 3 e 4	Ajuda dos Passos 5 e 6	Ajuda dos Passos 7 e 8	Sair da Ajuda	Parar Programa
<input type="button" value="F1"/>	<input type="button" value="F2"/>	<input type="button" value="F3"/>	<input type="button" value="F4"/>	<input type="button" value="F5"/>	<input type="button" value="F6"/>	<input type="button" value="F7"/>

Figura – Janela dos Passos 5 e 6

Passo 0 Passo 1 Passo 2 Passos 3 e 4 Passos 5 e 6 Passos 7 e 8 Sobre Dimensionamento Estático e Dinâmico de Sistemas Hidráulicos de Controle de Posição

$\omega_n^A \geq 3 \omega_n^{SIS} ?$

Sim. Então a Frequência Natural da Válvula deve ser maior que a Frequência Natural do Sistema.

*Relação adotada entre as Frequências Naturais*

3

$\omega_n^{VP} \geq 3 \omega_n^{SIS}$

*Tempo de Acomodação da Válvula (ms)*

$t_s^{VP}$  41,67

7  $\omega_n^{VP}$   
Frequência natural da válvula

8  $t_s^{VP} = 4\tau$   
Tempo de acomodação da válvula

FIM

Selecione com o mouse ou teclas de funções

Ajuda do Passo 1 (F1) Ajuda do Passo 2 (F2) Ajuda dos Passos 3 e 4 (F3) Ajuda dos Passos 5 e 6 (F4) Ajuda dos Passos 7 e 8 (F5) Sair da Ajuda (F6) Parar Programa (F7)

Figura – Janela dos Passos 7 e 8