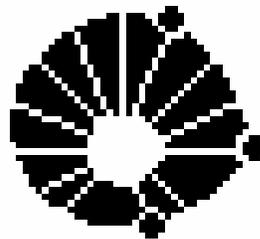


**Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica  
Departamento de Engenharia Biomédica**



## **DISSERTAÇÃO DE MESTRADO**

# **DESENVOLVIMENTO DE UM SOFTWARE PARA O TREINAMENTO DE TÉCNICOS DE MANUTENÇÃO DE EQUIPAMENTOS BIOMÉDICOS**

**Guilherme Araujo Wood**

**Orientador: Prof. Dr. Saide Jorge Calil**

**Comissão Julgadora:**

Saide Jorge Calil – FEEC/Unicamp – Presidente  
Antônio Fasolo Quevedo – FEEC/Unicamp  
Marcelo de Carvalho Reis – Cotuca/Unicamp

Dissertação submetida ao Departamento de Engenharia Biomédica como parte dos requisitos para a obtenção de grau de Mestre em Engenharia Elétrica na Universidade Estadual de Campinas

**28 de maio de 2003**

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

W85d Wood, Guilherme Araujo  
Desenvolvimento de um software para o treinamento de técnicos de manutenção de equipamentos biomédicos / Guilherme Araujo Wood.--Campinas, SP: [s.n.], 2003.

Orientador: Saide Jorge Calil  
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

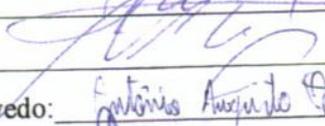
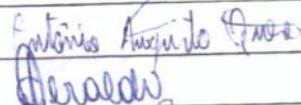
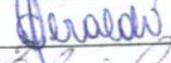
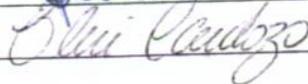
1. Ensino auxiliado por computador. 2. Eletrocardiografia. 3. Engenharia biomédica. 4. Hospitais – Departamento de engenharia. 5. Hospitais – Manutenção e reparos. I. Calil, Saide Jorge. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

## COMISSÃO JULGADORA – TESE DE MESTRADO

Candidato: GUILHERME ARAUJO WOOD

Data da Defesa: 28 de maio de 2003

Título da Tese: “Desenvolvimento de um Software para o Treinamento de Técnicos de Manutenção de Equipamentos Biomédicos”

Prof. Dr. Saide Jorge Calil (11.128-7):   
Prof. Dr. Marcelo de Carvalho Reis :   
Prof. Dr. Antonio Augusto Fasolo Queyedo:   
Secretária Giane Cristina Sales Geraldo:   
Coordenador PG Prof. Dr. Eleri Cardozo: 

## RESUMO

Desenvolvemos uma metodologia de projeto de aplicativos para o treinamento de técnicos de manutenção de equipamentos biomédicos. Como exemplo, criamos o aplicativo MONITOR, sobre monitores cardíacos e eletrocardiógrafos, que contém a simulação da operação, funcionamento e manutenção destes equipamentos. Escrevemos também o Arquivo de Ajuda “Manual do Monitor Cardíaco”, contendo explicações sobre Anatomia e Fisiologia do Coração, Operação, Funcionamento, Interferências, Segurança e Compatibilidade destes Equipamentos com o Desfibrilador e a Unidade Eletrocirúrgica.

## ABSTRACT

We developed a method for software design for biomedical equipment technician training. As an example, we created a software called MONITOR, which simulates the operation, functioning and maintaining of an ECG Monitor. We also wrote a Help File called “ECG Monitor Manual”, that contains explanations about Heart Anatomy and Physiology, Equipment Operation, Functioning, Safety, Interference and Compatibility with Defibrillators and Electrosurgery Units.

## OBSERVAÇÕES GERAIS

1) Muitas das palavras e expressões utilizadas em Informática são originárias do Inglês. Para algumas não existe uma tradução fiel, tais como *software*, *hardware*, *mouse*, *bit*, *byte*. Por isso, estas palavras no texto são mantidas sem tradução, destacadas em *itálico*. Algumas abreviaturas também são mantidas no original inglês (CAD, PC), pois são amplamente usadas em Engenharia e Informática.

2) Foram incluídas ao final duas tabelas contendo o significado das Abreviaturas utilizadas no texto e as Marcas Registradas citadas.

## ÍNDICE

DISSERTAÇÃO DE MESTRADO .....	i
<b>RESUMO .....</b>	<b>IV</b>
<b>ABSTRACT .....</b>	<b>IV</b>
<b>OBSERVAÇÕES GERAIS.....</b>	<b>IV</b>
<b>CAPÍTULO 1 - INTRODUÇÃO .....</b>	<b>1</b>
1.1.1 - <i>Vantagens do Treinamento Auxiliado por Computador (TAC)</i> .....	1
1.1.2 - <i>Métodos Utilizados em TAC</i> .....	2
1.1.3 - <i>Recursos Computacionais Utilizados em TAC</i> .....	3
1.2 - TAC DO USUÁRIO DE EQUIPAMENTOS BIOMÉDICOS.....	4
1.3 - TAC DO TÉCNICO DE MANUTENÇÃO DE EQUIPAMENTOS BIOMÉDICOS .....	6
1.4 - AS NECESSIDADES DE APRENDIZAGEM.....	8
<b>CAPÍTULO 2 - OBJETIVOS.....</b>	<b>10</b>
2.1 - OBJETIVO GERAL.....	10
2.2 - OBJETIVOS ESPECÍFICOS .....	10
<b>CAPÍTULO 3 - MATERIAIS E MÉTODOS.....</b>	<b>11</b>
3.1 - MODELO DE APRENDIZAGEM UTILIZADO PARA O DESENVOLVIMENTO DO APLICATIVO .....	11
3.1.1 - <i>A Documentação Técnica Através de Hipermídia</i> .....	11
3.1.2 - <i>A Simulação da Operação</i> .....	12
3.1.3 - <i>A Simulação do Funcionamento</i> .....	13
3.1.4 - <i>A Simulação de Defeitos</i> .....	14
3.1.5 - <i>A Simulação da Manutenção</i> .....	16
3.2 - RECURSOS COMPUTACIONAIS UTILIZADOS NO DESENVOLVIMENTO DO APLICATIVO.....	19
3.2.1 - <i>Escolha da Plataforma de Desenvolvimento</i> .....	19
3.2.2 - <i>Hardware e Sistemas Operacionais</i> .....	20
3.2.3 - <i>Escolha da Linguagem para o Desenvolvimento do Aplicativo</i> .....	20
3.3 - METODOLOGIAS DE TESTE.....	21
3.3.1 - <i>Teste de Assimilação</i> .....	21
3.3.2 - <i>Teste de Simulação</i> .....	22
3.3.3 - <i>Teste Comparativo</i> .....	22
<b>CAPÍTULO 4 - ESTUDO DE CASO.....</b>	<b>23</b>
4.1 - ESCOLHA DO EQUIPAMENTO PARA ESTUDO DE CASO .....	23
4.2 - A BASE DE DADOS “MONITOR.MDB” .....	23
4.2.1 - <i>A Tabela “SINAL”</i> .....	23
4.2.2 - <i>As Tabelas de Componentes</i> .....	24
4.2.3 - <i>As Tabelas de Defeitos</i> .....	25
4.2.4 - <i>A Tabela “MENSAGENS”</i> .....	26
4.3 - O ARQUIVO DE INICIALIZAÇÃO “MONITOR.INI”.....	26
4.4 - O APLICATIVO MONITOR.EXE .....	27
4.4.1 - <i>O Formulário Principal “MDIFMonitor”</i> .....	27
4.4.2 - <i>Os Formulários “Painel”</i> .....	27
4.4.3 - <i>Os Formulários “Diagramas de Blocos”</i> .....	29
4.4.4 - <i>Os Formulários “Circuitos Eletrônicos”</i> .....	30
4.4.5 - <i>O Formulário “Simulador de ECG”</i> .....	30

4.4.6 - O Formulário “Ordem de Serviço” .....	32
4.4.7 - O Controle Timer “TempVarre” .....	32
4.5 - O ARQUIVO DE AJUDA “MANUAL DO MONITOR” .....	33
4.5.1 - O Capítulo “Como Usar o Programa” .....	33
4.5.2 - O Capítulo “Introdução” .....	34
4.5.3 - O Capítulo “Operação” .....	35
4.5.4 - O Capítulo “Segurança Elétrica” .....	36
4.5.5 - O Capítulo “Funcionamento” .....	36
4.5.6 - O Capítulo “Interferências” .....	37
4.5.7 - O Capítulo “Manutenção” .....	38
4.5.8 - O Anexo “Outros Equipamentos” .....	39
4.6 - OS APLICATIVOS DE ILUSTRAÇÃO PARA A AJUDA.....	39
4.6.1 - O Aplicativo “PULSO” .....	39
4.6.2 - O Aplicativo “CONDUCAO” .....	40
4.6.3 - O Aplicativo “VETOR” .....	41
4.6.4 - O Aplicativo “ECGNORML” .....	42
4.7 - O RELATÓRIO TÉCNICO.TXT .....	42
<b>CAPÍTULO 5 - RESULTADOS.....</b>	<b>43</b>
5.1 - TESTE DE ASSIMILAÇÃO .....	43
5.2 - TESTE DE SIMULAÇÃO .....	44
<b>CAPÍTULO 6 - DISCUSSÃO .....</b>	<b>45</b>
<b>CAPÍTULO 7 - CONCLUSÕES .....</b>	<b>46</b>
7.1 - A SIMULAÇÃO DE FALHAS MÚLTIPLAS .....	46
7.2 - GENERALIZANDO O APLICATIVO.....	46
7.3 - COMPARANDO A APRENDIZAGEM COM A SALA DE AULAS.....	46
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>48</b>
<b>ABREVIATURAS .....</b>	<b>52</b>
<b>MARCAS REGISTRADAS.....</b>	<b>53</b>
<b>ANEXO A - PROGRAMAÇÃO EM VISUAL BASIC .....</b>	<b>54</b>
A.1 - O AMBIENTE OPERACIONAL MS-WINDOWS .....	54
A.2 - A LINGUAGEM VISUAL BASIC 3.0 .....	55
A.3 - OS OBJETOS DO MS-WINDOWS E DO VISUAL BASIC .....	55
A.4 - O AMBIENTE DE PROGRAMAÇÃO DO VISUAL BASIC 3.0 .....	56
A.5 - AS PROPRIEDADES DE UM OBJETO.....	58
A.6 - ARQUIVO DE PROJETO .....	60
A.7 - FORMULÁRIOS .....	61
A.8 - CONTROLES .....	61
A.9 - PROGRAMAÇÃO: EVENTOS, MÓDULOS E SEQÜÊNCIA DE EXECUÇÃO .....	62
A.10 - DISTRIBUINDO PROGRAMAS EM VISUAL BASIC 3.0.....	63
<b>ANEXO B - DESENVOLVIMENTO DE ARQUIVOS DE AJUDA.....</b>	<b>64</b>
B.1 - O GERENCIADOR DA AJUDA DO MS-WINDOWS .....	64
B.2 - REQUISITOS.....	65
B.3 - O PROCESSO DE CRIAÇÃO E COMPILAÇÃO .....	65
B.4 - CRIAÇÃO E EDIÇÃO DE FIGURAS .....	66
B.5 - CRIAÇÃO DE HIPERGRÁFICOS .....	66
B.6 - O ARQUIVO DE TÓPICOS .....	67
B.7 - MACROS .....	67

B.8 - ACRESCENTANDO FIGURAS.....	68
B.9 - O ARQUIVO DE PROJETO.....	69
B.10 - EXECUÇÃO DA COMPILAÇÃO .....	70
<b>ANEXO C - RELATÓRIOS DOS TESTES DE SIMULAÇÃO .....</b>	<b>71</b>

## **ÍNDICE DE TABELAS**

<b>TABELA 1 - AS PRINCIPAIS FALHAS DOS COMPONENTES.....</b>	<b>15</b>
<b>TABELA 2 - CONTEÚDO DO ARQUIVO DE INICIALIZAÇÃO MONITOR.INI.....</b>	<b>26</b>
<b>TABELA 3 - DESEMPENHO DOS ALUNOS NOS TESTES .....</b>	<b>44</b>
<b>TABELA 4 - ABREVIATURAS UTILIZADAS NA DISSERTAÇÃO OU NO SOFTWARE .....</b>	<b>52</b>
<b>TABELA 5 - MARCAS E PRODUTOS REGISTRADOS PELAS RESPECTIVAS EMPRESAS.....</b>	<b>53</b>
<b>TABELA 6 - EXEMPLOS DE PROPRIEDADES DE CMDOK .....</b>	<b>59</b>
<b>TABELA 7 - OS CONTROLES DO VISUAL BASIC USADOS NESTE TRABALHO.....</b>	<b>62</b>

## ÍNDICE DE FIGURAS

FIGURA 1 - DESMEMBRAMENTO DO EQUIPAMENTO EM BLOCOS FUNCIONAIS .....	14
FIGURA 2 - APLICATIVO PARA SELEÇÃO DE DEFEITOS .....	16
FIGURA 3 - O FORMULÁRIO "GERADOR DE SINAIS" .....	18
FIGURA 4 - O FORMULÁRIO "MULTÍMETRO" .....	18
FIGURA 5 - O FORMULÁRIO "OSCIOSCÓPIO" .....	19
FIGURA 6 - O FORMULÁRIO "PAINEL FRONTAL" DO MONITOR CARDÍACO .....	28
FIGURA 7 - O FORMULÁRIO "PAINEL TRASEIRO" DO MONITOR CARDÍACO.....	29
FIGURA 8 - EXEMPLO DE FORMULÁRIO DIAGRAMA DE BLOCOS.....	30
FIGURA 9 - EXEMPLO DE FORMULÁRIO CIRCUITO ELETRÔNICO .....	30
FIGURA 10 - O FORMULÁRIO "SIMULADOR DE ECG" .....	31
FIGURA 11 - O FORMULÁRIO ORDEM DE SERVIÇO .....	32
FIGURA 12 - EXEMPLO DE PÁGINA DO CAPÍTULO "INTRODUÇÃO" .....	34
FIGURA 13 - HIPERGRÁFICO DO CAPÍTULO SOBRE "OPERAÇÃO" .....	35
FIGURA 14 - HIPERGRÁFICO PRINCIPAL DO CAPÍTULO SOBRE "FUNCIONAMENTO" .....	36
FIGURA 15 - EXEMPLO DE HIPERGRÁFICO DE BLOCOS DO CAPÍTULO "FUNCIONAMENTO" .....	36
FIGURA 16 - "DESENHO EXPLODIDO" .....	38
FIGURA 17 - O APLICATIVO DE ANIMAÇÃO PULSO.EXE .....	39
FIGURA 18 - O APLICATIVO DE ANIMAÇÃO CONDUCAO.EXE .....	40
FIGURA 19 - O APLICATIVO VETOR.EXE .....	41
FIGURA 20 - O APLICATIVO EGNORML.EXE.....	42
FIGURA 21 - O GERENCIADOR DE PROGRAMAS .....	55
FIGURA 22 - O AMBIENTE DE PROGRAMAÇÃO DO VISUAL BASIC 3.0 .....	57
FIGURA 23 - O BOTÃO DE COMANDO CMDOK.....	59

## CAPÍTULO 1 - INTRODUÇÃO

O crescente desenvolvimento de tecnologia em saúde trouxe melhorias na qualidade de atendimento, mas gerou uma elevação nos custos da assistência, que devem ser otimizados com a operação correta e a manutenção adequada dos equipamentos. Isto exige um contínuo acompanhamento das inovações, por parte dos profissionais de saúde e de manutenção, através de reciclagens e treinamentos, dentro de um programa de educação continuada.

### 1.1.1 - Vantagens do Treinamento Auxiliado por Computador (TAC)

O intensivo desenvolvimento da informática permite hoje que a educação continuada de profissionais seja auxiliada pelo computador, com as seguintes vantagens em relação à educação convencional (Hoffer, 1990):

Interatividade: permitir ao usuário a simulação de situações reais, aprendendo a desenvolver habilidades na solução de problemas reais;

Realimentação Imediata: permitir que o usuário saiba, durante o treinamento, o resultado de sua ação;

Individualidade: permitir ao usuário (tanto ao aprendiz, quanto ao instrutor) selecionar os tópicos que são mais necessários ou interessantes para a aprendizagem;

Avaliação Objetiva: permitir ao instrutor avaliar os alunos através de indicadores de desempenho quantitativos, automaticamente calculados pelo computador.

### 1.1.2 - Métodos Utilizados em TAC

Com o desenvolvimento da informática, diversos métodos de treinamento foram criados, de acordo com os recursos computacionais disponíveis em cada época. A classificação apresentada abaixo não é absoluta, pois muitos programas combinam estas formas de Treinamento (Hoffer, 1990):

Lição e Prática: forma mais antiga de Treinamento, realizada através de apresentação de textos e avaliação através de questionários, após o ensino.

Tutorial: forma de Treinamento clássica, com testes que avaliam o usuário antes e depois da apresentação, a qual pode ser desenvolvida de acordo com as necessidades de aprendizagem do usuário, detectada automaticamente através dos testes.

Simulação: forma moderna de Treinamento, através da apresentação de objetos e eventos e avaliação por ações, que representam situações reais.

Consultor: forma inteligente de Treinamento, em que o programa apresenta dicas para o usuário, e utiliza um sistema especialista para simular a tomada de decisões de um profissional experiente.

Sistemas de Autoria: diversos programas de TAC são desenvolvidos com alguns poucos comandos simples, para possibilitar a criação de aulas por programadores leigos (em linguagens de programação) com grande flexibilidade. Os roteiros de Treinamento podem ser escritos em editores de texto, em linguagem simples, com comandos fáceis de aprender, e são analisados automaticamente pelo programa.

A metodologia desenvolvida neste trabalho combina Simulação e Consultor. Conforme discutido na Seção 5.3, seria interessante complementar com um Sistema de Autoria.

### 1.1.3 - Recursos Computacionais Utilizados em TAC

Os primeiros microcomputadores, na década de 1970, dividiam a tela do monitor em uma matriz de caracteres, o que restringia a construção de figuras e gráficos à combinação dos poucos símbolos da Tabela ASCII. Com o desenvolvimento dos padrões gráficos CGA, EGA e VGA, a tela foi dividida em pontos (*pixels*), e se permitiu o amplo uso de figuras, gráficos e animações em microcomputadores.

Em 1982, a Apple lançou o Macintosh, um computador com Sistema Operacional Gráfico, que utiliza Programação Orientada a Objetos e Eventos e o *mouse*, os quais tornam o uso do computador mais fácil e agradável. Por exemplo, quando se aciona (Evento) um Botão de Comando (Objeto) o computador executa uma determinada rotina, definida pelo programador. Seguiram-se o MS-Windows, para computadores compatíveis com o padrão IBM PC, o OpenWindows e o XWindows, para estações de trabalho com Sistema Operacional Unix.

Pliske & Psojka (1986) propõem o uso de Sistemas Baseados em Regras, Programação Orientada a Objetos e Interfaces Gráficas para o desenvolvimento de aplicativos TAC mais poderosos. Sistemas Baseados em Regras são programas que utilizam Bases de Conhecimento (estruturas de dados nas quais determinado tipo de conhecimento humano é codificado na forma de regras) para simular a tomada de decisões de um especialista. Podem ser utilizados na criação de programas do tipo Consultor.

Wiklund (1993) e Neri (1996) propõem o uso de Hipertexto e de Ilustrações para a criação de Documentação Eletrônica (Manuais de Operação e Manutenção *On-Line*), uma importante ferramenta na educação continuada de profissionais de saúde. Hipertexto é um tipo de arquivo de texto, no qual as páginas possuem referências, que podem ser utilizadas

para acesso direto a outra página, quando o usuário encontra algum assunto no texto que deseje aprofundar. A combinação de Hipertexto e outros recursos computacionais, como imagens, sons, animações, é chamada de Hiperídia.

## **1.2 - TAC do Usuário de Equipamentos Biomédicos**

Schwid (1987) desenvolveu um programa de simulação gráfica para treinamento de médicos sobre anestesia geral, na Universidade da Califórnia, em San Diego, utilizando computador VAX e linguagem FORTRAN. Além dos controles do aparelho de anestesia e de infusão de drogas, o sistema apresenta um monitor de parâmetros fisiológicos (sinal eletrocardiográfico, pressão arterial e pressão venosa central) e o paciente apresenta respostas, como alterações da cor da pele, abertura das pupilas e dos olhos.

Saliterman (1990) desenvolveu um sistema de autoria para a simulação clínica de pacientes em terapia intensiva, em plataforma Macintosh, no Hospital Metodista, em Minneapolis. Este sistema inclui um *phantom*<sup>1</sup> para treinamento sobre o uso de catéteres e a simulação de um monitor de parâmetros fisiológicos (sinal eletrocardiográfico, pressão arterial, pressão atrial direita ou ventricular direita via catéter).

---

<sup>1</sup> Boneco em tamanho natural e formato anatômico, para simulações.

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

Hooper & Cols. (1993) desenvolveu o EquipTeach, um aplicativo tutorial, específico para o treinamento dos profissionais de saúde na operação de equipamentos biomédicos. Foi desenvolvido no Hospital John Dempsey da Universidade de Connecticut, em plataforma Macintosh. Possui um sistema de autoria que permite a confecção de roteiros pelos instrutores, permitindo também que o sistema seja utilizado no treinamento sobre qualquer equipamento. Os tutoriais de EquipTeach consistem de cinco etapas:

- 1) Teste pré-Treinamento
- 2) Teoria de Funcionamento
- 3) Prática de Operação
- 4) Alarmes e Erros de Operação Comuns
- 5) Teste pós-Treinamento

O desempenho do aluno pode ser avaliado através da comparação dos resultados dos testes, que são idênticos nas etapas 1 e 5, dando uma medida precisa da retenção de conteúdo. Na etapa 2, o aluno é instruído sobre o objetivo clínico do equipamento e como ele funciona. Na etapa 3, ele aprende como realizar os diversos procedimentos de operação do equipamento, usando uma simulação gráfica de seus comandos. Finalmente, na etapa 4 é ensinado com identificar problemas com o equipamento, quais são suas possíveis condições de alarme, e como corrigir estes problemas.

Moniz & cols. (1995) desenvolveram no VA Medical Center, em Boston, um aplicativo TAC em ambiente MS-Windows, para usuários de desfibriladores. Este programa possui testes pré- e pós-treinamento, teoria de operação e explicações sobre todos os controles do equipamento.

Jiang & Sittig (1995) desenvolveram um sistema tutorial e de simulação, em plataforma OS/2, para o treinamento de profissionais de saúde sobre a operação de computadores utilizados no Sistema de Informações Hospitalares.

Moreira (1996) desenvolveu o SIMVEP - Simulador da Ventilação Pulmonar, um aplicativo para plataforma MS-Windows para o treinamento de profissionais de saúde sobre terapia ventilatória, incluindo os controles e modos de ventilação dos modernos ventiladores eletrônicos.

### **1.3 - TAC do Técnico de Manutenção de Equipamentos Biomédicos**

Um programa de TAC, utilizando Inteligência Artificial para *Troubleshooting* de equipamentos eletrônicos, foi desenvolvido em 1975 por Brown & cols. (citado em Wenger, 1987). SOPHIE (Sophisticated Education) utilizava perguntas e respostas (em linguagem natural) sobre a polarização e ganho dos componentes para treinar técnicos de manutenção eletrônica.

Pipitone (1986) desenvolveu para a Marinha dos Estados Unidos um sistema especialista para *Troubleshooting* de equipamentos eletrônicos, descrevendo seus módulos (blocos e componentes) pela sua conectividade e suas funções, calculando as probabilidades de falhas de cada módulo, e recomendando ao técnico o próximo teste a ser realizado.

Richardson & Jackson (1986) desenvolveram para a Força Aérea dos Estados Unidos um Consultor Inteligente de Manutenção, “uma ferramenta computacional que habilita o técnico novato a desempenhar tarefas como se fosse experiente, orienta os técnicos que precisam de maior desenvolvimento em suas habilidades técnicas e trabalha em cooperação com os técnicos experientes para incorporar heurísticas de diagnóstico...”,

combinando tutoriais, simulações e sistemas especialistas e de autoria. Como uma das principais fontes de conhecimento é o projeto do equipamento, uma interface com aplicativo CAD (Projeto Auxiliado por Computador) foi incorporada ao sistema. Trabalhando com o diagrama de blocos e esquemas dos circuitos, os engenheiros desenvolveram um modelo de conectividade dos módulos, e uma hierarquia de falhas (em forma de árvore), de modo a determinar onde realizar medidas, caminhando da raiz para as folhas, isolando a falha em nível de componente.

Mataban (1994) desenvolveu um sistema especialista para manutenção de bombas de infusão para a Força Aérea dos Estados Unidos, em ambiente MS-Windows, contendo a representação da estrutura do equipamento em nível de placas de circuito impresso, a exceção de componentes como cabos, conectores e baterias. A autora alega que o reparo destes equipamentos é feito através da troca das placas, devido ao alto custo da mão-de-obra e do uso de tecnologia de montagem de superfície, não justificando a representação dos equipamentos em todos os seus componentes. A base de conhecimentos desenvolvida continha uma lista padronizada de sintomas de defeito, e para cada sintoma uma lista dos componentes com um “fator de suspeita” de causar o problema, fator este entre 1 (pouco provável) e 5 (muito provável). Esta base fornecia os dados para um Consultor, que recomendava ao técnico a ação de diagnóstico ou reparo. O sistema ainda possuía diversas informações obtidas dos manuais de operação e manutenção, como regras de *troubleshooting* e figuras das peças de reposição. Ao final do Treinamento, emitia um relatório das atividades realizadas.

Nash (1995) propõe um método para isolar o defeito de um equipamento biomédico em nível de falha de componente, através dos seguintes passos: análise do sintoma,

localização da placa defeituosa, do bloco defeituoso e do componente defeituoso. Xu (1997) acrescentou 2 passos: substituir o componente defeituoso e testar novamente o funcionamento de todo o equipamento. Em seus trabalhos apresentaram exemplos do uso destas metodologias com monitores e unidades eletrocirúrgicas.

Fouladinejad & Roberts (1997) desenvolveram arquivos em multimídia na linguagem Toolbook, para computadores compatíveis com IBM-PC, com ambiente gráfico MS-Windows, para o treinamento de técnicos sobre a manutenção de ventiladores pulmonares eletrônicos, através de tutoriais e simulações.

#### **1.4 - As Necessidades de Aprendizagem**

O curso de Técnico de Equipamentos Médico-Hospitalares é muito recente no Brasil (as primeiras turmas foram formadas em 1993). A maioria dos técnicos formados em eletrônica que atuam na área hospitalar não tiveram qualquer contato com este tipo de equipamento em sua formação escolar. Os conceitos biomédicos e tecnológicos relacionados a estes equipamentos são aprendidos através da prática, de treinamentos, da leitura de manuais de operação e manutenção e do contato com colegas mais experientes.

Realizamos entrevistas com engenheiros e técnicos mais experientes do Centro de Engenharia Biomédica (CEB-Unicamp), para avaliar quais seriam as etapas de um processo ideal de aprendizagem:

- finalidade clínica (em termos simples, para que serve o equipamento?),
- operação (quais são os resultados de cada comando dado pelo usuário?),
- funcionamento (a princípio, através de diagrama de blocos, e com a experiência, através da análise do esquema eletrônico, ou mecânico, de cada bloco) e
- ações de manutenção (diagnóstico do defeito; ações corretivas; rotinas de teste e de manutenção preventiva).

Mesmo os técnicos eletrônicos que se especializaram através do curso de Equipamentos Médico-Hospitalares não possuem todos estes conceitos, a respeito de todos os equipamentos com os quais trabalham, pois não é possível abordar cada detalhe de operação, funcionamento e manutenção de cada modelo de equipamento. Por isso, também são candidatos para programas de educação continuada em seu trabalho.

Portanto, este trabalho visa o treinamento de qualquer técnico que trabalhe na área, não importando sua formação escolar e sua experiência. Para isso, todos os passos acima citados precisam ser implementados.

## **CAPÍTULO 2 - OBJETIVOS**

### **2.1 - Objetivo Geral**

Desenvolver uma metodologia de Treinamento Auxiliado por Computador para técnicos de manutenção de equipamentos médico-hospitalares, estabelecendo um processo de aprendizagem versátil, estruturado de acordo com as diferentes capacidades do aprendiz.

### **2.2 - Objetivos Específicos**

- Desenvolver um *software* baseado na metodologia TAC, utilizando como exemplo o Monitor Cardíaco;
- Avaliar os resultados do uso desta metodologia, através de testes com Técnicos de Equipamentos Biomédicos e alunos de Cursos Técnicos.

## **CAPÍTULO 3 - MATERIAIS E MÉTODOS**

Neste capítulo apresentamos a proposta genérica de metodologia de TAC para técnicos de manutenção de equipamentos biomédicos. Discorremos e justificamos também sobre as ferramentas computacionais escolhidas para o desenvolvimento do trabalho e a metodologia de testes.

No capítulo 4 apresentamos como Resultado do trabalho o aplicativo desenvolvido especificamente para o Monitor Cardíaco, com base na metodologia descrita a seguir.

### **3.1 - Modelo de Aprendizagem Utilizado para o Desenvolvimento do Aplicativo**

Para que o processo de aprendizagem satisfaça as sugestões apresentadas na seção 1.4, propomos que a representação do equipamento e de sua manutenção seja feita através de simulações, conforme serão descritas nas seções 3.1.2 a 3.1.5. Para a apresentação de documentação técnica propomos o uso de Hipermídia, a seguir.

#### **3.1.1 - A Documentação Técnica Através de Hipermídia**

Os Manuais de Operação e Manutenção contêm inúmeras informações técnicas relacionadas entre si, que podem ser representadas através de Hipertextos. Além disso, a representação gráfica de informações (como figuras, tabelas e animações) é importante para a aprendizagem de técnicos que atuam, ou atuarão, em manutenção de equipamentos em geral. Por isso, utilizamos Hipermídia para a representação da Documentação Técnica.

Para o TAC dos técnicos de manutenção de equipamentos biomédicos propomos a confecção de um arquivo de Hipermídia, contendo capítulos sobre os seguintes assuntos referentes aos equipamentos:

- 1) Finalidade Clínica e Princípio de Funcionamento;
- 2) Operação;
- 3) Funcionamento;
- 4) Segurança;
- 5) Interferências;
- 6) Procedimentos e Rotinas de Manutenção Preventiva e Corretiva.

Deste modo, o aprendiz tem à sua disposição informações de diferentes níveis de complexidade, de acordo com a experiência que ele já possui sobre o equipamento.

### **3.1.2 - A Simulação da Operação**

A operação consiste em um conjunto de ações executadas pelo usuário para controlar o funcionamento do equipamento. Mediante a observação de mostradores e outros dispositivos sinalizadores do equipamento, além de parâmetros externos (do paciente, por exemplo), o usuário pode acionar chaves, ajustar potenciômetros, conectar cabos, e outros. No entanto, só deve ser considerada ação de operação do aparelho aquela que não requer o uso de ferramentas. As ações que requerem o uso de ferramentas devem ser consideradas ações de calibração e de manutenção, que devem ser realizadas somente por pessoal habilitado para tal.

Um formulário gráfico pode representar o Painel de Controle de um equipamento biomédico, contendo objetos, tais como Botões de Comando, Etiquetas e Caixas de Figuras,

para simular os controles e mostradores do equipamento real. Quando o usuário aciona um Botão de Comando, o programa pode responder com uma alteração em uma Etiqueta ou Caixa de Figura, para simular a Operação do equipamento.

Como este trabalho propõe o desenvolvimento de um *software* para técnicos de manutenção, não é necessário simular as reações de um paciente à operação do equipamento. Entretanto, é necessário simular o equipamento de teste utilizado pelo técnico em suas ações de manutenção, como, por exemplo, o Simulador de ECG.

A simulação correta da operação é importante também para a simulação de defeitos, pois os controles do equipamento defeituoso, em muitos casos, não respondem ao acionamento pelo operador, ou o fazem de maneira inadequada.

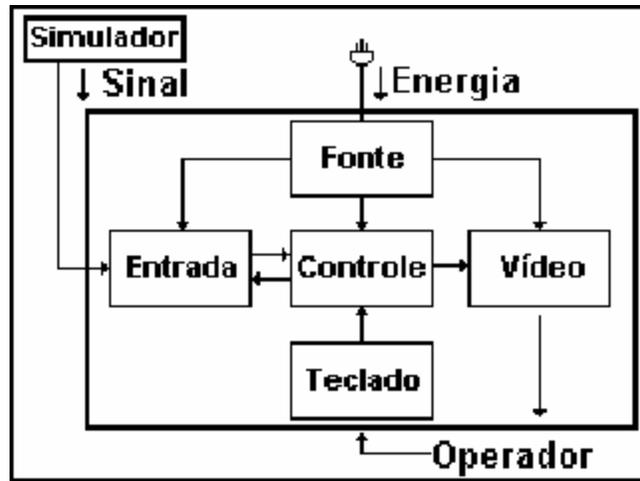
### **3.1.3 - A Simulação do Funcionamento**

Um equipamento pode ser representado como um grande sistema, com diversas entradas e saídas. Entretanto, um modelo global pode ser inviável, considerando a complexidade de uma função de transferência para todo o aparelho.

O equipamento deve ser desmembrado em diversos blocos (Figura 1), cujo modelo seja viável. Como a saída de um determinado bloco é entrada para outros (este é o conceito de conectividade), a simulação deve estar baseada em como ocorre a variação do sinal destas saídas em função do tempo.

Um bloco pode ser desmembrado em blocos menores, até o nível de circuitos. Define-se então a função de transferência destes circuitos, a qual determina o valor da saída do bloco em função de suas entradas.

Todos os nós entre componentes dos circuitos eletrônicos, inclusive as entradas e saídas dos blocos, podem ser representados por um vetor de variáveis reais (conhecido em



informática como *array*), que contêm o valor da tensão nestes pontos, em uma determinada amostra de tempo.

**Figura 1 - Desmembramento do Equipamento em Blocos Funcionais**

Os diversos blocos e seus componentes podem ser apresentados como formulários e objetos, respectivamente, cujas propriedades são alteradas de acordo com as entradas de cada bloco.

Como o computador é um dispositivo digital, o processamento dos parâmetros através do equipamento deve ser simulado através dos preceitos do Processamento Digital de Sinais, e o modelo deve ser realizado através de Sistemas de Variáveis Discretas.

### **3.1.4 - A Simulação de Defeitos**

As falhas de componentes eletrônicos (Loveday, 1982) causam a perda de desempenho do circuito e alterações das tensões de polarização dos componentes. Os fatores que causam as falhas podem ser:

- operacionais (tensão, corrente, potência, chaveamento), inerentes às características de funcionamento dos componentes no circuito, e que exigem um projeto adequado quanto às especificações de cada componente; ou
- ambientais (umidade, temperatura, pressão, vibrações e choques mecânicos, poeira e corrosivos no ar, surtos de sobretensão, operação inadequada do equipamento).

As falhas típicas dos componentes eletrônicos são apresentadas na Tabela 1.

<b>Componente</b>	<b>Falhas Típicas</b>
Resistor	Valor acima [da tolerância], circuito aberto.
Resistor Variável	Circuito aberto, contato intermitente.
Capacitor	Curto-circuito, circuito aberto.
Indutor	Circuito aberto, curto-circuito entre espiras, entre bobina e massa.
Semicondutores	Circuito aberto ou curto-circuito em qualquer das junções P-N.

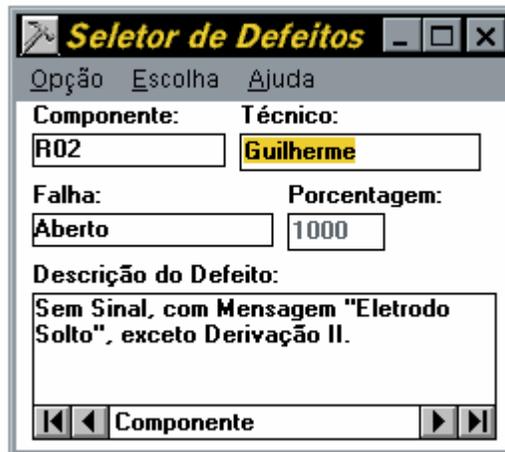
**Tabela 1 - As principais falhas dos Componentes**

As falhas de cada componente podem ser analisadas, através da verificação do tipo de defeito provocado no equipamento, pois diferentes falhas de componentes de um mesmo circuito podem provocar o mesmo defeito. Duas Tabelas, em uma Base de Dados, podem ser implementadas e interrelacionadas, uma contendo as descrições dos defeitos e a outra com os componentes, suas falhas e os defeitos que elas provocam.

De acordo com o tipo de falha, determinados parâmetros do componente têm seu valor alterado (por exemplo: quando a falha é “Curto Circuito”, a resistência do componente é reduzida a um valor muito pequeno, da ordem de miliohms), o que pode modificar a função de transferência do bloco a que pertence o componente, alterando o funcionamento do equipamento.

O **instrutor** pode escolher o defeito a ser simulado através de um aplicativo Seletor de Defeitos (Figura 2), que permite o acesso a estas Tabelas. A escolha pode ser:

- a) automática: aleatoriamente, usando pesos para a probabilidade de cada tipo de falha, ou
- b) manual, através de um Controle de Acesso a Dados, onde o Instrutor escolhe o defeito a ser simulado.



**Figura 2 - Aplicativo para Seleção de Defeitos**

### 3.1.5 - A Simulação da Manutenção

O processo de manutenção corretiva pode ser dividido em duas fases: o diagnóstico do defeito, através de ações de avaliação com equipamentos de teste, e a correção do mesmo, através de ações de correção, como ajuste ou substituição do componente defeituoso, por exemplo. As rotinas de manutenção preventiva incluem estes tipos de ações, mas são realizadas de acordo com um programa de inspeção e calibração periódicas do equipamento.

As ferramentas necessárias para o diagnóstico de defeitos de equipamentos eletrônicos são: documentação (manuais de operação e de serviço), equipamentos de teste

(multímetro, osciloscópio) e a metodologia (localizar circuito defeituoso, localizar componente defeituoso). Estas ferramentas podem ser simuladas da seguinte maneira:

- documentação técnica, através do Arquivo de Ajuda;
- equipamentos de teste, como formulários que simulam os painéis de equipamentos reais;
- metodologia, através de ações que o aprendiz deve realizar nos diversos formulários do aplicativo para conseguir resolver o problema do equipamento simulado.

A metodologia também é composta por regras heurísticas, para concentrar as ações de manutenção em determinado bloco ou componente, dependendo dos “sintomas” apresentados pelo equipamento - o chamado *Troubleshooting*, ou rotina de diagnóstico de falhas.

Os Formulários para Ações de Manutenção contêm os equipamentos de teste que o técnico utiliza no *troubleshooting* do equipamento: o Gerador de Sinais, o Multímetro, o Osciloscópio, e outros equipamentos (específicos para cada caso). Também está disponível um Botão de Comando que permite ao Usuário substituir um Componente.

O Formulário “Gerador de Sinais” (Figura 3) permite avaliar a resposta do equipamento a diferentes tipos de ondas: senoidais, quadradas, triangulares e aleatórias (ruído branco); em diferentes amplitudes e frequências. Testes como resposta em frequência, linearidade, e outros podem ser realizados, como explicado no Arquivo de Ajuda do programa.

Sinal	mVpp	Hz
Quadrada	0	0,1
Senoidal	0,1	0,2
Ruído	0,2	0,5
	0,5	1
	1	2
	2	5
	5	10
	10	20

Observação:  
Derivação II  
somente

dc -5 -4 -3 -2 -1 0 1 2 3 4 5mV

% 0 25 50 75 100%  
duty cycle

**Figura 3 - O Formulário "Gerador de Sinais"**

O Formulário “Multímetro” (Figura 4) permite medir tensão, intensidade de corrente, resistência, capacitância e o fator de amplificação  $\beta$  do transistor. Pode-se avaliar também o estado de funcionamento de um diodo. Tensão e corrente podem ser medidos com o equipamento ligado. Os demais parâmetros exigem do usuário que se desconecte o cabo de alimentação da tomada de energia elétrica.

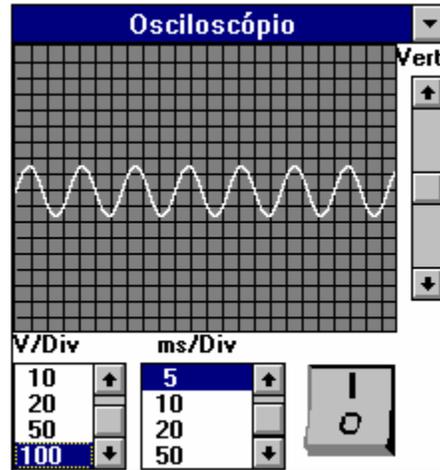
Multímetro

10mA Vcc Vca D

kOhm 10nF hFE

**Figura 4 - O Formulário "Multímetro"**

No Formulário “Osciloscópio” (Figura 5) é possível ajustar a escala de amplitude, em V/div (volts por divisão), cujo ajuste inicial (*default*) é de 100V/div; e a escala de tempo, em ms/div (milissegundos por divisão). Como em algumas escalas de tempo pode haver estouro (*overflow*), o programa gera mensagens de erro para escolhas indevidas.



**Figura 5 - O Formulário "Osciloscópio"**

As ações corretivas (ajustes, substituição de componentes, etc.) podem estar disponíveis para o aprendiz como opções de um objeto (menu, caixa de opções, botão de comando, ou outros). Após a escolha, o programa passa a simular o funcionamento do equipamento com o resultado desta ação, que pode ser correta ou não.

### **3.2 - Recursos Computacionais Utilizados no Desenvolvimento do Aplicativo**

#### **3.2.1 - Escolha da Plataforma de Desenvolvimento**

A Faculdade de Engenharia Elétrica da UNICAMP dispõe de várias plataformas para desenvolvimento de *software*: DOS, MS-Windows, OS/2, Macintosh, Unix, OpenWindows, XWindows. A plataforma MS-Windows foi escolhida para o desenvolvimento do aplicativo, para que seja possível obter a maior divulgação possível no Brasil, pois é líder brasileiro no mercado de Sistemas e Ambientes Operacionais. A Versão 3.1 desta plataforma foi utilizada pois o MS-Windows 95 ainda não havia sido lançado na época do início do desenvolvimento do programa.

### **3.2.2 - Hardware e Sistemas Operacionais**

O Aplicativo foi desenvolvido utilizando-se um microcomputador compatível com o padrão IBM-PC, com as seguintes características:

- Microprocessador Cx486DX2, com relógio de 66 MHz;
- Memória de Acesso Aleatório com capacidade de 8 MB;
- Disco Rígido com capacidade de 340 MB;
- Acionadores de Discos Flexíveis de 5,25 pol (até 1,2 MB) e 3,5 pol (até 1,44 MB);
- Placa de Vídeo de 1 MB, padrão SuperVGA.

A Plataforma utilizada no desenvolvimento do aplicativo é composta pelos sistemas operacionais MS-DOS, versão 6.22, e MS-Windows 3.11 para Workgroups, ambos em português.

Várias figuras utilizadas no Arquivo de Ajuda foram digitalizadas utilizando-se o *scanner* Microtek ScanMaker II.

### **3.2.3 - Escolha da Linguagem para o Desenvolvimento do Aplicativo**

Dentre as linguagens de programação para o ambiente MS-Windows, foram analisadas três possibilidades: a linguagem C++, o ambiente Access 2.0, de gerenciamento de bases de dados, e o ambiente Visual BASIC 3.0, de criação de aplicativos gráficos.

A linguagem C tem como vantagens a adaptação fácil para outras plataformas, o acesso a vários recursos do sistema operacional e do *hardware* e a execução muito mais veloz dos programas, mas exige um tempo muito maior para ser aprendida e o desenvolvimento de objetos e eventos requer maior habilidade do programador.

O ambiente Access, que dispõe de uma linguagem chamada Access BASIC, é muito fácil de se aprender, permite o uso de diversos recursos gráficos e facilita as operações com bases de dados, mas utiliza uma quantidade de recursos computacionais muito grande, o que exige uma configuração de *hardware* mais sofisticada. Além disso, alguns recursos de programação das outras linguagens não são disponíveis no Access, como animações, por exemplo. Ainda assim, este ambiente foi útil no desenvolvimento do programa, por facilitar a edição das tabelas das bases de dados.

O ambiente Visual BASIC foi escolhido por:

- a) necessitar de uma configuração de *hardware* menor, o que o torna mais acessível,
- b) permitir acesso fácil a recursos do sistema,
- c) trabalhar com bases de dados,
- d) possuir diversos objetos gráficos prontos (padronizados do ambiente MS-Windows), o que torna extremamente fácil o desenvolvimento de aplicativos,
- e) permitir a criação do aplicativo executável de fácil distribuição e instalação em outros microcomputadores.

### **3.3 - Metodologias de Teste**

Para elaboração dos testes da metodologia proposta, apresentamos três tipos de teste para avaliar o desempenho do *software* desenvolvido, do ponto de vista educacional.

#### **3.3.1 - Teste de Assimilação**

O aprendiz é submetido a um teste antes do treinamento, realiza a simulação, ou estuda o Arquivo de Ajuda, e é novamente submetido ao mesmo teste. Os resultados dos testes, antes e depois do Treinamento são então comparados. Este tipo de teste é

interessante para aprendizes com pouca ou nenhuma experiência com o assunto tratado.

Além disso, é importante delimitar o escopo do teste, para obter um resultado preciso.

Quanto mais amplo o assunto mais impreciso o resultado deste teste.

### **3.3.2 - Teste de Simulação**

Um problema é proposto ao aprendiz (um defeito, por exemplo), que tenta resolvê-lo através da simulação. Suas ações são então julgadas, avaliando-se sua rapidez e objetividade na solução do problema. Este teste é interessante para técnicos mais experientes, cujas necessidades de assimilação são menores. Pode também ser usado para se otimizar o Aplicativo, através das críticas que o técnico eventualmente fizer.

### **3.3.3 - Teste Comparativo**

Dois grupos de alunos podem aprender os mesmos conceitos através de métodos diferentes. Por exemplo, um grupo pode aprender sobre a Finalidade Clínica do Monitor Cardíaco através do Aplicativo, usando microcomputadores, e outro grupo pode assistir aula sobre o mesmo assunto, com transparências e quadro negro. Após o treinamento ambos os grupos são submetidos aos mesmos testes. Assim, pode se avaliar as vantagens e desvantagens de um método em relação ao outro. Este tipo de teste exige um longo prazo para sua correta avaliação. Ele não foi implementado no presente trabalho.

## **CAPÍTULO 4 - ESTUDO DE CASO**

### **4.1 - Escolha do Equipamento para Estudo de Caso**

Selecionamos como exemplo o Monitor Cardíaco, por ser equipamento de média complexidade e de amplo uso em hospitais, não importando seu tamanho ou especialidade.

### **4.2 - A Base de Dados “MONITOR.MDB”**

A Base de Dados MONITOR.MDB foi desenvolvida no ambiente Access para armazenar os dados de simulação para o Aplicativo MONITOR, o componente principal do *software*, descrito na seção 4.4. Esta base de dados é constituída das tabelas descritas a seguir.

#### **4.2.1 - A Tabela “SINAL”**

Esta tabela contém uma lista com 600 pontos (amostras) que representam um sinal eletrocardiográfico, na derivação II, dentro de um intervalo de tempo de 1 segundo. O valor de cada amostra do sinal é representado em números inteiros, onde cada unidade vale um centésimo de milivolt (0,01 mV). Exemplo: o valor 25 na tabela representa uma amostra do sinal que vale 0,25 mV. Estas amostras são utilizadas para criar o Sinal que é obtido de um Simulador de ECG virtual. Apenas a derivação II está armazenada, pois foi utilizada a metodologia do trabalho de Lucena (1989), conforme descrito na seção 4.4.5 - O Formulário “Simulador de ECG”.

Os valores utilizados para as amplitudes do sinal foram extraídos de Rawlings (1991). Os intervalos de tempo P-Q, QRS e S-T foram extraídos de Netter (1978).

#### 4.2.2 - As Tabelas de Componentes

As Tabelas CAPACITORES, DIODOS, RESISTORES e TRANSISTORES armazenam dados sobre os Componentes dos esquemas eletrônicos do Aplicativo MONITOR para a Simulação de Funcionamento.

A Tabela CAPACITORES possui os seguintes campos: Código (expressões que identificam os Capacitores nos formulários do Aplicativo MONITOR), Valor (valor da capacitância dos componentes), Pos e Neg (identificam os nós onde estão conectados os terminais do Capacitor).

A Tabela RESISTORES possui os seguintes campos: Código (expressões que identificam os Resistores nos formulários do Aplicativo MONITOR), Valor (valor da resistência dos componentes), Tolerância (número inteiro que indica a porcentagem de tolerância no valor da resistência), Term1 e Term2 (identificam os nós onde estão conectados os terminais dos Resistores).

A Tabela DIODOS possui os seguintes campos: Código (expressões que identificam os Diodos nos formulários do Aplicativo MONITOR), Tipo (letra que indica o tipo de diodo: S = “Silício”, L = “LED”, Diodo Emissor de Luz), Ânodo e Cátodo (identificam os nós onde estão conectados os terminais do Diodo).

A Tabela TRANSISTORES possui os seguintes campos: Código (expressões que identificam os Transistores nos formulários do Aplicativo MONITOR), Tipo (expressão que identifica o tipo de Transistor, conforme manuais de componentes) e Beta (número que indica o fator de amplificação do Transistor, na configuração de Emissor Comum).

#### 4.2.3 - As Tabelas de Defeitos

As Tabelas DEFEITOS e DESCRIÇÕES são utilizadas pelo Aplicativo MONITOR para armazenar dados para a Simulação de Defeitos. São tabelas codificadas e relacionadas através do programa, pois uma Descrição de Defeito pode servir para diversas Falhas de Componentes.

A Tabela DESCRIÇÕES contém apenas os campos: Código da Descrição (seqüência de seis letras) e Descrição do Defeito (texto apresentado no Aplicativo MONITOR).

A Tabela DEFEITO contém os seguintes campos: Código do Defeito (seqüência de seis letras), Código da Descrição (seqüência de seis letras, relacionado com o mesmo campo na Tabela DESCRIÇÕES), Componente (expressões que identificam os Componentes nos formulários do Aplicativo MONITOR), Falha (expressões que indicam o Tipo de Falha: “Curto”, “Aberto”, “Saturado”, e outros) e Peso (número de 1 a 10, que representa a probabilidade da ocorrência de determinada Falha no Componente selecionado, utilizado para a seleção automática de defeitos pelo Aplicativo Seletor de Defeitos).

Na simulação de um Curto-Circuito adotamos o valor de Resistência de  $1 \text{ m}\Omega$  (um miliohm) no cálculo da corrente que passaria pelo Componente defeituoso. Isto evita a possibilidade de ocorrer um erro do tipo divisão-por-zero neste cálculo. O valor escolhido é cem vezes menor que o menor valor comercial de resistência ( $0,1 \Omega$ ) e é da ordem de grandeza das resistências impostas por cabos condutores e trilhas de circuito impresso.

Da mesma forma, na simulação de um Componente em Aberto sua resistência é alterada para  $1 \text{ G}\Omega$  (um gigaohm). A resistência de isolamento de Componentes (como Capacitores ou Circuitos Integrados) e entre as trilhas é desta ordem de grandeza.

As Tabelas DEFEITOS e DESCRIÇÕES podem ser acessadas pelo instrutor através do Aplicativo Seletor de Defeitos (ver seção 3.1.4), para definir qual Defeito será simulado no Aplicativo MONITOR.

#### **4.2.4 - A Tabela “MENSAGENS”**

Esta tabela contém um conjunto de Mensagens que o Aplicativo MONITOR utiliza em Caixas de Mensagem, em diversas situações.

#### **4.3 - O Arquivo de Inicialização “MONITOR.INI”**

O Arquivo MONITOR.INI (Tabela 2) é um arquivo de texto no formato padrão dos Arquivos de Inicialização de aplicativos do ambiente MS-Windows. Contém algumas diretrizes para a execução do aplicativo MONITOR.EXE, de acordo com o Defeito escolhido pelo Instrutor no Aplicativo DEFEITO.EXE.

```
[Defeito]
Codigo=AO03AP
PorCento=1000

[Servico]
Tecnico=Guilherme
```

**Tabela 2 - Conteúdo do Arquivo de Inicialização MONITOR.INI**

Como os demais Arquivos de Inicialização, o arquivo MONITOR.INI apresenta as diretrizes, chamadas Chaves, reunidas em Seções, definidas por determinadas palavras, entre colchetes. A Seção Defeito contém as Chaves Código, cujo Valor é o código do defeito a ser simulado pelo Aplicativo, e PorCento, utilizado quando a falha do componente é do tipo Variação de Tolerância. A Seção Serviço contém a Chave Técnico, com o nome do aprendiz sob treinamento.

#### **4.4 - O Aplicativo MONITOR.EXE**

O Aplicativo MONITOR.EXE é o principal componente do *software* desenvolvido; seus formulários e controles são descritos nas seções a seguir.

##### **4.4.1 - O Formulário Principal “MDIFMonitor”**

A Interface tipo MDI controla as relações com o Sistema e entre os Formulários e o fluxo de simulação, através do Timer “TempVarre” (ver Seção 4.4.7). É o único Formulário que possui uma Barra de Menu, controlando a disposição das Janelas e o acesso ao Arquivo de Ajuda “Manual do Monitor Cardíaco”.

##### **4.4.2 - Os Formulários “Painel”**

Os Formulários “Painel Frontal” e “Painel Traseiro” emulam os controles, cabos e mostradores do Monitor Cardíaco. Através destes Formulários, é possível simular a Operação do equipamento.

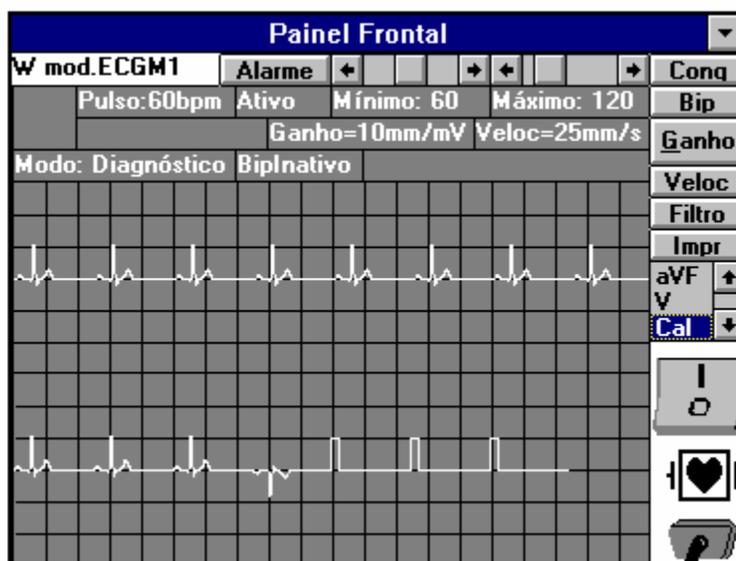
No Formulário “Painel Frontal” (Figura 6), duas Caixas de Figura emulam a tela do Monitor, apresentando o sinal eletrocardiográfico. Três Etiquetas, logo embaixo, apresentam o valor da Frequência Cardíaca e o estado do Filtro e do Alarme. Uma Caixa de Figura, à direita da tela, emula o Alarme Visual.

Uma Caixa de Combinação permite a seleção da Derivação a ser apresentada na tela. Duas Barras de Rolagem controlam os valores máximo e mínimo de Frequência Cardíaca para o Alarme.

Diversos Botões de Comando 3D emulam os controles do Painel do Monitor. O Botão “Inop” habilita e desabilita o Alarme Sonoro. “Cong” congela a imagem de uma das telas, e o sinal só é apresentado na outra. “Bip” habilita e desabilita o som do equipamento.

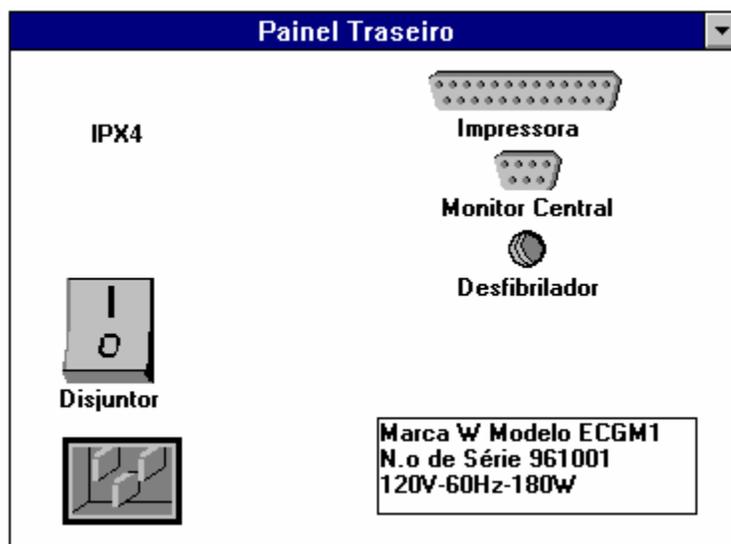
O Controle de Ganho (Sensibilidade) pode ser ajustado em ½, 1 ou 2 cm/mV. O Controle de Velocidade permite o ajuste em 25 ou 50 mm/s. E o Botão de Filtro habilita e desabilita o Filtro contra Interferências.

Um Controle do tipo Animação emula o Interruptor Liga-Desliga do equipamento. Um Objeto tipo Imagem apresenta o símbolo de Equipamento Tipo CF a Prova de Desfibrilador, conforme a Norma Técnica NBR-IEC 601-2-27 (ABNT - 1997).



**Figura 6 - O Formulário “Painel Frontal” do Monitor Cardíaco**

No Formulário “Painel Traseiro” (Figura 7), um Controle do tipo Animação emula o Conector do cabo de alimentação. Outro Controle do mesmo tipo emula o Disjuntor (similar ao Interruptor Liga-Desliga do Painel Frontal). Figuras representam a placa de identificação do equipamento e outros conectores, sem uso neste programa.



**Figura 7 - O Formulário “Painel Traseiro” do Monitor Cardíaco**

#### **4.4.3 - Os Formulários “Diagramas de Blocos”**

Para permitir a simulação, o Monitor Cardíaco foi desmembrado em seis grandes blocos: Fonte de Alimentação, Entrada do Sinal, Acoplamento, Processamento Analógico, Módulo Digital e Vídeo. Os dois últimos não foram implementados no presente trabalho.

Um Formulário Diagrama de Blocos (Figura 8) foi criado para cada grande bloco, nos quais eles estão desmembrados em blocos menores (os circuitos). Cada bloco menor é representado por um Botão de Comando, que abre o Formulário Circuito Eletrônico correspondente, quando acionado.

Botões de Opção indicam a presença de um nó (entrada ou saída de bloco) que pode ter sua forma-de-onda visualizada no Formulário “Osciloscópio”.

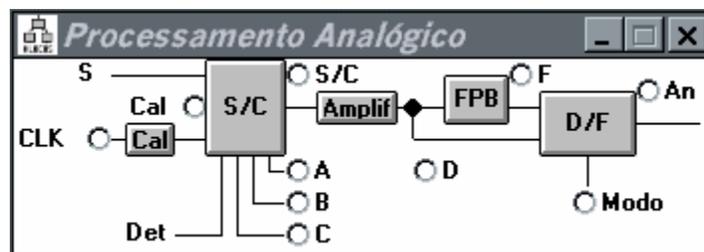


Figura 8 - Exemplo de Formulário Diagrama de Blocos

#### 4.4.4 - Os Formulários “Circuitos Eletrônicos”

Os Formulários de Circuitos Eletrônicos apresentam o esquema de Componentes dos diversos blocos do equipamento. Cada Componente é representado por um Botão de Comando 3D. Quando é selecionado um Componente, a Janela Ação de Manutenção permite avaliar os valores de seus parâmetros através do Multímetro. Alguns destes valores estão disponíveis na Base de Dados, outros são calculados durante a execução do Aplicativo.

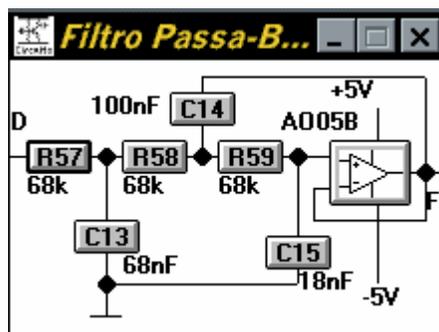


Figura 9 - Exemplo de Formulário Circuito Eletrônico

#### 4.4.5 - O Formulário “Simulador de ECG”

O “Simulador de ECG” (Figura 10) é um Formulário para Ações de Manutenção especificamente criado para o Monitor Cardíaco. Possui um Controle da Frequência

Cardíaca (0, 60 ou 120 bpm), e Controles para emular Interferências no sinal eletrocardiográfico. Seu funcionamento foi baseado no trabalho de Lucena (1989). O sinal da Derivação II é armazenado na memória do equipamento, num determinado número de amostras. Um conversor digital-analógico e um amplificador reproduzem o sinal eletrocardiográfico. E os sinais para as diversas posições dos eletrodos (RA, RL, LA, LL e V) são obtidos a partir de uma rede atenuadora composta de três resistores, agindo como divisores de tensão. As arritmias simuladas no trabalho citado não foram utilizadas no presente trabalho, por não serem importantes para o treinamento dos técnicos.



**Figura 10 - O Formulário “Simulador de ECG”**

#### 4.4.6 - O Formulário “Ordem de Serviço”

Este Formulário (Figura 11) apresenta uma Ordem de Serviço ao Aprendiz, um pedido de assistência técnica semelhante àquela do mundo real. Este Formulário é apresentado sempre que se inicie o Aplicativo, ou através da opção de Menu “Janela” - “Ordem de Serviço”.

<b>Ordem de Serviço</b>	
Abertura:	15/10/2002 15:31:13
Status:	Em Execução
Equipamento:	Monitor Cardíaco
Marca:	W
Modelo:	ecgm1
Num. de Série:	P/0001-96
Técnico:	Guilherme
<b>Descrição do Defeito:</b>	
Sem Sinal, com Mensagem "Eletrodo Solto", exceto Derivação II.	
<b>Encerramento:</b>	
<input type="button" value="OK"/>	

Figura 11 - O Formulário Ordem de Serviço

#### 4.4.7 - O Controle Timer “TempVarre”

Este Controle detém o código da simulação. Disparado a cada intervalo de 1 ms (um milissegundo), ele calcula o valor da tensão em cada nó dos circuitos eletrônicos do equipamento.

Praticamente todas as Funções de Transferência de blocos foram reduzidas a ganhos simples (amplificação ou divisão de tensão) ou funções lógicas representando chaveamentos. Isto se deve à baixa frequência de amostragem da simulação. Como

trabalhamos com uma taxa de 600 amostras por segundo, temos uma frequência de Nyquist de 300 Hz, compatível com a resposta em frequência de monitores cardíacos.

Apenas para o Filtro foi necessário um modelamento mais complexo, pois sua frequência de corte está abaixo da frequência de Nyquist. O Filtro de 3ª ordem teve sua Função de Transferência no domínio da frequência contínua transformada em equação a diferenças, no domínio do tempo discreto.

Utilizamos o algoritmo para detecção do complexo QRS de Abrantes & Nadal (1992), convertido para linguagem Visual Basic.

#### **4.5 - O Arquivo de Ajuda “MANUAL DO MONITOR”**

O Arquivo de Ajuda MONITOR.HLP, descrito a seguir, foi desenvolvido para conter as informações de um Manual de Operação e Manutenção do Monitor Cardíaco, contendo inclusive explicações sobre a finalidade clínica e princípio de funcionamento. O “Manual” foi dividido em alguns “Capítulos”, descritos a seguir.

##### **4.5.1 - O Capítulo “Como Usar o Programa”**

Este capítulo apresenta algumas informações sobre a utilização do programa, tais como: descrições das opções de menu, dos formulários (as “Janelas”) e de como utilizar o próprio Arquivo de Ajuda.

#### 4.5.2 - O Capítulo “Introdução”

Este capítulo apresenta explicações sobre os seguintes tópicos:

- Princípio de Funcionamento do Monitor Cardíaco;
- Explicações básicas sobre Anatomia e Fisiologia do Coração;
- Descrição do processo de Condução de Estímulos Elétricos nas Células;
- Descrição do processo de Condução de Estímulos Elétricos no Coração;
- Explicações sobre Frequência Cardíaca, Eletrodos (Figura 12) e Derivações;
- Explicações sobre o Sinal Eletrocardiográfico como resultado do processo de Condução de Estímulos Elétricos no Coração;
- Eletrocardiograma de pessoas normais;
- Exemplos de Anomalias do Sinal Eletrocardiográfico relacionados com doenças.

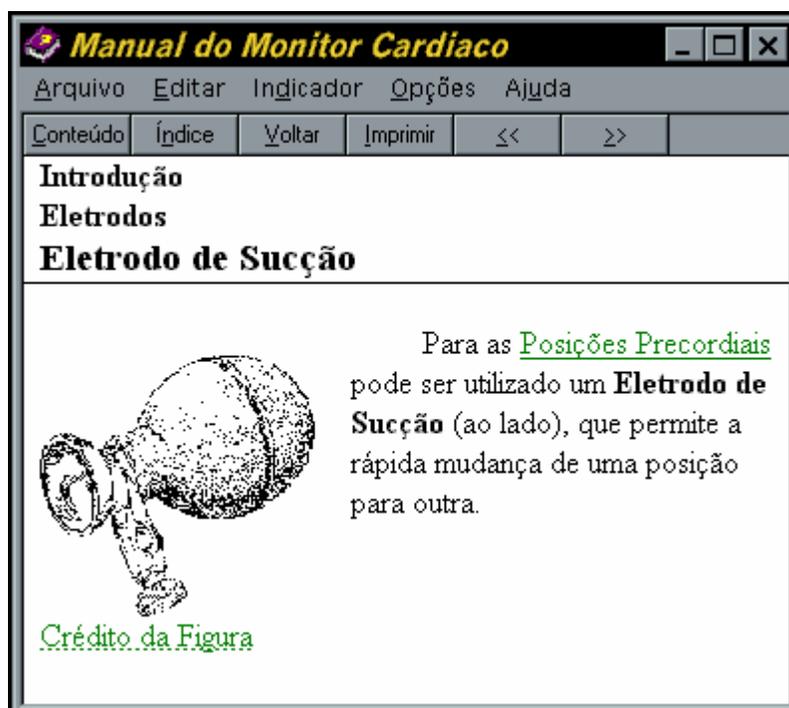
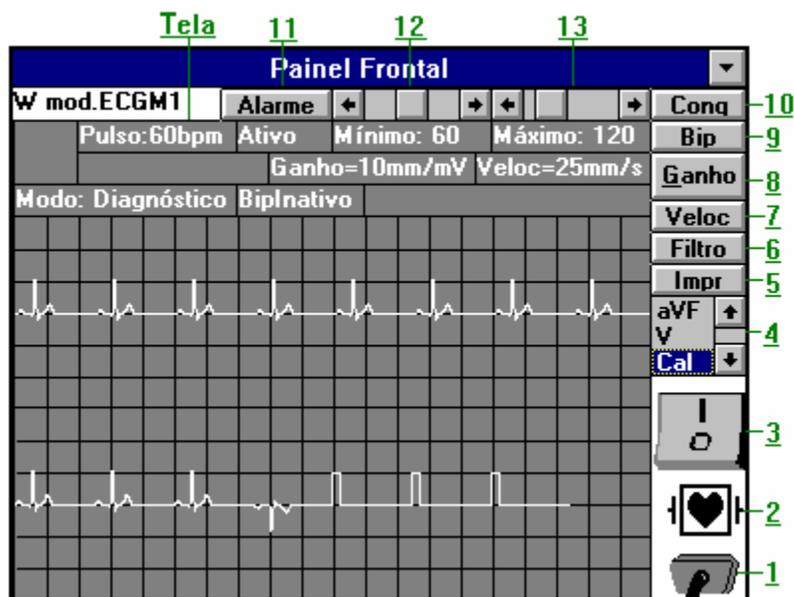


Figura 12 - Exemplo de Página do Capítulo “Introdução”

Além de diversas figuras, quatro aplicativos servem de ilustrações interativas para este Capítulo. Estes aplicativos são descritos na Seção 4.7.

#### 4.5.3 - O Capítulo “Operação”

Este Capítulo apresenta as funções dos diversos Controles presentes no Painel de Controle de um Monitor Cardíaco. Em sua Página inicial existe um hipergráfico (Figura 13) que permite o acesso direto às Páginas correspondentes aos Controles.



**Figura 13 - Hipergráfico do Capítulo sobre "Operação"**

Ao final do Capítulo são apresentados também alguns cuidados que se deve ter na Operação do Monitor Cardíaco. Embora alguns destes cuidados devam ser tomados apenas pelos profissionais de enfermagem, em seu trabalho com o paciente, o técnico deve conhecê-los, pois ele pode eventualmente participar de um treinamento para estes profissionais.

#### 4.5.4 - O Capítulo “Segurança Elétrica”

Este Capítulo apresenta os requisitos de Segurança Elétrica para Monitores Cardíacos, conforme a Normas Técnicas Brasileiras (ABNT - 1994 e 1997), com explicações para os símbolos de segurança presentes no equipamento.

#### 4.5.5 - O Capítulo “Funcionamento”

Este Capítulo apresenta as explicações sobre os diversos Diagramas de Blocos e Circuitos Eletrônicos simulados no Aplicativo. Nas Páginas sobre os Diagramas de Blocos estão disponíveis hipergráficos que permitem o acesso direto à Página do Circuito Eletrônico correspondente (Figura 14 e Figura 15).

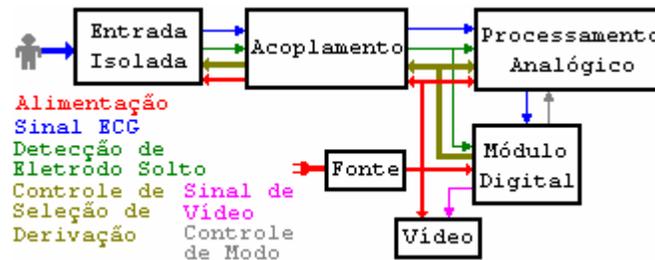


Figura 14 - Hipergráfico principal do Capítulo sobre "Funcionamento"

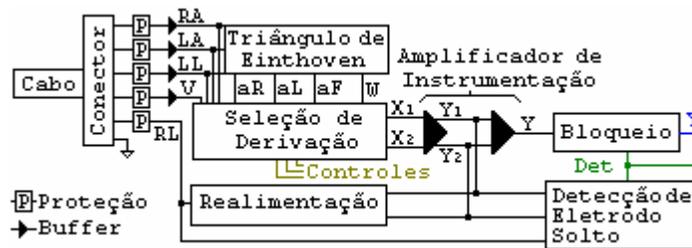


Figura 15 - Exemplo de Hipergráfico de Blocos do capítulo "Funcionamento"

#### **4.5.6 - O Capítulo “Interferências”**

Este Capítulo apresenta as principais fontes de Interferências que afetam o Funcionamento do Monitor Cardíaco:

- Artefato Respiratório
- Sinal Eletromiográfico
- Rede de Energia Elétrica
- Acionamento da Unidade Eletrocirúrgica
- Descarga do Desfibrilador
- Eletrodo Solto

#### 4.5.7- O Capítulo “Manutenção”

Este capítulo contém os detalhes de manutenção, que devem estar contidos na documentação técnica:

- Desenho Explodido (Figura 16);
- Equipamentos de Teste
- Rotina de Inspeção e Manutenção Preventiva;
- Cuidados - no transporte, montagem e instalação do equipamento;
- *Troubleshooting* (Diagnóstico de Falhas).

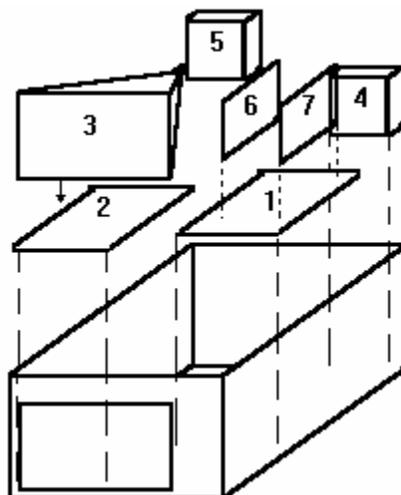


Figura 16 – “Desenho Explodido”

#### 4.5.8 - O Anexo “Outros Equipamentos”

Contém explicações sobre o Princípio de Funcionamento dos dois equipamentos citados no Arquivo de Ajuda: o Desfibrilador e a Unidade Eletrocirúrgica (também conhecido por Bisturi Elétrico).

#### 4.6 - Os Aplicativos de Ilustração para a Ajuda

Quatro aplicativos foram desenvolvidos, também em Visual BASIC, para servir como ilustrações interativas para o Arquivo de Ajuda. Eles são descritos a seguir.

##### 4.6.1 - O Aplicativo “PULSO”

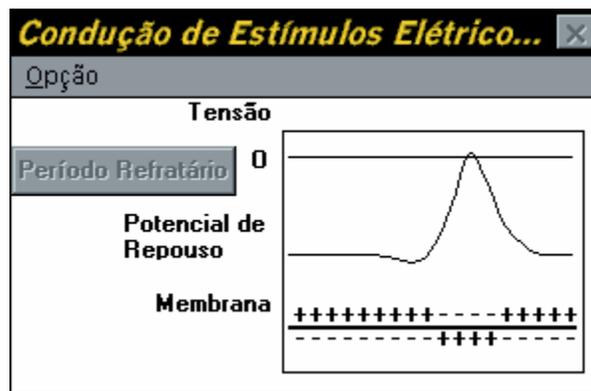
O Aplicativo PULSO.EXE (Figura 17) apresenta uma animação dos movimentos do coração, que representa seu funcionamento como bomba de sangue para o corpo. Quatro figuras são apresentadas alternadamente, mostrando o fluxo de sangue através do coração e os movimentos de abertura e fechamento das válvulas cardíacas. As figuras apresentadas foram desenvolvidas pelo autor, no Aplicativo PaintBrush.



Figura 17 - O Aplicativo de Animação PULSO.EXE

#### 4.6.2 - O Aplicativo “CONDUCAO”

O Aplicativo CONDUCAO.EXE (**Figura 18**) apresenta uma simulação da Condução de Estímulos Elétricos nas Células, ilustrando a página sobre o assunto no Arquivo de Ajuda Manual do Monitor. Ele apresenta uma seqüência de nove figuras, controlada por um Botão de Comando que simula o estímulo que chega à célula e dispara o processo de condução elétrica. Durante a exposição das figuras, este Botão de Comando fica desabilitado, simulando o chamado Período Refratário, durante o qual novos estímulos não conseguem disparar outros processos de condução. As figuras apresentadas foram desenvolvidas pelo autor, no Aplicativo PaintBrush.



**Figura 18 - O Aplicativo de Animação CONDUCAO.EXE**

#### 4.6.3 - O Aplicativo “VETOR”

O Aplicativo VETOR.EXE (Figura 19) apresenta, através de uma série de nove figuras, a Geração do Sinal Eletrocardiográfico a partir da projeção do Vetor Instantâneo Médio nas direções das Derivações I, II e III. O Vetor Instantâneo Médio indica a intensidade, direção e sentido da média dos estímulos elétricos cardíacos em cada momento. As figuras apresentadas foram modificadas de Netter (1978).

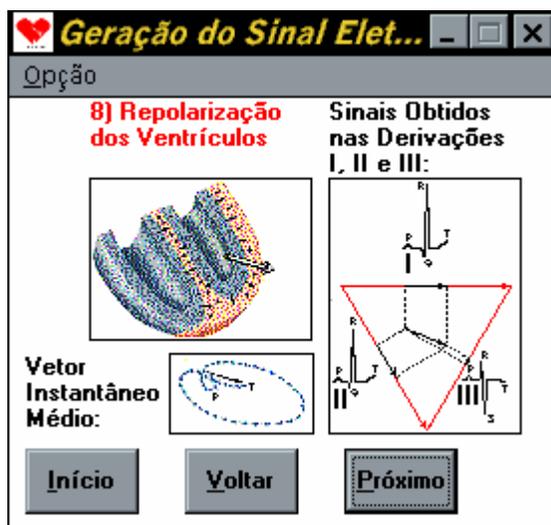
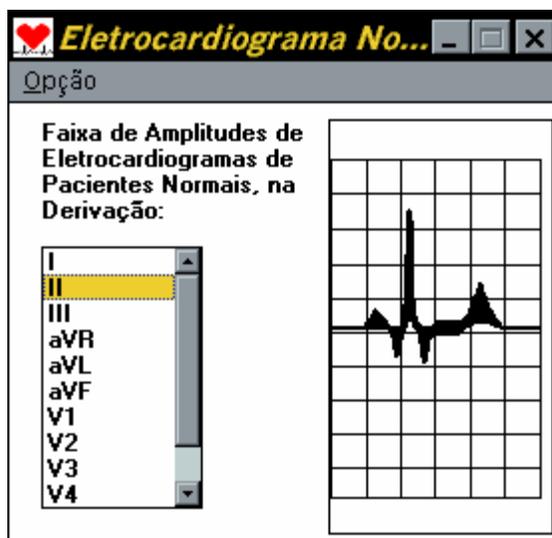


Figura 19 - O Aplicativo VETOR.EXE

#### 4.6.4 - O Aplicativo “ECGNORML”

O Aplicativo ECGNORML.EXE (**Figura 20**) apresenta em uma Caixa de Figura a faixa de amplitudes de um Eletrocardiograma de pessoa normal, na Derivação selecionada na Caixa de Listagem (as doze derivações: I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5 e V6). As figuras apresentadas foram modificadas de Rawlings (1991).



**Figura 20 - O Aplicativo EGNORML.EXE**

#### 4.7 - O Relatório Técnico.TXT

Um Arquivo de Texto é gerado durante a execução do Aplicativo, para armazenar todas as ações do Usuário. Este relatório pode ser útil na avaliação da simulação, conforme explicado na Seção 3.6.2.

## **CAPÍTULO 5 - RESULTADOS**

Foram realizados dois tipos de testes conforme descrito no Capítulo 3:

### **5.1 - Teste de Assimilação**

Onze alunos do curso de Técnico de Equipamentos Médico-Hospitalares, do Colégio Técnico de Campinas (Cotuca-Unicamp), da turma de 1999, foram submetidos ao treinamento, respondendo ao mesmo teste antes e depois de utilizar o *software*. As questões eram:

- 1) Quais são as opções de Ganho do Monitor Cardíaco? Como o Ganho pode ser aferido?
- 2) Pode-se desfibrilar o Paciente com o Monitor Cardíaco ligado? Porque?
- 3) Existe um Filtro que pode ser acionado, ou não, nos Monitores Cardíacos. Qual a sua função? Porque ele deve ser utilizado em algumas ocasiões, e não é aconselhável em outras?

Os testes foram realizados no mesmo dia: o pré-teste; o treinamento; e o pós-teste, quando os alunos ainda não tinham conhecimento das notas do pré-teste.

As respostas foram classificadas em: Corretas (2 pontos), Satisfatórias (1 ponto) e Erradas (sem pontos). A Tabela 3 apresenta o desempenho dos alunos.

ALUNO	TESTE PRÉ-TREINAMENTO				TESTE APÓS TREINAMENTO			
	Questão 1	Questão 2	Questão 3	Total	Questão 1	Questão 2	Questão 3	Total
A	0	1	1	2	0	1	2	3
B	0	2	0	2	1	2	0	3
C	2	0	0	2	1	2	1	4
D	1	2	1	4	2	2	2	6
E	1	2	2	5	2	2	1	5
F	2	2	2	6	2	2	2	6
G	1	0	1	2	2	1	2	5
H	1	1	1	3	1	2	2	5
I	0	0	1	1	1	2	2	5
J	0	2	1	3	2	2	2	6
K	1	0	2	3	2	0	2	4
<b>TOTAL</b>	9	12	12	33	16	18	18	52
<b>(%)</b>	<b>40,9</b>	<b>54,5</b>	<b>54,5</b>	<b>50,0</b>	<b>72,7</b>	<b>81,8</b>	<b>81,8</b>	<b>78,8</b>

Tabela 3 - Desempenho dos Alunos nos Testes

### 5.2 - Teste de Simulação

Vários alunos do curso de Técnico de Equipamentos Médico-Hospitalares, do Colégio Técnico de Campinas (Cotuca-Unicamp), têm sido submetidos a simulações após aulas convencionais, utilizando o Arquivo de Ajuda como ferramenta de ensino. A análise do arquivo RELATORIO.TXT permite avaliar a rapidez e objetividade do técnico em

procurar o defeito somente nos circuitos relacionados com a possível causa. O Anexo C apresenta exemplos de relatórios obtidos nas simulações.

## **CAPÍTULO 6 - DISCUSSÃO**

O teste de assimilação demonstrou um aprimoramento no desempenho de todos os alunos após o treinamento, aumentando a média da turma de 50,0 para 78,8. O teste deve ser repetido para um maior número de alunos, utilizando o Aplicativo por mais tempo, para uma melhor avaliação.

Os testes de simulação mostraram que o Aplicativo reproduz situações reais com razoável fidelidade.

## **CAPÍTULO 7 - CONCLUSÕES**

### **7.1 - A Simulação de Falhas Múltiplas**

Eventualmente, a falha de um determinado Componente pode produzir uma sobretensão, ou sobrecorrente, em outro Componente, levando-o também a uma falha.

O Aplicativo desenvolvido não permite a simulação de falhas múltiplas, isto é, de vários Componentes simultaneamente. Para fazê-lo, seriam necessárias algumas modificações nas tabelas da Base de Dados MONITOR.MDB, no Formulário Ação de Manutenção e no *Timer* TempVarre.

### **7.2 - Generalizando o Aplicativo**

O Aplicativo desenvolvido serve apenas para o treinamento sobre Monitores Cardíacos. Utilizando-se a mesma metodologia, é possível criar Aplicativos sobre outros equipamentos biomédicos. Entretanto, seria mais interessante tornar o Aplicativo genérico, transferindo os dados específicos de um determinado equipamento para Bases de Dados. Assim, para simular outros equipamentos bastaria desenvolver as Bases de Dados, com estruturas semelhantes, o que simplificaria o trabalho do Programador. Uma linguagem mais atual e voltada para a Internet pode ser usada, como, por exemplo, Java ou Python.

### **7.3 - Comparando a Aprendizagem com a Sala de Aulas**

Para verificar a validade do treinamento seria interessante utilizar a metodologia de teste comparativo, como descrito na Seção 3.9.3. Entretanto, esta metodologia exige um prazo longo para garantir sua precisão. Além disso, há o problema ético de se submeter um aluno a esta experiência, pois os dois grupos de alunos devem ser expostos aos mesmos

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

conteúdos. Para ser realizada seria interessante contar com a presença de um educador, graduado em Pedagogia. Este teste seria interessante para um trabalho futuro.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. **ABNT** Norma Técnica NBR-IEC 601.1 - Equipamento Eletromédico - Parte 1 - Prescrições Gerais para Segurança, 1994.
2. **ABNT** Norma Técnica NBR-IEC 601.2.27 - Equipamento Eletromédico - Parte 2 - Prescrições Particulares para Segurança de Equipamentos de Monitorização de Eletrocardiograma, 1997.
3. **ABRANTES, A.C.S. & NADAL, J.** Algoritmo para Detecção de Complexos QRS em Microcomputadores PC-AT. In: FÓRUM NACIONAL DE CIÊNCIA E TECNOLOGIA EM SAÚDE, 1., 1992, Caxambu - MG. Anais 668 p. p. 228-230.
4. **HOFFER, E. P. & BARNETT, G. O.** Computers in Medical Education, em **SHORTLIFFE, E. H., PERREAULT, L. E. & Cols.** Medical Informatics, cap.17, Addison-Wesley, 1990.
5. **HOOPER, J. A., BRONZINO, J. D., NOYES, N. T. & TAYLOR D.** EquipTeach: A Computer-Aided Instruction To Teach Users How To Operate Specific Medical Equipment. Biomedical Instrumentation and Technology, v. 27, p. 394-399, 1993.

6. **JIANG, M. Zh. & SITTING, D. F.** Developing Interactive Computer-Based Simulations: an Object-Oriented Development Methodology Enhances Computer-Assisted Instruction. Computer Methods and Programs in Biomedicine, v. 47, p. 189-196, 1995.
7. **LOVEDAY, G. C.** Electronic Fault Diagnosis. Pitman Publ. Ltd., 1982.
8. **LUCENA, Samuel E.** Um Simulador de ECG/Arritmia Cardíaca. Dissertação de Mestrado, DEB-FEE-UNICAMP, 1989.
9. **MATABAN, Bridget A. M.** Prototype Expert System for Infusion Pump Maintenance. Biomedical Instrumentation & Technology, v. 28, p. 19-29, 1994.
10. **MONIZ, M. J., CALVIN, J. & STANKIEWICZ, H.** Innovative Communication in a Biomedical Engineering Department. Journal of Clinical Engineering, v. 20, p. 212-217, 1995.
11. **MOREIRA, Cristiane M. M.** SIMVEP - Simulador da Ventilação Pulmonar. Dissertação de Mestrado, DEB-FEE-UNICAMP, 1996.
12. **NASH, Suzanne,** A Systematic Approach to Electronic Troubleshooting for Medical Equipment. Biomedical Instrumentation & Technology, v. 29, p. 236-242, 1995.

13. **NERI, Lori A.** From Books to Bytes. Healthcare Technology Management, v. 7, nº 3, p. 28-32, mar/1996.
14. **NETTER, Frank H.** The CIBA Collection of Medical Illustrations, Volume 5: Heart. CIBA, 1978.
15. **PIPITONE, Frank.** An Expert System for Electronics Troubleshooting Based on Qualitative Causal Reasoning. Journal of Computer-Based Instruction, v. 13, p. 39-42, 1986.
16. **PLISKE, D.B. & PSOTKA, J.** Exploratory Programming Environments for Designing ICAI. Journal of Computer-Based Instruction, v. 13, p. 52-57, 1986.
17. **RAWLINGS, Charles A.** Electrocardiography. SpaceLabs Inc., 1991. (Biophysical Measurement Series).
18. **RICHARDSON, J.J. & JACKSON, T.E.** Developing the Technology for Intelligent Maintenance Advisors. Journal of Computer-Based Instruction, v. 13, p. 47-51, 1986.

19. **SALITERMAN, Steven S.** A Computerized Simulator for Critical-Care Training: New Methodology for Medical Education. Mayo Clinic Proceedings, v. 65, p. 968-978, 1990.
  
20. **SCHWID, Howard A.** A Flight Simulator for General Anesthesia Training. Computers and Biomedical Research, v. 20, p. 64-75, 1987.
  
21. **WENGER, E.** Artificial Intelligence and Tutoring Systems. Morgan Kaufmann Publ. Inc., 1987.
  
22. **WIKLUND, Michael E.** Writing Effective User Manuals. Medical Device & Diagnostic Industry, p. 96-104, jan/1993.
  
23. **XU, Yixiong** The Art and Science of Troubleshooting Medical Equipment: A Model for Troubleshooting Medical Equipment down to the Component Level. Biomedical Instrumentation & Technology, v. 31, p. 129-136, 1997.

**ABREVIATURAS**

ABNT	Associação Brasileira de Normas Técnicas
ANSI	American National Standards Institute - Instituto Americano de Padrões
ASCII	American Standard Code for Information Interchange - Código Padrão Americano para Intercâmbio de Informações
BASIC	Beginners' All-Purpose Symbolic Instruction Code - Código Versátil de Instruções Simbólicas para Iniciantes
CAD	Computer Aided Design - Projeto Auxiliado por Computador
CD	<i>Compact Disk</i>
CGA	Color Graphics Adapter - Adaptador Gráfico a Cores
DDE	Dynamic Data Exchange - Troca Dinâmica de Dados
DLL	Dynamic Link Library - Biblioteca de Vínculo Dinâmico
DOS	Data Operating System
EGA	Extended Graphics Adapter - Adaptador Gráfico Ampliado
MDI	Multiple-Document Interface - Interface de Múltiplos Documentos
OLE	Object Linking and Embedding - Vinculação e Inserção de Objetos
PC	Personal Computer - Computador Pessoal
RAM	Random Access Memory - Memória de Acesso Aleatório
ROM	Read-Only Memory - Memória Restrita a Leitura
TAC	Treinamento Auxiliado por Computador - (Computer Aided Instruction)
VGA	Video Graphics Adapter - Adaptador de Gráfico para Vídeo

**Tabela 4 - Abreviaturas utilizadas na Dissertação ou no *Software***

**MARCAS REGISTRADAS**

<b>Produtos:</b>	<b>Empresa:</b>
PhotoShop	Adobe
PhotoStyler	Aldus
Apple, Macintosh	Apple Computer Inc.
Multimedia Toolbook	Asymetrix Inc.
IBM, PC	IBM - International Business Machines Co.
XWindows	Massachusetts Institute of Technology
MS-DOS, MS-Windows, Visual BASIC, Word, Access, PaintBrush	Microsoft Corporation
OpenWindows	Sun Microsystems, Inc.
Unix	Unix Systems Laboratories, Inc.

**Tabela 5 - Marcas e produtos registrados pelas respectivas empresas**

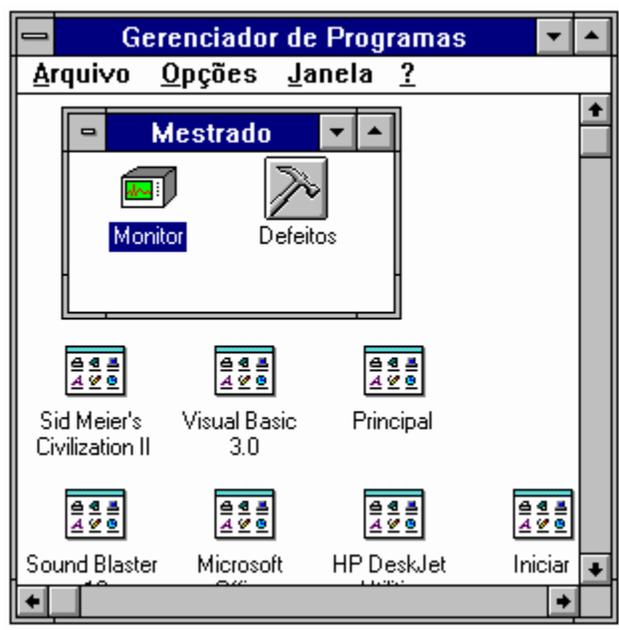
## **ANEXO A - PROGRAMAÇÃO EM VISUAL BASIC**

### **A.1 - O Ambiente Operacional MS-Windows**

O ambiente gráfico MS-Windows é constituído por um formulário principal, o Gerenciador de Programas, diversos aplicativos e um conjunto de recursos computacionais disponíveis para os desenvolvedores de programas. Para utilizar estes recursos o programador deve ter à disposição alguma linguagem de programação para este ambiente, como Visual C++ ou Visual BASIC, por exemplo.

O Gerenciador de Programas apresenta diversos conjuntos de ícones, símbolos que identificam programas que podem ser executados a partir deste Gerenciador, os chamados aplicativos.

Os aplicativos também são constituídos por formulários e têm diversas finalidades: edição de textos, processamento de imagens, gerenciamento de bases de dados, e outros. Em comum eles possuem o tipo de programação, orientada a Objetos e Eventos.



**Figura 21 - O Gerenciador de Programas**

### **A.2 - A Linguagem Visual BASIC 3.0**

O Visual BASIC é uma linguagem para programação orientada a objetos e eventos desenvolvida especificamente para o Ambiente Operacional MS-Windows. Encontra-se disponível no Brasil somente em inglês, mas permite o uso de acentos, til e cedilha.

### **A.3 - Os Objetos do MS-Windows e do Visual BASIC**

Existe uma hierarquia definida entre os diferentes Objetos do Ambiente Operacional MS-Windows. Um Aplicativo (*software*) é um Objeto do Sistema, assim como a Impressora ou um Arquivo, por exemplo. Um Formulário é um Objeto do Aplicativo e um Controle é um Objeto do Formulário.

Os Aplicativos podem se comunicar entre si através do Sistema. Através das Bibliotecas de Vínculo Dinâmico (DLL), podem compartilhar códigos de programação de uso geral no Sistema. Através da Troca Dinâmica de Dados (DDE), podem compartilhar

dados e processamento durante sua execução. Através da Vinculação e Incorporação de Objetos (OLE), podem compartilhar Objetos.

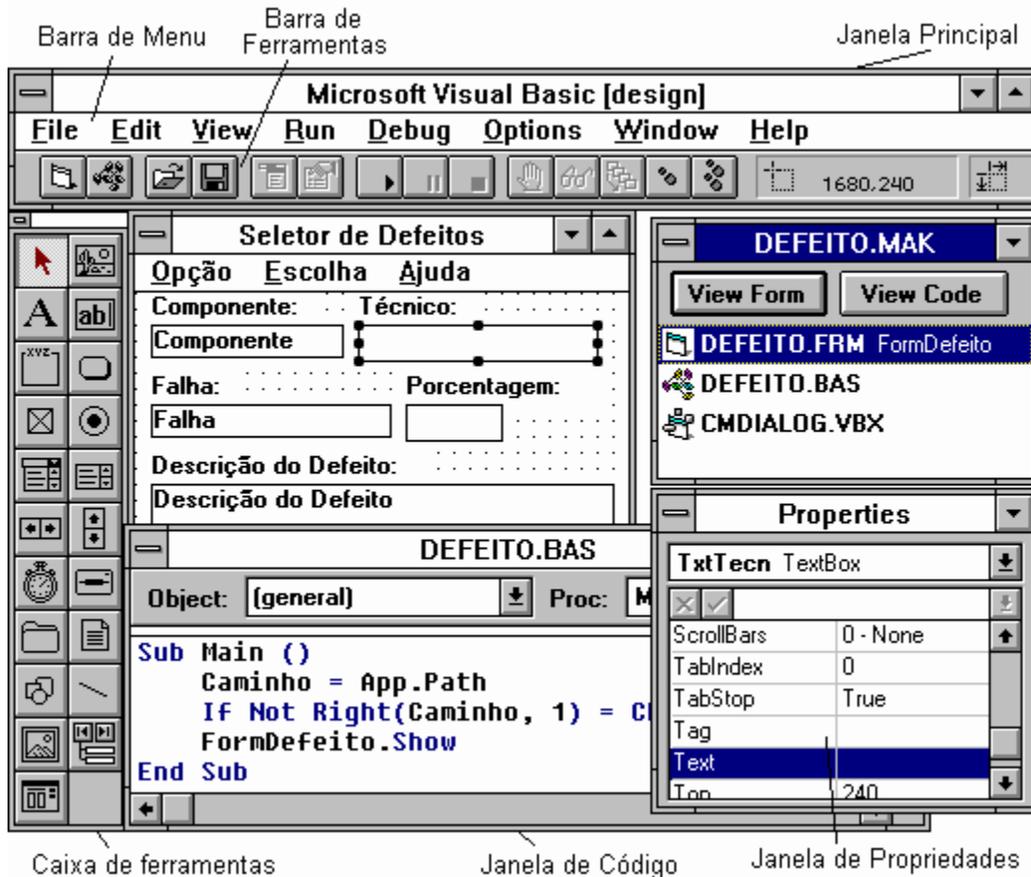
Um Aplicativo pode não possuir Formulários, executando apenas o Procedimento “Sub Main()”. Mas isto é uma subutilização dos recursos gráficos do MS-Windows. Assim, praticamente todos os Aplicativos possuem pelo menos um Formulário. Os Formulários no Visual BASIC são descritos na Seção A.7.

Cada Formulário possui um conjunto de Controles, como a Barra de Menu e os Botões de Comando, por exemplo. Uma descrição mais completa dos Controles do Visual BASIC é apresentada na Seção A.8.

O principal trabalho do Programador é desenvolver a interação entre todos estes Objetos, adequada às suas necessidades de apresentação.

#### **A.4 - O Ambiente de Programação do Visual BASIC 3.0**

O Visual BASIC também é um Aplicativo do MS-Windows, possuindo seus Formulários e Controles. A seguir, descrevemos resumidamente este Aplicativo.



**Figura 22 - O Ambiente de Programação do Visual BASIC 3.0**

A Janela Principal controla as demais Janelas e a interface com o Sistema. Os principais comandos do Menu estão também disponíveis graficamente na Barra de Ferramentas.

A Janela de Formulário apresenta graficamente como estarão disponíveis os Controles no Formulário real. Através da Janela de Controles, é possível inserir os Controles na Janela de Formulário. Maiores informações sobre Formulários estão disponíveis na Seção A.7.

A Janela de Projeto apresenta uma lista dos Arquivos que compõem o futuro Aplicativo (ver Seção A.6), permitindo acesso a eles. A Janela de Propriedades permite a

edição das Propriedades dos diversos Objetos (Seção A.5). Finalmente, a Janela de Código permite criar e editar as rotinas de programação de todos os Objetos do Projeto.

#### **A.5 - As Propriedades de um Objeto**

A estrutura de um Objeto é definida por um conjunto de Propriedades, que devem ser ajustadas de acordo com as necessidades do programador. Para cada tipo de Objeto existe um conjunto pré-definido de Propriedades.

Quando um Objeto é criado alguns valores padronizados (*Default*) são estabelecidos. As Propriedades de um Controle ou Formulário podem ser editadas através da Janela “Properties”.

O exemplo da figura abaixo mostra algumas Propriedades de um Botão de Comando:



**Figura 23 - O Botão de Comando CmdOK**

Propriedade	Significado	Valor
Name	Nome do Objeto	CmdOK
Caption	Título (texto apresentado no corpo do Objeto)	OK
Height	Altura	495
Width	Largura	1215
FontType	Tipo de Fonte do Título	MS Sans Serif
FontSize	Tamanho da Fonte do Título	12
FontBold	Fonte do Título em Negrito	True
FontItalic	Fonte do Título em Itálico	False

**Tabela 6 - Exemplos de Propriedades de CmdOK**

O Nome (*Name*) de um Objeto é uma Propriedade transparente ao Usuário, mas muito importante para o Programador, pois identifica o Objeto na programação dos Módulos e dos Eventos.

Outra Propriedade importante e também transparente ao Usuário é o Identificador de Contexto de Ajuda (*HelpContextID*), que permite se estabelecer um Vínculo entre o Objeto e uma determinada Página de um Arquivo de Ajuda, que pode ser acessada através da tecla

F1, quando o Objeto tem o Foco. Esta Propriedade não é disponível a todos os Objetos do Visual BASIC.

#### **A.6 - Arquivo de Projeto**

O Arquivo de Projeto, com extensão MAK, no formato de texto ASCII, contém uma lista de todos os arquivos que compõem o Aplicativo:

- **Formulários** - os arquivos com extensão FRM contêm uma descrição gráfica dos Formulários e seus Controles, inclusive seu ajuste de Propriedades, e os respectivos códigos de programação;
- **Módulos** - os arquivos com extensão BAS contêm os códigos de programação relativos a declarações de tipos, variáveis, constantes e procedimentos globais, isto é, no nível de Aplicativo;
- **Controles Extras** - os arquivos com extensão VBX (Visual BASIC Extensions) contêm novos tipos de Controles, incorporados ao Visual BASIC, com os respectivos conjuntos de Propriedades e Eventos.

O ajuste das Propriedades do Aplicativo fica também armazenado no Arquivo de Projeto. A Propriedade “Start Up Form” (Formulário Inicial) indica qual Formulário será aberto ao se iniciar o Aplicativo. A Propriedade “Help File” (Arquivo de Ajuda) indica qual Arquivo de Ajuda se deseja vincular com o Aplicativo. Estas propriedades podem ser editadas através da opção de Menu “Options” e “Project”.

É possível criar novos Formulários e Módulos dentro de um Projeto através dos respectivos comandos na Barra de Ferramentas ou de Menu do Visual BASIC. Para criar

novos Controles Extras, entretanto, faz-se necessário utilizar programas que trabalhem com a linguagem C, dentro do Sistema MS-Windows.

Também é possível adicionar, salvar ou remover os arquivos de um Arquivo de Projeto através dos respectivos comandos na Barra de Ferramentas ou de Menu do Visual BASIC.

Quando o Projeto está pronto, é possível criar o Aplicativo através da opção de Menu “File” e “Make EXE File”. O Aplicativo pode ser então executado a partir do Gerenciador de Programas ou do Gerenciador de Arquivos.

### **A.7 - Formulários**

O Formulário é uma interface gráfica do Aplicativo. Pode conter um número limitado de Controles (até 470). O Formulário controla a relação entre estes controles.

Algumas Propriedades do Formulário definem sua aparência, como Width (Largura), Height (Altura), BackColor (Cor do Fundo), e outros. Outras controlam o seu comportamento, como Visible (Visível), WindowState (Estado da Janela), que pode ser Normal, Maximizado (tela cheia) ou Minimizado (reduzido a um ícone).

Pode ser associado a cada Formulário uma Barra de Menu, com diversas opções hierarquizadas na forma de submenus.

Um tipo especial de Formulário é a Interface de Múltiplos Documentos (MDI), uma Janela que controla todas as outras do Aplicativo. Só pode ser criado um Formulário MDI por Projeto. Sempre que se utiliza diversos Formulários simultaneamente é interessante criar um MDI.

**A.8 - Controles**

Na tabela a seguir apresentamos alguns dos Controles do Visual BASIC, versão 3.0, com suas funções.

<b>Nome</b>	<b>Tradução</b>	<b>Função</b>
Picture Box	Caixa de Figura	Fornece uma área para criar e editar uma figura.
Image	Imagem	Armazena e mostra uma figura, mas não permite modificações, exceto via <i>software</i> .
Text Box	Caixa de Texto	Fornece uma área para criar e editar um texto.
Label	Etiqueta	Apresenta um texto, que só pode ser alterado via <i>software</i> .
Command Button	Botão de Comando	Emula um Botão que realiza um determinado comando, quando acionado.
Check Box	Caixa de Verificação	Apresenta uma Opção tipo Sim ou Não.
Option Button	Botão de Opção	Apresenta uma Opção dentro de um conjunto de opções, onde apenas uma pode ser escolhida.
List Box	Caixa de Listagem	Apresenta uma lista de itens que o usuário pode escolher.
Combo Box	Caixa de Combinação	Combina uma Caixa de Texto com uma Caixa de Listagem.
Horizontal Scroll Bar	Barra de Rolagem Horizontal	Permite ao Usuário escolher um Valor dentro de uma faixa de valores.
Data Control	Controle de Dados	Permite o acesso a dados de uma Tabela de uma Base de Dados.
Common Dialog Box	Caixa de Diálogo	Invisível para o Usuário, permite a interface com o Sistema para abrir e salvar Arquivos, acessar Arquivo de Ajuda, e outras ações.
Timer	<i>Timer</i>	Invisível para o Usuário, realiza uma rotina de programação em intervalos de tempo previamente definidos

Tabela 7 - Os Controles do Visual BASIC Usados neste Trabalho

Alguns Controles Adicionais incorporados ao Visual BASIC, em arquivos tipo VBX, foram utilizados neste trabalho. O Botão de Comando 3D permite variações em sua altura, como se fosse um *push-button* de várias posições. O Controle Animação permite que várias imagens sejam mostradas num mesmo Objeto.

### **A.9 - Programação: Eventos, Módulos e Seqüência de Execução**

A cada Objeto é associado um conjunto de Eventos, como o Click (acionamento) do Botão de Comando, ou o Change (mudança de conteúdo) da Caixa de Texto. Uma rotina de programação pode ser associada a cada Evento, sendo executada quando o evento ocorre.

Algumas rotinas podem ser compartilhadas por diversos Formulários. Elas podem formar uma biblioteca de rotinas num arquivo do tipo Módulo. Declarações de variáveis e constantes também podem ser compartilhadas.

Um Evento pode provocar outro simultaneamente. Ou pode ocorrer numa rotina de programação de outro Evento, eventualmente de outro Objeto. Por isso, é importante o Programador tomar cuidado com a ordem de execução das rotinas, para evitar resultados inesperados.

### **A.10 - Distribuindo Programas em Visual BASIC 3.0**

Os Aplicativos desenvolvidos em Visual BASIC não podem ser executados se alguns outros arquivos não estiverem disponíveis no diretório SYSTEM do Windows ou no diretório do próprio Aplicativo.

O requisito mínimo para um Aplicativo, neste caso, é a presença da Biblioteca VBRUN300.DLL, que contém a descrição dos Objetos do Visual BASIC. Caso se acrescente algum Controle Extra, devem ser adicionados os Arquivos VBX e DLL relativos a este Objeto.

Se for necessário trabalhar com Acesso a Bases de Dados ou Troca Dinâmica de Dados (DDE), são necessárias as respectivas Bibliotecas DLL. Caso algum Arquivo de Ajuda seja vinculado ao Aplicativo, ele deverá estar disponível no mesmo diretório.

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

O Programador deve tomar o cuidado de tornar disponíveis todos estes arquivos ao Usuário ao distribuir seu Aplicativo.

## **ANEXO B - DESENVOLVIMENTO DE ARQUIVOS DE AJUDA**

### **B.1 - O Gerenciador da Ajuda do MS-Windows**

O Gerenciador da Ajuda do ambiente MS-Windows (o programa WinHelp.EXE) é um aplicativo que permite a visualização dos Arquivos de Ajuda (aqueles com a extensão HLP). Estes arquivos podem possuir diversos recursos de hipermídia, como figuras, hipergráficos, sons.

Nos Arquivos de Ajuda, as expressões que indicam que um determinado assunto é tema de outra página do hipertexto são indicadas por cor diferenciada (o verde escuro é o padrão) e por um sublinhado. O sublinhado cheio indica transferência para a outra página. O sublinhado pontilhado indica que a outra página será apresentada como uma Caixa de Mensagem.

O formulário principal do Gerenciador da Ajuda possui um conjunto de Botões de Comando em sua porção superior com funções especiais:

- a) Conteúdo - apresenta a página principal do Arquivo de Ajuda;
- b) Localizar - abre um formulário contendo uma lista de Palavras-Chaves e uma lista dos Títulos de cada Palavra-Chave, permitindo o acesso a uma página selecionada;
- c) Histórico - abre um formulário contendo os Títulos das páginas já apresentadas;
- d) Voltar - retorna à última página apresentada.

É opcional para o desenvolvedor do Arquivo de Ajuda a inclusão dos Botões “>>” (Avançar) e “<<” (Retornar), que permitem apresentar as páginas do hipertexto em uma Seqüência, tal qual um livro real.

## **B.2 - Requisitos**

Para o Desenvolvimento de um Arquivo de Ajuda no Formato WinHelp, são necessários:

- 1) O Compilador da Ajuda (HC.EXE - Help Compiler), disponível em domínio público na Rede Mundial de Computadores INTERNET;
- 2) Um Processador de Textos que permita a edição de documentos no formato RTF, como o Microsoft Word ou o WordPerfect.

São opcionais:

- 1) Um Editor de Figuras do tipo *bitmap*, como o PaintBrush, por exemplo;
- 2) O Programa SHED, de criação de Hipergráficos, disponível no *pacote* do Visual BASIC, versão 3.0.

As figuras devem ser do tipo *bitmap*, pois o Compilador de Ajuda não aceita figuras dos tipos *vetoriais* e *postscript*.

## **B.3 - O Processo de Criação e Compilação**

O processo de criação do Arquivo de Ajuda é constituído dos seguintes passos:

- a) edição das figuras (opcional);
- b) edição dos hipergráficos, a partir das figuras seleccionadas (opcional);
- c) edição do texto do Arquivo de Tópicos (obrigatório);
- d) colocação das figuras e hipergráficos no Arquivo de Tópicos (opcional);
- e) edição do Arquivo de Projeto (obrigatório);
- f) execução do programa Compilador da Ajuda (obrigatório).

O fluxo entre as etapas listadas acima pode ser retornado em qualquer uma delas, quando for necessário.

#### **B.4 - Criação e Edição de Figuras**

As figuras a serem apresentadas no Arquivo de Ajuda podem ser criadas diretamente no editor de figuras ou obtidas através de dispositivos digitalizadores de imagens, os *scanners*, e ajustadas no editor.

Como o Compilador da Ajuda só pode processar arquivos de figuras no formato *bitmap*, as figuras nos formatos vetoriais e *postscript* devem ser convertidas em *bitmaps*, o que pode ser feito em alguns editores de imagens, como o PhotoStyler ou o PhotoShop, na opção de menu “Arquivo” - “Salvar Como...”.

#### **B.5 - Criação de Hipergráficos**

Através do aplicativo Editor de Hipergráficos (SHED.EXE), é possível criar e modificar hipergráficos a partir de figuras do tipo *bitmap*.

Para isso é necessário:

- a) abrir a figura no editor, através das opções de menu “Arquivo” - “Abrir”;
- b) marcar a região onde se deseja estabelecer um vínculo, arrastando o *mouse* com o botão esquerdo pressionado;
- c) escrever a expressão que identifica a página e selecionar o tipo de vínculo (Página, Caixa de Mensagem ou Macro), através das opções de menu “Editar” - “Atributos”; e
- d) salvar o hipergráfico com a extensão SHG.

Hipergráficos só podem ser acrescentados ao Arquivo de Tópicos através de vinculação.

## **B.6 - O Arquivo de Tópicos**

Um arquivo contendo o texto a ser exibido no Arquivo de Ajuda deve ser criado no formato RTF (Rich Text Format).

Após a edição do texto, a criação de referências às páginas (vínculos, palavras-chaves) pode ser feita através da introdução de notas-de-rodapé especiais no início de cada página. A nota-de-rodapé “#” é obrigatória e indica a expressão através da qual se estabelece o vínculo de outras páginas com a atual. A nota “\$” indica o Título da página, através do qual ela é conhecida nas janelas Histórico e Localizar. A nota “k” indica uma lista de palavras-chaves para a página, separadas por “;” (ponto-e-vírgula). A nota “+” indica que a página pertence a uma Seqüência. A nota “!” (ponto de exclamação) inicia a execução de uma Macro.

As palavras do texto podem ser transformadas em vínculos através do Sublinhado. O Sublinhado Simples estabelece um vínculo que abre uma página como Caixa de Mensagem. O Sublinhado Duplo cria um vínculo que transfere para outra página do Hipertexto. Em ambos os casos a expressão que identifica a página (aquela da nota-de-rodapé “#”) deve ser escrita logo após o texto sublinhado (sem espaços), em formato Oculto. Caso se deseje executar uma Macro, antes do nome da mesma deve ser colocado um ponto de exclamação (!), também no formato Oculto.

## **B.7 - Macros**

Macros são rotinas padronizadas de programação para o Gerenciador de Ajuda. Por exemplo, pode-se executar um outro aplicativo do ambiente Windows, ou abrir outro Arquivo de Ajuda, através de uma Macro. O Manual do Compilador de Ajuda contém a

listagem completa de Macros que podem ser utilizadas na compilação de um Arquivo de Ajuda.

As Macros podem ser chamadas de três formas:

- a) colocadas no Arquivo de Projeto, na Seção CONFIG, sendo executadas quando se abre o Arquivo de Ajuda;
- b) colocadas no Arquivo de Tópicos, numa nota-de-rodapé “!”, sendo executadas na apresentação da página que contém a nota; ou
- c) também no Arquivo de Tópicos, após uma expressão Sublinhada, sendo executada quando o usuário aciona o botão esquerdo do *mouse*, com o cursor sobre esta expressão no Arquivo de Ajuda.

### **B.8 - Acrescentando Figuras**

As figuras do tipo *bitmap* podem ser acrescentadas ao Arquivo de Tópicos de dois modos: inserção e vinculação.

Para inserir uma figura, usa-se a seqüência de opções de menu “Inserir” - “Figura”. Embora torne o processo de compilação muito mais rápido, a inserção apresenta duas desvantagens importantes: aumenta o tamanho do Arquivo de Tópicos e não transfere automaticamente modificações da figura original para a figura inserida.

O vínculo entre o Arquivo de Tópicos e uma figura é criado utilizando-se uma palavra especial (bmc, bml ou bmr), seguida do nome do arquivo, entre chaves({ }). BMC coloca a figura como se fosse um caractere, BML coloca-a à esquerda do texto e BMR, á direita. Exemplo: “{bml fig.shg}” coloca o hipergráfico “fig” à esquerda do texto.

Modificações de figuras vinculadas desta forma são introduzidas no Arquivo de Ajuda na compilação seguinte à modificação.

### **B.9 - O Arquivo de Projeto**

O Arquivo de Projeto é um arquivo de texto que deve ser criado com a extensão PRJ. Este arquivo contém as informações que o compilador analisa para o processo de criação do Arquivo de Ajuda.

Este arquivo é composto por Seções, estabelecidas por palavras especiais, entre colchetes ([ ]). As Seções de um Arquivo de Projeto são:

- a) FILES - lista dos Arquivos de Tópicos a serem compilados (obrigatória);
- b) OPTIONS - diretrizes para a compilação:
  - b.1) BMROOT - especifica o (s) diretório (s) que contém as figuras a serem acrescentadas ao Arquivo de Ajuda, por vinculação;
  - b.2) COMPRESS - indica se o programador deseja realizar a compressão do Arquivo de Ajuda durante sua criação
  - b.3) COPYRIGHT - especifica uma mensagem contendo os autores do Arquivo de Ajuda;
  - b.4) ERRORLOG - especifica um arquivo para receber as mensagens de erro durante a compilação;
  - b.5) REPORT - indica se as mensagens de erro serão apresentadas na tela durante a execução (ON) ou não (OFF);
  - b.6) WARNING - indica o nível de erros que o Compilador apresentará na tela durante sua execução (de 1 - apenas os mais graves - até 3 - todos);

- b.7) TITLE - indica a expressão que identifica o Arquivo de Ajuda no formulário do Gerenciador de Ajuda.
- c) BITMAPS - lista das figuras a serem acrescentadas, por vinculação, ao Arquivo de Ajuda, devendo estar no mesmo diretório do Arquivo de Projeto (recomenda-se substituir por CONFIG - BMROOT, caso o número de figuras seja grande);
- d) MAP - tabela que associa números às expressões que identificam as páginas do Arquivo de Tópicos, para serem utilizados na propriedade HelpContextID de objetos de aplicativos criados em Visual BASIC, Access ou outros;
- e) ALIAS - tabela que associa outras expressões àquelas que identificam as páginas do Arquivo de Tópicos, permitindo que estas sejam substituídas pelas primeiras;
- f) CONFIG - lista de Macros a serem executadas na abertura do Arquivo de Ajuda (por exemplo: BrowseButtons(), acrescenta os botões de Sequência ao Gerenciador de Ajuda);
- g) WINDOWS - especifica o tamanho dos formulários do Gerenciador de Ajuda.

### **B.10 - Execução da Compilação**

O Compilador de Ajuda deve ser executado no ambiente DOS, podendo ser chamado através de:

```
C:> HC NOME_DO_ARQUIVO.PRJ
```

Durante a compilação podem ocorrer erros, com os quais estão associadas diversas mensagens do compilador. A lista completa de mensagens de erro do Compilador está disponível no Manual do Compilador de Ajuda [Ref]. Os erros devem ser corrigidos no ambiente Windows antes da nova tentativa de compilação.

## ANEXO C - RELATÓRIOS DOS TESTES DE SIMULAÇÃO

### Relatório de Treinamento

Tecnico: William

Abertura: 9/12/1998 19:40:37

Descrição do Defeito: Sem Sinal, com Mensagem "Eletrodo Solto", exceto Derivação II.

Componente Defeituoso: ContatoLA; Falha: Aberto

Inicializando:

Osciloscópio: Escala de Tensão = 100

Osciloscópio: Escala de Tempo = ,1

Derivação = II

Consulta Ordem de Serviço

Ações Efetuadas:

Liga Chave

Desliga Chave

Conecta Cabo de Paciente

Liga Osciloscópio

Liga Chave

Conecta Cabo de Alimentação

Simulador: Pulso = 60

Simulador: Pulso = 120

Simulador: Pulso = 60

Derivação = II

Derivação = I

Derivação = III

Derivação = I

Derivação = II

Desconecta Cabo de Alimentação

Verificando Cabo: RabichoRA

Multímetro: kOhm = ,000

Verificando Cabo: RabichoLA

Multímetro: kOhm = ,000

Verificando Cabo: RabichoLL

Multímetro: kOhm = ,000

Verificando Cabo: RabichoLA

Verificando Resistor: R02

Multímetro: kOhm = 100

Multímetro: Vcc = ,000

Verificando Cabo: CaboLA

Multímetro: kOhm = ,000

Verificando Conector: ContatoLA

Multímetro: kOhm = +EEE

Substituição Correta: Conector: ContatoLA

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

Encerramento: 9/12/1998 19:50:48

Consulta Ordem de Serviço  
Conecta Cabo de Alimentação  
Derivação = I  
Derivação = III  
Desliga Chave

### **Relatório de Treinamento**

Tecnico: William  
Abertura: 9/12/1998 19:53:45  
Descrição do Defeito: Não Liga.  
Componente Defeituoso: T3-Bob.P1; Falha: Aberto

Inicializando:

Osciloscópio: Escala de Tensão = 100  
Osciloscópio: Escala de Tempo = ,1  
Derivação = II  
Consulta Ordem de Serviço

Ações Efetuadas:

Conecta Cabo de Alimentação  
Conecta Cabo de Paciente  
Liga Chave  
Liga Osciloscópio  
Desliga Osciloscópio  
Desliga Chave  
Verificando Tensão: Tom  
Multímetro: Vca = 125  
Verificando Tensão: Prim  
Verificando Disjuntor: Disj01  
Desconecta Cabo de Alimentação  
Verificando Cabo: CaboF  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboT  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboN  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoF  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoN  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoT  
Multímetro: kOhm = ,000  
Verificando Bobina: T3-Bob.P1  
Multímetro: kOhm = +EEE  
Substituição Correta: Bobina: T3-Bob.P1

Encerramento: 9/12/1998 19:57:50

Consulta Ordem de Serviço  
Conecta Cabo de Alimentação  
Liga Chave  
Simulador: Pulso = 60

### **Relatório de Treinamento**

Tecnico: Mauro  
Abertura: 9/12/1998 20:01:12  
Descrição do Defeito: Não Liga.  
Componente Defeituoso: ContatoF; Falha: Aberto

#### Inicializando:

Osciloscópio: Escala de Tensão = 100  
Osciloscópio: Escala de Tempo = ,1  
Derivação = II  
Consulta Ordem de Serviço

#### Ações Efetuadas:

Verificando Cabo: CaboF  
Multímetro: kOhm = ,000  
Verificando Disjuntor: Disj01  
Multímetro: kOhm = ,000  
Verificando Chave: Chv01  
Multímetro: kOhm = +EEE  
Liga Chave  
Multímetro: kOhm = ,000  
Verificando Bobina: T3-Bob.P1  
Multímetro: kOhm = ,000  
Verificando Bobina: T3-Bob.S1  
Multímetro: kOhm = ,000  
Verificando Bobina: T3-Bob.S2  
Multímetro: kOhm = ,000  
Verificando Diodo: D24  
Multímetro: D = ,600  
Verificando Diodo: D26  
Multímetro: D = ,600  
Verificando Diodo: D25  
Multímetro: D = ,600  
Verificando Diodo: D27  
Multímetro: D = ,600  
Verificando Capacitor: C18  
Multímetro: 10nF = 220  
Verificando Capacitor: C19  
Multímetro: 10nF = 220  
Verificando Conector: ContatoF  
Multímetro: kOhm = +EEE  
Verificando Conector: ContatoN  
Multímetro: kOhm = ,000

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

Verificando Conector: ContatoT  
Verificando Conector: ContatoF  
Multímetro: kOhm = +EEE  
Substituição Correta: Conector: ContatoF

Encerramento: 9/12/1998 20:07:07

Consulta Ordem de Serviço  
Conecta Cabo de Alimentação  
Conecta Cabo de Paciente  
Simulador: Pulso = 60  
Simulador: Pulso = 0

### **Relatório de Treinamento**

Tecnico: Mauro  
Abertura: 9/12/1998 20:10:39  
Descrição do Defeito: Sem Sinal, com Mensagem "Eletrodo Solto", exceto Derivação I.  
Componente Defeituoso: RabichoLL; Falha: Aberto

Inicializando:

Osciloscópio: Escala de Tensão = 100  
Osciloscópio: Escala de Tempo = ,1  
Derivação = II  
Consulta Ordem de Serviço

Ações Efetuadas:

Verificando Ponto: LL  
Verificando Cabo: RabichoLL  
Multímetro: kOhm = +EEE  
Verificando Cabo: RabichoLA  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoLL  
Multímetro: kOhm = +EEE  
Substituição Correta: Cabo: RabichoLL

Encerramento: 9/12/1998 20:14:26

Consulta Ordem de Serviço  
Conecta Cabo de Paciente  
Simulador: Pulso = 60  
Liga Chave  
Desliga Chave  
Conecta Cabo de Alimentação  
Liga Chave

### **Relatório de Treinamento**

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

Tecnico: Rodrigo

Abertura: 9/12/1998 20:17:54

Descrição do Defeito: Sem Sinal, com Mensagem "Eletrodo Solto", exceto Derivação III.

Componente Defeituoso: R07; Falha: Aberto

Inicializando:

Osciloscópio: Escala de Tensão = 100

Osciloscópio: Escala de Tempo = ,1

Derivação = II

Consulta Ordem de Serviço

Ações Efetuadas:

Consulta Ordem de Serviço

Verificando Cabo: CaboRA

Liga Osciloscópio

Desliga Osciloscópio

Multímetro: kOhm = ,000

Verificando Cabo: CaboLA

Multímetro: kOhm = ,000

Verificando Cabo: CaboLL

Multímetro: kOhm = ,000

Verificando Cabo: CaboRL

Multímetro: kOhm = ,000

Verificando Cabo: CaboVn

Multímetro: kOhm = ,000

Verificando Cabo: CaboRL

Multímetro: kOhm = ,000

Verificando Cabo: CaboLL

Multímetro: kOhm = ,000

Verificando Cabo: CaboLA

Multímetro: kOhm = ,000

Verificando Cabo: RabichoRA

Multímetro: kOhm = ,000

Verificando Cabo: RabichoLA

Multímetro: kOhm = ,000

Verificando Cabo: RabichoLL

Multímetro: kOhm = ,000

Verificando Cabo: RabichoRL

Multímetro: kOhm = ,000

Verificando Cabo: RabichoVn

Multímetro: kOhm = ,000

Verificando Cabo: CaboVn

Multímetro: kOhm = ,000

Consulta Ordem de Serviço

Verificando Conector: ContatoRA

Multímetro: kOhm = ,000

Verificando Conector: ContatoLA

Multímetro: kOhm = ,000

Verificando Conector: ContatoLA

Verificando Conector: ContatoLL

Multímetro: kOhm = ,000

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

Verificando Conector: ContatoRL  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoTF  
Multímetro: kOhm = ,000  
Verificando Resistor: R11  
Verificando Resistor: R17  
Multímetro: kOhm = 30,0  
Verificando Resistor: R24  
Multímetro: kOhm = 30,0  
Verificando Resistor: R18  
Multímetro: kOhm = 30,0  
Verificando Resistor: R17  
Multímetro: kOhm = 30,0  
Verificando Resistor: R21  
Multímetro: kOhm = 30,0  
Verificando Resistor: R22  
Multímetro: kOhm = 30,0  
Verificando Resistor: R25  
Multímetro: kOhm = 30,0  
Verificando Resistor: R19  
Multímetro: kOhm = 30,0  
Verificando Resistor: R23  
Multímetro: kOhm = 30,0  
Verificando Resistor: R20  
Multímetro: kOhm = 30,0  
Verificando CI: CAnalog02  
Verificando Ponto: W  
Verificando Ponto: aF  
Verificando Ponto: aL  
Consulta Ordem de Serviço  
Verificando Conector: ContatoVn  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoRA  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoLA  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoLL  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoLL  
Verificando Conector: ContatoRL  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoRL  
Verificando Conector: ContatoRL  
Verificando Conector: ContatoRL  
Verificando Conector: ContatoTF  
Multímetro: kOhm = ,000  
Liga Osciloscópio  
Desliga Osciloscópio  
Liga Osciloscópio  
Desliga Osciloscópio  
Verificando Ponto: RA  
Verificando Ponto: LA  
Verificando Ponto: LL  
Verificando CI: CAnalog02  
Verificando CI: CAnalog01

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

Verificando CI: CAnalog02  
Verificando Ponto: V  
Verificando Conector: ContatoVn  
Verificando Cabo: CaboRA  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboLA  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboLL  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboRL  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoRA  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoLA  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoLL  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoRA  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoLA  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoLL  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoRL  
Multímetro: kOhm = ,000  
Verificando Cabo: RabichoVn  
Multímetro: kOhm = ,000  
Verificando Resistor: R05  
Multímetro: kOhm = 100  
Verificando Resistor: R04  
Multímetro: kOhm = 100  
Verificando Resistor: R03  
Multímetro: kOhm = 100  
Verificando Resistor: R02  
Multímetro: kOhm = 100  
Verificando Resistor: R01  
Multímetro: kOhm = 100  
Verificando Resistor: R02  
Multímetro: kOhm = 100  
Verificando Cabo: CaboLL  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboRL  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboLA  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboRA  
Multímetro: kOhm = ,000  
Verificando Cabo: CaboLA  
Multímetro: kOhm = ,000  
Verificando Resistor: R06  
Verificando Conector: ContatoRA  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoLA  
Multímetro: kOhm = ,000  
Verificando Conector: ContatoLL

## DISSERTAÇÃO DE MESTRADO - GUILHERME ARAUJO WOOD

Multímetro: kOhm = ,000  
Verificando Conector: ContatoRL  
Multímetro: kOhm = ,000  
Verificando Resistor: R09  
Multímetro: kOhm = 100  
Verificando Resistor: R15  
Multímetro: kOhm = +EEE  
Verificando Resistor: R06  
Multímetro: kOhm = 100  
Verificando Resistor: R09  
Multímetro: kOhm = 100  
Verificando Resistor: R08  
Multímetro: kOhm = 100  
Verificando Resistor: R07  
Multímetro: kOhm = +EEE  
Substituição Correta: Resistor: R07

Encerramento: 9/12/1998 20:53:09

Consulta Ordem de Serviço