

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIA E TECNOLOGIA

CURSO DE MESTRADO EM ENGENHARIA ELÉTRICA

*SISTEMA DE AQUISIÇÃO E
ANÁLISE DE DADOS EÓLICOS*

CLÁUDIO AFONSO FLEURY

Campina Grande - PB
Julho - 1994

SISTEMA DE AQUISIÇÃO E ANÁLISE DE DADOS EÓLICOS

CLÁUDIO AFONSO FLEURY

Dissertação apresentada à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do grau de Mestre.

Área de Concentração: Processamento da Informação / Automação

Misael Elias de Moraes, Dr. -Ing.
Orientador

Francisco José Simões, Ph.D.
Co-orientador

Campina Grande - PB
Julho - 1994



F618s Fleury, Cláudio Afonso.
Sistema de aquisição e análise de dados eólicos /
Cláudio Afonso Fleury. - Campina Grande, 1994.
99 f.

Dissertação (Mestrado em Engenharia Elétrica) -
Universidade Federal da Paraíba, Centro de Ciências e
Tecnologia, 1994.
"Orientação : Prof. Dr. Misael Elias de Moraes, Prof.
Dr. Francisco José Simões".".
Referências.

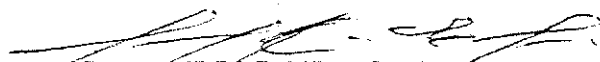
1. Energia Eólica. 2. Climatologia - Dados Eólicos. 3.
Agrometeorologia. 4. Dissertação - Engenharia Elétrica. I.
Moraes, Misael Elias de. II. Simões, Francisco José. III.
Universidade Federal da Paraíba - Campina Grande (PB). IV.
Título

CDU 621.548(043)

SISTEMA DE AQUISIÇÃO E ANÁLISE DE DADOS EÓLICOS

CLÁUDIO AFONSO FLEURY

Dissertação Aprovada em 07.07.1994



MISAEEL ELIAS DE MORAIS, Dr.-Ing, UFPB

Orientador



FRANCISCO JOSÉ SIMÕES, Ph.D., UFPB

Orientador



RAIMUNDO CARLOS SILVÉRIO FREIRE, Dr., UFPB

Componente da Banca



TANTRAVAHÍ VENKATA RAMANA RAO, Ph.D., UFPB

Componente da Banca

CAMPINA GRANDE - PB

Julho - 1994

DEDICATÓRIA

Aos meus pais Ademício e Ana Maria, pelo amor, dedicação e confiança dispensados a todos os seus filhos igualmente, e as minhas irmãs Adeane e Kênya pelo incentivo e carinho.

À minha esposa Ana Cláudia, por seu apoio, paciência e carinho para comigo e para com nosso filho Alexandre.

E à memória do meu filho Alexandre, que de onde estiver, sabe que eu gostaria de ter dedicado a ele muito mais que este trabalho, — a **vida**.

AGRADECIMENTOS

Ao encerrar os trabalhos desta dissertação, tenho o prazer de registrar alguns agradecimentos especiais a todos os que, de forma direta ou indireta, acabaram por auxiliar, influenciar, opinar e corrigir o trabalho presente durante todo o tempo de curso do Mestrado.

Ao Prof^o Misael Elias de Moraes, pela orientação deste trabalho e apoio nos momentos mais difíceis.

Ao Prof^o Francisco José Simões, por sua paciência e boa vontade na transmissão dos conceitos de energia eólica, e acompanhamento contínuo dos trabalhos.

Ao companheiro de todas as horas Prof^o Ilton Luiz Barbacena (ETFGO), por sua inestimável amizade, e à sua digníssima família pelos momentos bons que passamos.

À Escola Técnica Federal de Goiás, particularmente nas pessoas do Diretor Prof^o José Alves e de seus assessores Prof^o Luiz Gonzaga e Prof^o Wagner Bento, e aos demais colegas professores do Curso de Telecomunicações que propiciaram a minha liberação.

Aos companheiros de jornada, dos quais sempre guardarei boas lembranças: Meteorologista Fernando Moreira da Silva, Prof^o Giovanni, Prof^o Paulo Cortez, Anal. de Sist. José Bezerra e família, Prof^a Laurinda, Prof^o Walinton, Eng^o Rinaldo e Maroni (LAPS), Eng^o Aléssio, Eng^o César e Eng^o Rogério.

À minha esposa amada Ana Cláudia, por sua dedicação ao lar, por suas revisões ortográficas e gramaticais, e por sua enorme força de vida, muitas vezes camuflada na fragilidade de seu ser.

RESUMO

Apresenta-se um sistema de aquisição e análise de dados eólicos com aplicação em climatologia, em estações sinóticas, em agrometeorologia, e orientado para projetos de sistemas conversores de energia eólica. Descreve-se suas características técnicas e operacionais, e o hardware/software de seus componentes: sub-sistema de amostragem e registro, e sub-sistema de processamento das amostras. O primeiro componente tem a função de coletar amostras da velocidade do vento (intensidade e direção) na região que se pretende avaliar o regime dos ventos, para os mais variados objetivos, como prever a produção anual de energia elétrica gerada por sistemas conversores de energia eólica. As amostras coletadas no campo e transportadas para o sub-sistema de processamento, são compiladas, e em sequência, realiza-se uma avaliação estatística dos dados, cujos resultados obtém-se através de relatórios impressos e gráficos em vídeo. Um protótipo do sistema proposto é implementado para a realização de testes em laboratório.

ABSTRACT

An acquisition and analysis wind data system suitable to climatology, to synoptic stations, to agrometeorology, and oriented to design of wind energy converter systems, is presented here. Its technical and operational characteristics as also its components' hardware/software are all described. The system is composed of: a sample and record sub-system, and a sample processing sub-system. The first one has the function of collecting velocity wind samples (intensity and direction) in the local where the wind regime is searched, in general to many purposes, such as preview annual electric energy production generated by wind energy converter systems. The collected samples are transported to the processing sub-system where they are compiled, and following this processing, a statistical analysis is carried out and its results are showed on printed reports and through graphics on screen. A prototype is implemented to do laboratory tests.

SUMÁRIO

RESUMO	iii
ABSTRACT	iv
SUMÁRIO	v
LISTA DE FIGURAS E TABELAS	viii
LISTA DE ABREVIATURAS E SÍMBOLOS	xi
INTRODUÇÃO	1
Comentários Iniciais	1
Motivação do Trabalho	5
Proposta do Trabalho	8
Organização do Texto	9
CAPÍTULO 1 - FUNDAMENTOS TEÓRICOS SOBRE VENTOS	10
1.1 - Energia e Potência Eólica	10
1.2 - Regime de Ventos	12
1.3 - Turbina Eólica	14
1.4 - Medição dos Ventos	17
CAPÍTULO 2 - SISTEMA PROPOSTO	18
2.1 - Descrição	18
2.2 - Características	21
2.3 - Operação	23

CAPÍTULO 3 - AQUISIÇÃO E TRANSFERÊNCIA DE DADOS	27
3.1 - Módulo Portátil de Aquisição de Dados	27
3.1.1 - Arquitetura Básica	27
3.1.2 - Considerações de Hardware	29
3.1.3 - Considerações de Software	31
3.2 - Módulo de Leitura e Gravação de Memória	34
3.2.1 - Arquitetura Básica	34
3.2.2 - Considerações de Hardware e Software	35
CAPÍTULO 4 - SOFTWARE DE ANÁLISE DE DADOS EÓLICOS	40
4.1 - Funcionalidade	40
4.2 - Metodologia de Projeto	43
4.2.1 - Cenário-Base	44
4.2.2 - Aplicação Específica	45
4.3 - Arquitetura e Estrutura de Dados	46
4.4 - Recursos Utilizados	56
4.4.1 - Para Implementação	56
4.4.2 - Para Execução	57
4.5 - Interface com o Usuário	57
4.5.1 - Opção Arquivo	58
4.5.2 - Opção Edição	59
4.5.3 - Opção Estatística	60
4.5.4 - Opção Gráfico	61
4.5.5 - Opção Turbina	61
4.5.6 - Opção Cartucho	62
CAPÍTULO 5 - TESTES E RESULTADOS	63
5.1 - Testes Preliminares	63
5.2 - Ambientes de Programação	64
5.3 - Instrumentos de Medição	68
5.4 - Material Usado	69
5.5 - Resultados	70

CONSIDERAÇÕES FINAIS	74
Sinopse do Trabalho	74
Conclusão	75
Sugestões para Continuidade do Trabalho	76
APÊNDICE A - DISTRIBUIÇÃO DE WEIBULL	78
A.1 - Introdução	78
A.2 - Formulação Matemática	79
A.3 - Estimação dos Parâmetros de Weibull	80
APÊNDICE B - DESENHOS ESQUEMÁTICOS	83
B.1 - Introdução	83
B.2 - Módulo Portátil de Aquisição de Dados	84
B.3 - Módulo de Leitura e Gravação de Memória	89
B.4 - Cartucho de Memória	92
APÊNDICE C - PROGRAMAÇÃO ORIENTADA A OBJETOS	93
C.1 - Introdução	93
C.2 - Conceitos Básicos	94
C.3 - Linguagem de Programação C++	95
REFERÊNCIAS BIBLIOGRÁFICAS	97

LISTA DE FIGURAS E TABELAS

Figuras:

Figura I.1 - Sensibilidade do Custo da Eletricidade p/ Vários Parâmetros	5
Figura 1.1 - Fluxo de um volume ($V \cdot A$) de ar por unidade de tempo	12
Figura 1.2 - Função Densidade de Probabilidade de Weibull.....	13
Figura 1.3 - Potência Extraída por um Rotor Eólico.....	14
Figura 1.4 - Coeficientes de Potência e Torque	16
Figura 2.1 - Sistema de Aquisição de Dados Eólicos - SAADE	18
Figura 2.2 - Diagrama de Blocos do SAADE	19
Figura 2.3 - Módulos do SAADE	20
Figura 2.4 - Etiqueta de Dados de Controle do Cartucho	24
Figura 2.5a - Fluxograma da Sequência de Desligamento da URD	25
Figura 2.5b - Fluxograma da Sequência de Ligação da URD	26
Figura 3.1 - Arquitetura Básica do MPAD	28
Figura 3.2 - Disposição de um Conjunto-amostra na Memória	32
Figura 3.3 - Algoritmo do Software Básico do MPAD	33
Figura 3.4 - Arquitetura Básica do MLGM	35
Figura 3.5 - Cabo de Comunicação Serial do MLGM com a UAD	36
Figura 3.6 - Formato do Bloco de Dados da Comunicação Serial	36
Figura 3.7 - Algoritmo do Software Básico do MLGM	38
Figura 4.1 - Fluxo de Dados no SAADE	41
Figura 4.2 - Formato dos Registros dos Arquivos do Sistema	42

Figura 4.3 - Classes e Hierarquia	44
Figura 4.4 - Opções do Software de Análise Estatística	45
Figura 4.5 - Arquitetura do Software de Análise	46
Figura 4.6 - Módulos de Software Auxiliares	47
Figura 4.7 - Faixas de Velocidade da Distribuição de Frequência	60
Figura 5.1a - Distribuição Temporal	66
Figura 5.1b - Distribuição de Frequência	66
Figura 5.1c - Distribuição de Frequência Acumulada	67
Figura 5.1d - Distribuição de Weibull	67
Figura 5.1e - Distribuição de Frequência Direcional (Rosa dos Ventos)	68
Figura 5.2a - Relatório de Estatística Geral	70
Figura 5.2b - Relatório de Médias Horárias (m/s)	71
Figura 5.2c - Relatório de Distribuição de Frequência	72
Figura 5.2d - Relatório de Amostras	73
Figura B.1 - Microcontrolador - MPAD	84
Figura B.2 - Memória de Código e Amostras - MPAD	85
Figura B.3 - Seleção de Dispositivos - MPAD	86
Figura B.4 - Amostrador - MPAD	87
Figura B.5 - Despertador - MPAD	88
Figura B.6 - Microcontrolador - MLGM	89
Figura B.7 - Memória de Código e Amostras - MLGM	90
Figura B.8 - Comunicação Serial - MLGM	91
Figura B.9 - Cartucho de Memória	92

Tabelas:

Tabela I.1 - Evolução do Homem e o Consumo de Energia	1
Tabela I.2 - Comparativo de Custos entre Tecnologias de Prod. de Eletric. nos EUA	4
Tabela 2.1 - Características Técnicas do SAADE	22
Tabela 2.2 - Cabeçalho de Parâmetros Gravados no Cartucho	23
Tabela 3.1 - Código dos Comandos Implementados do MLGM	39
Tabela 5.1 - Velocidades Médias Horária do Aeroporto de Praia, Junho de 1975	65
Tabela 5.2a - Componentes do Módulo MLGM	69
Tabela 5.2b - Componentes do Módulo MPAD	69
Tabela 5.2c - Componentes Usados no Protótipo	70

LISTA DE ABREVIATURAS E SÍMBOLOS

Abreviaturas:

μ C	- microcontrolador
A/D	- Analógico / Digital
AMS	- extensão do nome de arquivo de dados com amostras
ASCII	- American Standard Code for Information Interchange
BIOS	- Basic Input-Output System
CMOS	- Complementary Metal-Oxide Semiconductor
CPU	- Central Processor Unit
DIR	- Direção do vento
DOS	- Disk Operating System
EPROM	- Erasable Programmable Read-Only Memory
EEPROM	- Electrically Erasable Programmable Read-Only Memory
fdp	- função densidade de probabilidade
FDP	- Função Distribuição de Probabilidade Acumulada
GND	- Referência de sinal
HMOS	- Handling Metal-Oxide Semiconductor
I/O	- Input / Output
IDE	- Integrated Development Environment
LSB	- Least Significant Byte/Bit
MAX	- extensão do nome de arquivo de dados com velocidades máximas
MED	- extensão do nome de arquivo de dados com velocidades médias
MLGM	- Módulo de Leitura e Gravação de Memória

MPAD	- Módulo Portátil de Aquisição de Dados
MSB	- Most Significant Byte/Bit
$p()$	- função densidade de probabilidade
$P()$	- Função Distribuição de Probabilidade Acumulada
PC	- Personal Computer (microcomputador da IBM)
POO	- Programação Orientada a Objeto
RAM	- Random-Access Memory
rpm	- rotações por minuto
RS-232	- Padrão de especificações mecânicas e elétricas p/ comunicação serial
RTC	- Real Time Clock
RV	- Regime de Ventos
RXD	- Sinal/pino de recepção
SAADE	- Sistema de Aquisição e Análise de Dados Eólicos
SCEE	- Sistema Conversor de Energia Eólica
SIU	- Sistema Internacional de Unidades
STANDBY-	Espera, reserva
TTL	- Transistor-Transistor Logic
TXD	- Sinal/pino de transmissão
UAD	- Unidade de Análise de Dados
UC	- Unidade de Controle
URD	- Unidade de Registro de Dados
VGA	- Video Graphics Array
WMO	- World Meteorological Organization

Símbolos:

ρ	- densidade do ar, medida em kilogramas por metro cúbico
λ	- razão de velocidade de extremidade, adimensional
Ω	- velocidade rotacional, medida em radianos por segundo
λ_1	- razão de velocidade de torque máximo, adimensional
ω_1	- velocidade rotacional p/ máxima potência, medida em radianos por segundo
λ_2	- razão de velocidade de projeto, adimensional
ω_2	- velocidade rotacional p/ torque máximo, medida em radianos por segundo
Δt	- intervalo de tempo, medido em segundos
A	- área de uma seção transversal, medida em metros quadrado
c	- fator de escala da curva de Weibull
C_P	- coeficiente de potência, adimensional
C_Q	- coeficiente de torque, adimensional
E_c	- energia cinética, medida em Joules
k	- fator de forma da curva de Weibull
m	- massa, medida em kilogramas
P	- potência cinética, medida em Watts
$P_{m\acute{a}x}$	- potência máxima, medida em Watts
Q	- torque, medido em Newtons metro
$Q_{m\acute{a}x}$	- torque máximo, medido em Newtons metro
V	- intensidade da velocidade do vento, medida em metros por segundo
V_m	- intensidade de velocidade média, medida em metros por segundo

INTRODUÇÃO

Comentários Iniciais

A Antropologia Social, bem como outras ciências sociais contemporâneas, utilizam a quantidade de energia consumida por uma comunidade como um dos indicadores do nível de desenvolvimento sócio-econômico atingido por aquela comunidade. E verificando a relação entre o consumo de energia durante as várias fases de evolução da humanidade, podemos observar, conforme mostrado na **Tabela I. 1**, que o homem tem melhorado sua condição de vida ao lado de um crescente consumo de energia. É de se esperar portanto, que comunidades que dispõem de maior facilidade de obtenção de energia do que outras, possuam melhores condições para se desenvolverem.

São vários os fatores que propiciam o maior ou menor consumo de energia por uma determinada comunidade nos dias de hoje, dentre os quais podemos salientar: a disponibilidade de recursos e reservas energéticas na região de estabelecimento da comunidade, o domínio da tecnologia de exploração destes recursos, o armazenamento e conversão dos recursos energéticos, o uso racionalizado dos recursos não-renováveis, e a disponibilidade mercadológica de energia em suas várias formas.

Estágio de Evolução da Humanidade	Época	Consumo de Energia (kcal / dia)
Homem Primitivo	1.000.000 a.C.	2.000
Homem Nômade	1.000 a.C.	5.000
Homem Agrícola (Europa)	1.400 d.C.	26.000
Homem Industrial (Inglaterra)	1.850 d.C.	77.000
Homem Atual (E.U.A.)	1.980 d.C.	230.000

Fonte: GOLDEMBERG (1979) - Energia no Brasil

Tabela I. 1 - Evolução do Homem e o Consumo de Energia

Relacionado à obtenção do índice de qualidade de vida de uma determinada comunidade aparece também, mais recentemente, a questão ambiental, a ecologia. A comunidade global alcançou mais um estágio na caminhada para a preservação do planeta para as futuras gerações, com a realização da Conferência da Terra (Rio-92), não somente por causa da assinatura de cartas de intenções entre nações, mas principalmente pela postura mundial frente às questões preservacionistas. Muitas conclusões foram tiradas a respeito de tópicos ecológicos fundamentais, tais como o lema de desenvolvimento auto-sustentado, que preconiza entre outras coisas, o uso de fontes renováveis de energia limpa [Lopes, 1991], ou seja, fontes energéticas que não poluam o meio-ambiente em todas as suas etapas de exploração, conversão, armazenamento e uso.

A Energia Eólica, que há pelo menos dois mil anos já é explorada pelo homem, e portanto com tecnologia disponível e testada, é uma forma de energia alternativa que se apresenta, dentro do atual momento histórico, como forma alternativa adequada por ser uma energia *limpa e democrática*. Democrática porque a distribuição de ventos é bastante homogênea em quase todo o planeta conforme estudos de Putnam [Golding, 1976], ou seja, não existem grandes reservas privilegiando territórios específicos, como ocorre no caso de fontes de energia provenientes de reservas minerais (urânio) e orgânicas (petróleo, biomassa); e limpa ecologicamente porque não agride o meio-ambiente nas várias etapas de seu aproveitamento:

- i. *exploração*: feita por turbinas eólicas, que não emitem gases/substâncias poluentes;
- ii. *conversão*: através de geradores elétricos ou sistemas mecânicos de bombas hidráulicas;
- iii. *armazenamento*: obtido através de baterias ou reservatórios d'água, inexistindo o risco de desastres ecológicos como ocorre no armazenamento de combustíveis (petróleo, urânio,...);
- iv. *uso*: seja na forma de energia elétrica ou água bombeada.

Existe uma linha de estudiosos ecológicos que afirmam ser a energia eólica não tão "limpa" como querem outros cientistas. Eles assentam suas críticas principalmente na poluição visual, na poluição sonora e na interferência em telecomunicações provocadas pelas enormes pás dos aerogeradores. Mas, mesmo estes mais céticos, acreditam que se trata de uma boa alternativa para a questão energética mundial.

É preciso estarmos cientes da necessidade de outras formas de energia serem usadas como parceiras da energia eólica, uma vez que estudos já demonstraram que somente algo em torno de 2% do consumo anual mundial de energia poderia ser suprido por energia eólica cinética praticamente recuperável, ou seja, algo em torno de $3 \text{ a } 30 \times 10^{17} \text{ J/ano}$ [Silva, 1988].

Devemos também considerar nesta análise os fatores custo de instalação e custo de produção de eletricidade através dos ventos. Nas últimas décadas têm-se notado uma contínua queda nos custos nominais das turbinas eólicas e aparatos afins, e uma sensível melhoria na qualidade destes equipamentos e seus periféricos. Hoje, a energia anemoelétrica é totalmente competitiva em relação a várias fontes de geração de eletricidade [Craveiro, 1991], como mostra a **Tabela I. 2** . Dentre as fontes de energia renováveis ela é a de menor custo médio de instalação, e a terceira de menor custo médio de produção dentre àquelas citadas na **Tabela I.2**.

A energia oriunda dos ventos, é portanto, uma opção que precisa ser considerada como solução viável para se obter uma melhoria da qualidade de vida de uma determinada comunidade de modo auto-sustentado e a custos módicos.

"A energia eólica mostra-se, atualmente no Mundo, uma das formas alternativas mais promissoras..." [Silva, 1988].

Fonte de Geração	Renovável	Custo Médio de Instalação (US\$ / kW)	Custo Médio de Produção (US\$ / MWh)
Nucleoelétrica	Não	3.500,00	75,00
Hidroelétrica	Não	2.500,00	50,00
Termoelétrica Carvão-Mineral	Não	1.600,00	58,00
Termoelétrica Gás Natural	Não	850,00	55,00
Central Fotovoltaica	Sim	5.250,00	155,00
Termoelétrica Piscina Solar	Sim	3.430,00	62,00
Termoelétrica Combustão Biomassa	Sim	1.950,00	61,00
Termoelétrica Concentrador Solar	Sim	1.700,00	69,00
Termoelétrica Gaseific. Biomassa	Sim	1.570,00	51,00
Anemoelétrica (Eólica)	Sim	1.000,00	55,00

Fonte: Energy Technology Status Report - California Energy Commission (1991)

Tabela I. 2 - Comparativo de Custos entre Tecnologias de Produção de Eletricidade nos EUA

Motivação do Trabalho

A energia eólica apresenta aspectos de difusão e de aleatoriedade provocados pelo deslocamento da atmosfera. Devido a estas características, e também por questões econômicas, a estimação de recursos é a primeira etapa a ser cumprida antes da exploração de potenciais de fontes energéticas deste tipo. Diferentemente do que ocorre com as reservas de combustíveis fósseis, a quantidade de energia disponível nos ventos sofre variações a longo e curto prazos, como por exemplo, variações com as estações do ano e com a hora do dia [Justus, 1978].

A estimativa de potenciais eólicos é baseada no estudo do comportamento temporal dos ventos em uma dada região, chamado de **Regime dos Ventos (RV)**. O RV é caracterizado pela grandeza vetorial *velocidade* do vento. Essa grandeza possui comportamento dependente da atmosfera, da topografia, da aspereza do solo na região, das construções e da vegetação no local de investigação do RV [Simões, 1987].

São vários os fatores que influenciam o custo da energia produzida por um **Sistema Conversor de Energia Eólica (SCEE)**, das quais citamos: velocidade média (V_m), operação e manutenção do sistema (O&M), taxa de juros (r), disponibilidade da máquina (D), preço do capital (C/W), expectativa de vida (n), etc. Prof^o Swift-Hook, 1989, em seu curso sobre o custo de geração de energia para turbinas inglesas típicas, em localidades de muito vento, deixa claro através de seu gráfico-aranha, a alta sensibilidade do custo de produção de eletricidade através da energia eólica, com a

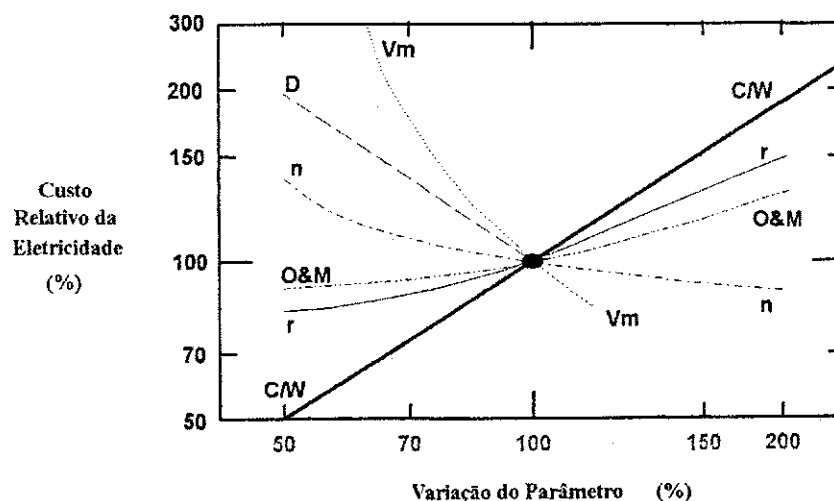


Figura I.1 - Sensibilidade do Custo da Eletricidade p/ Vários Parâmetros

variação do parâmetro Velocidade Média do Vento. Esta sensibilidade pode ser observada no gráfico-aranha mostrado na **Figura I.1**, que mostra a curva do parâmetro V_m como a mais inclinada, que leva a grandes variações do custo de produção de eletricidade para pequenas alterações na intensidade deste parâmetro. Ainda sobre o gráfico-aranha, podemos afirmar que a Velocidade Média é o parâmetro mais influente no custo final da eletricidade produzida por SCEE's.

Em geral não se pode assegurar uma locação ótima (local de maior velocidade média do vento) para instalação de um SCEE tomando-se como base apenas os dados da estação meteorológica mais próxima [Bandeira, 1990]. Um levantamento detalhado do RV na região de provável instalação do SCEE se faz imprescindível para que se possa realizar análises estatísticas fundamentadas em grandes massas de dados coletadas durante períodos de tempo consideráveis (12 ou mais meses), através de sistemas de aquisição e registro de dados do vento.

Vários desses sistemas de registro de amostras, genericamente conhecidos por **data loggers**, têm sido desenvolvidos por países com tradição na exploração de energia eólica, como os dinamarqueses, onde estão concentrados os mais importantes produtores mundiais de artefatos eólicos para "fazendas de cataventos" [Silva, 1988]. Alemães, ingleses e norte-americanos, também têm desenvolvido sistemas de alta qualidade, confiabilidade e precisão [Catálogos Técnicos].

Como o intervalo de tempo para levantamento do RV em determinada região é geralmente de um ano para a obtenção de uma boa estimativa (erro de estimação em torno de 10%, como em Lysen, 1983), realiza-se na prática a instalação de várias torres com instrumentos de medição de vento e sistemas de registro de intensidade e direção do vento, em pontos estrategicamente escolhidos na área de investigação, para que o levantamento seja realizado de forma simultânea.

A instalação de vários sistemas de aquisição de dados do vento, para a realização dessas medições, torna-se onerosa devido a necessidade de um bom número de instrumentos e equipamentos eletrônicos (dependendo da precisão desejada da estimativa e da extensão da área investigada). Este aparato, na maioria das vezes, está disponível somente nos centros mais avançados e com custo relativamente elevado, fatores primordiais na aquisição de sistemas deste tipo por países em desenvolvimento.

Visando atender estas necessidades é que se considerou a possibilidade de desenvolvimento de um sistema de medição e registro de dados de vento, que fosse de baixo custo de produção, mas que ainda mantivesse as características mínimas exigidas em um equipamento desta natureza: facilidade de operação, precisão na medição das amostras, confiabilidade no registro e na manutenção dos dados, portabilidade e versatilidade de programação da frequência de amostragem, e autonomia de tempo compatível com a natureza de levantamento de dados eólicos.

A priori era este o único propósito deste projeto, mas a necessidade de um software para realizar a redução dos dados colhidos acabaria por ser a demanda seguinte na evolução do processo. Nasceu então a proposta de elaboração de um sistema que contemplasse todos estes requisitos: coletar, registrar e reduzir dados do vento, e que atendesse a todos os requisitos citados anteriormente.

Proposta do Trabalho

O Sistema de Aquisição e Análise de Dados Eólicos (SAADE) aqui discutido tem como objetivo atender os requisitos principais de um instrumento de medição do vento. O SAADE foi concebido em duas partes:

- **aquisição de dados**, que compreende o hardware do data logger propriamente dito e o software básico para controle de suas funções de amostragem e armazenamento.
- **análise de dados e saída de avaliações**, constituída de um software desenvolvido em linguagem de alto nível, um microcomputador tipo PC e uma unidade leitora de cartucho de memória.

O sub-sistema de aquisição de dados fará amostragens em intervalos previamente programados pelo operador do sistema, e o armazenamento destas amostras se dará em memória de estado sólido (circuito integrado) que após ser preenchida com amostras será transportada manualmente para uma unidade que procederá a transferência das amostras para o microcomputador tipo PC.

No microcomputador teremos o processamento das amostras brutas (amostras em unidade não convencional: pulsos/seg.), reduzindo-as a planilhas totalizadoras de dados. Estas planilhas serão analisadas conforme os procedimentos usuais como propostos em Justus, 1978 e em Lysen, 1983.

Os resultados da análise estatística dos dados do vento poderão ser visualizados através de gráficos no vídeo do microcomputador e através de relatórios impressos variados, e poderão ser armazenados para consultas posteriores, ou poderão ser usados para avaliação de produção de energia elétrica de SCEE's comerciais, previamente cadastrados no banco de dados de turbinas, e que hipoteticamente poderiam ser instalados no local cujo RV foi investigado.

Organização do Texto

Depois desta breve introdução, o texto segue, no **Capítulo 1**, com uma revisão de alguns conceitos sobre ventos, aproveitamento de energia eólica, potência eólica, caracterização de regime de ventos e sistemas conversores de energia eólica, com o intuito de relembrar tópicos teóricos que são usados no restante da dissertação.

O sistema proposto é apresentado em linhas gerais no **Capítulo 2**, através da descrição de suas características como um todo, e também da descrição de seus dois componentes básicos: o sub-sistema de coleta dos dados e o sub-sistema de análise dos dados.

No **Capítulo 3** são mostrados, com maior detalhes, os módulos componentes do hardware do projeto. São tecidas considerações sobre a arquitetura dos módulos, e algumas observações a respeito de hardware e software básico são colocadas também. O objetivo deste capítulo é fazer a especificação técnica do sistema e discutir artificios empregados no projeto do hardware.

E no **Capítulo 4** é apresentada a etapa de análise de dados eólicos do sistema proposto, bem como o projeto do software de análise e sua operação. Neste ponto são abordadas manipulações estatísticas necessárias à redução de dados e à estimativa de produção anual de energia elétrica.

Algumas avaliações de resultados obtidos com o protótipo, montado para validação a nível de experimento em laboratório, são colocadas no **Capítulo 5**. Este capítulo tem como função principal apresentar a metodologia empregada no projeto e construção do protótipo, o material utilizado e as condições nas quais foram realizados os testes de bancada.

As **Considerações Finais** enumeram conclusões a respeito do sistema proposto e sugestões para trabalhos futuros, e fecha a dissertação. Definições básicas sobre a Função de Distribuição Cumulativa de Weibull são colocadas no **Apêndice A**, no **Apêndice B** estão os esquemas do hardware e no **Apêndice C** uma breve apresentação dos conceitos de programação orientada a objeto (O.O.P.).

CAPÍTULO 1

FUNDAMENTOS TEÓRICOS SOBRE VENTOS

1.1 - Energia e Potência Eólica:

A **energia eólica** é proveniente da transformação de parte do efeito térmico solar em energia cinética da atmosfera. Os diferentes níveis de radiação solar sobre regiões distintas do planeta provocam o deslocamento de camadas de ar, e este deslocamento é denominado **vento**.

A velocidade do vento é caracterizada por duas grandezas que variam com o tempo: **magnitude** e **direção**, de forma instantânea (rajadas), diária, e ainda sazonalmente. Além disto os ventos também variam com a localização geográfica (latitude e longitude), com a altitude, com a topografia do relevo, com a aspereza do solo e com a existência de obstáculos artificiais (construções) e naturais (vegetação).

Chamamos de **fenômeno aleatório** àquele experimento que pode ser representado por uma função, de um ou mais parâmetros, e cujo resultado é não-determinístico. A descrição de uma seção qualquer de um fenômeno aleatório é chamada de *função-amostra*, e a coleção de todas as funções amostras possíveis do fenômeno dá-se o nome de *processo estocástico*. A descrição completa de um processo estocástico é feita através da função densidade de probabilidade (fdp) conjunta do processo, mas na prática, poucos são os processos para os quais pode-se determinar a fdp conjunta. Neste tipo de processo a descrição só é feita parcialmente através da determinação de alguns momentos estatísticos [Siqueira, 1991].

As flutuações do vento constituem um fenômeno aleatório e o potencial eólico em um local pode ser estimado a partir de curvas de distribuição das velocidades do vento no tempo. Segundo os padrões da World Meteorological Organization - WMO, as medições devem ser efetuadas a uma altitude de 10 (dez) metros em relação ao nível

do solo, em intervalos de 10 (dez) minutos (proposta baseada na análise espectral da velocidade de vento realizada por Van der Hoven) [Silva, 1988].

Mesmo tendo uma baixa densidade, o ar possui massa, e quando essa massa flui com velocidade V através de uma área A , como esboçado na **Figura 1.1**, isto representará uma taxa ($m/\Delta t$) de fluxo de volume de ar ($V \cdot A$) por unidade de tempo dada por:

$$\frac{m}{\Delta t} = \rho \cdot A \cdot V \quad (\text{kg/s}) \quad (1.1.1)$$

onde: V - velocidade da massa de ar,
 A - área da secção transversal,
 ρ - massa de ar por unidade de volume,
 m - massa,
 Δt - variação de tempo

O fluxo de energia cinética (E_c) por unidade de tempo, ou potência cinética (P) da massa de ar, é função do cubo da velocidade do vento, e portanto, pequenas variações da velocidade do vento influenciam de modo contundente a geração de potência cinética:

$$\frac{E_c}{\Delta t} = P = \frac{1}{2} \cdot (\rho \cdot A \cdot V) \cdot V^2 = \frac{\rho}{2} \cdot A \cdot V^3 \quad (\text{W}) \quad (1.1.2)$$

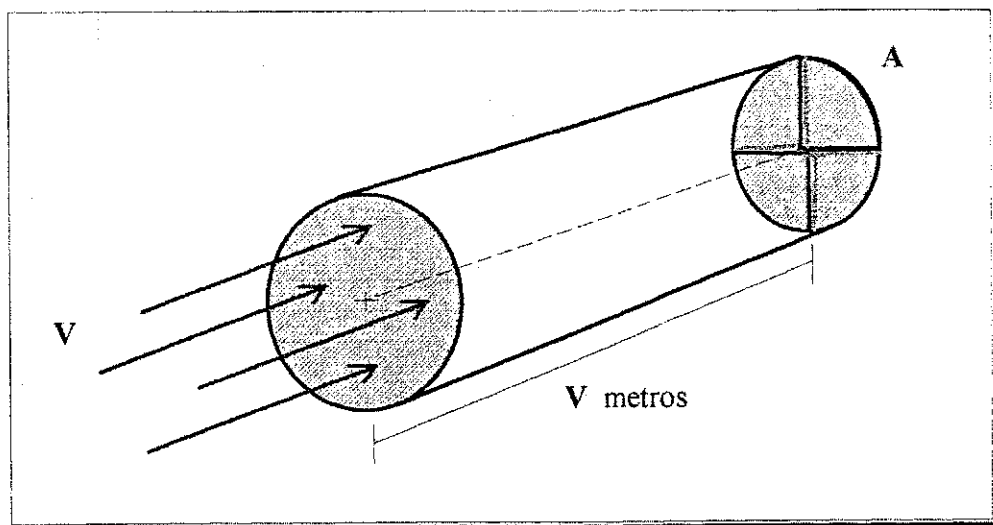


Figura 1.1 - Fluxo de um volume ($V \cdot A$) de ar por unidade de tempo.

A extração da energia eólica é normalmente realizada por um rotor acionado por vento, e Betz mostrou em 1927 que a extração máxima de potência do vento por um rotor ideal, é alcançada quando a velocidade do vento atrás do rotor é 1/3 da velocidade do vento sem perturbação (Lysen, 1983), o que corresponde a uma fração de 59,3% ou 16/27 da potência cinética total disponível no vento. Por causa das imperfeições aerodinâmicas, como ocorre em qualquer máquina real, e das perdas mecânicas e elétricas, a potência extraída na prática raramente alcança a marca de 40% da potência cinética dada pela equação 1.1.2 [Golding, 1976].

1.2 - Regime de Ventos

Quando consideramos o uso de vento para propósitos de geração de eletricidade surgem algumas questões importantes que precisam ser avaliadas para a viabilização econômica. A primeira delas seria a quantidade de energia anual que pode ser esperada; outra questão seria a distribuição do vento no tempo, durante o dia, mes ou ano, e até mesmo sobre períodos maiores com o interesse de previsão de suprimento de energia elétrica. A velocidade de rajadas é outra questão importante para o projeto estrutural de turbinas eólicas, e ainda uma outra informação relacionada à duração de calmarias e de ventos com altas velocidades torna-se necessária para estudos de armazenamento e complementação de energia elétrica via conversão de outras formas de energia.

As respostas para estas questões podem ser obtidas através da análise dos padrões de ventos e sua caracterização por números e gráficos, que em geral se define como **Regime de Ventos (RV)**.

O regime de ventos (RV) de um determinado local pode ser representado por funções de distribuição da velocidade de vento (V) em um período estipulado e por funções de distribuição de ocorrência da velocidade de vento. Esta última distribuição tem sido modelada através de vários estudos, sendo que a curva de Weibull tem se adaptado satisfatoriamente a dados experimentais com apenas dois parâmetros: o fator de forma (k) e o fator de escala (c) [Justus, 1978]. A equação da função densidade de probabilidade de Weibull é dada por:

$$f(V) = \frac{k}{c} \cdot \left(\frac{V}{c}\right)^{k-1} \cdot e^{-\left(\frac{V}{c}\right)^k} \quad (1.2.1)$$

O termo velocidade reduzida de vento (x) sempre aparece nas expressões da distribuição de Weibull, que assume a seguinte forma quando em função apenas do fator de forma (k):

$$f(x) = k \Gamma^k \left(1 + \frac{1}{k}\right) x^{k-1} \exp\left(-\Gamma^k \left(1 + \frac{1}{k}\right) x^k\right) \quad (1.2.2)$$

onde: $x = V / V_m$ é a velocidade reduzida,
 V_m é a velocidade média.

Em função do fator de forma (k), pode-se fazer uma análise rápida do comportamento da distribuição dos ventos. Como podemos ver nos gráficos plotados na **Figura 1.2**, quanto maior o valor deste parâmetro mais concentrada se encontra a distribuição dos ventos em torno da velocidade média ($k = 4$), caracterizando uma região de ventos uniformes com velocidades próximas à média.

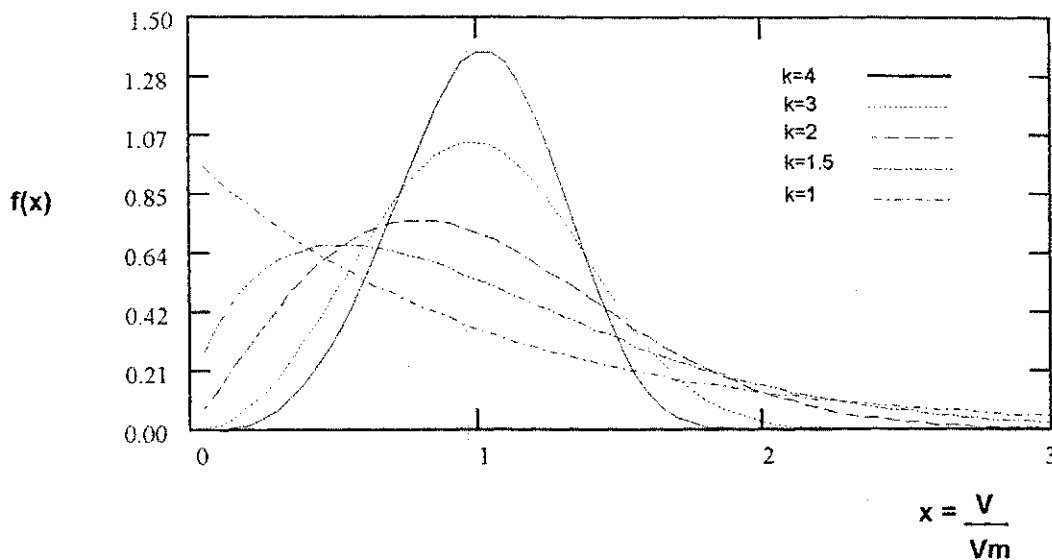


Figura 1.2 - Função Densidade de Probabilidade de Weibull

Medidas da velocidade e direção do vento efetuadas a longo prazo, usualmente a uma altura de 10 (dez) metros em relação ao solo, são feitas em quase todo o mundo por serviços de meteorologia de vários países. Estas medidas têm sido utilizadas para afirmar até que ponto uma dada localidade é indicada para a utilização da energia eólica.

1.3 - Turbina Eólica

As duas principais classes de máquinas acionadas por vento (turbinas eólicas) que podem ser distinguidas são: máquinas de eixo horizontal (eixo na direção do vento) e máquinas de eixo vertical (eixo em um plano perpendicular a direção do vento) [Golding, 1976].

Um rotor eólico pode extrair potência do vento porque ele provoca uma diminuição da velocidade do vento que atravessa suas pás, ou seja, na parte de trás do rotor a velocidade é menor do que na parte da frente. Se o rotor diminuir a velocidade demasiadamente, então a massa de ar fluirá em volta da área do rotor ao invés de atravessá-la. Estando parado, o rotor obviamente não produz potência e a altas velocidades de rotação o ar é, de certa forma, bloqueado pelo rotor e novamente nenhuma potência é produzida. Torna-se claro que entre os dois extremos existe uma velocidade rotacional ótima (ω_2), na qual a potência extraída é também máxima conforme mostra a **Figura 1.3** (para uma dada velocidade de vento) [Lysen, 1983].

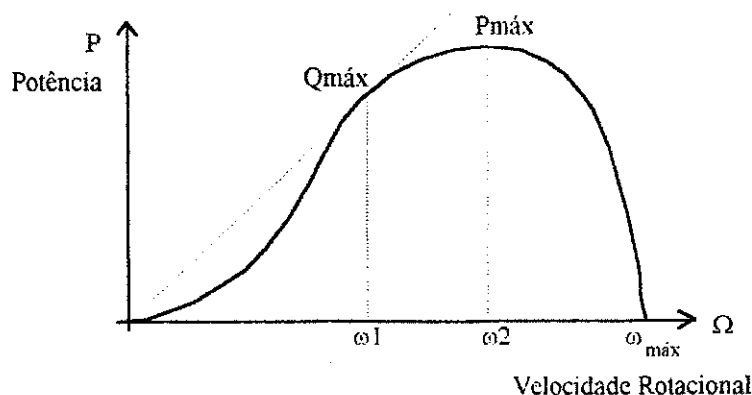


Figura 1.3 - Potência Extraída por um Rotor Eólico

Por questões de acoplamento de carga mecânica ao eixo das máquinas, é necessário conhecer também a variação do torque Q (Nm) do rotor eólico em função da velocidade rotacional Ω (rad/s). O torque se relaciona com a potência P (W) pela seguinte lei:

$$P = Q \cdot \Omega, \quad (1.3.1)$$

Portanto, podemos obter a curva *torque x velocidade rotacional* através da tangente ($Q = P / \Omega$) em cada ponto da curva *potência x velocidade rotacional*, e isto explica porque o torque máximo $Q_{\text{máx}}$ é alcançado em uma velocidade menor ($\omega 1$) do que aquela ($\omega 2$) que produz a potência máxima $P_{\text{máx}}$ conforme mostra a **Figura 1.3**. Para cada velocidade de vento diferente tem-se um par de curvas, *torque x veloc. rotacional* e *potência x veloc. rotacional*, diferentes.

Para contornar o inconveniente manuseio destas curvas que variam com a velocidade do vento, com o raio do rotor, e com a densidade do ar, é que se trabalha com coeficientes adimensionais para potência, torque e velocidade.

O coeficiente de potência C_P , o coeficiente de torque C_Q e a razão de velocidade de extremidade λ (tip speed ratio) são definidos por:

$$C_P = \frac{P}{\frac{\rho}{2} \cdot A \cdot V^3} \quad (1.3.2)$$

$$C_Q = \frac{Q}{\frac{\rho}{2} \cdot A \cdot V^2 \cdot R} \quad (1.3.3)$$

$$\lambda = \frac{\Omega \cdot R}{V} \quad (1.3.4)$$

onde: R - raio do rotor eólico

V - velocidade linear do vento

Ω - velocidade angular do rotor

Substituindo as expressões anteriores em 1.3.1 resulta:

$$C_p = C_Q \cdot \lambda \quad (1.3.5)$$

A vantagem imediata desta análise, é que o comportamento de rotores com diferentes dimensões e submetidos a diferentes velocidades, pode ser representado por apenas duas curvas: $C_p - \lambda$ e $C_Q - \lambda$ [Lysen, 1983] mostradas na **Figura 1.4**. O parâmetro λ é denominada razão de velocidade de projeto de sistemas conversores de energia eólica (SCEE's), e quando operando nesta razão, a turbina apresenta eficiência aerodinâmica máxima, retirando do vento a máxima potência que ela consegue. Quando $\lambda = \lambda_1$, a turbina apresenta torque máximo. A região estável de operação da turbina corresponde ao intervalo:

$$\lambda_1 < \lambda < \lambda_{\text{máx}}$$

onde $\lambda_{\text{máx}}$ representa a máxima razão de velocidade que a turbina pode alcançar [Lima, 1993].

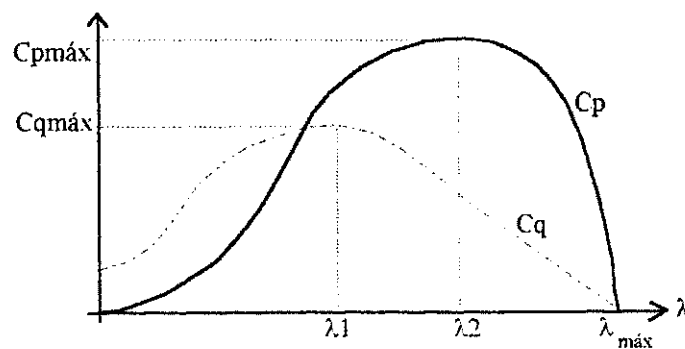


Figura 1.4 - Coeficientes de Potência e de Torque

1.4 - Medição dos Ventos:

Uma corrente de ar coloca em movimento sistemas susceptíveis de rotação sobre um eixo, produz um aumento ou diminuição de pressão em diferentes porções de um corpo fixo, esfria superfícies com temperaturas mais altas por convecção, acelera evaporação, emite sons musicais de diferentes tons e intensidades de aparelho sensível a vibrações; permitindo várias formas de obtenção das características dos ventos [Golding, 1976].

Todos estes métodos têm sido usados, embora alguns tenham se tornado mais comuns que outros. Em termos de trabalhos de pesquisa os instrumentos baseados no primeiro efeito (rotação sobre eixo), usado em conjunção com várias formas de registro, são os mais desenvolvidos. Nesta classe se encaixa o anemômetro rotacional de conchas (cup), que é o instrumento mais usado para medição da velocidade de vento no mundo todo, e também, é o tipo empregado neste projeto.

A despeito de sua aparente simplicidade, sua teoria aerodinâmica é complexa e fórmulas para o relacionamento entre a velocidade de rotação e a velocidade do vento, sob condições de variação, continuam sendo empíricas.

Antes do uso e periodicamente (pelo menos uma vez por ano) quando em operação, um anemômetro deve ser calibrado em túnel de vento. Isto é feito pela observação da velocidade de rotação indicada contra velocidades conhecidas do túnel de ventos, e avaliação da relação linear que melhor se adapte às duas velocidades. É difícil determinar com precisão, em testes no túnel de ventos, a velocidade inicial de disparo do movimento das conchas, mas valores típicos estão geralmente na faixa de 0,6 a 1,2 m/s [Justus, 1978].

Se o número inteiro de voltas de um anemômetro de conchas é tomado dentro de um período médio de tempo desejado (ex. 1 minuto), e a direção é a média sobre o mesmo período, então a velocidade efetiva para aquele intervalo será superestimada se for assumido que a velocidade medida represente o fluxo médio na direção média. Um método de estimação do erro de processamento dos dados na forma citada anteriormente, é apresentado por Justus, 1978.

CAPÍTULO 2

SISTEMA PROPOSTO

Descreveremos o sistema de aquisição e análise de dados eólicos proposto neste trabalho, através do esboço de sua composição, da apresentação de suas características e de sua operação.

2.1 - Descrição:

Com o propósito de elucidação, dividimos o SAADE em dois sub-sistemas, a saber: um sub-sistema coletor de dados (Unidade de Registro de Dados - URD) e um outro sub-sistema processador de dados (Unidade de Análise de Dados - UAD). Na **Figura 2.1** temos um esquema do SAADE e na **Figura 2.2** o SAADE é mostrado em diagrama de blocos.

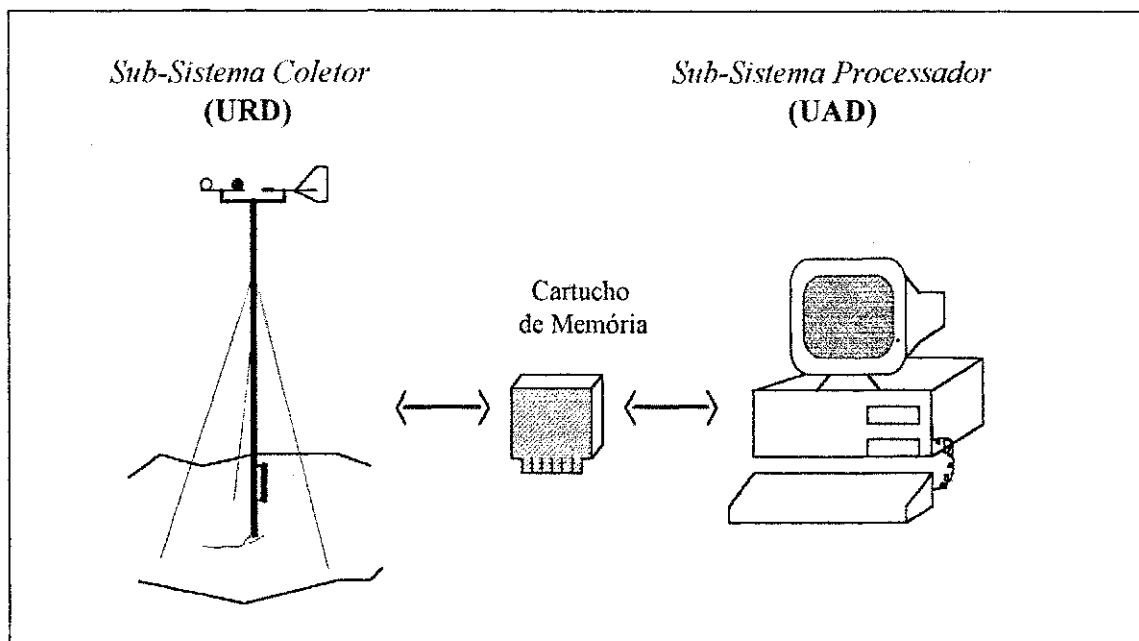


Figura 2.1 - Sistema de Aquisição e Análise de Dados Eólicos - SAADE

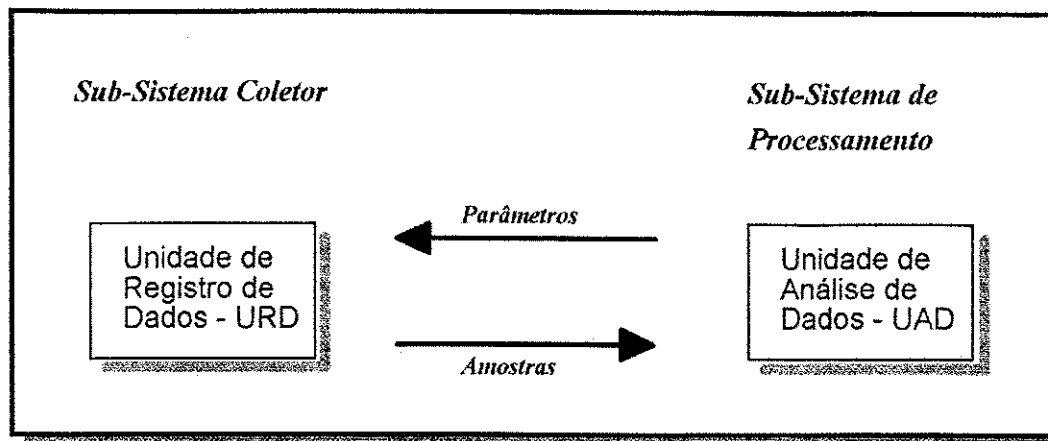


Figura 2.2 - Diagrama de Blocos do SAADE

O sub-sistema coletor permite a aquisição de amostras da velocidade do vento (direção mais frequente e magnitude média e máxima) e o seu armazenamento em memórias de estado sólido do tipo permanente (EEPROM), segundo um período de tempo pré-programado pelo usuário (1 a 255min.). Estas memórias estão acondicionadas em cartuchos rígidos (cartridges) para facilitar a instalação e retirada das mesmas pelo usuário. A instalação deste sub-sistema em locais remotos se torna plenamente possível por ele ser totalmente autônomo, portátil e automático. Construído exclusivamente com circuitos integrados de tecnologia CMOS, família de baixo consumo de energia, o SAADE tem autonomia de até 4 meses quando mantido por bateria de chumbo comumente usada em motocicletas.

Passado o período de aquisição de dados, o cartucho de memória com os registros das amostras coletadas no campo deve ser levado manualmente até o sub-sistema processador de dados, para a realização da análise estatística das amostras. O sub-sistema processador de dados consiste de um dispositivo leitor de cartucho de memória e transmissor de dados (via interface serial RS-232), e de um microcomputador compatível com o IBM PC.

O programa computacional de análise dos dados, ordena as amostras em formato de registros diários e as analisa estatisticamente, fornecendo os padrões de variação diária e sazonal, histograma de duração de velocidades, os parâmetros da curva de distribuição de Weibull, a rosa dos ventos e a densidade de potência eólica disponível. A partir das características de algumas turbinas eólicas comerciais, armazenadas previamente num banco de dados, pode-se também estimar a energia elétrica que seria

produzida pela turbina selecionada e, daí, calcular o custo do kW-h produzido pelo conjunto na região pesquisada.

Em termos de hardware o SAADE é constituído de módulos (construídos em placas de circuito impresso separadas) com funções bastante específicas. A função do sub-sistema coletor é realizada pelo Módulo Portátil de Aquisição de Dados - MPAD, e as tarefas do sub-sistema de processamento são divididas entre o Módulo de Leitura e Gravação de Memória - MLGM e o microcomputador, conforme pode ser visto no diagrama mostrado na **Figura 2.3**.

Enquanto o MPAD foi concebido para ser totalmente independente de outros dispositivos, o MLGM foi projetado para ser usado em conjunto com uma unidade de processamento que possua porta de comunicação serial do tipo RS-232. O Capítulo 3 descreve detalhadamente o hardware e o software básico (softwares de controle) de ambos módulos.

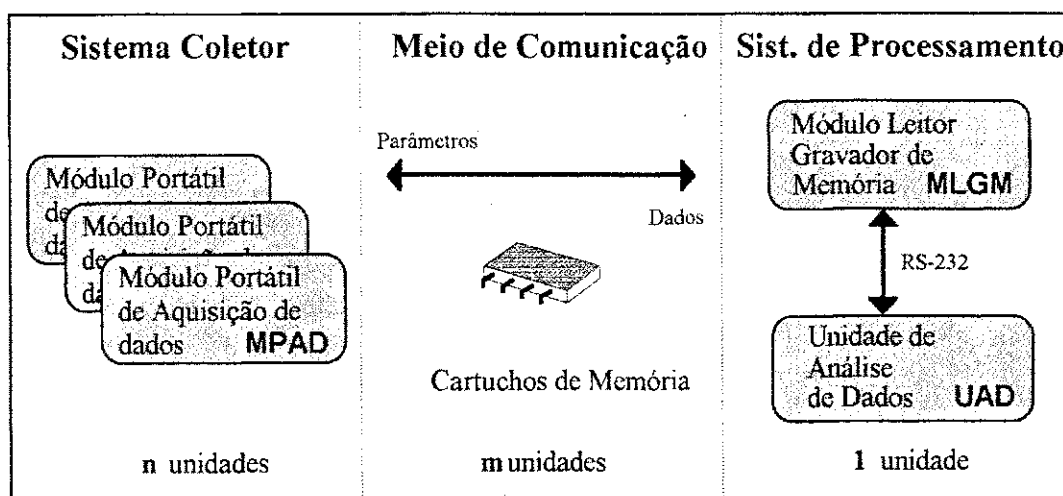


Figura 2.3 - Módulos do SAADE

2.2 - Características:

Portabilidade, facilidade de operação, autonomia (ditada pela potência da bateria e pelo intervalo de tempo de gravação das amostras) e baixo custo são as principais características do SAADE. Dentre outras características, não menos importantes, podemos citar ainda: facilidade de alteração do software de análise dos dados e manutenção de hardware simples e barata.

O sistema proposto oferece a opção de duas medidas simultâneas do módulo da velocidade (anemômetros dispostos em níveis diferentes de altitude) e uma medida da direção do vento. A quantidade de anemômetros a serem usados em cada URD é informada através de um byte gravado no cartucho pela UAD (ver **Tabela 2.2**). Cada cartucho de memória usa duas EEPROMs, de 32kbytes cada, o que proporciona espaço para até 32760 amostras. Supondo o uso de apenas um anemômetro e 10min. como intervalo de gravação de amostras, teremos 2340 grupos de 14 bytes ocupados por 6 conjuntos-amostra e uma máxima horária, ou o equivalente a 97,5 dias de medição. Este limite é imposto apenas pela capacidade de memória, visto que até 110 dias de medição são suportados com segurança por uma bateria comum de motocicleta (12Vcc, 7Ah).

A parte de processamento do circuito da URD fica "*dormindo*" durante a maior parte do tempo (99,8%, considerando intervalo de 10min. para gravação de amostras) por questões de economia de energia. Durante este tempo, apenas os contadores de pulsos dos anemômetros e o "*circuito de despertar*" ficam ativos. A cada minuto o restante do circuito é acordado e as contagens e a direção são processadas e armazenadas se necessário, e logo após o circuito de processamento volta a hibernar. A URD consome apenas 2,5 mA quando dormindo e 70 mA quando acordada., resultando em um consumo médio menor que 2,6 mA.

O módulo da velocidade é obtido em rpm através da contagem dos pulsos fornecido pelo anemômetro e totalizado pelo microcontrolador a cada minuto. A direção também é contabilizada a cada minuto. Após *n* minutos (intervalo de gravação de amostra) calcula-se o valor médio do módulo da velocidade, que então é gravado juntamente com a direção mais frequente obtida no mesmo intervalo de tempo de acumulo dos pulsos. As características técnicas do sistema proposto e do protótipo montado estão sumarizadas na **Tabela 2.1**.

CARACTERÍSTICA	SISTEMA PROPOSTO	PROTÓTIPO MONTADO
Entradas dos Dados		
Velocidade	2 canais	1 canal
Direção (cód. Gray)	4 bits	3 bits
Memória (kbytes)		
URD programa	8	32
dados	64	8
UAD (microcomput)	640 (mín.)	640
Portabilidade	Sim	Sim
Autonomia de memória conj. amostra¹		
com 1 anemômetro	32760 conjuntos	4088 conjuntos
com 2 anemômetros	16380 conjuntos	—
Autonomia de bateria (12Vcc, 7Ah)	4 meses	—
Consumo da URD		
ativo	70,0 mA	150,0 mA
dormindo	2,5 mA	40,0 mA
Precisão das Medidas		
Velocidade	4 pulsos/rotação	4 pulsos/rotação
Direção	22,5° (4 bits)	45° (3 bits)
Valores Limites		
Velocidade	4096 pulsos/min (= 101 m/s)	4096 pulsos/min (= 101 m/s)
Intervalo de Amostragem	1 min.	1 min.
Intervalo de Gravação	de 1 a 255 min.	de 1 a 60 min.
Saídas dos Dados		
URD	cartucho memória	CI de memória
UAD	relatórios, vídeo, disquete, ...	relatórios, vídeo, disquete, ...
Interface de Comunicação Serial	RS-232	RS-232
Temperatura de Operação	entre 0° e 65° C	entre 0° e 65° C
Fonte de Alimentação	bateria de chumbo, 12Vcc, 7Ah	Fonte CA-CC, 5V, 12V, -12V, 500mA
Proteção Contra Descargas Atmosféricas	fusível anti-surto e supressor de transientes de estado sólido	—

Tabela 2.1 - Características Técnicas do SAADE

¹ Valores calculados tomando-se 10 minutos como intervalo de gravação de amostras.

2.3 - Operação:

A operação do SAADE se resume em quatro etapas:

- i. configuração da URD,
- ii. transporte e instalação do cartucho de memória na URD,
- iii. retorno do cartucho com as amostras para a UAD, e
- iv. execução do software de análise na UAD.

A configuração se dá através da gravação de parâmetros nos primeiros 16 bytes do cartucho de memória que será utilizado para registro das amostras coletadas pela URD. A **Tabela 2.2** ilustra a disposição e significado de cada parâmetro. Esta gravação é realizada com o auxílio do software executado na UAD, conforme descrito adiante no Capítulo 4.

ENDEREÇO	PARÂMETRO
00 hex	Número identificador da URD
01 hex	Quantidade de anemômetros
02 hex	Taxa de amostragem: de 1 a 255 min
03-04 hex	MSB-LSB do último endereço utilizável da EEPROM
05-06 hex	MSB-LSB do próximo ender. do conjunto-amostra a ser gravado
07 hex	contador de uso da EEPROM (incrementado na retirada)
08-0F hex	reservado

Tabela 2.2 - Cabeçalho de Parâmetros Gravados no Cartucho

Posteriormente à configuração da URD o cartucho é levado manualmente até o local onde se encontra instalados a URD e o sistema de medidores: torre de sustentação (canos galvanizados, estais, cabos, hastes de aterramento, etc), anemômetro(s), medidor de direção e a caixa de acondicionamento da URD.

Se a unidade já se encontrar em funcionamento, deve-se proceder de acordo com a sequência de desligamento, que está colocada em forma de fluxograma na **Figura 2.5a**. Depois de desligada a URD, o cartucho usado deve ser retirado e o operador deve também anotar na etiqueta de dados de controle do cartucho (**Figura 2.4**) o dia e a hora corrente no momento da operação de retirada/colocação do mesmo.

<u>Etiqueta de Dados</u>		Cartucho: _____
1. Instalado em: _____ / _____ / _____	às _____:_____	Cód. de Erro: _____
Nº URD: _____	Retirado em: _____ / _____ / _____	às _____:_____
2. Instalado em: _____ / _____ / _____	às _____:_____	Cód. de Erro: _____
Nº URD: _____	Retirado em: _____ / _____ / _____	às _____:_____
3. Instalado em: _____ / _____ / _____	às _____:_____	Cód. de Erro: _____
Nº URD: _____	Retirado em: _____ / _____ / _____	às _____:_____

Figura 2.4 - Etiqueta de Dados de Controle do Cartucho

A seguir deve-se posicionar na URD o cartucho que ficará instalado para mais uma temporada de medição, e só então procede-se com a sequência de ligação da URD (Figura 2.5b). Se existir algum problema, um sinal luminoso de advertência (emitido por um led) piscará durante alguns segundos indicando o tipo do erro. O operador deve anotar o código do erro (número de acendimentos) na etiqueta de dados do cartucho e condená-lo até que se faça a correção do problema, que pode estar na unidade ou mesmo no próprio cartucho.

As etapas anteriores devem ser seguidas para outras unidades que porventura estejam na região, e ao finalizá-las os cartuchos contendo as amostras devem ser levados até o local onde se encontra a UAD, observando cuidados básicos, tais como: não deixar os cartuchos em locais úmidos, expostos diretamente ao sol ou a campos magnéticos, ou ainda sofrer descargas eletrostáticas, quedas ou pressão de volumes maiores, etc.

A leitura dos cartuchos se dá de forma sequencial na UAD, sendo que para cada cartucho lido é criado um arquivo de dados contendo as amostras, ainda em rpm. Na fase de processamento deste arquivo, o programa pede os parâmetros da reta de conversão de rpm para m/s, característica a cada anemômetro utilizado. Depois disto todas as análises estatísticas são feitas em função dos valores médios e máximos da velocidade em m/s.

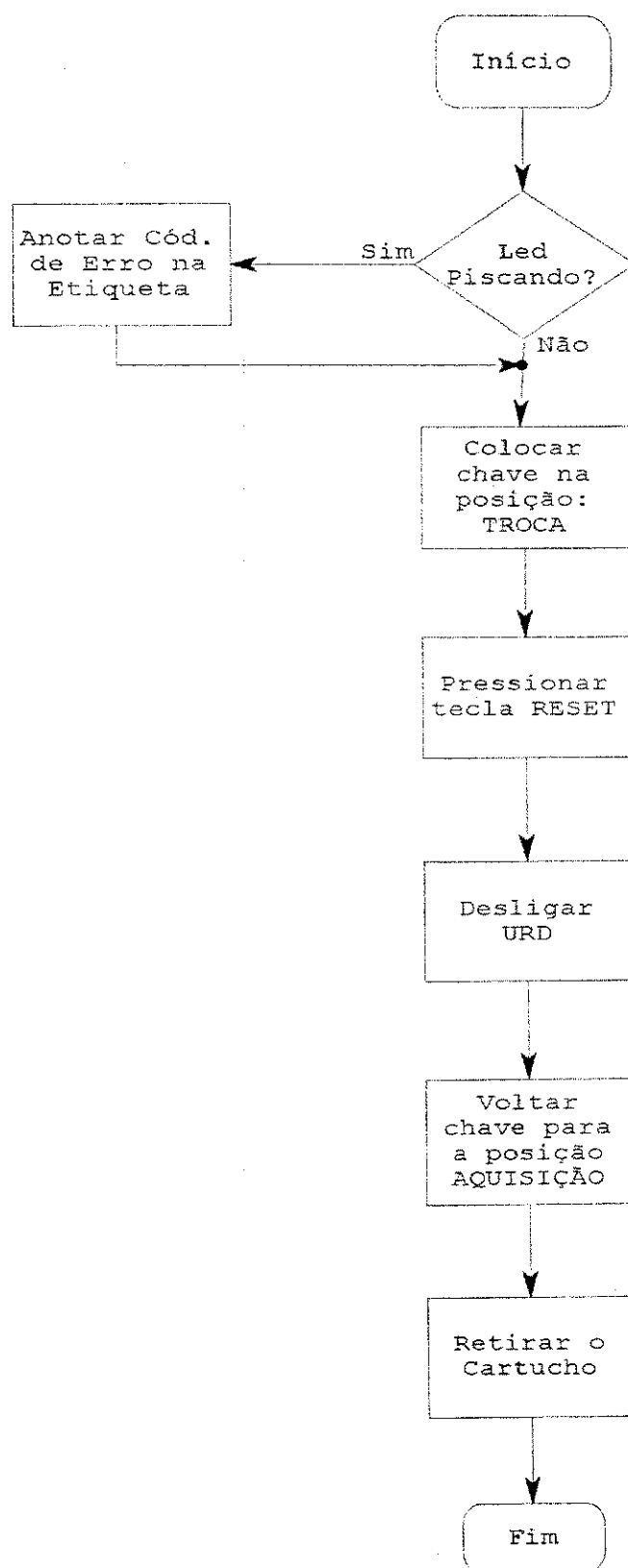


Figura 2.5a - Fluxograma da Sequência de Desligamento da URD

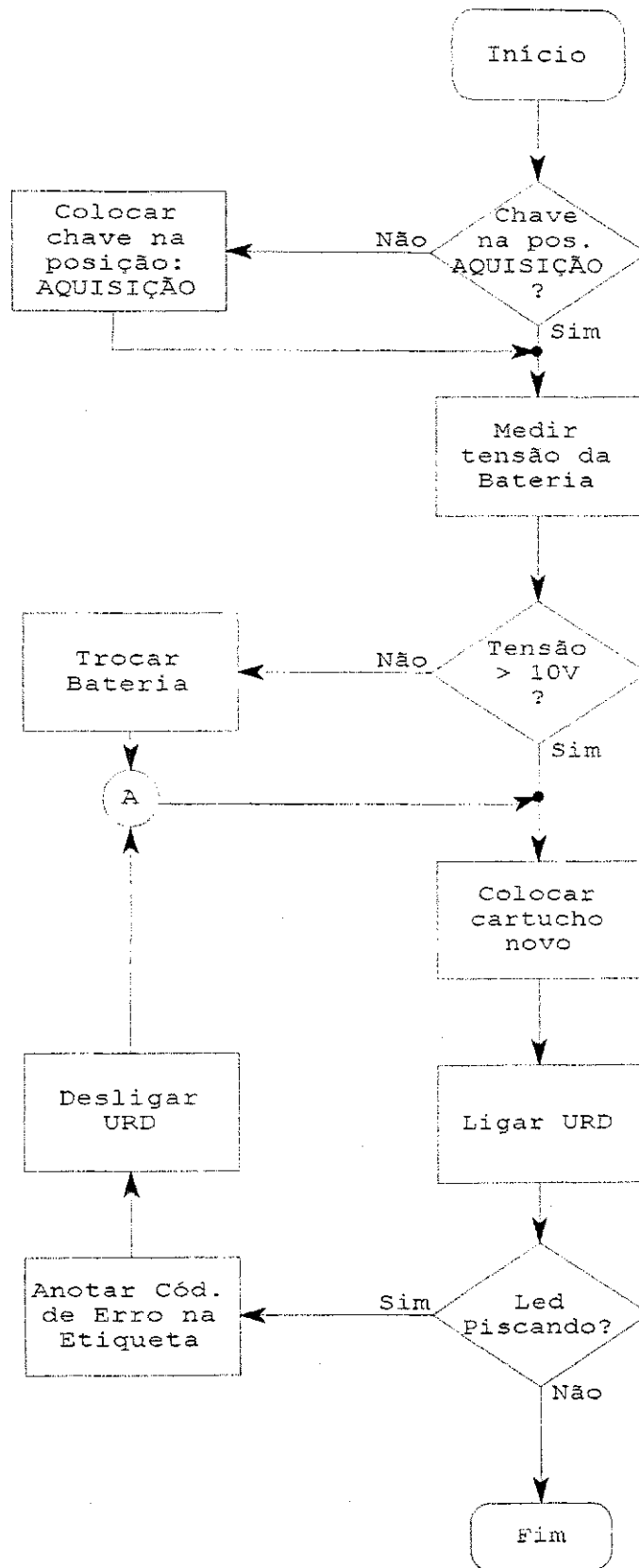


Figura 2.5b - Fluxograma da Sequência de Ligação da URD

CAPÍTULO 3

MÓDULOS DO SISTEMA

São apresentadas, nos itens seguintes, descrições detalhadas dos módulos de hardware que compõem o SAADE: o Módulo Portátil de Aquisição de Dados (**MPAD**) e o Módulo de Leitura e Gravação de Memória (**MLGM**).

3.1 - Módulo Portátil de Aquisição de Dados:

A função de medição e acúmulo das amostras de velocidade do vento (módulo e direção) é desempenhada por este módulo. Utilizamos um microcontrolador de 8 bits como unidade de controle (UC) do módulo, um relógio de tempo real (RTC) e um monoestável como circuitos auxiliares para despertar o microcontrolador (μ C) de seu estado Power-Down (estado de economia de consumo de energia).

Os instrumentos (anemômetro e windvane ou medidor de direção) usados como referência para o projeto deste módulo foram criados, construídos e aferidos pelo Núcleo de Energia - NERG do CCT da UFPb Campus II.

3.1.1 - Arquitetura Básica:

Devido aos principais pontos norteadores de concepção do SAADE (confiabilidade, custo reduzido e autonomia), este módulo dispõe de uma arquitetura simples e robusta, baseada em um microcontrolador de 8 bits e alguns dispositivos auxiliares: memória de código (EPROM), memória de amostras (EEPROM), contadores de pulsos e circuito despertador (RTC). A **Figura 3.1** ilustra a arquitetura básica do MPAD.

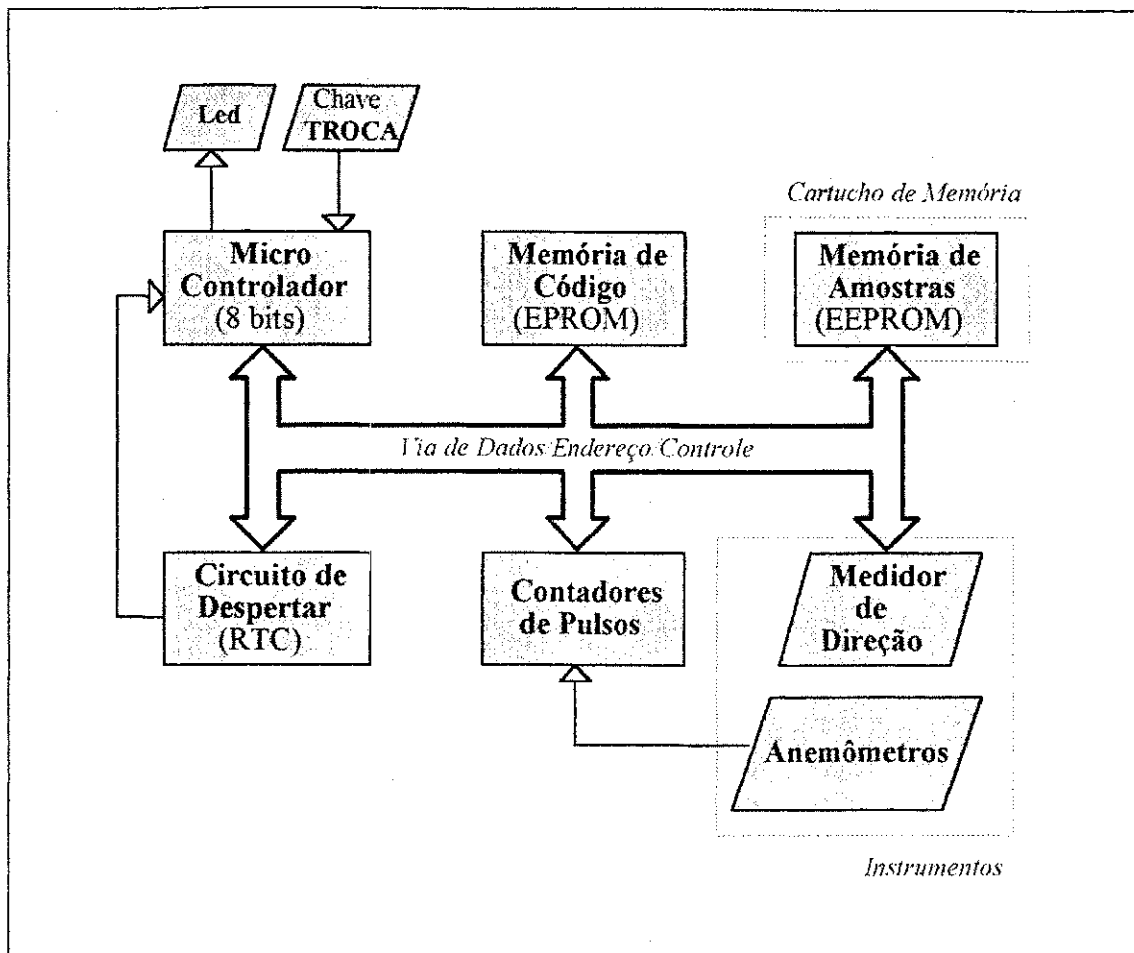


Figura 3.1 - Arquitetura Básica do MPAD

O MPAD usa um cartucho como unidade de entrada de parâmetros (primeiros 16 bytes do espaço linear de endereçamento) e ao mesmo tempo como memória secundária para armazenamento das amostras. A chave indicadora de troca de cartucho e os instrumentos de medição (anemômetros e medidor de direção) funcionam também como unidades de entrada para o MPAD. Apenas um led faz o papel de unidade de saída, e cuja finalidade exclusiva é comunicar ao mundo exterior a existência de alguma anormalidade com o módulo. O led piscará um certo número de vezes correspondente ao valor do código de erro ocorrido, permanecendo, em seguida, apagado durante quatro segundos. Este ciclo se repetirá indefinidamente, até que a unidade seja desligada para a correção do eventual problema.

A memória do cartucho está dividida em duas áreas distintas de dados: área de cabeçalho e área de amostras. A área do cabeçalho fica na parte inicial da memória do cartucho (16 primeiros bytes), e contém informações que determinam a configuração de funcionamento do MPAD com relação à quantidade de anemômetros a serem usados, à frequência com que as amostras devem ser gravadas, ao número de identificação do MPAD, ao tamanho da memória de amostras, e à outras informações como mostrado na **Tabela 2.2**. A área de amostras consiste em todo o espaço de endereçamento linear restante, e conterá amostras de intensidade e direção da velocidade do vento.

3.1.2 - Considerações de Hardware

Com o objetivo de alcançar a maior autonomia possível de carga da bateria, usamos uma metodologia no projeto de hardware para reduzir o consumo global de corrente do módulo, que consiste em colocar a unidade de controle (UC) (que possui consumo considerável de corrente para os nossos propósitos) funcionando apenas o tempo necessário para a leitura dos instrumentos e gravação das amostras na memória, e logo a seguir desativá-la parcialmente, deixando para os circuitos auxiliares, com menor consumo de corrente, as tarefas que não prescindam da intervenção da UC. Após um certo período de tempo, imposto pelo projeto, a UC é reativada por um circuito despertador para um novo ciclo de processamento. Este ciclo se repetirá enquanto o módulo se encontrar em perfeito funcionamento.

Adotamos como bloco de controle do módulo um microcontrolador (μC), devido à flexibilidade de hardware resultante de sua aplicação em sistemas de controle, de automação e de medição. Por possuir unidade de processamento, portas de entrada e saída (I/O), temporizadores/contadores, comunicação serial e conversores A/D, todos juntos em uma única pastilha, estes CI's tornam o projeto de hardware mais simples e versátil quanto às futuras modificações e à programação.

Existem várias famílias de microcontroladores, desenvolvidas por fabricantes consagrados de dispositivos semicondutores, cada uma com melhor performance em alguns pontos do que outras. Para este módulo optamos pela família Intel 8051, mais especificamente, pelo microcontrolador Intel 80C31BH (versão CMOS sem ROM interna). Dentre os fatores que determinaram esta escolha, citamos como principal o fato deste componente possuir um modo de funcionamento especial, chamado

POWER DOWN. Neste modo de funcionamento, que é ativado via software, o μC deixa de executar suas tarefas e entra em estado de hibernação, exigindo somente a potência necessária para manter sua RAM interna, reduzindo de 25mA para 10 μA o seu consumo de corrente [Silva, 1990 e Silva Jr, 1988].

O estado de hibernação do μC só é encerrado via RESET de hardware, e no módulo descrito neste item, um circuito despertador, baseado em um RTC (CMOS) de baixo consumo, é quem realiza a tarefa de reativar o μC .

O código executável do software básico de aquisição e gravação das amostras está armazenado em EPROM (CMOS), que também possui a característica de economia de energia quando operada no seu modo STANDBY, reduzindo de 25mA para 100 μA seu consumo de corrente [AMD, 1991]. Durante o período de hibernação do μC a EPROM também é colocada em seu modo STANDBY.

Por questões de segurança e de facilidade de transporte dos dados coletados, as amostras são armazenadas em memória permanente do tipo EEPROM (CMOS), também com características de baixo consumo.

Os circuitos contadores de pulsos estão sempre ativos, mas são também de consumo reduzido (CMOS). Admitindo a baixíssima probabilidade de ventos com velocidade superior a 100m/s, utilizamos este valor como limite máximo para o projeto dos contadores. Os anemômetros usados giram até mil vezes por minuto (1000 rpm) para ventos nesta velocidade limite [NERG-1, 1989] e, considerando ainda que os mesmos possuem precisão de um pulso a cada 1/4 (um quarto) de volta, teremos a cada minuto como contagem máxima:

$$\text{contagem}_{\text{máx}} \cong 4 \times 1000 = 4000 \text{ rpm} \quad (3.1)$$

Portanto, usamos contadores de 12 bits comercialmente comuns (contagem até 4096), suficientes para cobrir o limite máximo dado em (3.1). Estes contadores serão zerados a cada minuto para que não ocorra perda ou mascaramento de informação por estouro de contagem. Deste modo, estabelecemos o período de tempo de hibernação do μC em 1 minuto.

3.1.3 - Considerações de Software:

A primeira tarefa do software básico deste módulo é verificar a existência de cartucho de memória no slot para prosseguir para a segunda tarefa, a de distinguir se o início do programa foi causado por um RESET de hardware ou pela ligação da alimentação do módulo. Isto é feito através da comparação de uma seqüência de identificação, que é armazenada em local apropriado da RAM do μ C assim que o módulo é energizado.

Além da gravação desta seqüência de identificação, algumas variáveis contadoras/acumuladoras são iniciadas, os contadores são zerados, os parâmetros de configuração são lidos do cartucho de memória, e a identificação do módulo é checada. Se o início do programa é provocado pelo RESET de hardware, então as tarefas anteriores não são executadas.

A rotina principal do software se resume em realizar a leitura dos contadores de pulsos e acúmulo destes valores para cálculo do valor médio da velocidade do vento quando o intervalo de tempo para gravação for alcançado. A direção também é lida e a frequência de ocorrência da mesma dentro do intervalo é calculada, para ser gravada junta com a velocidade média, formando assim um conjunto-amostra.

A velocidade média é calculada também em rpm, e portanto, necessitando de até 10 bits ($2^{10} = 1024$) para o seu armazenamento na memória de amostras. Já a direção, que é medida com precisão de $\pi/4$, necessita apenas de 4 bits para ser armazenada, consumindo um total de 2 bytes para cada conjunto-amostra. A disposição de cada conjunto-amostra na memória é mostrada na **Figura 3.2**. Podemos observar nesta figura que os oito bits menos significativos (LSB) da velocidade média são armazenados no endereço de menor ordem, e no endereço seguinte temos os outros dois bits mais significativos (MSB) posicionado no nibble inferior, e no nibble superior do mesmo endereço são colocados os 4 bits da direção mais freqüente (DIR).

Antes de provocar uma nova hibernação do μ C, o programa verifica se a chave de TROCA foi acionada pelo usuário para indicar a intenção de retirada do cartucho de memória do módulo. Em caso da tecla ter sido acionada, o programa

executará alguns passos finais para depois autorizar o desligamento do MPAD, sinalizado através do acendimento do led, para que o usuário possa retirar o cartucho.

Todas as tarefas executadas pelo software de controle do módulo MPAD estão colocadas em forma de algoritmo conforme a **Figura 3.3**.

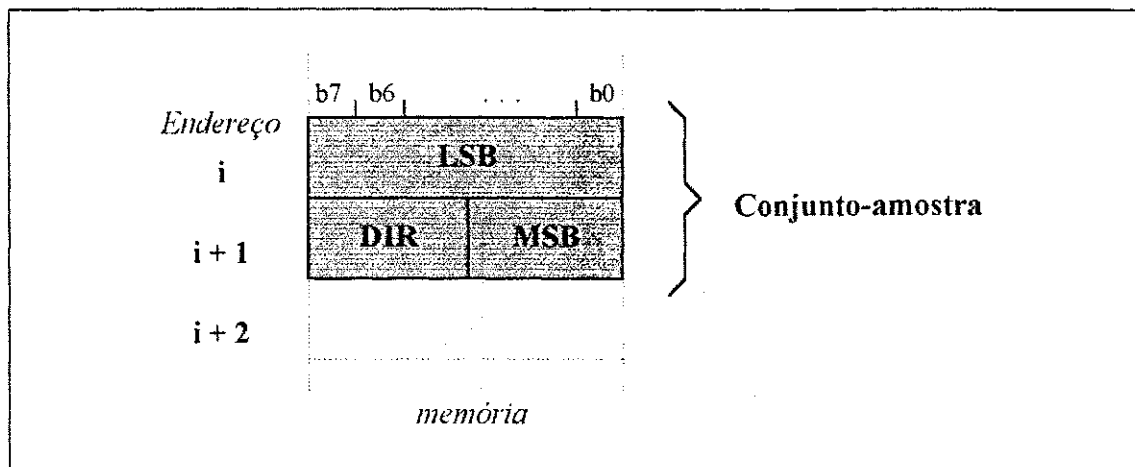


Figura 3.2 - Disposição de um Conjunto-amostra na Memória

```

início

// Declaração de variáveis
caracter    memram[10],           // flag de memoria RAM (teste de RESET period.)
            cont,                // contador de minutos p/ gravacao de amostra
            min,                // contador de minutos p/ grav. de max. horaria
            dir,                // direcao em codigo GRAY (nibble superior)
            maxhd;              // direcao em que ocorreu a max. horaria

inteiro     anem,                // contagem de pulsos do anemometro
            maxhv,              // contagem maxima de pulsos dentro de uma hora
            soma,              // soma de pulsos em cada interv. de amostragem
            ender,             // endereco corrente p/ armaz. das amostras
            media;             // amostra a ser gravada na EEPROM

// Inicializacoes realizadas somente ao se ligar a alimentacao do MPAD
se memram <> "ALEXANDRE"
    memram = "ALEXANDRE";
se le_ccprom(0) <> NIMPAD           // Cartucho correto p/ esta unidade?
    erro(1);
fim-se
se TROCA = 0                       // Chave de Troca acionada
    erro(2);
    
```

```

fim-se
prog_rtc(); // inicia timer de reset periódico
soma = 0; cont = 0; // inicia variáveis contadoras
maxhv = 0; maxhd = 0; min = 0;
fim-se
ender = le_eeprom(5)*256 + le_eeprom(6); // ender. do prox. conj.-amostra

// Amostragem
LOCKCNT = 1; // leva contagens para latches
RSTCNT = 1; // zera contadores
anem = le_contagem();
soma = soma + anem / 4; // acumula em rpm
se anem > maxhv
    maxhv = anem;
    maxhd = le_direcao();
fim-se

// Gravacao do conjunto-amostra: velocidade media e direcao do vento
cont = cont + 1;
se cont >= le_eeprom(2) // alcançou interv. p/ gravação?
    media = soma / cont; // em rpm
    dir = le_direcao();
    grava_eeprom( ender, media ); // grava velocidade
    grava_eeprom( ender + 1, dir ); // código GRAY de 4 bits
    soma = 0; cont = 0;
fim-se

// Gravacao da maxima horaria: velocidade e direção
min = min + 1;
se min >= le_eeprom(2) / 60
    grava_eeprom( ender, maxhv );
    grava_eeprom( ender+1, maxhd );
    min = 0; maxhd = 0; maxhv = 0;
fim-se
ender = ender + 2; // endereço do prox conj.de amostras
grava_eeprom( 5, ender );
se ender > le_eeprom(3) * 256 + le_eeprom(4)
    erro( 3 ); // nao ha mais espaco na EEPROM
fim-se

// Troca Cartucho de Memória?
se TROCA = 0
    erro(4); // código p/ desligamento do MPAD
fim-se
ativa_PD(); // vai hibernar...

fim-algoritmo

```

Figura 3.3 - Algoritmo do Software Básico do MPAD

3.2 - Módulo de Leitura e Gravação de Memória:

Este módulo é utilizado junto à UAD e tem como propósito realizar as seguintes funções: a leitura das amostras contidas no cartucho de memória, o apagamento destas amostras para reutilização do cartucho e a gravação dos parâmetros de configuração do MPAD.

Utilizamos um μC de 8 bits da mesma família como unidade de controle (UC) do módulo, e a sua característica de comunicação serial para a ligação do MLGM com a UAD.

O protocolo de comunicação empregado é assíncrono, e o módulo funciona sempre como escravo, aguardando os comandos da unidade de processamento. Como os microcomputadores IBM-PC e compatíveis em geral possuem portas seriais do tipo RS-232, adotamos então este padrão para o projeto do MLGM, visando obter versatilidade no interfaceamento do módulo com a UAD.

A alimentação deste módulo é fornecida por uma fonte comum de corrente contínua, e portanto a preocupação de consumo existente na concepção do MPAD não é fator restritivo para o projeto do MLGM.

3.2.1 - Arquitetura Básica:

Mantendo a mesma linha de raciocínio utilizado no projeto do MPAD, utilizamos um μC da família Intel 8051 como unidade de controle, mas como neste caso inexistente a necessidade de consumo baixo, adotamos o 8031 construído com tecnologia HMOS que é de menor custo. Este μC não possui ROM / EPROM interna, por isso usamos novamente uma EPROM para armazenar o código do programa de comunicação.

Para indicar o estado funcional corrente do MLGM usamos um led como unidade de saída. Um par de circuitos conversores de nível de tensão TTL para RS-232, e vice-versa, foram utilizados para a construção da porta serial. Esta porta serial

pode ser considerada como unidade de entrada e saída deste módulo. A Figura 3.4 mostra a arquitetura básica do MGLM.

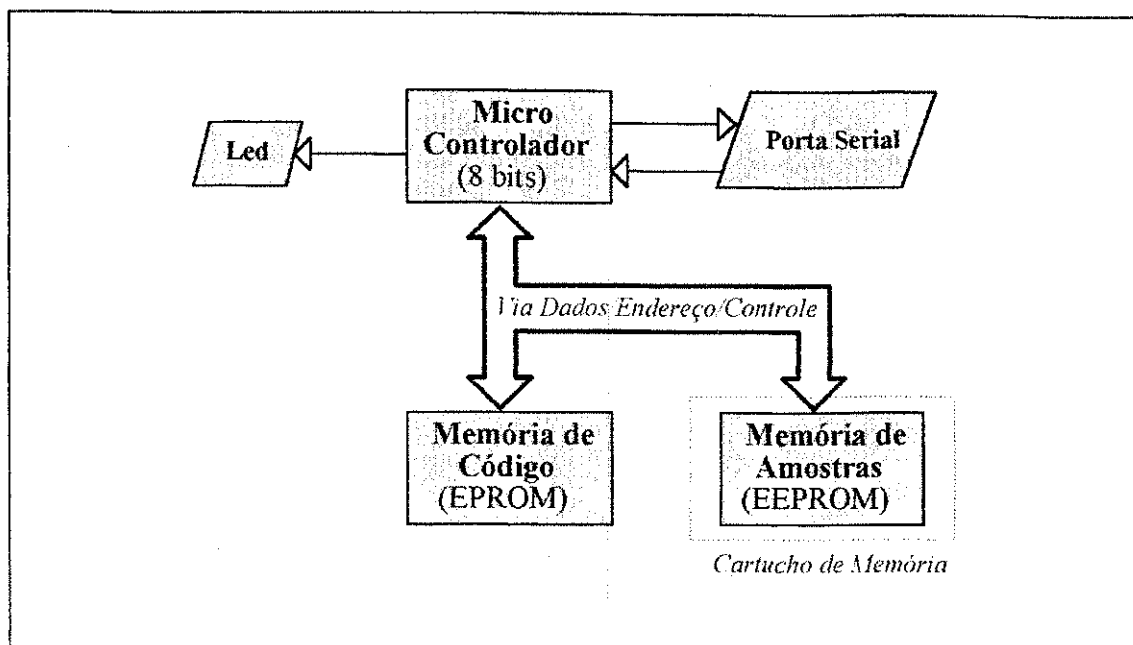


Figura 3.4 - Arquitetura Básica do MLGM

3.2.2 - Considerações de Hardware e Software:

O μC 8031 possui várias facilidades para realização de comunicação serial, dentre as quais está a de auto-geração da taxa de transmissão (baud rate) através de um de seus temporizadores, evitando a necessidade de se acrescentar circuito auxiliar para o desempenho de tal função.

A porta serial apresenta apenas os três sinais fundamentais para comunicação serial, no conector: TXD - sinal de transmissão de dados, RXD - sinal de recepção de dados, e GND - referência do sinal e terra do circuito. Os demais sinais existentes no padrão RS-232 não são utilizados. O cabo de conexão deve ser configurado como mostra a Figura 3.5.

O protocolo de comunicação implementado pelos programas usados no MGLM (sistema *escravo*) e na UAD (sistema *mestre*) utiliza a conhecida prática do

"bem-mandado", ou seja, o *escravo* fica sempre aguardando uma ordem (bloco de dados) do *mestre*. Esta ordem é a unidade básica de troca de dados adotada, e cada unidade desta é composta de 19 bytes divididos em três campos funcionais como é mostrado na **Figura 3.6**. Após a recepção e cumprimento da ordem o *escravo* retorna ao estado de espera de nova ordem do *mestre*.

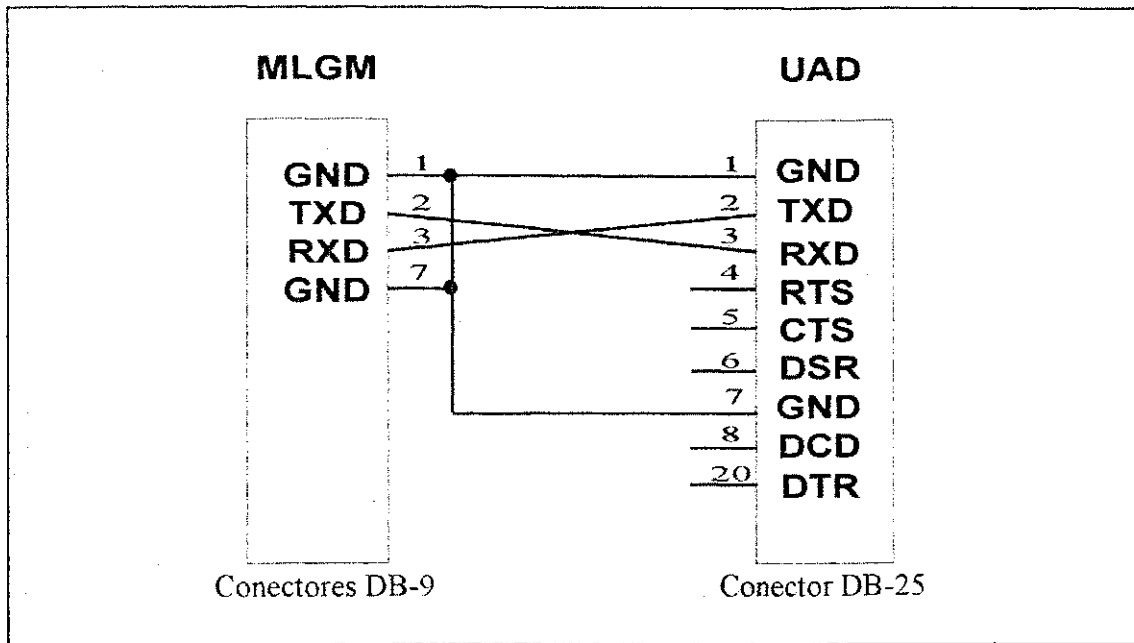


Figura 3.5 - Cabo de Comunicação Serial do MLGM com a UAD

No início do bloco temos um campo de identificação do comando, com a função solicitada pelo *mestre*. O campo seguinte é usado para comandos que fazem referência a um endereço específico da memória de amostras, como no caso de leitura ou gravação de informação em uma determinada posição da memória (ex. cabeçalho de parâmetros). O último campo é um vetor de 16 bytes de informação, que tem sua função dependendo do comando emitido pelo *mestre*.

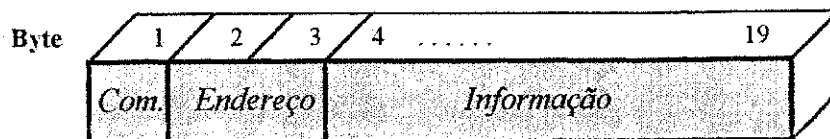


Figura 3.6 - Formato do Bloco de Dados da Comunicação Serial

A transmissão do bloco é feita sempre com a duplicação de bytes recebidos, que poderíamos chamar de handshaking por software. Após o *mestre* transmitir um byte ele espera o retorno do mesmo byte para que nova transmissão possa ser executada. Assim que o byte é lido pelo *escravo* na porta serial o mesmo é retornado para o *mestre* indicando que o *escravo* está pronto para receber outro byte. O processo se repete até que o bloco todo tenha sido transmitido. No caso de algum byte não ter sido retornado, ou até mesmo retornado com erro, a transmissão do bloco é cancelada, e após um certo tempo (time-out) o *mestre* tenta enviar novamente o mesmo bloco.

Antes de ser iniciada qualquer comunicação, os sistemas trocam alguns bytes de testes do canal serial, e com isto a verificação de erros intrínsecos a comunicação serial, tais como configuração diferente das portas seriais (taxa de transmissão, paridade, bits de parada,...), conexões e cabos, podem ser indicados ao usuário pelo sistema.

Todas as tarefas executadas pelo software de comunicação do módulo MLGM estão colocadas em forma de algoritmo na **Figura 3.7**.

```

início
// Declaração de variáveis
bit          fim;
character    comando;

// Inicia porta serial
inicia_serial();                // Porta Serial no modo 1, a 2400,8,N,1

// Verifica se cartucho esta posicionado
se existe_cart() = 0
    erro(1);
senão
    led = 1;                    // acende led: operando normalmente
fim-se

// Estabelecendo comunicacao serial com a URA (Unidade de Registro e Analise)
grava_serial( ler_serial() );   // teste da comunicação serial
fim = 0;
enquanto fim = 0                // Aguarda comando da UAD
    comando = ler_serial();
    grava_serial( comando );
    escolha comando
        caso 0xF2 : envia();    // leitura de amostras
        caso 0xF4 : recebe();   // gravacao de cabec./dados
        caso 0xF5 : apaga();    // apagamento da EEPROM
        caso 0xF1 : fim = 1;    // abandona comunicacao

```

```

    fim-escolha
    fim_enquanto
    led = 0; // apaga led
    fim-algoritmo
    global inteiro bufs[19];

    função recebe() // PC para 8031
    inteiro n, ender;
    recebe_bloco();
    ender = bufs[0] + bufs[1] * 256;
    para n = 2 até n < 18
        grava_ceprom( ender, bufs[n] );
        n = n + 1; ender = ender + 1;
    fim-para
    fim-função

    função envia() // 8031 para PC
    inteiro n, ender;
    recebe_bloco();
    ender = bufs[0] + bufs[1] * 256;
    para n = 2 até n < 18
        bufs[n] = le_ceprom( ender );
        n = n + 1; ender = ender + 1;
    fim-para
    ler_serial(grava_serial( 0xF2 ));
    para n = 0 até n < 18 // envia bloco c/ informacao
        ler_serial(grava_serial( bufs[n] ));
        n = n + 1;
    fim-para
    fim-função

    função apaga()
    inteiro n, ender_i, ender_f;
    recebe_bloco();
    ender_i = bufs[0] + bufs[1] * 256;
    ender_f = bufs[2] + bufs[3] * 256;
    enquanto ender_i <= ender_f
        grava_ceprom( ender_i, 0 ); ender_i = ender_i + 1;
    fim-enquanto
    fim-função

    função recebe_bloco()
    inteiro n;
    para n = 0 até n < 18 // recebe e retorna bloco
        bufs[n] = ler_serial();
        grava_serial( bufs[n] ); n = n + 1;
    fim-para
    fim-função

```

Figura 3.7 - Algoritmo do Software Básico do MLGM

Os comandos implementados no software de comunicação usado pelo MLGM e pela UAD dizem respeito a leitura, gravação e apagamento de dados no cartucho de memória. A **Tabela 3.1** lista os comandos implementados com respectivos códigos identificadores e formato dos campos do bloco de dados usado como unidade básica na comunicação serial entre os equipamentos..

COMANDO	BLOCO DE DADOS (ver Figura 3.5)		
	Código	Endereço	Informação
Abandono de comunicação	F1 hex	NA	NA
Leitura de amostras	F2 hex	início da área de mem. a ler	NA
Gravação de cabeçalho	F4 hex	0000 hex	cabeçalho (Tabela 2.2)
Apagamento de memória	F5 hex	início da área de mem. a apagar	fim da área de mem. a apagar

Legenda: NA - não se aplica

Tabela 3.1 - Código dos Comandos Implementados do MGLM

CAPÍTULO 4

SOFTWARE DE ANÁLISE DE DADOS EÓLICOS

Apresentamos neste capítulo o software de análise estatística de dados eólicos do SAADE, usado na UAD. Inicialmente descrevemos sua funcionalidade dentro do sistema, depois a metodologia empregada e em seguida exploramos sua arquitetura. Também são detalhados a implementação, os recursos mínimos (software e hardware) exigidos para sua execução, e por fim a interface com o usuário.

4.1 - Funcionalidade:

A função principal deste software é reduzir as amostras de velocidade do vento, coletadas em campo pelo MPAD, em informações úteis ao usuário do sistema: valores médios e máximos da velocidade do vento, desvio-padrão das amostras, histogramas das distribuições de frequência e temporal dos ventos, previsão de produção anual de energia elétrica, etc.

Após a etapa de aquisição e transporte das amostras (gravadas no cartucho de memória), entra em ação os módulos MLGM e UAD para a realização da etapa de processamento gerenciada por este software.

A operação do programa pressupõe uma certa sequencialidade nas tarefas a serem executadas para dados não processados. Em um primeiro instante, as amostras precisam ser lidas do cartucho de memória e gravadas em arquivos de dados no winchester. Continuando o processamento, estes arquivos serão reduzidos em dados médios, e a seguir, cálculos estatísticos são processados. A partir deste instante, todas as opções do software se tornam disponíveis, e o usuário poderá executá-las na ordem em que preferir: impressão de resultados, visualização de gráficos, selecionamento de turbina eólica para previsão de produção de energia, preparação de cartucho para nova aquisição, etc.

Para dados pré-processados, ou seja, que já foram lidos e gravados em arquivos anteriormente, a sequencialidade no uso das opções do software se resume apenas na execução do processamento estatístico, realizado na opção *Estatística* (vide item 4.3 para maiores detalhes). E novamente, como no caso de dados não processados, após o cumprimento destas sequências, todas as opções estarão disponíveis ao usuário, para serem usadas em qualquer ordem.

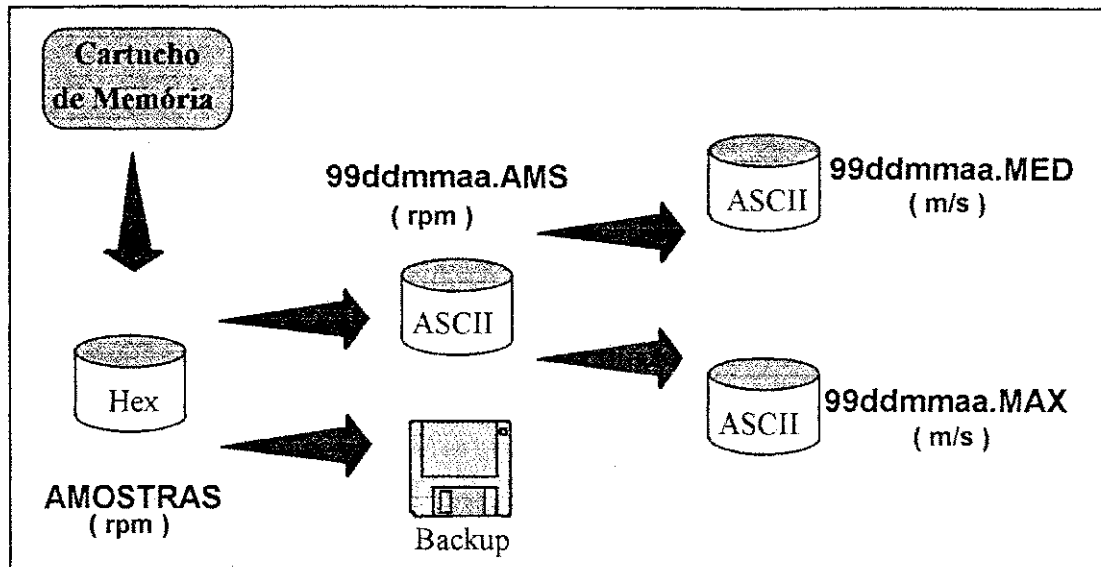


Figura 4.1 - Fluxo de Dados no SAADE

Na **Figura 4.1** temos ilustrado o fluxo dos dados através do sistema, desde a coleta das amostras no campo até a saída de avaliações estatísticas. Como podemos observar na referida figura, durante a leitura das amostras gravadas no cartucho de memória, é criado um arquivo denominado AMOSTRAS, contendo uma cópia idêntica dos dados do cartucho. Este arquivo temporário é usado apenas como entrada para a conversão dos dados representados no sistema binário para o sistema decimal utilizando um formatação tipo texto (ASCII). Um novo arquivo é criado com o nome formado pela seguinte regra:

-
- i. Os dois primeiros caracteres são obtidos do número de identificação do MPAD.
 - ii. Os seis caracteres seguintes são obtidos da data de início da aquisição dos dados.
 - iii. A extensão do nome do arquivo é AMS .
-

A função deste arquivo é permitir a realização de crítica dos dados por parte do usuário, através da edição do arquivo em tela-cheia. O formato dos registros deste arquivo .AMS está representado na **Figura 4.2**. Devemos observar que a unidade de velocidade usada nas amostras coletadas no MPAD é rotações por minuto (rpm), e por conseguinte, as velocidades gravadas neste arquivo também estão em rpm. No **Capítulo 5** temos um exemplo típico deste arquivo, obtido nos testes preliminares em bancada do SAADE, e ilustrado em forma de relatório na **Figura 5.2d**.

<u>Arquivo</u>	<u>Registro-tipo</u>	
AMOSTRAS	Data - Hora	16 bytes
	Cabeçalho da EEPROM	1° reg.
	veloc1 dir1 veloc2 dir2 veloc8 dir8	demais
99ddmmaa.AMS	Nidentif, Data, Hora, Qtd. de Anem., Intervalo de Gravação	66 bytes
	1° reg.	
	veloc1 dir1, veloc2 dir2, veloc8 dir8,	demais
99ddmmaa.MED 99ddmmaa.MAX	Local, Nidentif, Coef. Ang., Coef. Linear, Data Inic., Data Final,	226 bytes
	1° reg.	
	data, veloc1 dir1, veloc2 dir2, veloc24 dir24,	demais

Figura 4.2 - Formato dos Registros dos Arquivos do Sistema

Nesta fase do processamento, é solicitado ao usuário a descrição do local onde está instalada a unidade de aquisição de dados de onde foi retirado o cartucho, bem como os parâmetros da reta característica de conversão de rpm para m/s, do anemômetro(s) executor(es) das medidas.

Na etapa seguinte os dados gravados no arquivo .AMS são reduzidos a velocidades médias horárias e velocidades máximas horárias, e gravados em dois outros arquivos cujos registros-tipo estão também mostrados na **Figura 4.2**. Os dados da reta característica de conversão de unidades são usados para a obtenção dos valores em unidades do SIU. Os arquivos criados mantêm o mesmo nome, apenas a extensão é mudada para MED no caso do arquivo de médias, e para MAX no caso do arquivo de máximas.

A partir deste ponto, os dados são processados somente na memória interna, e portanto todo valor calculado só estará disponível enquanto permanecer ligada a UAD. Ou seja, se o usuário precisar pesquisar estes dados em uma outra ocasião, ele precisará solicitar o reprocessamento estatístico dos dados gravados nos arquivos MED e MAX.

O sistema foi concebido para usuários que caracterizam-se pela posse de conhecimentos de ciências atmosféricas, especificamente, conhecimentos sobre ventos e sua exploração, tais como: especialistas em SCEE's, agrometeorologistas e projetistas, dentre outros.

4.2 - Metodologia de Projeto:

Algumas características exigidas para o projeto de hardware também são comuns ao projeto do software de análise estatística: confiabilidade e facilidade de operação. Além destas, facilidade de manutenção e reutilização de código são requisitos fundamentais em qualquer software atual, pois sabemos que quanto mais um software é usado, maior é a probabilidade de ocorrerem pedidos, oriundos dos usuários, de alterações e/ou inclusão de novos recursos [Weiskamp, 1991].

A primeira das características exigidas, a confiabilidade, em geral só é alcançada após intensivos testes práticos e uso crítico da "versão Beta" do software por especialistas. A segunda característica já está relacionada com questões de interfaceamento homem-máquina, e adotamos a fórmula *janela-menu* como elementos estruturais para facilitar a operação do sistema. As demais características, facilidade de manutenção e reutilização de código, são as mais difíceis de serem atendidas por metodologias convencionais de programação. Mas estas características estão exatamente entre as principais vantagens oferecidas pela Programação Orientada a Objeto - POO [Khoshafian, 1990], e por isto adotamos este caminho para o desenvolvimento do software de análise do SAADE.

O sistema foi construído em duas etapas: projeto da interface homem-máquina, e projeto da aplicação específica. Inicialmente desenvolvemos um cenário-base que fornecesse ferramentas elementares para a interface homem-máquina, e depois passamos ao desenvolvimento do software de análise estatística de dados eólicos. Nos itens que se seguem discutimos sobre estas etapas de construção do projeto.

4.2.1 - Cenário-Base:

O cenário-base para o desenvolver da aplicação específica, consiste de classes de objetos elementares como: janelas retangulares com acessórios (sombreamento, títulos, movimentação, redimensionamento), menus de opções com barra de seleção e teclas-quentes, lista de seleção, página gráfica, gráficos de barra e circular, recursos do sistema operacional, arquivos em disco e em impressora, e diretório de arquivos.

As classes de objetos criadas para este projeto, e seus relacionamentos (hierarquia), são mostrados na **Figura 4.3**. Devemos observar que os métodos e atributos adotados para cada classe de objetos desenvolvidas para o projeto, não esgotaram todas as possibilidades de comportamento e qualificação existentes para as entidades representadas, mas foram levadas em consideração apenas as necessidades mínimas exigidas para o sistema em questão. Para uma revisão dos conceitos básicos de programação orientada a objeto, tais como: classe, métodos, herança, derivação; uma breve introdução sobre o assunto é apresentada no **Apêndice C**.

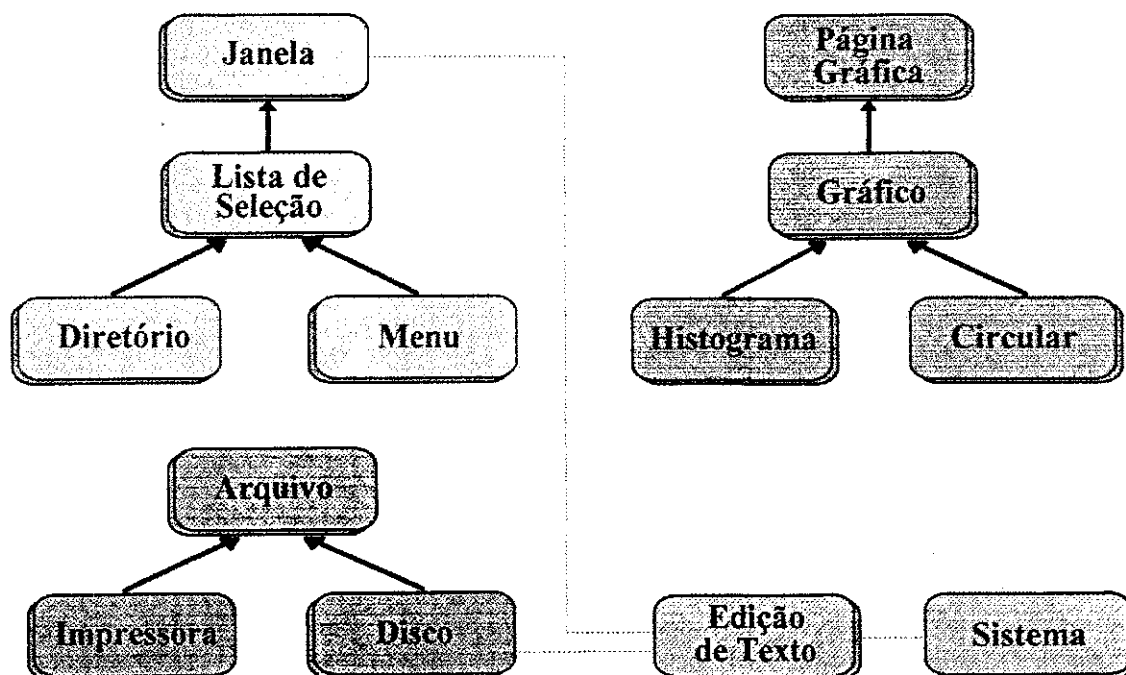


Figura 4.3 - Classes e Hierarquia

As classes *Diretório* e *Menu* herdam atributos e métodos da classe *Lista de Seleção*, a qual, por sua vez, é derivada da classe *Janela*. A mesma hierarquia é observada para as classes de objetos gráficos. Existe ainda uma classe de objetos com informações do sistema operacional (drive e sub-diretório correntes, hora e data do sistema, dados de arquivos do diretório, etc) que não deriva e nem é ascendente de outra qualquer outra classe. A classe *Edição de Texto* está mostrada no diagrama do projeto relacionando-se através de composição com as classes *Janela* e *Disco*, e ela representa objetos textos com métodos de manipulação dos mesmos em tela cheia.

4.2.2 - Aplicação Específica:

Terminado o projeto do cenário, o desenvolvimento da aplicação específica se tornou mais simples com a decomposição funcional para a modelagem da espinha dorsal do programa. Desta estruturação funcional, surgiram módulos de software gerenciadores que atendem às opções disponíveis nesta versão do programa, mostradas em diagrama de blocos na **Figura 4.4** e discutidas no item 4.4.

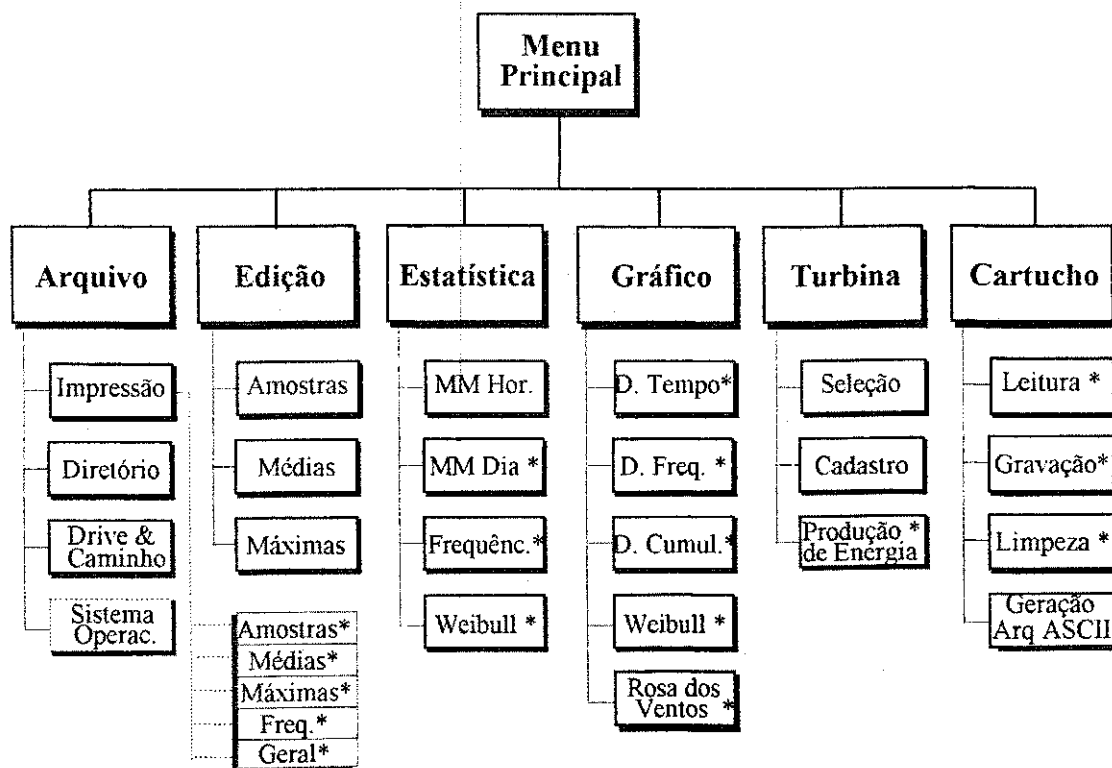


Figura 4.4 - Opções do Software de Análise Estatística

Nestes módulos gerenciadores é que são criados dinamicamente os objetos, de acordo com as opções escolhidas pelo usuário durante a execução do software; e um módulo denominado principal faz a abertura da primeira tela do programa e a geração de algumas variáveis globais, como também o gerenciamento dos demais módulos. Funções específicas de totalização e cálculos estatísticos também são adicionadas aos módulos gerenciadores.

4.3 - Arquitetura e Estruturas de Dados:

A arquitetura do software é fundamentada em módulos gerenciadores e executores das tarefas acessíveis ao usuário pelo menu principal de opções. O módulo principal coordena a execução das tarefas requisitadas no menu principal (menu horizontal), através da chamada de módulos secundários responsáveis por funções específicas de acordo com a opção escolhida. Para uma melhor compreensão da arquitetura do sistema e da subordinação dos módulos, apresentamos um diagrama de blocos com os módulos gerenciadores do software da UAD na Figura 4.5, e na Figura 4.6 os módulos de suporte desenvolvidos para o cenário-base.

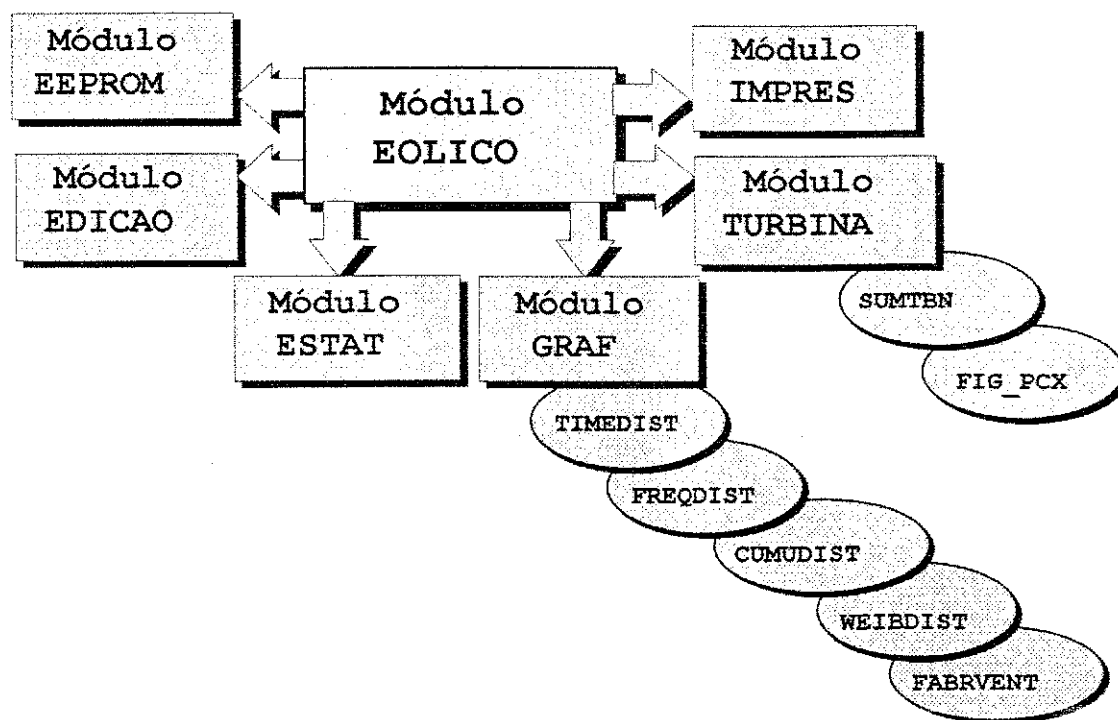


Figura 4.5 - Arquitetura do Software de Análise

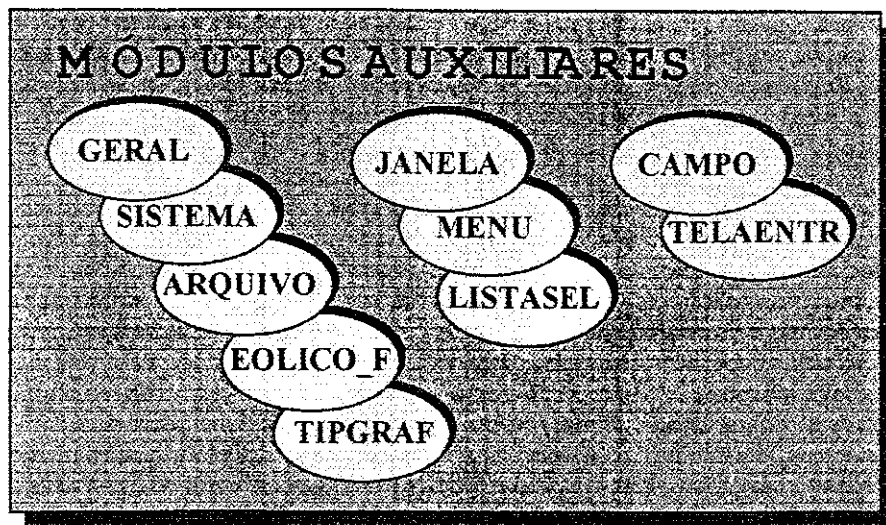


Figura 4.6 - Módulos de Software Auxiliares

A seguir, passamos a descrever a função, o arquivo de definições, a quantidade de linhas de código, as estruturas de dados utilizadas, as variáveis e objetos criados e/ou recebidos como parâmetros, de cada módulo (principais e auxiliares).

Módulo:	EOLICO.CPP	
Função:	Gerenciar o menu principal, apresentar tela-inicial, e ativar outros mód.	
Definições:	EOLICO.H	
Linhas Cód:	146	
Objetos globais:	Sistema s;	// objeto de sistema operacional
	TelaGraf TelaGrafica;	// objeto de acesso à tela gráfica
Objetos locais:	Menu princ, arq, graf;	// sub-menus das respectivas opções
	Janela fundo, status, logot;	// janelas iniciais
Variáveis globais:	int gfreq[26],	// 26 classes de freq. de velocidade
	gdir[8][4],	// 4 classes de freq. direcional
	gestat = 0;	// flag de analise estatistica
	float gmed[24],	// media horaria mensal
	gmedd[31],	// media diaria no mes
	gmedia,	// media geral
	gdpadr,	// desvio-padrao geral
	gmaxim,	// maxima geral
	gmax[24],	// maxima horaria mensal
	gmaxd[31],	// maxima diaria no mes
	weib_k, weib_c;	// parametros da curva de Weibull
	char gport,	// porta serial usada p/ comunicacao
	gmes,	// mes analisado
	estatisticamente	
	gunid[3];	// num. de identif. unidade analisada

Módulo: IMPRES.CPP

Função: Gerenciar a impressão de relatórios em arquivo ou na impressora.
 Definições: IMPRES.H
 Linhas Cód: 391
 Chamado por: main()
 Chama: imp_amost(), imp_media(), imp_maxima(), imp_freq(), imp_geral();
 Objetos locais: Menu imp, resp, dest; // menu de impres, resposta e destino
 Janela nome; // janelas para entr. nome de arquivo
 Campo org; // campo p/ edicao nome do arquivo
 Variáveis globais: FILE arqprn; // handle do arq. de impressao

Módulo: TURBINA.CPP

Função: Gerenciar a impressão de relatórios em arquivo ou na impressora.
 Definições: TURBINA.H
 Linhas Cód: 319
 Chamado por: turbinas()
 Chama: seleciona(), cadastra(), cadgeral(), sumario_tbn(), cadcurva(), preve();
 Objetos locais: ListaSel turb; // lista de selecao de turbinas
 Janela cad; // janelas para cadastro
 TelaEntr tc; // tela de cadastro de turbinas
 Variáveis: struct Turbina s; // dados da turbina selecionada

Módulo: EDIÇÃO.CPP

Função: Gerenciar a edição dos arquivos de amostras e médias/máximas.
 Definições: EDICAO.H
 Linhas Cód: 55
 Chamado por: main()
 Chama: mensagem();
 Objetos locais: Menu ed; // menu de tipo de arq. a editar

Módulo: EEPROM.CPP

Função: Gerenciar a comunicacao serial e gerar arquivo de amostras (ASCII).
 Definições: EEPROM.H
 Linhas Cód: 416
 Chamado por: main()
 Chama: receb_eeprom(), envia_eeprom(), zera_eeprom(), geracao();
 Objetos locais: Menu cart; // opcoes de tarefas c/ o cartucho
 Janela coms; // janela para comunic. serial
 Variáveis: struct bloco; // definicao de bloco de comunic.

Módulo: ESTAT.CPP

Função: Gerenciar a realizacao de estatísticas sobre os dados coletados.
 Linhas Cód: 497
 Chamado por: main()

Módulo: GRAF.CPP

Função: Gerenciar as opções de gráficos do sistema.
Definições: GRAF.H
Linhas Cód: 254
Chamado por: main()
Objetos locais: Menu graf; // menu de gráficos

Módulo: TIMEDIST.CPP

Função: Gerar e mostrar o gráfico da distribuição temporal dos dados coletados.
Definições: TIPGRAF.H
Linhas Cód: 27
Chamado por: graf()
Objetos locais: GrafBar g; // objeto gráfico de barras

Módulo: FREQDIST.CPP

Função: Gerar e mostrar o gráfico da distribuição de frequência dos dados colet.
Definições: TIPGRAF.H
Linhas Cód: 32
Chamado por: graf()
Objetos locais: GrafBar g; // objeto gráfico de barras

Módulo: CUMUDIST.CPP

Função: Gerar e mostrar o gráfico da distribuição cumulativa dos dados colet.
Definições: TIPGRAF.H
Linhas Cód: 30
Chamado por: graf()
Objetos locais: GrafBar g; // objeto gráfico de barras

Módulo: WEIBDIST.CPP

Função: Gerar e mostrar o gráfico da distribuição de Weibull dos dados colet.
Definições: TIPGRAF.H
Linhas Cód: 46
Chamado por: graf()
Chama: fdp_weib(k, c); // calculo da fdp de Weibull
Objetos locais: GrafBar g; // objeto gráfico de barras

Módulo: FABRVENT.CPP

Função: Gerar e mostrar o gráfico da distribuição direcional dos dados coletad.
Definições: TIPGRAF.H
Linhas Cód: 19
Chamado por: graf()
Objetos locais: GrafWindFabr g; // objeto gráfico Rosa dos Ventos

A seguir, temos os módulos auxiliares, contendo as classes de objetos elementares, funcionando como suporte à interface do sistema com o usuário.

Módulo: GERAL.CPP

Função: Fornecer funções diversas ao sistema.
 Definições: GERAL.H
 Linhas Cód: 119
 Chamado por: vários módulos
 Possui: strtrim(), som(), stof(), cursor(), cget(), cput(), goto_xy();

Módulo: EOLICO_F.CPP

Função: Realizar saída para o shell e acessar diretório de arquivos.
 Definições: EOLICO_F.H
 Linhas Cód: 289
 Chamado por: vários módulos
 Possui: diretorio(), drivesubd(), sistop();

Módulo: SISTEMA.CPP

Função: Declarar a classe Sistema, que fornece acesso aos recursos do S.O.
 Definições: TIPGRAF.H
 Linhas Cód: 89
 Classe: private:
 char cwd[MAXPATH];
 int gdriver, // drive grafico: CGA, EGA, VGA, DETECT
 gmode, // modo grafico
 vmode, // modo do video corrente
 vmax, // maior modo de video suportado
 vmin; // menor modo de video suportado
 char *vmem; // memoria de video
 public:
 Sistema(); // construtor
 int getdisc() { return cwd[0] - 'A'; } // retorna o drive de disco
 void setdisc(int drive); // ajusta o novo drive
 char *getpath() { return (char *) (cwd+2); } // retorna o path
 void setpath(char *path); // ajusta o novo path
 int getvmode() { return vmode; } // retorna o modo de video
 void setvmode(int vmode); // ajusta o novo modo grafic
 char *getvmem() { return vmem; } // retorna o end. de video
 void setdct(char *cwd); // ajusta dir. cor. trab.
 char *getdct() { return cwd; } // retorna dir. cor. trab.
 void Erro(char *classe, char *funcao, char *msg, int codigo=0);

Módulo: **ARQUIVO.CPP**

Função: Definir classe Arquivo, usada para acessar arquivos sequenciais.

Definições: ARQUIVO.H

Linhas Cód: 192

```

Classe: private:
        char    nome[MAXFILE];    // nome
        char    ext[MAXEXT];      // extensao
        char    dir[MAXDIR];      // diretorio onde esta(ra) o arquivo
        char    drive[MAXDRIVE];  // drive onde esta(ra) o arquivo
        char    *reg;             // registro corrente
        char    fim;              // flag de final de arquivo
        long    poscur;           // posicao corrente
        int     compr;            // comprimento maximo do registro
        FILE    *fd;              // leito usado nas operacoes do arq.
        void    Cria();           // cria arquivo se ele nao existir
        void    temlivre();       // procura 1o. registro excluido
                                     // se nao encontrar, vai p/ o fim

public:
        Arquivo( char *a, char *b=NULL, char *c=NULL, char *d=NULL,
                unsigned e=COMPR );

        Arquivo( char *A, unsigned B=COMPR );    // Construtor
// Parametros:    a - nome do arquivo
//                b - ext do arquivo
//                c - diretorio do arquivo
//                b - ext do arquivo
//                c - diretorio do arquivo
//                d - drive do arquivo
//                e - comprimento maximo dos registros
//                A - rota (drive, diretorio, nome e extensao)
//                B - comprimento maximo dos registros
~Arquivo() { delete reg; }
void Abre();                // Abrir e posicionar no 1o. reg.
void Fecha();              // Fechar o arquivo
int Existe();               // Verificar se o arquivo ja existe
int Fim() { return (int) fim; } // var.indicadora de final de arquivo
char* Reg() { return reg; }
char* Reg( char *p );      // buffer com os dados do reg. cor.
int ComprReg() { return compr; } // Retornar o compr. do reg.
long ComprArq();           // Retornar o comprimento do arq.
void Le( int a=0 );        // Ler o reg. apontado por poscur
void Grava( int a=0 );     // Gravar o reg. na posicao atual
void Posiciona( enum movimenta a=PROXIMO, long b=11 );
void Descarrega() { fflush( fd ); } // Fazer o flush do buffer de disco
void Deleta();             // Coloca marca de exclusao
void Trocanome( char *n ) { fnsplit( n, drive, dir, nome, ext ); }
void Erro( char *classe, char *funcao, char *msg, int codigo=0 );

```

Módulo: CAMPO.CPP

Função: Definir classe Campo, usada para editar campos de entrada de dados.

Definições: CAMPO.H

Linhas Cód: 159

Classe: private:

```

char x, y, // posicao do 1o. caracter do
campo
    largc, // largura total do campo
    *masc, // mascara na entrada/edicao
    *ent, // valor inicial
    *sai; // var. c/ valor apos entrada/edicao

void mostra( char *p, char pc, char qtde );
public:
    Campo( char a, char b, char *c, char *d, char *e=NULL, char f=0 );
// Funcao: construtor
// Parametros: a - coluna da tela do primeiro caracter do campo
//              b - linha da tela do primeiro caracter do campo
//              c - variavel destino
//              d - valor inicial a ser mostrado no campo
//              e - mascara a ser usada na edicao
//              f - largura do campo
    char edita( char a, char b, char *c, char *d, char *e, char f );
// Funcao: Permitir a edicao do campo
// Parametros: a - coluna da tela do primeiro caracter do campo
//              b - linha da tela do primeiro caracter do campo
//              c - variavel destino
//              d - valor inicial a ser mostrado no campo
//              e - mascara a ser usada na edicao
//              f - largura do campo
char *conteudo() { return sai; } // Valor editado no campo
void ve() { mostra( sai, 0, largc ); } // Mostra o valor atual do campo

```

Módulo: TELAENTR.CPP

Função: Definir classe Telaentr, usada para editar tela de campos de entrada de dados em tela cheia, tipo texto.

Definições: TELAENTR.H

Linhas Cód: 87

Objetos: referência à vetor de objetos Campo

Classe: private:

```

Campo *campos[MAXCAMPOS]; // ender. dos objetos de campo
struct {
    char col, // coluna do label
        lin, // linha do label
} tit[MAXCAMPOS]; // titulos dos campos de edicao
char n; // numero atual de campos na tela
char campocur; // campo correntemente em edicao

```

```

public:
  Telaentr() { campocur = n = 0; }; // Construtor
  ~Telaentr(); // Destrutor
  void insere( char *t, char x, char y, char a, char b, char *c, char *d,
              char *e, char f);
// Funcao: Inserir campo a ser editado na tela
// Parametros: t - titulo do campo
// x - coluna do primeiro caracter do titulo
// y - linha do primeiro caracter do titulo
// a - coluna do primeiro caracter do campo
// b - linha da tela do primeiro caracter do campo
// c - variavel destino
// d - valor inicial a ser mostrado no campo
// e - mascara a ser usada na edicao
// f - largura do campo
char edita( int a=0 ); // Editar a tela
char *campo( int a=0 ) { return campos[a]->conteudo(); }
void erro( char *classe, char *funcao, char *msg );

```

Módulo: **JANELA.CPP**

Função: Definir classe Janela, usada para criar e manipular de janelas de texto.

Definições: JANELA.H

Linhas Cód: 396

```

Classe: protected:
  int x, y; // canto superior esquerdo
  int larg, alt; // largura e altura
  char moldura; // existente se diferente de zero
  char sombra; // existente se diferente de zero
  char *titulo; // titulo da janela
  paljan cores; // cores usadas na janela
  int *buffer; // buffer da area sobreposta da tela
  char curx, cury; // posicao corrente do cursor
  text_info sobreposta; // info's da janela sobreposta
  void poemoldura(char pisca=0); // desenha moldura em volta
  void poesombra(); // desenha sombra externamente
public:
  Janela(int x, int y, int larg, int alt, char *tit, char ctxt, char cmol,
         char mold, char s);
  ~Janela() { delete conteudo; delete buffer; }
  int move( void );
  int mudatam( void );
  void ativa( void );
  void desativa( void );
  void preenche( char c );
  void mostraxy ( char *texto, int x=0, int y=0 );
  void mostraxy ( unsigned No, int x=0, int y=0 );
  void mostraxy ( float f, int x=0, int y=0, char l=4, char p=1 );

```

```

Módulo:      MENU.CPP
-----
Função:      Definir classe Menu, usada para manipular opcoes em janelas de texto.
Definições:  MENU.H
Linhas Cód:  204
Objetos:     referência à objetos Janela
Classe:      protected:
              private:
                char  nopc;                // numero de opcoes
                char  tipo;                // menu vertical / horizontal (=1)
                opcao *opcoes;            // vetor com as opcoes
                char  corrente;            // opcao com a barra de selecao
                char  primeira;            // primeira opcao
                char  *teclq;              // vetor de teclas- quente
                palmenu cores;              // cores
                void  mostraopc();          // funcao para mostrar as opcoes
              public :
                Menu( char A, char B, char C, char D, char a, opcao *b, char c=1,
                    char *E=NULL, char F=WHITE, char G=WHITE, char d=0x70,
                    char e=GREEN, char f=1, char g=1, char *h=NULL, char H=1,
                    char I=1 );
              // Parametros:      a - numero de opcoes
              //                   b - vetor de estrutura das opcoes
              //                   c - tipo
              //                   d - cor da barra de selecao
              //                   e - cor das teclas- quente
              //                   f - opcao corrente (default: 1a. opcao)
              //                   g - primeira opcao da janela (default: 1a. opcao)
              //                   h - vetor de teclas- quente (default: nao ha teclas-q)
              //                   A - absc. do 1o. carac. da 1a. opcao, relativa a tela
              //                   B - ordenada 1o. carac. da 1a. opcao, relativa a tela
              //                   C - largura da janela em no. de colunas (caracteres)
              //                   D - altura da janela em no. de linhas de texto
              //                   E - titulo da janela
              //                   F - cor do texto
              //                   G - cor da moldura e do titulo
              //                   H - moldura externamente `a janela (= 0 nao coloca)
              //                   I - sombra `a moldura/janela (=0 nao coloca)
              ~Menu() { }                // Destrutor
              void opcaodefault( char opc ); // opcao inicialmente selecionada
              void ativa();                // Mostrar as opcoes
              void desativa();              // Retirar o menu
              void move();                  // Mudar a posicao da janela
              int escolhe();                // Permitir a escolha das opcoes
              void ligaopc( char n=-1 );
              void deslopc( char n=-1 );    // ativa/desativa a n-esima opcao
              char opcfuncional( char n );  // n-esima opcao funcional

```

Módulo: **LISTASEL.CPP**

Função: Definir classe ListaSel, usada para manipular listas de informacoes em janelas de tela texto, com barra de selecao.

Definições: LISTASEL.H

Linhas Cód: 130

Objetos: referência à objetos Janela

Classe:

```

private:
    int nel;                // qtde de elem. na lista de selecao
    char **elem;           // vetor de pont. p/ elementos
    char topo;             // elemento no topo da janela
    char corrente;         // elemento corrente
    char corbarra;         // cor da barra de selecao
    void mostraelem();     // funcao para mostrar os
elementos
    int pesquisa( char c ); // pesquisa pelas primeiras letras
public :
    Listasel( char *a[], char A, char B, char C, char D, char *E=NULL,
              char F=WHITE, char G=WHITE, char H=RED );
// Parametros:   a - elementos da lista
//               A - absc. do 1o. carac. da 1a. opcao, relativa a tela
//               B - ordenada 1o. carac. da 1a. opcao, relativa a tela
//               C - largura da janela em no. de colunas (caracteres)
//               D - altura da janela em no. de linhas de texto
//               E - titulo da janela
//               F - cor do texto
//               G - cor da moldura e do titulo
//               H - cor da barra de selecao
~Listasel() { }           // Destrutor
void ativa();             // Mostra opcoes dentro da janela
void desativa();         // Retira a janela
int escolhe();
// Funcao:       Permitir a escolha de alguma das opcoes do menu
//               retornando o numero do elemento escolhido ou zero
//               se nao foi escolhido nenhum (ESC).

```


4.4 - Recursos Utilizados:

4.4.1 - Para Implementação:

Das linguagens atuais orientadas a objeto a mais popular, sem dúvidas, é a linguagem C++, e isto se deveu ao grande número de usuários C que viram na linguagem C++ uma maneira natural de migrar para esta nova metodologia de concepção de programas. Outras linguagens, SmallTalk, Objlog e Modula, com certeza apresentam melhores estruturas orientadas a objetos, visto terem sido desenvolvidas especificamente para esta nova metodologia, diferentemente da linguagem C++ que surgiu de uma adaptação da ling. C, mas em compensação, como acontece com sua ancestral, ela continua gerando códigos mais rápidos que as demais. Outro ponto a favor da linguagem C++ é o grande número de fornecedores de compiladores existente no mercado, consolidando definitivamente a linguagem como a opção adotada pela maioria de software-houses e desenvolvedores de aplicativos [Weiskamp, 1991].

Pelo exposto no parágrafo anterior, resolvemos adotar a linguagem C++ como ferramenta de desenvolvimento do código do programa de análise do SAADE. Mais especificamente, fizemos uso do compilador e ambiente de desenvolvimento integrado (IDE) da Borland Inc., o Turbo C++, Vs.2.0 para DOS. Conforme podemos ver no **item 4.3**, a descrição das estruturas de dados e classes foram feitas usando-se a sintaxe da linguagem C++. Indicamos na *bibliografia consultada* algumas obras que tratam de programação em C++, e que poderão ajudar o leitor iniciante no assunto.

Entre arquivos de declarações (.H), de projetos (.PRJ), de desktop (.DSK), de definições de classes e programas (.CPP e .C), de arquivos de dados (.AMS, .MED, .MAX, .PRN), de arquivos de programas objetos (.OBJ), de suporte de interface gráfica (.BGI e .CHR), e executáveis (.EXE), temos um total de aproximadamente 1.250 kBytes de dados armazenados em memória de massa, e que foram gerados ao longo de mais de 5 meses de codificação e depuração.

Gostaríamos de estimular a reutilização do código por qualquer usuário que demonstre interesse no aproveitamento deste trabalho na sua totalidade, ou em partes (classes de objetos elementares: janela, menu, seleção de lista, gráficos, etc), devido a facilidade de geração de novas aplicações utilizando as classes de objetos. O

código executável, e o código-fonte de todos os módulos do software de análise do SAADE, estão disponíveis na Coordenação de Pós-Graduação em Engenharia Elétrica da UFPb - Campus II, e podem ser usados sem qualquer prévia autorização do autor.

4.4.2 - Para Execução:

A plataforma de hardware mínima para executar o software de análise estatística do SAADE deve ter CPU compatível com Intel 8086 / 8088 / 80286 / 80386 / 80486, memória interna de pelo menos 640 kBytes, memória de massa (disco rígido) com 2 MBytes de espaço livre, monitor de vídeo com resolução de 640 x 480 pontos (padrão VGA), porta de saída serial RS-232, impressora matricial de 80 colunas. Recomenda-se a utilização de CPU 80386 por questões de velocidade de processamento, principalmente na elaboração de gráficos, que em geral envolve funções de manipulação de operações aritméticas com ponto flutuante.

O software foi desenvolvido para trabalhar em qualquer sistema operacional compatível com o MS-DOS, Vs. 6.0, mas "deve" rodar em versões mais antigas.

A plataforma utilizada para execução do software, nos testes preliminares, constava de um microcomputador com CPU 386DX40, com memória RAM de 4MBytes, com winchester de 120MBytes, com monitor SVGA Color (800 x 600 pixels), com 2 (duas) portas seriais RS-232 e uma impressora matricial de 80 colunas (10cpi).

4.5 - Interface com o Usuário:

O sistema utiliza o terminal de vídeo (monitor de vídeo e teclado) como hardware de interação para o usuário acessar as informações de processamento disponíveis.

O software do SAADE possui 7 (sete) opções apresentadas na primeira tela (tela de apresentação) em forma de menu horizontal. Cada uma destas opções possui um módulo de software distinto, exceto a opção *Fim*. Todas as opções iniciais e respectivos sub-menus estão mostradas na **Figura 4.4**.

As opções do programa não estão disponíveis a todo momento para o usuário. Existe uma seqüência de operações que deve ser seguida para ativar as opções condicionadas. No software implementado essas opções condicionadas são mostradas nos menus em cores diferentes quando não estão ativas, e mesmo que a barra de seleção seja colocada sobre uma opção inativa a função da escolha não é executada. As opções condicionadas estão marcadas com um asterisco na **Figura 4.4**.

Nos sub-itens seguintes detalhamos a função e características peculiares a cada uma das opções do software de análise estatística.

4.5.1 - Opção Arquivo:

Este sub-menu oferece outras 4 (quatro) opções, sendo a última delas invisível inicialmente ao usuário (opção indicada com quadro tracejado na **Figura 4.2**), que precisa levar o cursor até o final da janela e forçar mais uma posição para poder acessá-la. Esta última opção, *Sistema Operacional*, permite a saída momentânea do software para o shell. Para retornar ao SAADE o usuário deve usar o comando *EXIT* do sistema operacional.

A primeira opção neste sub-menu é a de *Impressão* de dados, que dá acesso a outro sub-menu com mais 5 (cinco) opções para impressão: *Amostras* para descarga do arquivo *.AMS*, *Médias* para impressão das velocidades médias horária, *Máximas* para impressão de um relatório contendo as veloc. máximas horária, *Frequência* para impressão da distribuição de frequência de ocorrência das velocidades distribuídas em 26 classes, e *Geral* para impressão dos momentos estatísticos gerais e parâmetros da curva de distribuição de Weibull (ver **Apêndice A.1**). Se alguma opção se encontrar inativa, então a execução da opção *Estatística* do menu principal reverterá esta situação.

Na opção seguinte do sub-menu *Arquivo, Diretório*, o usuário tem acesso à lista de arquivos presentes nos disco e subdiretório correntes, que podem ser mudados através da terceira opção, *Drive & Caminho*, para o disco e/ou subdiretório que o usuário desejar manter como corrente.

4.5.2 - Opção Edição:

Esta opção tem o objetivo de permitir total acesso (visualização, alteração, inclusão, e exclusão) aos dados coletados e reduzidos. A escolha desta opção oferece outro sub-menu com mais 3 (três) opções: *Amostras* que permite editar os dados do arquivo *.AMS* que o usuário escolher.

As outras duas outras opções, *Médias*, e *Máximas*, da mesma forma como na opção anterior, acessam os arquivos de médias e máximas horárias, respectivamente. Também como na primeira opção deste sub-menu, deve-se escolher o arquivo *.MED* (ou *.MAX*) antes de proceder com a edição¹. Como discutido em item anterior, estes arquivos são gerados pelo sistema através de processamento disparado pela opção *Estatística* do menu principal.

É importante salientarmos que o programa faz o reconhecimento das informações dos arquivos de dados através das posições dos campos/dados, sempre tendo como referência o início de cada registro (linha no editor de texto). Portanto, é fundamental que o usuário mantenha a posição e o formato dos dados dentro de cada registro após a edição do arquivo escolhido, para que o programa continue a reconhecer o arquivo corretamente.

¹ Nesta versão do software, utiliza-se o editor *dBFast* para edição dos arquivos. Este editor aceita os comandos do *Wordstar*, e também possui menu de comandos acessível ao se pressionar a tecla *ESC*.

4.5.3 - Opção Estatística:

Outro sub-menu com mais 4 (quatro) opções são mostradas ao se acessar esta opção que, pode ser considerada como o coração do sistema, tanto que, ela é a primeira opção a ser posicionada no menu principal ao se carregar o programa. Todas as reduções de dados e a maioria dos cálculos feitos pelo sistema são realizados ao se escolher suas opções.

Ao se iniciar um processamento de amostras, somente a opção do topo do sub-menu *Estatística* estará ativa. As demais serão ativadas após a execução da função da primeira opção. Antes de se ter acesso ao sub-menu desta opção, o usuário terá que selecionar um arquivo de amostras para ser processado. Depois desta escolha, o software analisa o arquivo e mostra o(s) mes(es) com dados disponíveis para análise. O usuário então determinará, segundo seu interesse, o período desejado para o processamento das amostras. Após a determinação do período o sub-menu é mostrado.

A primeira opção, *Méd / Máx. Horária*, faz a transformação dos dados colhidos (total de amostras por hora = 60 / intervalo de gravação) em dados horários: velocidades médias e máximas, com respectivas direções. A opção seguinte, *Méd / Máx. Diária*, faz o mesmo trabalho, mudando apenas na totalização dos dados por dia.

A classificação das velocidades em 26 níveis, como mostrado na **Figura 4.5**, é obtida com a execução da opção *Distribuição de Frequência*, que calcula a ocorrência de cada faixa de velocidade, em horas. Na última opção, *Distribuição de Weibull*, são calculados os fatores de forma e de escala desta curva de distribuição. No **Apêndice A** é colocado uma fundamentação teórica sobre a Distribuição de Weibull.

FAIXA	VELOCIDADE (m/s)	OBSERVAÇÃO
1ª	0	calmaria
2ª	0 — 1	
3ª	1 — 2	
25ª	23 — 24	
26ª	24 —	acima de 24 m/s

Figura 4.7 - Faixas de Velocidade da Distribuição de Frequência

4.5.4 - Opção Gráfico:

O acesso ao sub-menu desta opção só é conseguido após a execução da opção *Estatística*. O sub-menu possui mais 5 (cinco) opções de gráficos. A primeira delas, gera um gráfico de distribuição das velocidades no tempo, ou seja, as flutuações diurnas da velocidade do vento. Na abscissa estão as horas do dia, e nas ordenadas a velocidade média de cada hora, e em linha tracejada é mostrado o valor da velocidade média global (todas as amostras do período escolhido na opção *Estatística*).

Os gráficos gerados pelas duas opções seguintes, *Distribuição de Frequência* e *Distribuição Cumulativa*, apresentam as funções densidade de probabilidade (fdp) e probabilidade acumulada (FDP), em forma de histogramas, dos dados do período.

A fdp de Weibull, acessada pela opção *Distribuição de Weibull*, é mostrada justaposta sobre a fdp dos dados medidos, para uma melhor comparação da aproximação das curvas em termos gráficos.

E por último a opção *Rosa dos Ventos* mostra a distribuição de frequência das velocidades do vento classificada por direção.

4.5.5 - Opção Turbina:

Esta opção permite ao usuário *Selecionar* uma das turbinas existentes no banco de dados de turbinas comerciais, para realizar uma previsão de produção anual de energia (opção *Produção de Energia*), caso a turbina selecionada fosse instalada na região onde foram colhidas as amostras.

Através da opção *Cadastro*, o usuário pode ampliar o banco de dados com novas turbinas, ou até modificar os dados de alguma turbina cadastrada anteriormente. Isto atribui ao software a qualidade de expansibilidade. As informações cadastradas dão uma visão geral das características de cada máquina (potência nominal, fabricante, torre, diâmetro e quantidade de pás, tipo de gerador etc) e das curvas de potência e produção ideal anual de energia.

amostras (120 valores médios + 24 valores máximos) armazenadas na EEPROM foram lidas e enviadas à UAD pelo MLGM via comunicação serial.

Para testar o software de análise, optamos por trabalhar com uma massa de dados conhecida e cujos cálculos estatísticos também já fossem conhecidos. Escolhemos os dados de velocidade média horária do mês de Junho/1975, no aeroporto da capital Praia, das ilhas da República do Cabo Verde [Lysen, 1983], pelo simples fato destes dados retratarem um período básico de análise de 1 (um) mes de levantamento de dados.

Estes dados estão dispostos em forma de planilha na **Tabela 5.1**, sendo que as informações de direção não são mostradas porque foram geradas de forma randômica, pois a planilha original de dados possuía somente informações de velocidade. Os resultados obtidos do processamento destes dados, utilizando o software de análise estatística, forneceram os mesmos valores obtidos por Lysen. Na **Figura 5.1** temos os gráficos que podem ser obtidos com o programa de análise estatística, e na **Figura 5.2** os relatórios, todos obtidos com os dados deste teste.

5.2 - Ambientes de Programação:

Para o desenvolvimento do software básico e do software de análise estatística, utilizamos as seguintes ferramentas de software:

1. Compilador C-51, Vs. 3.40, Keil Elektronik GmbH
2. Linkeditor L51, Vs. 3.00, Keil Elektronik GmbH
3. Simulador/Depurador AVSIM51, 1986, Ken Anderson Software
4. Editor de programas do sistema operacional MS-DOS
5. Ambiente de desenvolvimento Turbo C++ for DOS da Borland, Vs.2.0.

O software básico foi desenvolvido em linguagem C específica para microcontroladores da família Intel 8051. O código fonte do módulo MPAD possui 112 linhas, enquanto que o arquivo fonte do MLGM tem 140 linhas, valores obtidos sem se considerar os arquivos de cabeçalho (.H)

Hora	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Média
Dia 1	9,4	10,0	8,8	9,4	9,7	10,0	10,9	11,4	11,9	13,9	13,9	14,2	13,9	13,3	13,6	13,6	13,1	11,1	11,1	10,6	10,6	10,9	10,9	9,7	11,5
2	9,4	8,3	8,6	9,2	9,2	8,9	10,3	10,6	11,1	12,2	11,1	11,4	11,4	9,7	9,4	9,1	8,6	8,6	9,1	8,6	8,6	8,3	8,1	7,5	9,5
3	6,7	7,5	8,3	6,9	6,1	7,9	9,7	10,9	10,8	10,8	10,0	10,3	10,3	10,3	10,3	10,3	9,2	9,2	9,6	8,6	8,3	6,4	5,8	5,0	8,7
4	4,4	4,4	4,4	4,2	4,4	5,6	6,9	9,2	9,8	11,1	10,8	11,1	9,6	9,8	9,7	9,2	9,2	9,2	7,2	7,5	6,1	5,3	4,4	4,2	7,4
5	3,9	3,9	4,4	3,9	4,2	4,7	5,3	6,7	8,6	10,3	9,7	10,0	10,8	11,6	12,5	12,2	10,8	8,3	6,1	6,4	7,2	6,7	6,9	8,6	7,7
6	8,1	6,9	7,2	7,9	8,9	8,1	8,6	9,4	10,8	11,7	11,9	12,5	12,2	12,5	11,1	10,8	10,0	9,7	10,0	10,3	10,0	9,2	9,5	9,7	9,9
7	7,2	6,1	6,1	4,2	3,1	3,3	5,3	8,1	10,0	10,0	11,9	9,7	7,5	6,7	4,2	3,3	3,0	1,4	1,7	2,5	2,2	4,7	5,3	4,2	5,5
8	3,3	5,0	4,4	2,5	4,2	3,3	4,4	6,4	6,9	8,6	8,9	8,9	9,2	9,7	8,6	10,3	10,0	8,6	7,8	8,9	5,8	5,8	5,0	6,7	6,8
9	3,6	4,4	6,4	6,1	6,4	7,5	7,9	8,9	10,0	9,4	8,3	8,6	10,0	10,6	9,2	10,0	7,5	6,9	5,8	6,4	5,0	4,4	4,7	4,4	7,2
10	4,4	5,0	5,0	4,4	5,0	4,4	4,7	5,6	6,9	8,6	8,9	8,6	8,6	7,2	7,8	7,8	7,2	6,4	5,8	6,1	4,7	5,0	5,3	5,0	6,2
11	5,3	5,3	4,2	4,7	5,9	3,9	5,6	5,9	6,9	7,2	7,5	6,1	4,7	6,1	7,8	8,3	8,3	6,9	6,1	5,3	5,3	5,0	5,3	5,0	5,9
12	5,3	4,7	4,7	4,7	4,2	4,2	5,9	6,9	8,3	9,2	8,6	9,2	8,9	8,1	8,1	9,4	8,9	7,5	7,2	6,7	5,8	6,9	6,9	6,1	6,9
13	4,7	5,3	5,0	5,6	5,3	5,3	7,5	7,2	9,4	9,7	10,0	9,7	9,4	9,7	9,7	9,4	9,4	8,9	7,5	6,4	6,9	5,8	6,4	6,4	7,5
14	5,3	6,7	8,3	8,1	7,9	7,5	8,3	9,7	9,4	11,1	9,7	9,7	8,9	8,6	7,8	8,1	8,1	7,8	7,5	7,2	6,1	5,0	5,0	4,4	7,8
15	4,7	6,1	7,5	8,3	7,6	6,8	7,2	7,5	8,3	10,3	10,0	8,6	9,4	9,4	10,0	10,0	9,6	8,3	6,7	6,4	6,7	6,1	6,4	6,7	7,9
16	6,9	6,9	6,4	5,9	6,4	7,5	7,5	8,6	9,2	9,7	9,7	8,9	9,4	9,1	9,7	9,7	9,2	8,6	8,1	6,4	4,7	4,7	4,4	4,7	7,6
17	4,4	5,6	5,3	5,0	5,3	5,6	6,3	6,9	8,6	10,0	10,3	10,3	10,3	10,3	10,6	9,7	6,4	5,8	5,6	5,8	6,1	4,2	2,5	3,1	6,8
18	4,4	5,0	6,1	5,9	6,1	6,9	6,1	6,4	5,8	8,9	9,2	9,7	9,2	8,1	5,6	5,9	6,1	8,9	7,8	5,6	5,9	6,7	6,4	6,4	6,8
19	5,6	5,6	5,3	4,4	7,5	8,1	7,8	6,7	8,3	9,4	8,1	6,4	6,7	5,6	4,7	3,4	3,1	6,7	6,1	6,7	5,9	5,6	5,9	5,6	6,2
20	6,1	5,2	3,1	3,3	4,2	5,0	5,9	6,1	6,7	6,4	8,1	8,1	8,1	8,9	8,9	5,6	6,4	7,8	8,1	9,2	8,6	8,4	7,8	7,2	6,8
21	6,9	6,4	7,9	8,3	8,9	7,5	6,4	7,5	7,2	8,9	8,3	7,2	5,9	5,9	5,8	9,7	9,2	8,9	11,1	11,1	10,9	11,4	11,1	11,1	8,5
22	10,0	9,7	10,0	10,3	10,0	10,8	10,8	11,4	11,7	11,1	12,2	11,1	10,3	8,9	9,2	9,7	9,4	9,4	9,1	9,7	9,1	9,1	8,3	7,5	10,0
23	6,4	6,1	6,7	6,9	7,2	5,9	6,7	6,9	6,9	8,9	6,4	5,0	5,3	6,1	6,7	5,3	4,4	6,2	7,5	8,1	8,3	6,8	5,3	5,0	6,6
24	5,8	5,3	6,1	6,1	5,8	6,2	7,8	7,8	6,9	6,4	5,8	5,9	5,9	3,3	2,8	4,2	3,9	2,5	5,9	6,9	6,4	5,3	2,2	1,9	5,3
25	1,4	1,4	2,2	3,1	3,4	4,2	3,9	6,1	6,7	5,8	4,7	5,6	3,6	4,2	4,4	3,1	1,9	2,2	3,1	5,0	5,0	6,7	6,9	7,2	4,2
26	5,6	4,4	5,0	5,3	4,4	2,8	2,8	5,8	6,7	7,2	9,2	9,2	8,1	6,7	6,7	4,2	6,4	8,1	6,9	5,7	6,1	5,9	5,3	4,4	6,0
27	4,2	3,9	3,1	3,1	2,2	3,3	4,7	3,1	4,4	6,9	6,1	5,6	5,6	4,7	4,2	4,4	3,8	4,4	5,6	5,6	5,3	5,9	4,7	5,0	4,6
28	5,0	5,3	3,6	4,2	4,2	2,8	5,0	4,7	4,2	6,1	8,3	11,1	11,1	13,6	13,1	12,5	12,5	11,9	11,4	11,4	10,6	10,6	11,1	11,1	8,6
29	10,6	11,1	11,4	11,1	10,8	10,3	11,4	12,2	12,2	11,7	11,9	11,6	11,1	10,0	11,1	11,1	10,0	9,4	9,4	10,0	8,6	6,9	6,1	5,6	10,2
30	5,6	5,3	5,8	5,6	5,9	6,2	6,7	6,9	6,9	6,9	8,1	8,6	6,9	5,3	3,9	6,9	6,9	6,9	5,6	5,9	6,4	5,6	5,9	4,7	6,2

Tabela 5,1 - Velocidades Médias Horária do Aeroporto de Praia (m/s), Junho de 1975

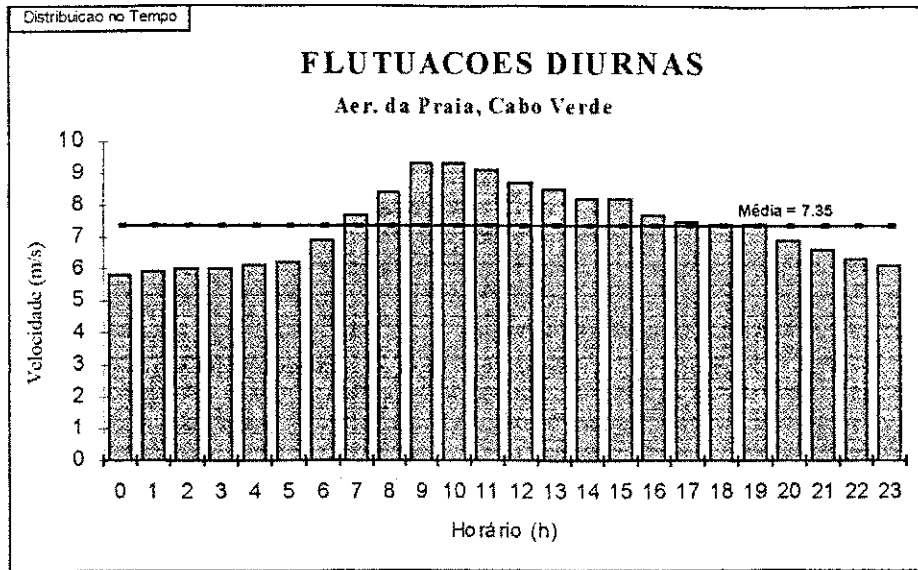


Figura 5.1a - Distribuição Temporal

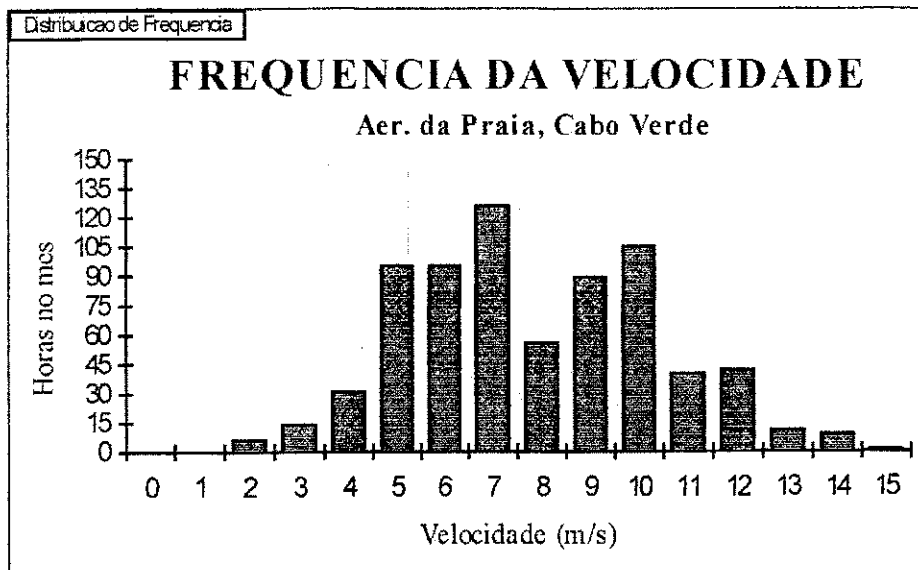


Figura 5.1b - Distribuição de Frequência

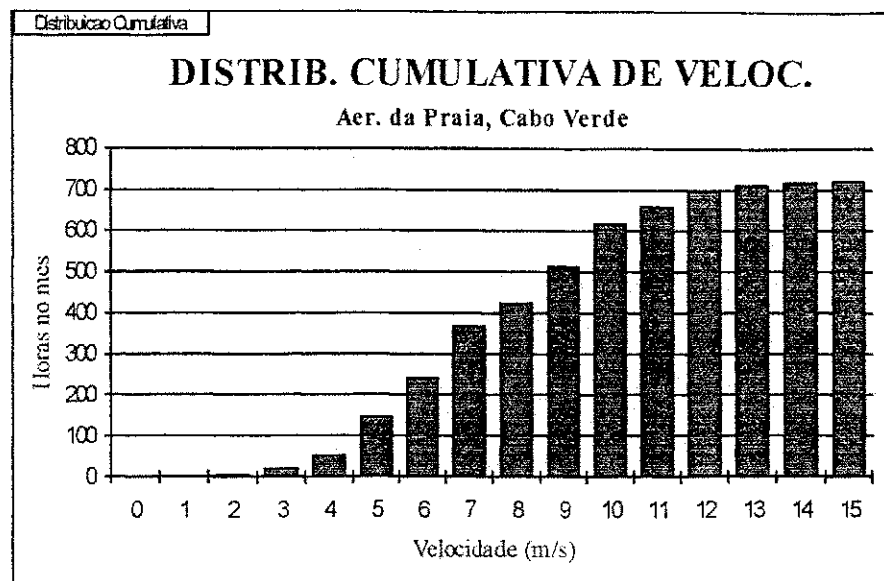


Figura 5.1c - Distribuição de Frequência Acumulada

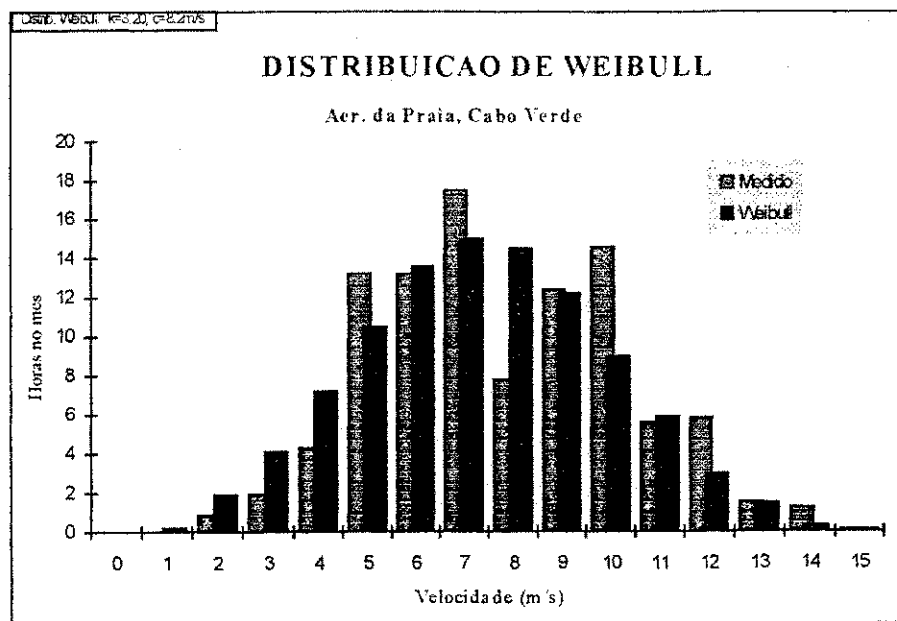


Figura 5.1d - Distribuição de Weibull

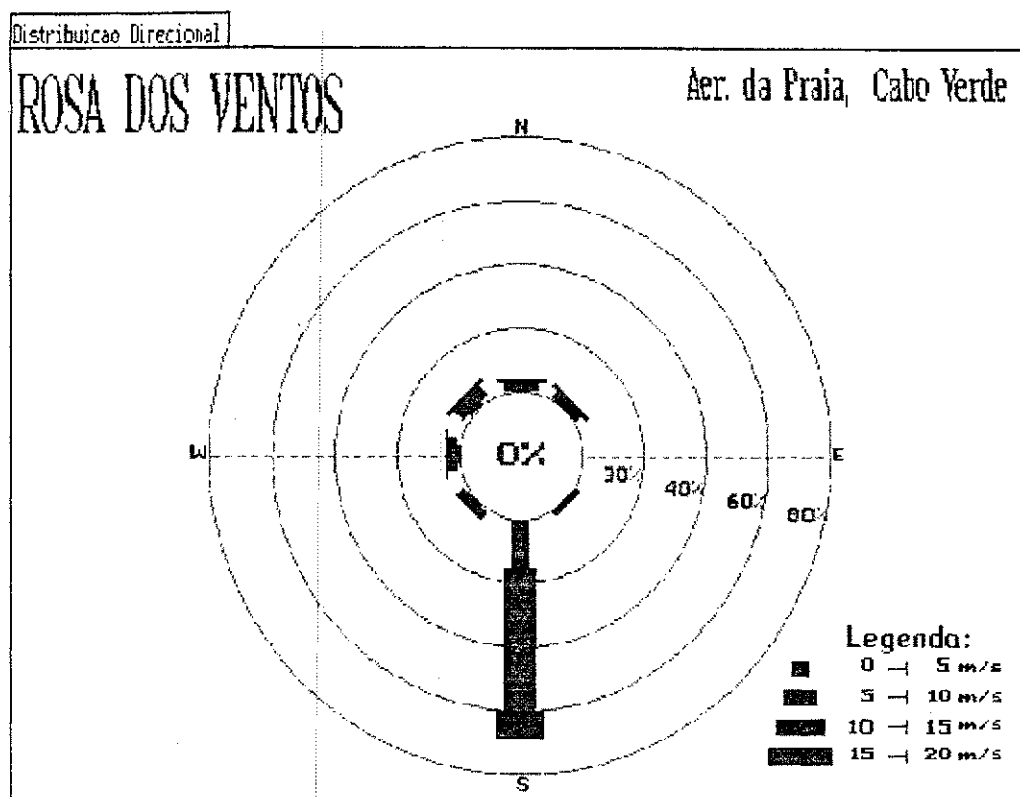


Figura 5.1e - Distribuição de Frequência Direcional (Rosa dos Ventos)

5.3 - Instrumentos de Medição:

O anemômetro utilizado no protótipo para os testes preliminares foi inteiramente desenvolvido pelo Núcleo de Energia (NERG) da UFPb, bem como o instrumento de indicação de direção.

Este anemômetro é do tipo rotacional com 4 (quatro) copos equidistantes diagonalmente, e fornece um sinal digital com variação de tensão (alta ou baixa) para cada 1/4 (um quarto) de volta que é completada. Ele trabalha com um elemento eletrônico baseado no efeito Hall, ou seja, sensível a variação de campo magnético. No instrumento, este campo é fornecido, por 8 (oito) pequenos ímãs permanentes, colocados em um anel giratório preso ao suporte dos copos.

O indicador de direção funciona com o mesmo tipo de sensor posicionado na base giratória que suporta também o elemento direcionador (grimpa). A direção é codificada usando-se o código Gray para representação das 8 (oito) posições possíveis.

5.4 - Material Usado:

Na Tabela 5.2a e na Tabela 5.2b, temos listado os componentes dos módulos de hardware de acordo com o projeto apresentado no Apêndice B, e na Tabela 5.2c estão enumerados os componentes usados na montagem do protótipo, além de bornes, fio de wire-wrap e soquetes.

DESCRIÇÃO	QUANTIDADE	NOME(S)
capacitor	03	C1 - C3
resistor	03	R1 - R3
diodo	01	D1
transistor	01	Q1
led	01	L1
crystal	01	Y1
conector	02	CN1 - CN2
microcontrolador	01	U1
trava (latch)	01	U2
memória EPROM	01	U3
conversor TTL-RS232	02	U4 - U5
chave pressão	01	S1

Tabela 5.2a - Componentes do Módulo MLGM

DESCRIÇÃO	QUANTIDADE	NOME(S)
capacitor	06	C1 - C6
resistor	07	R1 - R7
diodo	01	D1
transistor	01	Q1
led	01	L1
crystal	02	Y1 - Y2
conector	02	CN1 - CN2
microcontrolador	01	U1
trava (latch)	05	U2, U9 - U12
memória EPROM	01	U3
conversor TTL-RS232	02	U4 - U5
selecionador (demux)	01	U6
porta lógica	02	U7 - U8
contador assíncrono	02	U13 - U14
relógio de tempo real	01	U15
chave (liga-desl., pressão)	02	S1 - S2

Tabela 5.2b - Componentes do Módulo MPAD

DESCRIÇÃO	QUANTIDADE	NOME(S)
capacitor	42	C1 - C42
led	01	L1
fusível	01	F1
diodo	05	D1 - D5
potenciômetro	01	P1
resistor	07	R1 - R7
conector 8 pinos	01	CN1
cristal	02	Y1 - Y2
memória EPROM	01	U3
memória EEPROM	01	U4
selecionador (demux)	01	U6
trava (latch)	05	U2, U9 - U12
microcontrolador	01	U1
transformador de tensão	01	T1
contador assíncrono	01	U14
transistor	01	Q1
display de cristal líquido	01	DSP1
conversor TTL-RS232	02	U21 - U22
regulador de tensão	03	M1 - M3
relógio de tempo real	01	U15
chave (liga-desl, pressão)	03	S1 - S2

Tabela 5.2c - Componentes Usados no Protótipo

5.5 - Resultados:

31/05/94	Demonstracao (nome da empresa)	11:25
ESTATISTICA GERAL		
Unidade: 94	Periodo-mes: 6	
. Local : Aer. da Praia, Cabo Verde	. Data Inicio: 01/06/75	
. Elevacao: 530 m (Nível do Mar)	. Data Final : 30/06/75	
. Altura : 10 m (Nível do Solo)		
Reta de Conversao (rpm para m/s):	Coef. Angular : 0,024688	
	Coef. Linear : 0,023660	
Arquivos: 94010675.AMS/.MED/.MAX		
Veloc. Media: 7,35 m/s	Weibull (forma): k= 3,20	
Desv.-Padrao: 2,512 m/s	Weibull(escala) : c= 8,20	

Figura 5.2a - Relatório de Estatística Geral

31/05/94	Demonstracao (nome da empresa)		11:25																						
RELATORIO DE MEDIAS (m/s)																									
Unidade: 94	Periodo-mes: 6																								
. Local : Aer. da Praia, Cabo Verde	. Data Inicio: 01/06/75																								
. Elevacao: 530 m (Nivel do Mar)	. Data Final : 30/06/75																								
. Altura : 10 m (Nivel do Solo)	Coef. Angular : 0,024688																								
Reta de Conversao (rpm para m/s):	Coef. Linear : 0,023660																								
Hora Local																									
Des	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Media
1	9,4	10,0	8,8	9,4	9,7	10,0	10,3	11,4	11,3	13,9	13,3	14,2	13,3	13,6	13,6	13,1	11,1	11,1	10,6	10,6	10,9	10,9	9,7	11,50	
2	9,4	8,3	8,6	8,2	9,2	8,9	10,8	10,6	11,1	12,2	11,1	11,4	11,4	9,7	9,4	9,1	8,6	8,6	9,1	8,6	8,6	8,3	9,1	7,5	9,47
3	6,7	7,3	8,3	6,9	6,1	7,9	9,7	10,9	10,8	10,0	10,3	10,3	10,3	10,3	10,3	9,2	9,2	9,6	8,6	8,3	6,4	5,8	5,0	8,72	
4	4,4	4,4	4,4	4,2	4,4	5,6	6,9	8,2	9,8	11,1	10,8	11,1	9,6	9,3	9,7	9,2	9,2	9,2	7,2	7,5	6,1	5,3	4,4	4,2	7,40
5	3,2	3,3	4,4	3,9	4,2	4,7	5,3	6,7	8,6	10,3	9,7	10,0	10,8	11,3	12,3	12,2	11,8	8,3	8,1	6,4	7,2	6,9	6,9	8,6	7,35
6	8,1	6,9	7,2	7,9	8,9	9,1	8,6	9,4	10,8	11,7	11,9	12,5	12,2	11,5	11,1	10,8	10,0	9,7	10,0	10,3	10,0	8,2	8,5	9,7	8,89
7	7,2	6,1	6,1	4,2	3,1	3,3	5,3	8,1	10,0	10,0	11,9	9,7	7,8	4,7	4,2	3,3	3,0	1,4	1,7	2,5	2,2	4,7	5,3	4,2	5,49
8	3,3	5,3	4,4	3,5	4,2	3,3	4,4	6,4	6,9	8,6	8,3	8,2	8,2	8,7	8,6	10,3	10,0	8,6	7,8	8,9	5,8	5,8	5,0	6,7	6,80
9	3,6	8,4	6,4	6,1	6,4	7,5	7,3	8,9	10,0	8,4	8,3	8,8	10,8	10,3	9,2	10,0	7,5	6,9	8,2	6,4	5,0	4,4	4,7	4,4	7,18
10	4,4	5,9	5,3	4,4	5,0	4,4	4,7	5,6	6,9	8,6	8,2	8,5	8,6	7,2	7,8	7,2	7,2	6,4	5,8	6,1	4,7	5,0	5,3	5,0	6,18
11	5,3	5,3	4,2	4,7	5,9	3,9	5,6	6,9	6,9	7,2	7,5	6,1	4,7	5,1	7,8	8,3	8,3	6,9	6,1	5,3	5,3	5,0	5,3	5,0	5,34
12	5,3	4,7	4,7	4,7	4,2	4,2	5,9	6,9	9,3	9,2	8,6	9,2	8,9	8,1	8,1	9,4	8,9	7,5	7,2	6,7	5,9	6,9	6,9	6,1	6,35
13	4,7	6,3	5,3	5,6	5,3	5,3	7,6	7,2	8,4	8,7	10,0	8,7	8,4	8,7	9,4	9,4	8,9	7,8	6,4	6,9	5,8	6,4	6,4	6,4	7,32
14	5,3	6,7	5,3	3,1	7,9	7,6	8,3	3,7	9,4	11,1	9,7	9,7	8,9	8,6	7,8	5,1	9,1	7,8	7,8	7,2	6,1	5,0	5,0	4,4	7,78
15	4,7	6,1	7,5	5,3	7,6	6,8	7,2	7,5	8,3	10,3	10,0	8,6	9,4	8,4	10,0	10,0	9,6	8,3	6,7	6,4	6,7	6,1	6,4	6,7	7,26
16	6,3	6,3	6,4	5,9	6,4	7,5	7,5	8,6	8,2	9,7	9,7	8,9	8,4	8,1	9,7	9,7	9,2	8,6	8,1	6,4	4,7	4,7	4,4	4,7	7,00
17	4,4	5,6	5,3	5,0	5,3	5,6	6,3	6,9	8,6	10,0	10,3	10,3	10,3	10,3	10,6	9,7	6,4	5,8	5,8	5,8	6,1	4,2	2,5	3,1	6,83
18	4,4	5,3	6,1	5,9	6,1	6,9	6,1	6,4	5,8	8,9	8,2	9,7	8,2	5,1	5,6	5,2	6,1	8,9	7,2	5,6	5,9	6,7	6,4	6,4	6,20
19	5,6	5,6	7,5	4,4	7,5	8,1	7,8	6,7	9,3	9,4	8,1	6,4	6,7	5,6	4,7	3,4	3,1	6,7	6,1	6,7	5,9	5,6	5,9	5,6	6,22
20	6,1	5,2	3,1	3,3	4,2	5,0	5,9	6,1	6,7	6,4	8,1	8,1	5,1	5,8	8,9	5,6	6,4	7,8	8,1	8,3	8,6	6,4	7,8	7,2	6,30
21	6,3	6,4	7,2	6,3	8,9	7,5	6,4	7,5	7,2	8,9	8,3	7,2	5,3	5,3	5,8	9,7	8,2	8,9	11,1	11,1	10,9	11,4	11,1	11,1	8,43
22	10,9	9,7	10,3	10,3	10,0	10,8	10,8	11,4	11,7	11,1	10,2	11,1	10,3	9,9	9,2	9,7	9,4	9,4	9,1	9,7	9,1	9,1	9,3	7,5	9,36
23	6,4	6,1	6,7	6,9	7,2	3,9	6,7	6,9	6,9	8,9	6,4	5,0	5,3	6,1	6,7	5,3	4,4	6,2	7,5	5,1	8,3	8,8	5,3	5,0	6,46
24	5,9	5,3	6,1	6,1	5,8	6,2	7,8	7,8	6,9	6,4	5,8	5,9	5,3	3,3	2,8	4,2	3,9	2,5	5,9	6,9	6,4	5,3	2,2	1,9	5,30
25	1,4	1,4	1,2	3,1	3,4	4,2	5,9	6,1	6,7	5,8	4,7	5,6	3,6	4,2	4,4	3,1	1,9	2,2	3,1	5,0	5,0	6,7	6,9	7,2	4,24
26	6,6	4,4	5,0	5,3	4,4	2,8	2,8	5,8	6,7	7,2	9,2	9,2	6,1	6,7	6,7	4,2	6,4	8,1	6,9	5,7	6,1	5,9	5,3	4,4	5,96
27	4,2	5,9	3,1	3,1	2,2	3,3	4,7	3,1	4,4	6,9	6,1	5,6	5,6	4,7	4,2	4,4	3,8	4,4	5,6	5,6	5,3	5,9	4,7	5,0	4,58
28	5,0	5,3	3,6	4,2	4,2	2,8	5,0	4,7	4,2	6,1	8,3	11,1	11,1	13,6	13,1	12,5	12,5	11,3	11,4	11,4	10,6	10,6	11,1	11,1	8,56
29	10,6	11,1	11,4	11,1	10,8	10,3	11,4	12,2	12,2	11,7	11,9	12,6	11,1	10,0	11,1	11,1	9,0	9,4	9,4	10,0	8,6	6,9	6,1	5,6	10,23
30	5,6	5,3	5,8	5,6	5,9	6,2	6,7	6,9	6,9	8,1	8,8	6,9	5,3	3,9	6,9	6,9	6,9	6,9	5,5	5,3	6,4	5,8	5,9	4,7	6,23
Med	5,9	5,9	6,0	6,0	6,1	6,2	6,9	7,7	8,4	9,3	9,3	9,1	8,7	8,5	8,2	8,2	7,7	7,5	7,4	7,4	6,9	6,6	6,3	6,1	7,35

----- Fim.

Figura 5.2b - Relatório de Médias Horárias (m/s)

31/05/94		Demonstracao (nome da empresa)		11:25	
DISTRIBUICAO DE FREQUENCIA					
Unidade: 94		Periodo-mes: 6			
. Local: Aer. da Praia, Cabo Verde					
. Elevacao: 530 m (Nível do Mar)					
. Altura: 10 m (Nível do Solo)					
Faixa Veloc (m/s)	Horas (hrs)	Frequencia Relativa (%)	Horas Acum. (hrs)	Freq.Rel. Acumulada (%)	
= 0	0	0,000	0	0,000	
< 1	0	0,000	0	0,000	
< 2	6	0,833	6	0,833	
< 3	14	1,944	20	2,778	
< 4	31	4,306	51	7,083	
< 5	95	13,194	146	20,278	
< 6	95	13,194	241	33,472	
< 7	126	17,500	367	50,972	
< 8	56	7,778	423	58,750	
< 9	89	12,361	512	71,111	
< 10	105	14,583	617	85,694	
< 11	40	5,556	657	91,250	
< 12	42	5,833	699	97,083	
< 13	11	1,528	710	98,611	
< 14	9	1,250	719	99,861	
< 15	1	0,139	720	100,000	
< 16	0	0,000	720	100,000	
< 17	0	0,000	720	100,000	
< 18	0	0,000	720	100,000	
< 19	0	0,000	720	100,000	
< 20	0	0,000	720	100,000	
< 21	0	0,000	720	100,000	
< 22	0	0,000	720	100,000	
< 23	0	0,000	720	100,000	
< 24	0	0,000	720	100,000	
≥ 24	0	0,000	720	100,000	
----- Fim.					

Figura 5.2c - Relatório de Distribuição de Frequência

06/06/94	Demonstracao (nome da empresa)	10:48
RELATORIO DE AMOSTRAS		
Unidade: 94	Amostragem:	
. Local:	. Data Inicio: 18/04/94	
. Elevacao: 530 m (Nivel do Mar)	. Hora Inicio: 19:00	
. Altura: 10 m (Nivel do Solo)	. Intervalo: 1 min.	
. Anemomet: 1		
Velocidade em rpm		
Conj. 0001: 0000/6, 0010/6, 0032/6, 0022/6, 0025/6, 0009/6, 0032-6, 0029/6, Conj. 0002: 0023/6, 0014/6, 0031/6, 0006/6, 0002/6, 0031-6, 0277/6, 0053/6, Conj. 0003: 0001/6, 0009/6, 0009/6, 0011/6, 0277-6, 0003/6, 0007/6, 0013/6, Conj. 0004: 0008/6, 0015/6, 0051/6, 0051-6, 0063/6, 0040/6, 0018/6, 0025/6, Conj. 0005: 0040/6, 0026/6, 0063-6, 0013/6, 0002/6, 0000/6, 0000/6, 0000/6, Conj. 0006: 0003/6, 0013-6, 0011/6, 0000/6, 0000/6, 0011/6, 0000/6, 0000/6, Conj. 0007: 0011-6, 0009/6, 0021/6, 0002/6, 0000/6, 0000/6, 0000/6, 0021-6, Conj. 0008: 0000/6, 0016/6, 0000/6, 0000/6, 0000/6, 0006/6, 0016-6, 0022/6, Conj. 0009: 0010/6, 0004/6, 0025/6, 0005/6, 0002/6, 0025-6, 0000/6, 0010/6, Conj. 0010: 0001/6, 0001/6, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0011: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0012: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0013: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0014: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0058: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0059: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0060: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0061: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0062: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0063: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, Conj. 0064: 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, 0000/0, ----- Fim.		

Figura 5.2d - Relatório de Amostras

CONSIDERAÇÕES FINAIS

Estando sempre propensos a obter um projeto de pesquisa que redundasse em um produto aplicável, útil e prático, assumimos uma conduta de trabalho receptiva e aberta às sugestões e críticas que contribuíssem para alcançar este intuito. Mas, como todo trabalho de pesquisa, ou qualquer projeto fundamentado em planejamento prévio, também este sofreu mutações em seus objetivos preliminares, durante o seu desenrolar. Se por um lado, esta postura de abertura, trouxe benefícios como a obtenção de um instrumento mais prático, de outro, ela também aflorou dificuldades inerentes às mudanças de percurso, as quais, algumas vezes, chegaram a alterar a estrutura do projeto. Sendo assim, só nos resta concordar integralmente com a lúcida afirmação de Fisher, 1989:

"Temos que aceitar a instabilidade dos requisitos como um fato da vida,
e não condená-la como o resultado de um raciocínio malconduzido."

Estamos cientes das limitações do trabalho apresentado nesta dissertação, que como toda pesquisa, também esta não poderia dissecar totalmente o assunto focalizado.

Diante do exposto, dos fatos, e tentando analisar com imparcialidade, chegamos ao final com a impressão do dever cumprido e de termos conseguido o melhor, dentro do contexto em que se conduziu a pesquisa.

Sinopse do Trabalho

Inicialmente colocamos a necessidade de se pesquisar e aproveitar fontes alternativas de energia, e de preferência, fontes ecologicamente limpas. Mostramos também a relação existente entre a evolução da humanidade e o seu crescente consumo

de energia, e por isso mesmo, a racionalização do uso e exploração, aliada à pesquisa, serão uma constante na história humana.

Seguindo, fizemos uma revisão teórica sobre ventos, aproveitamento de energia eólica, potência eólica, caracterização de regime de ventos e sistemas conversores de energia eólica; conceitos essenciais ao desenvolvimento do trabalho.

A partir deste ponto, apresentamos o sistema proposto em linhas gerais, através da descrição de suas características como um todo, e de sua divisão em dois sub-sistemas funcionalmente distintos: sub-sistema coletor de dados, responsável pela amostragem da velocidade do vento, e sub-sistema de processamento, redutor de dados brutos.

Continuando com a descrição do sistema, detalhamos os módulos componentes do hardware do projeto. Apresentamos a arquitetura dos módulos em diagramas de bloco, e de modo mais aprofundado, colocamos algumas observações técnicas a respeito de hardware e software básico do SAADE.

Descrevemos e discutimos o sub-sistema de processamento em sua totalidade na etapa seguinte deste trabalho, composto principalmente pelo software de análise estatística. A Metodologia de programação/projeto, a operação, o cálculo e os resultados são abordados em detalhes.

Também realizamos avaliações de resultados obtidos com o protótipo implementado, e ao mesmo tempo, apresentamos observações críticas em relação ao projeto proposto e ao sistema testado no laboratório.

Conclusão

Ainda que falte muito para ser consolidado como produto industrial, acreditamos que o SAADE contemplou às características preliminares impostas ao projeto, na sua totalidade, e que, com um pouco mais de trabalho, o sistema pode se tornar um instrumento útil para aplicações na climatologia, em estações sinóticas, em trabalhos de pesquisas meteorológicas, na agrometeorologia, no projeto de SCEE's, etc.

O sistema SAADE, concretizado pelo protótipo construído, que o implementou parcialmente, realizou todas as suas tarefas: aquisição de dados, acondicionamento de dados, transporte de dados, redução e análise de dados. Confiabilidade e robustez não puderam ser avaliadas devido às restrições de ordem financeira durante a construção do protótipo, apesar de seu baixíssimo custo quando comparado a similares estrangeiros.

As informações mais utilizadas por projetistas e especialistas em sistemas conversores de energia eólica, tais como valores médios, desvios-padrão, histogramas de distribuição de frequência, rosa dos ventos, etc, são geradas com base em padrões internacionais, através da utilização de menus de escolha do software da UAD, que facilitam enormemente o acesso a todas elas.

É importante salientarmos que o projeto se propõe a realizar análise do regime de ventos em microescala em uma determinada região, trabalhando com valores de velocidades médias horárias, ou seja, com uma massa considerável de dados, e portanto microcomputadores com boa performance de processamento e memória de armazenamento razoável são recomendados. Para a realização de análises sinóticas devemos consultar outros trabalhos, como por exemplo Bandeira, 1990.

Sugestões para Continuidade do Trabalho

Podemos dividir as sugestões de melhoria do sistema, em dois grupos distintos, classificadas de acordo com suas complexidades de realização, esforço e tempo para adaptação:

Melhorias imediatas:

1. Acrescentar ao protótipo do sistema desenvolvido, todas as funções-comando propostas no projeto de hardware.

2. Acrescentar novos formatos gráficos para as figuras de turbinas eólicas no software de análise de dados eólicos (somente o formato PCX é aceito pelo software da UAD).
3. Construir um editor de dados ASCII próprio para edição dos dados do sistema.
4. Acrescentar uma classe de tratamento de erros, no software de análise, que evite ou recupere algumas situações geradoras de interrupção do programa.
5. Acrescentar uma classe de tratamento de eventos gerados por dispositivos apontadores (mouse).

Melhorias complexas:

1. Acrescentar novos sensores, que permitam a realização de medidas de temperatura, de umidade, de radiação solar e insolação, de pressão, de evaporação, e de precipitação.
2. Fornecer novas opções de transporte de dados coletados: via modem, via rádio transceptor, etc.
3. Colocar painel solar (matriz de células fotovoltaicas) para carregamento automático da bateria.
4. Colocar display e teclado para manipulação remota da unidade de aquisição de dados.
5. Criar artifícios para endereçamento de quantidade de memórias maiores (até 1 Mbyte) na unidade coletora.

APÊNDICE A

DISTRIBUIÇÃO DE WEIBULL

A.1 - Introdução:

Após desenhar um número razoável de histogramas de duração de velocidades ou histogramas de frequência de velocidade e aproximá-los por curvas suaves, nota-se que a forma destas curvas é muito similar. E isto fica mais claro ainda se os valores de velocidade do vento são divididos pela velocidade média da distribuição em particular, ou seja, se eles se tornam adimensionais [Lysen, 1983].

Em situações como esta, os pesquisadores costumam procurar funções matemáticas que se aproximem das curvas de frequência e duração, tanto quanto possível, para serem usadas como ferramentas para predizer a saída de turbinas eólicas.

Vários autores [Lysen,1983; Justus,1978] demonstram a boa aproximação aos dados experimentais, obtida com a Distribuição de Weibull, classificada como distribuição contínua de valores extremos [Soong, 1986]. Este apêndice trata da distribuição de Weibull para velocidade de vento, usada extensivamente em aplicações de cargas de vento e recentemente para estudos de energia eólica [Justus, 1978].

Esta distribuição permite a avaliação de várias importantes propriedades da distribuição do vento em termos, apenas, de dois parâmetros (c e k), conhecidos como fator de escala e fator de forma da curva de Weibull, respectivamente.

Nos itens seguintes apresentamos a formulação matemática adotada na representação e manipulação deste modelo de distribuição, bem como a estimação dos seus parâmetros (como utilizada no programa de análise estatística do SAADE).

A.2 - Formulação Matemática:

A distribuição de Weibull, especificamente a função densidade de probabilidade (fdp) $p(V)$, é dada por:

$$p(V) = \frac{k}{c} \left(\frac{V}{c} \right)^{k-1} \cdot e^{-\left(\frac{V}{c}\right)^k} \quad (\text{A.2.1})$$

onde c é chamado fator de escala (em unidade de velocidade) e k é o fator de forma (adimensional). A função de probabilidade acumulada (FDP) equivalente é dada por:

$$P(V \leq V_x) = \int_0^{V_x} p(V) \cdot dV = 1 - e^{-\left(\frac{V_x}{c}\right)^k} \quad (\text{A.2.2})$$

A probabilidade de ocorrência de ventos com velocidade abaixo de algum nível de interesse V_x é dada pela equação (A.2.2), e similarmente, a probabilidade de velocidades acima de V_x é dada por:

$$P(V \geq V_x) = \int_{V_x}^{\infty} p(V) \cdot dV = e^{-\left(\frac{V_x}{c}\right)^k} \quad (\text{A.2.3})$$

e a probabilidade de se ter velocidades entre dois limites de interesse V_x e V_y é:

$$P(V_x \leq V \leq V_y) = \int_{V_x}^{V_y} p(V) \cdot dV = e^{-\left(\frac{V_x}{c}\right)^k} - e^{-\left(\frac{V_y}{c}\right)^k} \quad (\text{A.2.4})$$

Igualando-se a 1/2 a equação (A.2.3), obtemos a velocidade de vento mediana (percentil 50) $V_{mediana}$:

$$V_{mediana} = c \cdot (\ln 2)^{1/k} \quad (\text{A.2.5})$$

A velocidade média V_m (ou \bar{V}) pode ser encontrada da distribuição de Weibull por:

$$\bar{V} = \int_0^{\infty} V \cdot p(V) \cdot dV = c \cdot \Gamma(1+1/k) \quad (\text{A.2.6})$$

onde $\Gamma(x)$ é função gama usual definida por [Butkov, 1983]:

$$\Gamma(x) = \int_0^{\infty} \varphi^{x-1} \cdot e^{-\varphi} \cdot d\varphi \quad (\text{A.2.7})$$

O desvio-padrão σ da distribuição é dado por:

$$\sigma^2 = \int_0^{\infty} (V - \bar{V})^2 \cdot p(V) \cdot dV = c^2 \cdot [\Gamma(1+2/k) - \Gamma^2(1+1/k)] \quad (\text{A.2.8})$$

E a média do cubo da velocidade do vento, por:

$$\langle V^3 \rangle = \int_0^{\infty} V^3 \cdot p(V) \cdot dV = c^3 \cdot \Gamma(1+3/k) \quad (\text{A.2.9})$$

De posse destas equações podemos calcular a energia eólica média disponível (equação 1.1.2), o fator padrão de energia $\langle V^3 \rangle / V_m$, a potência extraível ou fator de capacidade de um dado projeto de SCEE, e a probabilidade da velocidade do vento estar em certos intervalos de interesse (e.g., abaixo da veloc. cut-in, acima da veloc. nominal, acima da veloc. cut-out) [Justus, 1978].

A.3 - Estimação dos Parâmetros de Weibull:

A distribuição de Weibull mostra sua utilidade quando os dados de vento de uma estação de referência estão sendo usados para prever o regime de ventos em regiões ao redor daquela estação. A idéia é que somente as velocidades média anual ou mensal sejam suficientes para prever a distribuição de frequência completa do ano ou do mês [Justus, 1978]. Se as séries temporais de dados medidos estão disponíveis na localidade

e altura desejadas, então, é claro, não existe muita necessidade de análises em termos de distribuições de probabilidade [Lysen, 1983].

Existem pelo menos três métodos de se extrair os parâmetros da distribuição de Weibull de um determinado conjunto de dados: papel de Weibull, análise do fator padrão de energia e análise do desvio-padrão [Lysen, 1983]. Por questões de facilidade de cálculos computacionais, adotamos o terceiro método, para a estimação destes parâmetros, no programa de análise estatística do SAADE. Todos os três métodos citados anteriormente são discutidos por Lysen, 1983.

A obtenção do fator de forma k da curva de Weibull é feita através da relação entre o desvio-padrão (equação A.2.8) e a velocidade média do conjunto de dados analisados (equação A.2.6), como a seguir:

$$\frac{\sigma}{\bar{V}} = \sqrt{\frac{\Gamma(1+2/k) - \Gamma^2(1+1/k)}{\Gamma(1+1/k)}} \quad (\text{A.3.1})$$

Computacionalmente, a função Gama pode ser obtida através de tabelas previamente calculadas e armazenadas, por demoradas computações de integrais, ou por função analítica que aproxime a função original. Lysen, 1983, usa:

$$\Gamma(1+1/k) = G(k) = 0.568 + \frac{0.434}{k} \quad (\text{A.3.2})$$

com precisão melhor do que 0,2% para $1 < k < 3,6$.

Após algumas manipulações algébricas chega-se à seguinte equação transcendental:

$$\frac{2}{k} \left[\ln(G(k)) + \ln \left(\left(\frac{\sigma}{\bar{V}} \right)^2 + 1 \right) \right] = \ln(G(k)) \quad (\text{A.3.3})$$

cuja raiz pode ser obtida numericamente através de vários métodos: das tentativas, das tentativas com interpolação linear, de Newton-Raphson, de Graeffe, dentre outros

[Pacitti, 1977]. O método das tentativas apresenta esforço computacional exponencialmente proporcional à precisão desejada das raízes procuradas, mas ainda assim, foi adotado no programa do SAADE por sua simplicidade de implementação e por não necessitarmos de raiz com precisão maior que décimos do valor original, pois, valores do fator de forma k com mais de uma casa decimal são raramente utilizados.

Depois da obtenção do fator de forma k , o fator de escala c pode ser calculado a partir da equação A.2.6:

$$c = \frac{\bar{V}}{\Gamma(1+1/k)} \quad (\text{A.3.4})$$

Todas as referências indicadas na bibliografia e que tratam de energia eólica, devem ser consultadas para discussões mais aprofundadas sobre o assunto presente.

APÊNDICE B

DESENHOS ESQUEMÁTICOS

B.1 - Introdução:

Apresentamos os circuitos desenvolvidos para a realização das funções do SAADE, inicialmente os desenhos do **MPAD**, depois do **MLGM** e por último os desenhos esquemáticos do protótipo usado nos testes preliminares.

Os desenhos esquemáticos foram divididos em pranchas de formato A4 de acordo com a função desempenhada por cada parte do circuito, e obedecendo critérios subjetivos de estética e de densidade de componentes por prancha.

B.2 - Módulo Portátil de Aquisição de Dados

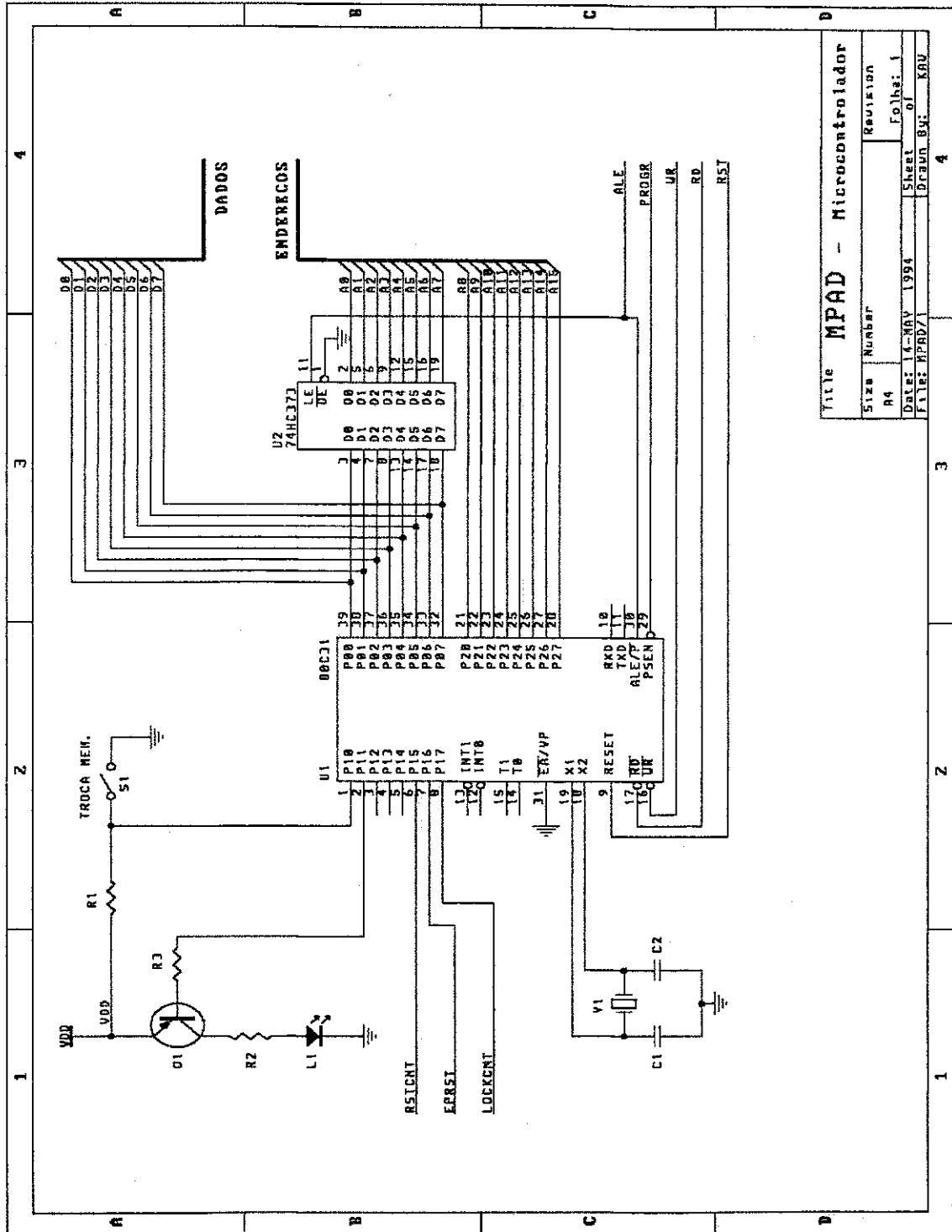


Figura B.1 - Microcontrolador - MPAD

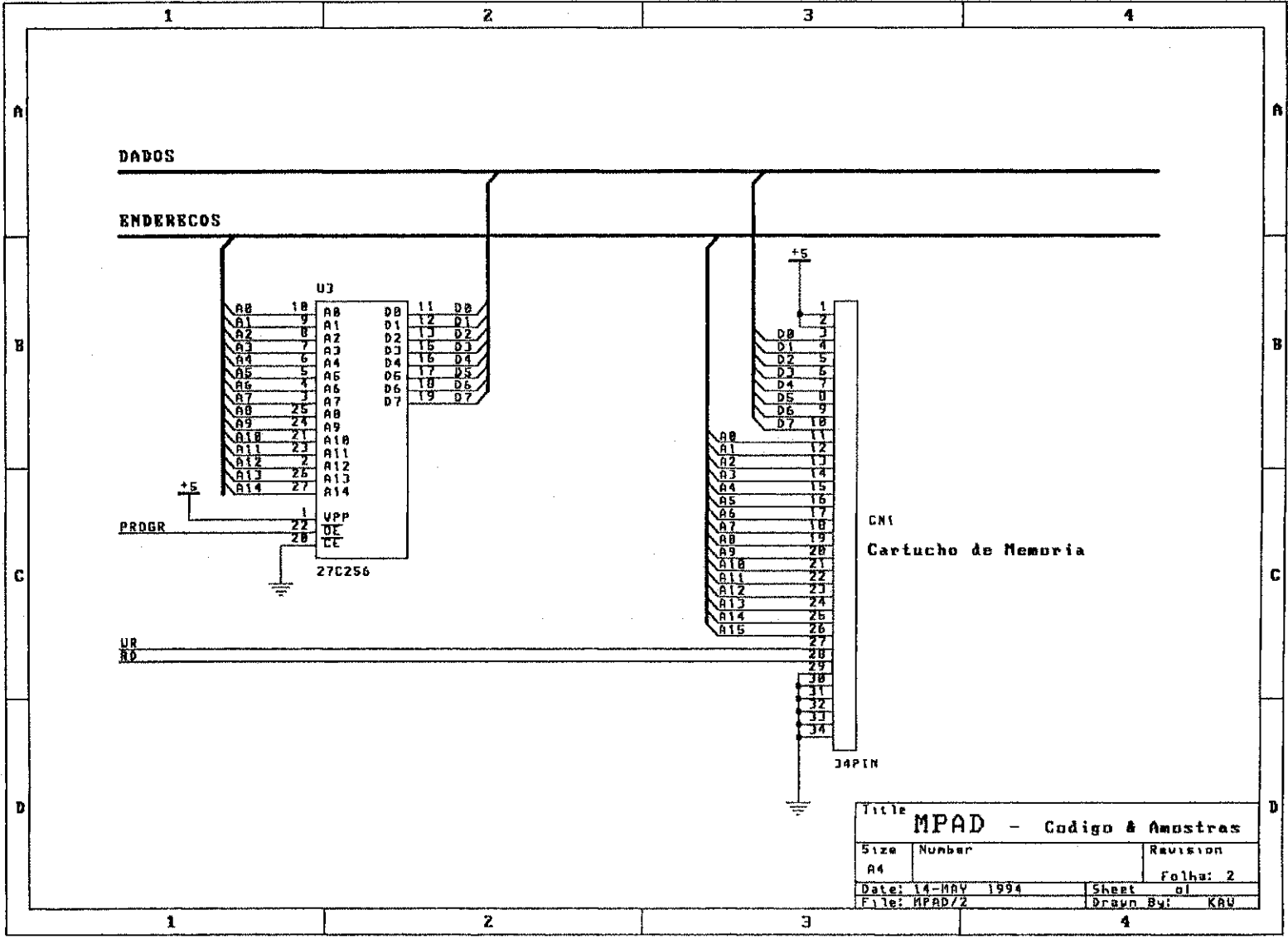
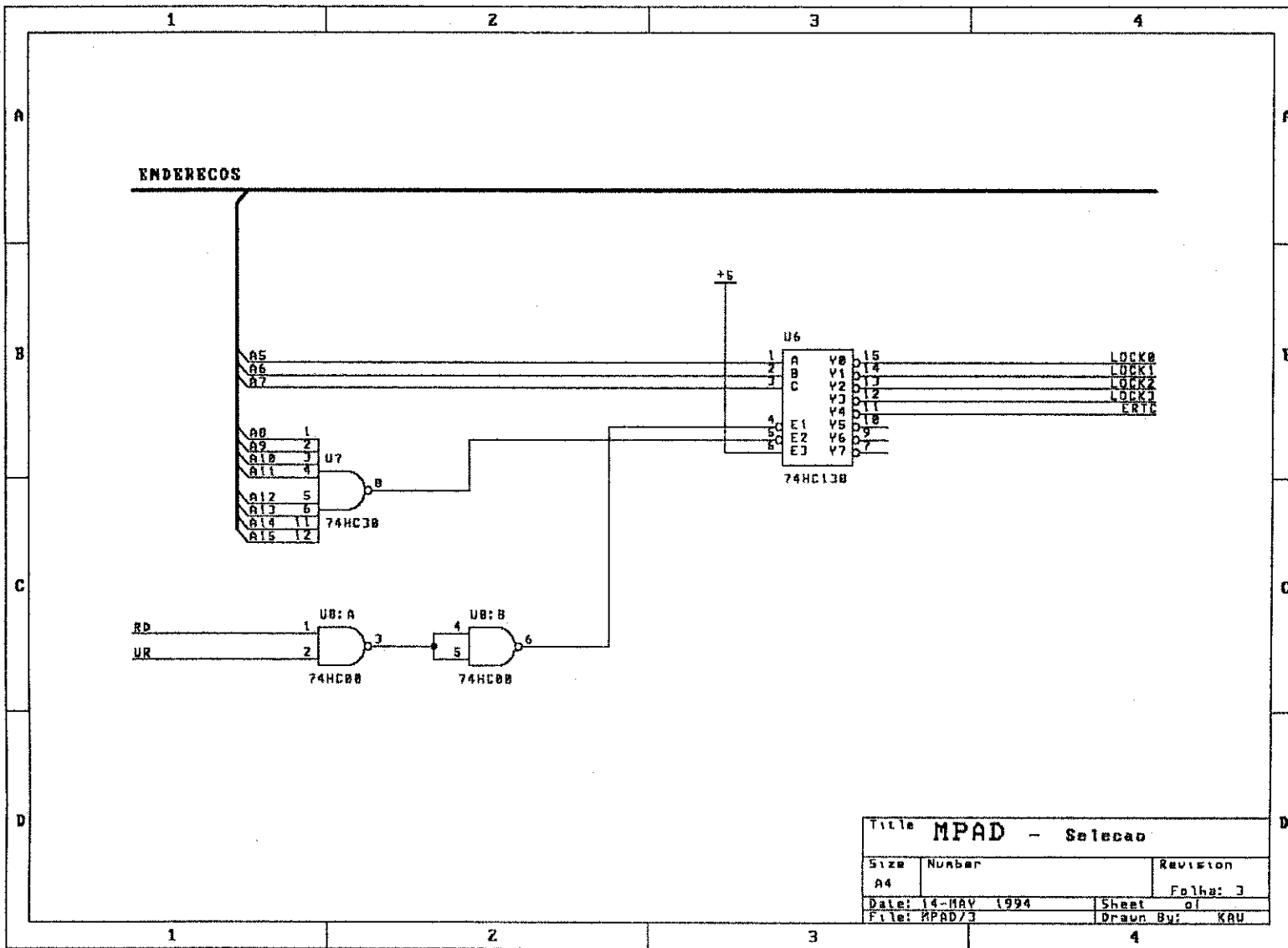


Figura B.2 - Memória de Código e Amostras - MPAD

Figura B.3 - Seleção de Dispositivos - MPAD



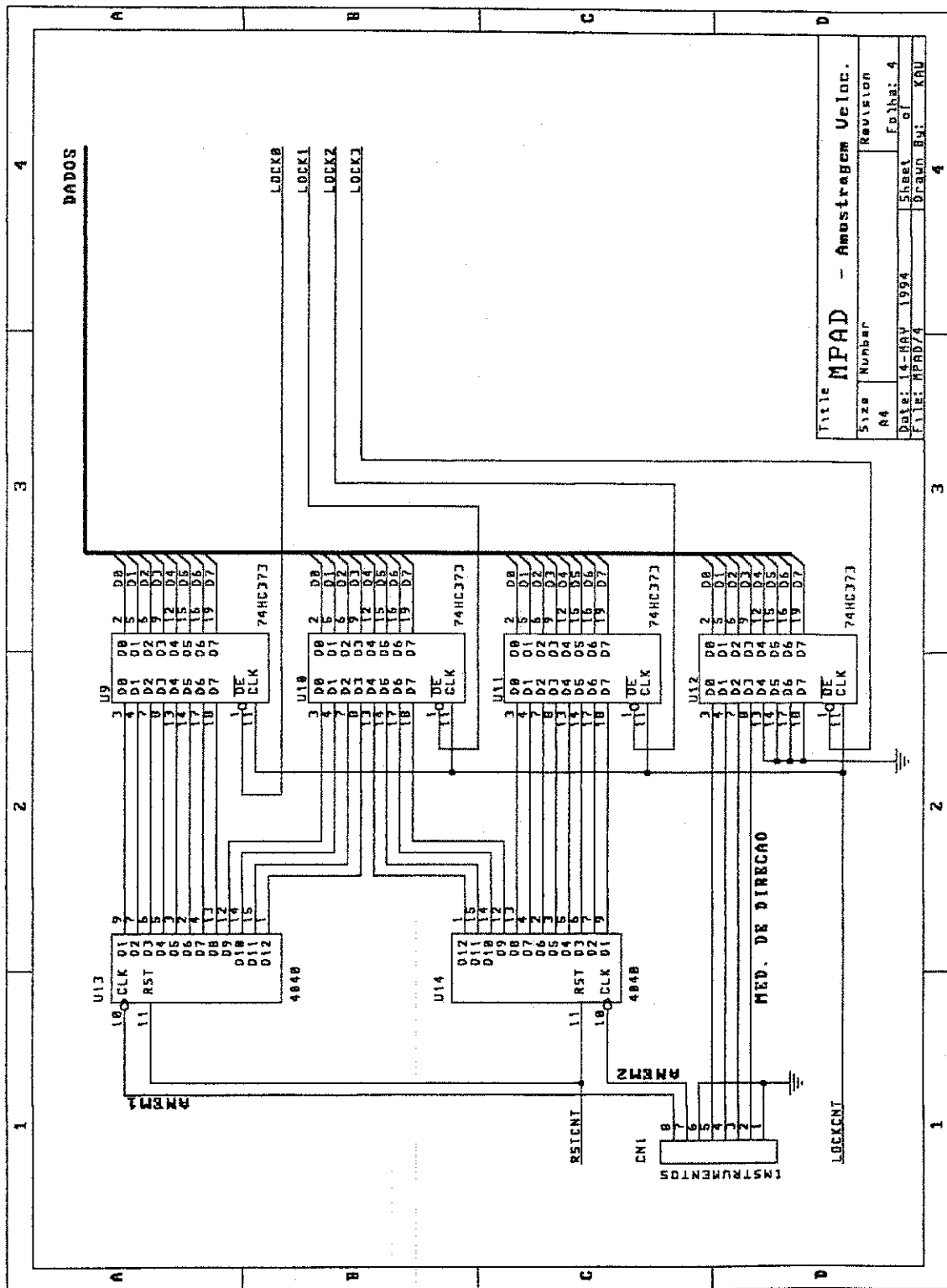


Figura B.4 - Amostrador - MPAD

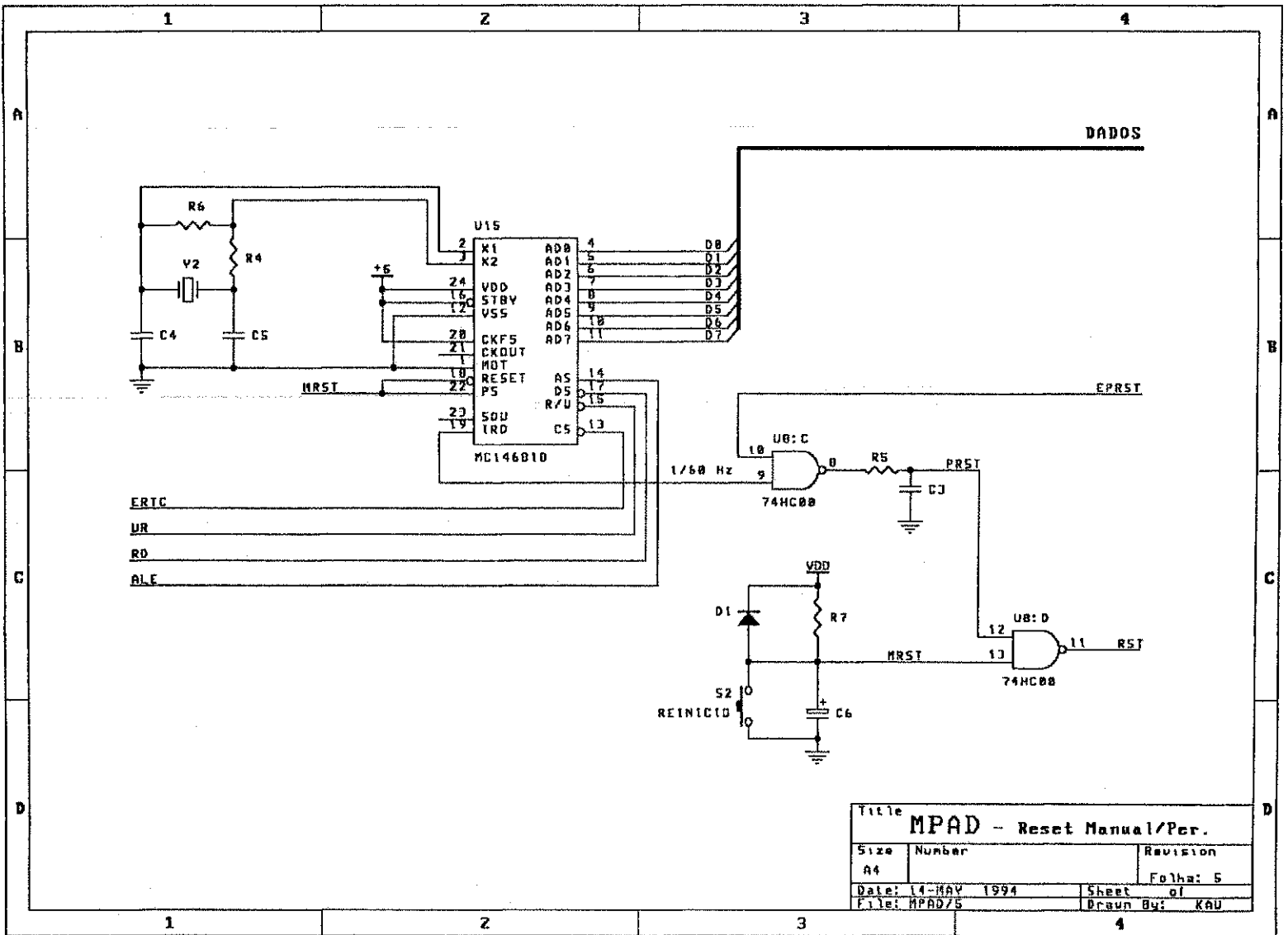


Figura B.5 - Despertador - MPAD

Title		
MPAD - Reset Manual/Per.		
Size	Number	Revision
A4		Folha: 5
Date: 14-MAY 1994	Sheet	01
File: MPAD75	Drawn By:	KAU

B.3 - Módulo de Leitura e Gravação de Memória:

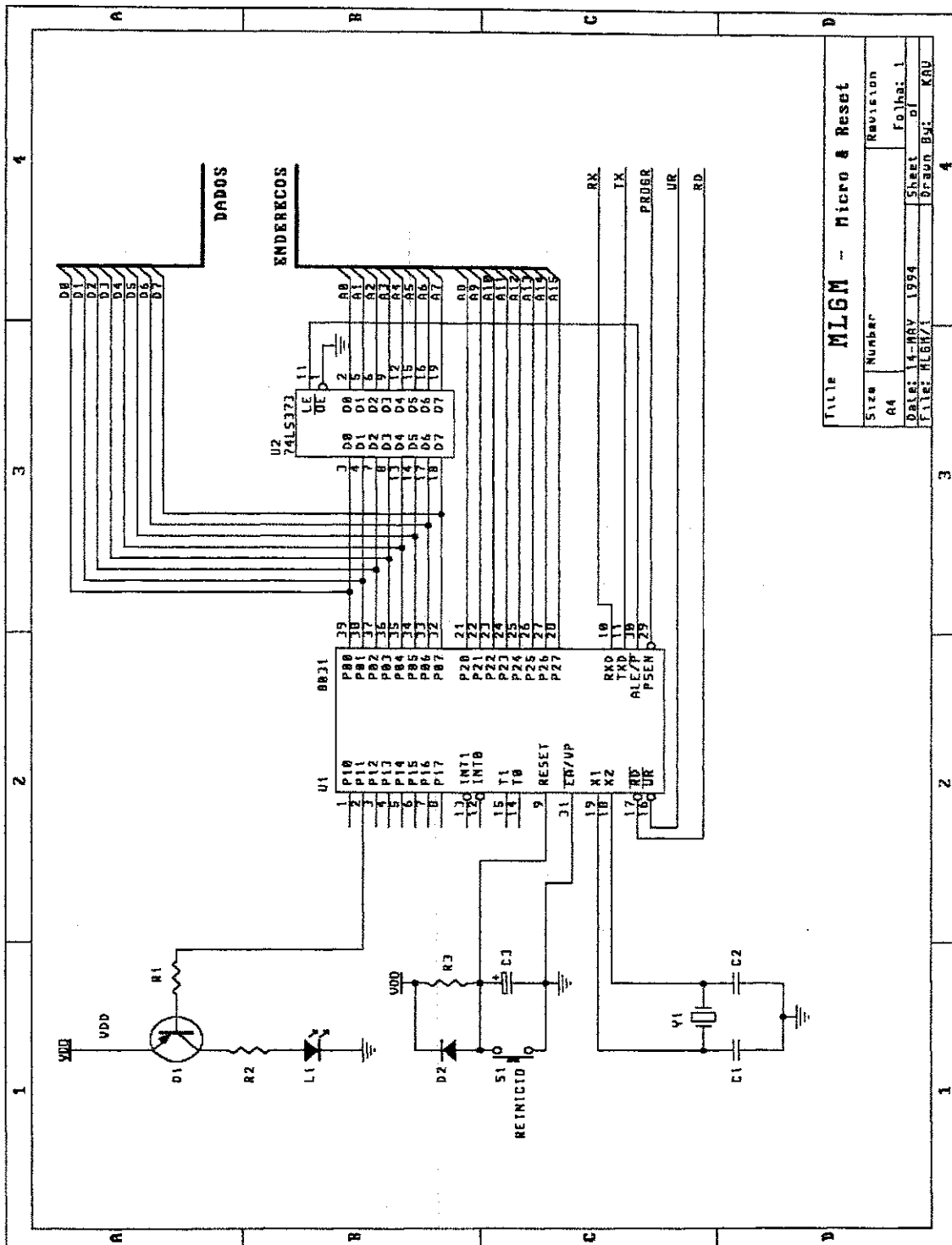


Figura B.6 - Microcontrolador - MLGM

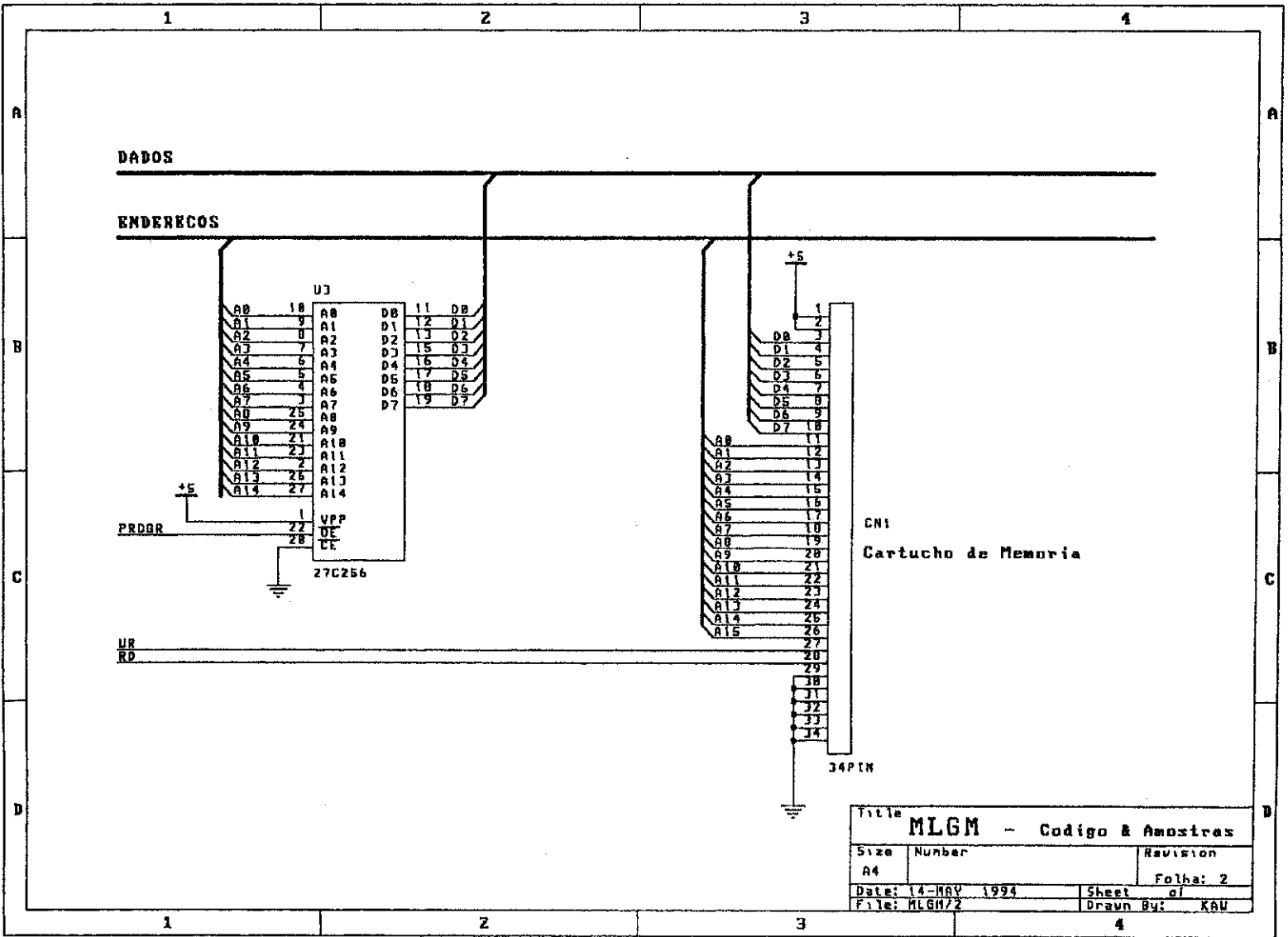


Figura B.7 - Memórias de Código e Amostras - MLGM

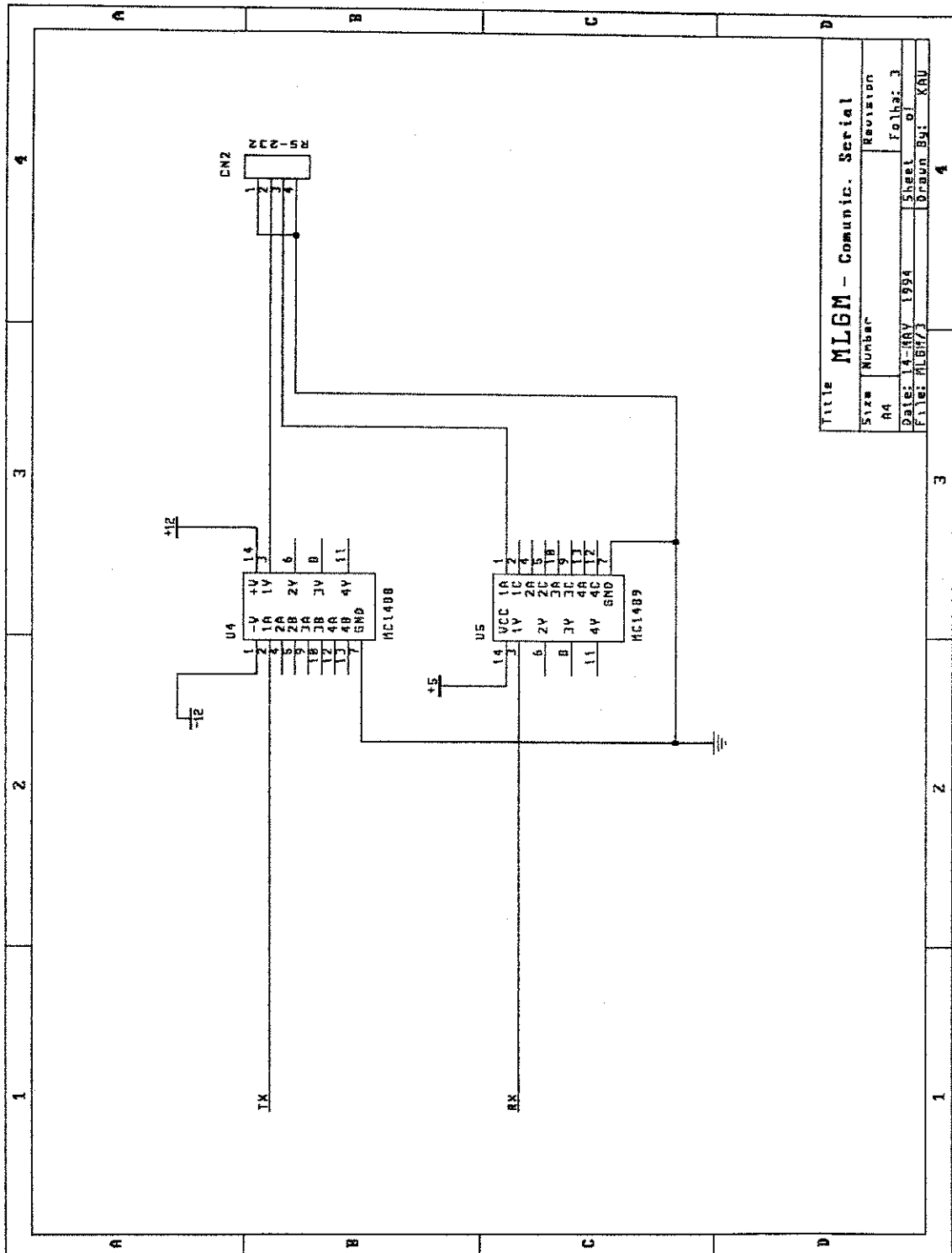


Figura B.8 - Comunicação Serial - MLGM

B.4 - Cartucho de Memória (64 KBytes):

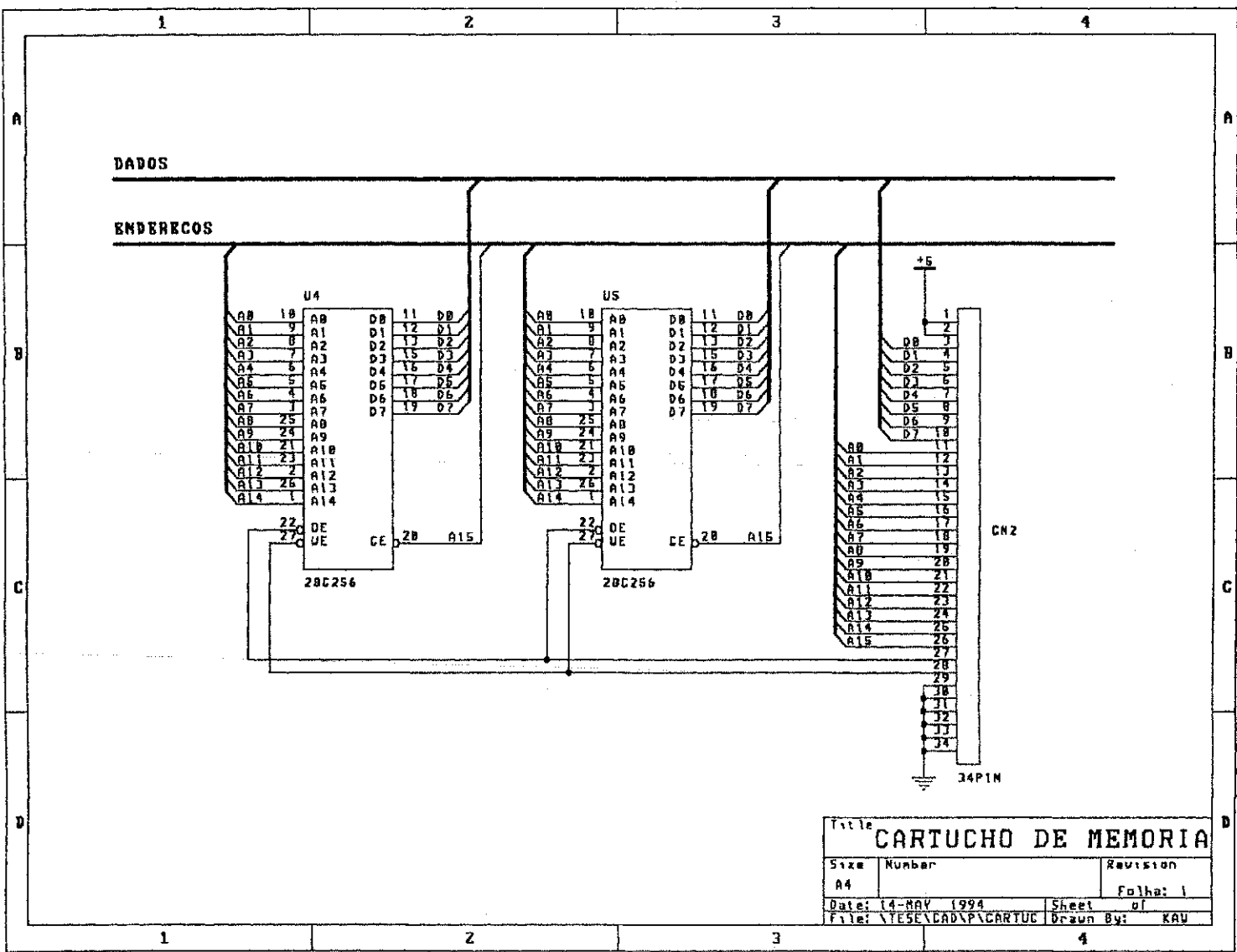


Figura B.9 - Cartucho de Memória

APÊNDICE C

PROGRAMAÇÃO ORIENTADA A OBJETOS

C.1 - Introdução:

Programação Orientação a Objetos (POO) é uma metodologia de programação que busca fornecer melhores paradigmas e ferramentas para atender a demandas recentes de se obter maior funcionalidade dos sistemas de computação. Estes paradigmas e ferramentas serão usados com vários objetivos: para modelar o mundo real tão próximo a perspectiva do usuário quanto possível, para construir componentes reutilizáveis de software e bibliotecas facilmente expansíveis, e para facilitar a modificação e extensão de implementações de componentes de software, sem que seja preciso recodificar a partir do zero.

A POO tenta satisfazer as necessidades de usuários finais tanto quanto as dos desenvolvedores de software. Nesta metodologia de trabalho os dados são o ponto principal, em oposição a outras metodologias que enfocam procedimentos e funções como entidades centrais.

Os três aspectos mais fundamentais da orientação a objetos são *tipo de dados abstrato*, *herança* e *identidade de objeto*. Existe outras propriedades e noções associadas a estes aspectos formando a disciplina POO. Bancos de dados, linguagens, e sistemas orientados a objetos estão disponíveis no mercado, mas em geral, eles enfatizam um ou dois destes conceitos sem suportar outros diretamente [Khoshafian, 1990]. Por exemplo, a linguagem Ada não suporta *herança*, enquanto *identidade de objetos* é limitada nas linguagens C++ e Eiffel, e SmallTalk apresenta alguns problemas no suporte a *encapsulamento*.

C.2 - Conceitos Básicos:

• Tipo Abstrato de Dados

Tipos de Dados descrevem um conjunto de objetos¹ com a mesma representação, e geralmente, existe um certo número de operações associadas com cada tipo de dados. *Tipo de Dados Abstrato* estende a noção de tipo de dados, escondendo a implementação de operações definidas pelo usuário e associadas ao tipo de dados.

Linguagens de programação que suportam *tipo de dados abstrato* possuem mecanismos diretos de definição de estruturas de dados e de operações usadas para manipular ocorrências (instâncias) destas estruturas de dados. E ainda mais, todas as manipulações das instâncias do tipo de dados são feitas exclusivamente através das operações associadas com o tipo de dados [Khoshafian, 1990].

• Herança

Através deste conceito, projetistas podem construir novos módulos de software (tais como classes) sobre uma hierarquia de módulos já existente. A herança de comportamento de objetos possibilita o compartilhamento de código entre os módulos de software e compartilhamento de estruturas de dados entre os objetos de dados. A combinação destes dois tipos de herança fornecem uma poderosa estratégia de modelagem e desenvolvimento de software.

• Identidade de Objeto

Identidade de objeto é a propriedade de um objeto que o distingue dos demais. Com a identidade os objetos podem conter outros objetos ou fazerem referências a outros.

¹ Objeto aqui é usado como termo corriqueiro do dia-a-dia, e definido por A. B. de Holanda como: s. m. tudo que é perceptível por qualquer dos sentidos.

Enquanto *herança* e *tipo de dados abstrato* são conceitos usados para modelar e organizar os tipos de objetos (classes), a *identidade de objeto* organiza os objetos do espaço de objetos manipulados pelo programa orientado a objeto.

C.3 - Linguagem de Programação C++

Esta linguagem foi desenvolvida por Bjarne Stroustrup da AT&T no início dos anos 80 [Stroustrup, 1987], e hoje é uma das linguagens de programação orientada a objetos mais populares no mundo. A primeira implementação desta linguagem tratava-se de um pré-processador para compiladores C, e na sua evolução continuou sendo uma extensão orientada a objeto da linguagem C [Khoshafian, 1990].

C++ possui duas construções para declaração de classes: o primeiro método é uma extensão do construtor *struct* da sua linguagem ancestral C, e o outro método surge através do novo construtor implementado *class*.

A hierarquia de classes é suportado em C++ através do operador ':' (dois-pontos) e permite subclasses para acessar métodos e variáveis de instância de outras classes na sua hierarquia. Na versão mais recente da linguagem pode-se implementar heranças múltiplas através do operador ',' (vírgula).

O polimorfismo é permitido usando-se o sobrecarregamento de nome de funções e de operadores. Nome de funções podem ser sobrecarregadas para se ter número variado de argumentos e tipos diferentes. Os operadores padrões do sistema, tais como +, -, * e outros, também podem ser sobrecarregados para classes definidas pelo usuário. C++ suporta ligação dinâmica (em tempo de execução) através da definição de funções virtuais.

Um dos grandes benefícios da sobrecarga de operadores vem com os canais de entrada / saída. Com este novo método de E/S, usuários podem sobrecarregar os operadores de canais para permitirem manipulação direta de tipos de dados definidos pelo usuário.

Algumas das principais vantagens desta linguagem são enumeradas a seguir:

1. Extensão verdadeira da linguagem C: facilidade de migração de milhares de usuários para a metodologia de orientação a objetos. Qualquer compilador C++ aceita programas escritos em C.
2. Boa performance: programas escritos em C++ não sofrem de problemas de performance associados às primeiras implementações das linguagens de POO. Existe uma leve degradação da performance com funções virtuais, desde que alguns compiladores as implementam com um nível extra de indireção.
3. Popularidade e variedade de fornecedores: hoje a C++ é de longe mais utilizada que SmallTalk, Objective-C, Eiffel, ou Object Pascal; e de olho neste filão do mercado, a grande maioria de fornecedores de compiladores colocaram a disposição dos usuários dezenas de ambientes de desenvolvimento de software utilizando a linguagem C++.

Mas diferentemente da linguagem SmallTalk, C++ não traz uma grande coleção de classes hierárquicas pré-definidas para facilitar o desenvolvimento de aplicações. Felizmente esta questão têm sido contornadas pelos fornecedores de compiladores e ambiente de desenvolvimentos de aplicações que estão embutidas em seus pacotes várias classes genéricas, como por exemplo classes para tratamento de interface gráfica com usuário fornecida pela Borland Inc. com seu produto Object-Window Library (OWL).

Khoshafian lista uma série de referências bibliográficas a respeito do assunto, na seção 9.8 de seu título Orientação a Objeto: Conceitos, Linguagens, Bancos de Dados e Interfaces para Usuários.

REFERÊNCIAS BIBLIOGRÁFICAS

[AMD, 1991]

Advanced Micro Devices; "CMOS Memory Products", Data Book, Sunnyvale - Ca, USA, 1991.

[Bandeira, 1990]

Bandeira, M. M.; "Regionalização do Regime Eólico no Nordeste do Brasil (Período: 1977-1981)", Dissertação de Mestrado, DCA/UFPb, Campina Grande-Pb, Brasil, 1990.

[Butkov, 1983]

Butkov, E.; "Física Matemática", Guanabara Dois, Rio de Janeiro, Brasil, 1983.

[Craveiro, 1991]

Craveiro, P. M. A. et al; Palestra: "Energia Eólica: Uma Alternativa Real", Cia. de Eletricidade do Ceará - COELCE, Fortaleza, Brasil, 1991.

[Golding, 1976]

Golding, E. W.; "The generation of Electricity by wind power", Halsted Press Book, New York, USA, 1976.

[Justus, 1978]

Justus, G. G.; "Winds and Wind System Performance", Georgia Institute of Technology, The Franklin Institute Press, Atlanta - Georgia, USA, 1978.

[Khoshafian, 1990]

Khoshafian, S. et Razmik, A.; "Object Orientation - Concepts, Languages, Databases, User's Interface", John Wiley & Sons Inc., USA, 1990.

[Lima, 1993]

Lima, A. M. N. et al; "Análise Estática de Sistemas de Conversão de Energia Eólica", Artigo submetido ao COBEM/93, DEE-UFPb, Campina Grande, Brasil, 1993.

[Lopes, 1991]

Lopes, C.; "O Desafio do Desenvolvimento Sustentável", Revista Ecologia e Desenvolvimento, Nº 10, Dezembro/91.

[Lysen, 1983]

Lysen, E. H.; "Introduction to Wind Energy", Consultant Services Wind Energy Developing Countries - Wind Energy Group of the Eindhoven University of Technology, The Netherlands, 1983, 2nd Edition.

[Pacitti, 1977]

Pacitti, T.; "Programação e Métodos Computacionais", Livros Técnicos e Científicos, Rio de Janeiro, Brasil, 1977.

[Silva, 1988]

Silva, S. R.; "Sistema Eólico de Geração de Energia Elétrica", Tese de Doutorado, COPELE/UFPA, Campina Grande-PB, Brasil, 1988.

[Silva, 1990]

Silva Jr, V. P. da; "Microcontrolador 8051", Editora Érica, São Paulo, Brasil, 1990.

[Silva Jr, 1988]

Silva Jr, V. P. da; "Microcontroladores", Editora Érica, São Paulo, Brasil, 1988.

[Simões, 1987]

Simões, F. J.; "II Curso de Planejamento Energético", Sudene/Atecel/Nerg.CCT.UFPB, Campina Grande, Brasil, 1987.

[Siqueira, 1991]

Siqueira, G. L.; "Sistemas Rádio Celulares", CETUC-PUC-Rio, Rio de Janeiro, Brasil, 1991.

[Soong, 1986]

Soong, T. T.; "Probabilistic Modeling and Analysis in Science and Engineering", LTC Editora S/A (tradução), Rio de Janeiro, Brasil, 1986.

[Stroustrup, 1987]

Stroustrup, B.; "The C++ Programming Language", Addison-Wesley Publishing Company, New Jersey, USA, 1987.

[Swift-Hook, 1989]

Swift-Hook, D. T.; "Introduction - WECS Economics", Curso: Principles of Wind Energy Conversion, Dept. Science and Technology, Univ. of London, UK, July 3-8, 1989.

[Weiskamp, 1991]

Weiskamp, K. et al; "Programação Orientada para Objeto com Turbo C++", Makron Books do Brasil Editora Ltda (tradução), São Paulo, Brasil, 1991.

Bibliografia Consultada:

1. "A Terra em transe na Eco-92", **Revista VEJA**, Nº 23, pp.52-101, Editora Abril S/A, São Paulo, Brasil, 03/06/1992.
2. CIMA - Comissão Internacional para a preparação da **CNUMAD - Conferência das Nações Unidas sobre o Meio Ambiente e Desenvolvimento**; "Subsídios Técnicos para a elaboração do Relatório Nacional do Brasil para a CNUMAD", Versão Preliminar, Imprensa Nacional, Julho-91.
3. Catálogos técnicos de produtos afins:
 - Wind Classifier, AERODYN, Damendorf, West Germany;
 - Wind Turbines, VESTAS/DWT, Lem, Denmark;
 - NRG Logger, NRG SYSTEMS, Vermont, USA
 - Wind Recording System, NEW (Northumbrian Energy Workshop LTD), Northumberland, U.K.
4. "8051 Family of Single Chip Microcomputers", Microcomputer Components, Users Manual, **Intel Corporation**, Santa Clara - CA, USA, 1981.
5. **Araújo, A. A.**; "Sistema de Aquisição de Dados para um Campo de Testes de Turbinas Eólicas", Dissertação de Mestrado, DEE/UFPb, Campina Grande - Pb, Brasil, 1981.
6. **Berry, J. T.**; "The Waite Group's C++ Programming", Howard W. Sams & Company, Indiana, USA, 1989.
7. **Cheremisinoff, N. P.**; "Fundamentals of Wind Energy", Ann Arbor Science Publishers Inc., Michigan, USA, 1979.
7. **Ezzel, B.**; "Programação Gráfica em Turbo C++", Ciência Moderna Ltda, Rio de Janeiro, Brasil, 1991
8. **Ferraro, R. F.**; "Guia do Programador Para as Placas EGA e VGA", Ciência Moderna Ltda, Rio de Janeiro, Brasil, 1990
9. **Oliveira, L. H. de, et al**; "O Mundo de Cada Um (Ambiente)", Revista Super Interessante, Nº 4, pp.16-22, Editora Abril S/A, São Paulo, Brasil, Abril -1990.
10. **Lancaster, D.**; "CMOS Cookbook", Howard W. Sams & Co. Inc., Indianapolis - ID, USA, 1977.
11. **Xicor Data Book Supplement**; "Memory Products", Xicor Inc., Milpitas - Ca, USA, 1989.