

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**  
(Bacharelado)

**PROTÓTIPO DE SISTEMA DE SEGURANÇA PREDIAL  
ATRAVÉS DE MONITORAMENTO UTILIZANDO RECURSOS  
DA INTERNET**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**ERASMO KRÜGER**

BLUMENAU, DEZEMBRO/2002.

2002/2-21

# **PROTÓTIPO DE SISTEMA DE SEGURANÇA PREDIAL ATRAVÉS DE MONITORAMENTO UTILIZANDO RECURSOS DA INTERNET**

**ERASMO KRÜGER**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Antonio Carlos Tavares — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

**BANCA EXAMINADORA**

---

Prof. Antonio Carlos Tavares

---

Prof. Lúcio Bareta Todorov

---

Prof. Miguel A. Wisintainer

Dedico este trabalho aos meus pais, Siegfried e Josefa, e a minha noiva, Sheila, pelo incentivo recebido durante o desenvolvimento deste trabalho.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por todas as oportunidades e desafios que Ele tem colocado em meu caminho.

Aos professores do curso de Bacharelado em Ciência da Computação, que contribuíram de forma indispensável na minha formação pessoal e profissional. Em especial ao professor Antonio Carlos Tavares, que me orientou no desenvolvimento deste trabalho.

Aos meus familiares e colegas, que sempre me apoiaram e incentivaram durante toda a minha formação acadêmica.

# SUMÁRIO

AGRADECIMENTOS .....	IV
LISTA DE FIGURAS .....	VII
LISTA DE TABELAS .....	VIII
LISTA DE QUADROS .....	IX
RESUMO .....	X
ABSTRACT .....	XI
1 INTRODUÇÃO .....	1
1.1 OBJETIVOS DO TRABALHO .....	1
1.2 ESTRUTURA DO TRABALHO .....	2
2 DOMÓTICA .....	3
2.1 FUNÇÕES DOMÓTICAS .....	4
2.1.1 FUNÇÃO DE GESTÃO .....	4
2.1.2 FUNÇÃO DE CONTROLE-COMANDO .....	5
2.1.3 FUNÇÃO DE COMUNICAÇÃO .....	7
2.2 CARACTERÍSTICAS DOS SISTEMAS DOMÓTICOS .....	8
2.2.1 CONTROLADORES LOCAIS .....	9
2.2.2 ESTAÇÕES DE GERENCIAMENTO .....	10
2.2.3 REDES DOMÓTICAS .....	11
3 DESENVOLVIMENTO DO PROTÓTIPO .....	15
3.1 TECNOLOGIAS RELACIONADAS AO PROTÓTIPO .....	15
3.1.1 SENSORES .....	15
3.1.2 ATUADORES .....	16

3.1.3 COMUNICAÇÃO SOFTWARE / HARDWARE.....	18
3.1.4 SOFTWARE DE CONTROLE .....	19
3.2 MÉTODO DE CONSTRUÇÃO DO PROTÓTIPO.....	23
3.3 REQUISITOS DO SOFTWARE .....	23
3.4 ESPECIFICAÇÕES DO SISTEMA .....	24
3.4.1 ESPECIFICAÇÃO DA APLICAÇÃO .....	25
3.4.2 ESPECIFICAÇÃO DO HARDWARE.....	34
3.5 IMPLEMENTAÇÃO .....	36
3.5.1 FERRAMENTAS UTILIZADAS NA IMPLEMENTAÇÃO .....	36
3.5.2 PRINCIPAIS FUNÇÕES DA APLICAÇÃO .....	37
3.5.3 PRINCIPAIS FUNÇÕES DO HARDWARE.....	43
3.5.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	46
3.6 TESTE E VALIDAÇÃO .....	53
3.6.1 FUNÇÕES ESPECIAIS.....	55
4 CONCLUSÕES .....	57
4.1 EXTENSÕES .....	57
ANEXO I.....	58
REFERÊNCIAS BIBLIOGRÁFICAS .....	60

## LISTA DE FIGURAS

Figura 1: Conectores de um cabo paralelo, [HOW02].	18
Figura 2: Relação dos sinais dos pinos dos conectores paralelos, [HOW02].	19
Figura 3: Estados e transições para uma tarefa em tempo real.	22
Figura 4: Visão geral do sistema	24
Figura 5: Diagrama de contexto	26
Figura 6: Modelo Entidade x Relacionamento Lógico.	26
Figura 7: Modelo Entidade x Relacionamento Físico	27
Figura 8: Diagrama de fluxo de dados (DFD) do evento "Manter Configurações"	27
Figura 9: DFD do evento "Manter Emails"	28
Figura 10: DFD do evento "Manter sinais de status"	28
Figura 11: DFD do evento "Manter dispositivos"	28
Figura 12: DFD do evento "Manter programações por horário"	29
Figura 13: DFD do evento "Manter programações por evento"	29
Figura 14: DFD do evento "Gerar arquivo de LOG"	30
Figura 15: DFD do evento "Enviar mensagens pela internet"	30
Figura 16: DFD do evento "Gerar comandos de acionamento"	31
Figura 17: Macro fluxo da rotina de monitoramento e controle.	31
Figura 18: Esquemático simplificado do circuito dos atuadores na placa centralizadora.	35
Figura 19: Esquemático simplificado do circuito dos sensores na placa centralizadora.	36
Figura 20: Foto da placa centralizadora (Componentes).	45
Figura 21: Foto da placa centralizadora (trilhas)	45
Figura 22: Tela principal do aplicativo.	47

Figura 23: Tela de cadastro dos sensores. ....	47
Figura 24: Tela de cadastro dos dispositivos atuadores. ....	47
Figura 25: Catálogo de endereços de e-mail. ....	48
Figura 26: Tela de configurações gerais.....	48
Figura 27: Tela de cadastro de programações por horário. ....	49
Figura 28: Tela de cadastro e alteração de uma programação por horário.....	50
Figura 29: Tela de cadastro das programações por evento.....	50
Figura 30: Tela de cadastro e alteração de uma programação por evento.....	51
Figura 31: Tela de status da rotina de monitoramento e controle. ....	52
Figura 32: Tela do painel de testes. ....	53
Figura 33: Exemplo de e-mail enviado pelo sistema.....	54

## **LISTA DE TABELAS**

Tabela 1: Redes públicas e serviços em relação a rede e serviços no setor residencial. ....	7
Tabela 2: Padrões Domóticos e suas regiões de utilização .....	14
Tabela 3: Elementos sensores e variáveis a medir mais comuns, [ANG93].....	16
Tabela 4: Comparação entre relés mecânicos e eletrônicos, [OSR87].....	17
Tabela 5: Lista de eventos da Aplicação .....	25
Tabela 6: Lista de atributos das tabelas do sistema.....	32
Tabela 7: Dicionário dos fluxos de dados. ....	33
Tabela 8: Endereços da porta paralela no microcomputador. ....	34
Tabela 9: Utilização dos pinos da porta paralela. ....	44

## LISTA DE QUADROS

Quadro 1: Funções de leitura e escrita na porta paralela. ....	37
Quadro 2: Funções de controle da porta paralela. ....	39
Quadro 3: Rotina de envio de e-mail utilizando Thread. ....	41
Quadro 4: Rotina de geração de LOG. ....	43
Quadro 5: Exemplo de LOG gerado pelo sistema. ....	55
Quadro 6: Rotina principal do módulo de monitoramento e controle. ....	58

## **RESUMO**

Este trabalho apresenta as técnicas de automação especificamente aplicadas à área de automação predial ou Domótica. Apresenta também a implementação de um protótipo de monitoração e controle predial voltado para segurança, envolvendo as variáveis presença e supervisão de portas e janelas, enviando informações pela internet. O protótipo opera em um microcomputador pessoal do tipo PC e utiliza a porta paralela para controle das variáveis envolvidas.

## **ABSTRACT**

This work shows the automation techniques applied on the building automation or Domotica. It shows too an implementation of a prototype of building supervision and control directed to security, with presence, doors and windows supervision variables, sending information through internet. The prototype works on a personal computer and uses the parallel printer port to control involved variables.

# 1 INTRODUÇÃO

Hoje em dia, cada vez mais, vem sendo necessária a utilização de produtos e serviços relacionados a segurança e automação predial. Com isso a indústria nesta área vem crescendo muito rapidamente, lançando no mercado inúmeras opções conforme as necessidades dos consumidores. Porém os produtos encontrados no mercado são em muitos casos inacessíveis financeiramente, como por exemplo os circuitos fechados de TV, ou têm pouca flexibilidade como os alarmes.

Tendo em vista esta situação, tem-se como desafio encontrar uma solução que reúna baixo custo, acessibilidade e seja uma ferramenta eficaz que se adapte às mais diversas situações e necessidades, que com o passar do tempo possa ir evoluindo, exigindo assim, uma solução dinâmica.

Pode-se citar então, a Domótica, que faz a integração do espaço arquitetônico, da informática e das telecomunicações, objetivando maior qualidade de vida através da redução de trabalho doméstico, aumento da segurança e do bem estar, e a racionalização dos consumos [ANG93].

Este trabalho apresenta uma síntese do estudo realizado sobre os conceitos da Domótica, e se complementa com a construção de um protótipo para controle e monitoração de algumas variáveis que normalmente são encontradas em uma habitação, e com maior ênfase em segurança.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo principal deste trabalho é especificar e desenvolver um protótipo de sistema de segurança predial através de monitoramento, utilizando componentes simplificados e de baixo custo, capaz de realizar operações básicas de segurança associadas a recursos da internet.

Os objetivos específicos do trabalho são:

- a) desenvolver módulos para monitoramento de portas, janelas e de presença, que se comunicarão com o microcomputador pessoal (PC) através da porta paralela;
- b) desenvolver um interruptor eletrônico acionado pela porta paralela;

- c) desenvolver software que utilizará os módulos de monitoramento como base para tomar decisões pré-configuradas pelo usuário;
- d) fazer uma análise da melhor maneira de interação entre todos os componentes do sistema, com o objetivo de tornar a configuração simples e ter um funcionamento satisfatório.

## 1.2 ESTRUTURA DO TRABALHO

O primeiro capítulo após a introdução é dedicado ao estudo da Domótica. Nele são apresentadas as funções domóticas através dos serviços básicos de gestão, de controle e de comunicação. Ainda neste capítulo são apresentadas as principais características dos sistemas domóticos segundo a arquitetura do projeto.

Em um sistema domótico são implementados vários controles sobre os equipamentos e sistemas, desta forma faz-se necessário o estudo das técnicas de controle, as mesmas utilizadas em controle automático em indústrias, automóveis e outros. Para maior esclarecimento e embasamento teórico, foram incluídos no capítulo alguns aspectos dos sistemas de controle de processos. Neles são apresentados os conceitos básicos de um sistema de controle, as teorias envolvidas e algumas configurações normalmente empregadas.

O capítulo seguinte é destinado a demonstrar todas as características inerentes ao desenvolvimento do protótipo, como as tecnologias envolvidas, as especificações do *software* e do *hardware* desenvolvido, detalhes das implementações e funcionamento do aplicativo.

Por último será feita uma conclusão, abordando também as dificuldades encontradas no desenvolvimento do protótipo, suas limitações e as possíveis extensões que podem ser aplicadas ao protótipo.

## 2 DOMÓTICA

O termo Domótica foi adotado na Europa para designar o campo de aplicação tecnológica que visa a integração do espaço arquitetônico, da informática e das telecomunicações [ANG93]. Nos Estados Unidos e Japão optou-se pela expressão *intelligent building* para designar esta área. Além dos termos já citados pode-se citar ainda *smart building* etc, comprovando que nenhum termo ainda foi adotado a nível internacional.

O termo *smart building* fora utilizado já no ano de 1977 nos EUA quando da construção do World Trade Center em Nova York. A utilização da palavra *smart* lembrava “estar na moda, ser elegante e inteligente”, porém, tinha outro significado no sentido de “doloroso, que faz mal”, o que gerava confusões e os faziam lembrar de *smart bomb* utilizada na segunda guerra mundial. Por isso o termo *smart building* foi substituído por *intelligent building* consensuado e utilizado em todos os EUA e também no Japão.

Segundo [ANG93], a Domótica tem basicamente como objetivo oferecer uma maior qualidade de vida do espaço cotidiano, assumindo papéis de controle, gerenciamento e comunicação. Esta disciplina integradora visa dar respostas a serviços necessários ao homem, tais como:

- a) Economia de energia: dispor de uma temperatura confortável em todas as estações do ano;
- b) Conforto visual: dispor de iluminação suficiente e bem distribuída nas zonas de atividade;
- c) Conforto acústico, estar protegido de perturbações acústicas, tanto interiores como exteriores;
- d) Qualidade do ar, conforto olfativo e conforto higrotérmico, para manter o ar puro, com umidade adequada, sem correntes de ar, apesar das atividades domésticas interiores: higiênicas, lúdicas ou profissionais;
- e) Conforto dado pela segurança, para estar protegido de intrusos garantindo os bens e as pessoas (conforto moral);
- f) Conforto de atividade, assegurando múltiplas tarefas domésticas, tais como limpeza e conservação dos locais e armazenamento dos alimentos;
- g) Conforto higiênico-saudável para assegurar as atividades do corpo, tais como comer, dormir, lavar-se, manter a forma;

- h) Distrair-se;
- i) Comunicar-se, tanto com o interior como exterior da casa;
- j) Mantimento e otimização dos gastos gerenciando o conjunto. Prover-se dos materiais e fluidos necessários;
- k) Eventualmente trabalhar e estudar, todo ou parte do tempo em casa.

## **2.1 FUNÇÕES DOMÓTICAS**

Ainda de acordo com [ANG93], as funções domóticas permitem satisfazer um número considerável de necessidades acima relacionadas. Segundo o tipo de “serviço”, estas funções se dirigem à Gestão, Controle e Comunicação. Para maior esclarecimento destes serviços são estudadas cada função em separado, para tanto, decompõe-se em sub-funções elementares, como segue.

### **2.1.1 FUNÇÃO DE GESTÃO**

Esta função tem como objetivo automatizar certas atividades sistemáticas principalmente voltadas ao conforto. Estas automações podem ser através de programação, controle de consumo e um mantimento preventivo. A função de Gestão é composta por quatro sub-funções a saber:

- a) Gestão de iluminação: fornece um dos principais elementos voltados ao conforto, adequando o ambiente segundo a necessidade de cada um segundo a idade, capacidade físico-motoras, uso dos espaços ao longo do dia e repercussão sobre a ocupação dos espaços. Outro aspecto importante desta sub-função é a otimização do uso e gastos de eletricidade sem deixar de lado o conforto dos usuários. Como serviços auxiliares dessa sub-função, pode-se citar a temporização, a variação de intensidade, o acendimento e apagamento automático programado ou comandado à distância. Como objetivo futuro dessa sub-área, pode-se citar o comando vocal.
- b) Gestão de calefação, ar condicionado e ventilação: esta sub-função visa permitir ao usuário medir e controlar sua calefação, como também as cargas elétricas e seu próprio conforto. Esta sub-função teve grande impulso com a evolução dos sensores e com necessidade de racionalização de energia e encontra-se em estado bem avançado. Como necessidades de conforto voltadas a esta sub-função, pode-se citar

a otimização em relação ao meio externo; a auto-adaptação em relação aos equipamentos; a gestão, ambiente por ambiente. Sob o aspecto dos requisitos possíveis desta sub-função podemos citar o controle à distância, passagem automática do regime “conforto” para o regime “redução” no caso da ausência de pessoas, dentre outras;

- c) Gestão de qualidade do ar: através desta sub-função a domótica pode prover controle total do ar no ambiente, não somente da temperatura e da umidade acima citadas, mas também da existência de gases tóxicos como por exemplo o gás de cozinha;
- d) Funcionalidade dos espaços: a partir das modificações nos grupos familiares, novos modos de vida e adaptação às novas necessidades é que se exige a flexibilidade do ambiente. Tal flexibilidade visa permitir atualizações futuras nos sistemas atuais para instalação de novos sistemas ou para interconexão dos mesmos, permitindo a evolução das necessidades dos usuários. Para a Domótica, a flexibilidade dos espaços, as instalações e os equipamentos são de vital importância. Para tanto, é importante garantir a adaptabilidade no futuro, mediante um projeto e um equipamento adequado, Deve-se levar em conta a estrutura do edifício, fachadas flexíveis, a construção de tetos e pisos flexíveis. Os requisitos voltados à funcionalidade dos espaços visam principalmente o aumento da produtividade e segurança, o emprego dos recursos com maior eficiência e a flexibilidade de adequação a novos usos.

## **2.1.2 FUNÇÃO DE CONTROLE-COMANDO**

Esta função visa agrupar os vários produtos técnicos instalados na casa em uma unidade de tratamento, permitindo a visualização e intervenção dos estados do funcionamento dos mesmos. Deve permitir controlar ou prevenir defeitos. Comandar intervenções de manutenção, detectar anomalias através de parâmetros previamente definidos.

As informações geradas pelo imóvel são visualizadas nas unidades de tratamento, podendo-se então criar registros dos parâmetros e, eventualmente, a indução de comandos corretivos.

A seguir são apresentadas as sub-funções desta função:

- a) Controle técnico: esta sub-função visa auxiliar o usuário a fazer o uso dos equipamentos, dispositivos e instalações mais confiáveis, assim como prover auto diagnóstico dos mesmos. Cabe a esta sub-função realizar, dentre outras, o controle dos diferentes fluídos utilizados na casa, das presenças anormais devidas a intrusos, dos parâmetros fundamentais para verificação do estado de saúde de um membro da família. É de extrema necessidade a confiabilidade dos valores apresentados, para que possam ser utilizados como assistência ao usuário, sendo estes acessíveis (ergonomicamente) a todos os indivíduos da casa. Como exemplo de controle técnico pode-se citar: visualização das informações em um monitor de vídeo, de TV ou outro indicador de controle; desligamento seletivo de cargas para evitar sobrecargas no sistema; informações de consumo de água, gás, de eletricidade, custos dos mesmos e horários de sua aplicação; comandos únicos que atuam sobre diversos equipamentos, como o comando de saída de férias que desligaria as lâmpadas, ativaria o sistema de monitoramento de invasão e alarme, cortaria o suprimento de água e luz. Outro exemplo da sub-função de controle técnico poderia ser o controle da higiene e saúde do corpo, onde por exemplo existem sensores monitorando a pulsação, pressão arterial e nível de açúcar na urina.
- b) Segurança e Tele-transmissão: esta sub-função é associada a segurança dos bens materiais das pessoas, o que inclui a prevenção de intrusos, incêndios e acidente domésticos tele-transmitindo alarmes em caso de ocorrência de um deles. Como fatores importantes desta sub-função, cita-se a confiabilidade do sistema, evitando por exemplo, falsos alarmes, e fácil operação podendo ser manuseado por qualquer membro da família. Dentro desta sub-função pode-se considerar: o controle de acesso, detecção e controle das pessoas que entram e saem do meio; a detecção de eventuais acidentes como incêndios, fuga de gás e água; detecção de intrusos, e tele-vigilância e tele-assistência.
- c) Assistência-saúde: permite ao usuário sua conexão através de um computador pessoal com centros de assistências, que asseguram o controle e acompanhamento da evolução de certos casos graves, segundo cada paciente.

### 2.1.3 FUNÇÃO DE COMUNICAÇÃO

Uma das características possibilitadas pela comunicação é a interatividade. Através desta é permitido o telecomando e a programação desde que os sistemas envolvidos obedeçam a uma certa padronização.

A função de comunicação pode ser classificada segundo dois ângulos: a comunicação sem significados, por exemplo sinais de áudio e vídeo, onde se busca a maior fidelidade possível; e a comunicação onde visa-se o intercâmbio de informações de controle do ambiente. A Tabela 1 apresenta os diferentes serviços e aplicações das comunicações no setor residencial.

**Tabela 1: Redes públicas e serviços em relação a rede e serviços no setor residencial.**

	Exterior	Interior
Redes	<ul style="list-style-type: none"> <li>- Rede telefônica / RDSI e TV a cabo, separados</li> <li>- Idem na combinação com fibras ópticas</li> <li>- RDSI de banda alta</li> </ul>	<ul style="list-style-type: none"> <li>- Intercâmbio telefônico</li> <li>- TV a cabo adicionais</li> <li>- Nós RDSI</li> <li>- Bus doméstico (eletrônica de consumo)</li> <li>- Nós RDSI de banda alta</li> </ul>
Serviços e Aplicações	<ul style="list-style-type: none"> <li>- Telefone</li> <li>- Telex</li> <li>- Correio eletrônico</li> <li>- Fac-símile</li> <li>- Tele-vídeo</li> <li>- Videotexto</li> <li>- Banco de dados</li> <li>- Telecompras</li> <li>- TV</li> <li>- Tele-texto</li> <li>- Telemetria</li> <li>- Telecontrole</li> </ul>	<ul style="list-style-type: none"> <li>- Todas as comunicações entre terminais gravadores e reprodutores no local (funções de segurança, controle, áudio, vídeo)</li> <li>- Todas as uniões entre terminais de gravação e reprodução, e a rede de telecomunicação pública (vídeo gravação, CD, câmeras, etc.)</li> </ul>

A seguir são analisadas as três sub-funções da função de comunicação:

- a) Comunicação de controle: com a utilização da Rede Integrada de Serviços Integrados (RDSI), que utiliza uma técnica comum para serviços de voz, texto, dados e imagens, a função de controle pode realizar sua função principal, interligar os vários dispositivos entre si e com o operador do sistema. Além dos serviços já citados esta rede disponibilizará também a troca de comando entre os vários equipamentos e o operador.
- b) Comunicação de espaços: dentre os serviços oferecidos por esta sub-função pode-se citar a possibilidade do relacionamento da família com o ambiente externo e os serviços coletivos dos imóveis. Para realização destes serviços coletivos tem-se a necessidade de interconectar os aparelhos de áudio e vídeo-comunicação da casa, fazendo-os comunicarem entre si, permitindo assim um melhor serviço de som e imagem, aumentando o conforto.
- c) Comunicação-serviços: esta sub-função visa a conexão da rede interna de áudio e vídeo com uma rede exterior, podendo esta ser pública ou privada. Nos objetivos deste serviço estão incluídos: o tele-trabalho, a tele-educação, o tele-mantimento, a tele-vigilância-assistencia.

## **2.2 CARACTERÍSTICAS DOS SISTEMAS DOMÓTICOS**

No início as instalações domóticas apresentavam, basicamente, uma central de processamento da qual partiam todas as conexões físicas com os equipamentos. Com esta configuração, porém, ocorriam certos problemas com pouca tolerância a falhas, pois se o elemento centralizador e processador falhasse, todo o sistema viria a baixo, além do custo das instalações tornar-se elevado, pois o centralizador nem sempre poderia estar próximo do equipamento controlado. Portanto, conforme [LIM95], com a evolução da domótica, permitiu-se pouco a pouco a descentralização dos sistemas. Estes passaram a contar com unidades centralizadora dos sensores e atuadores, com pouca capacidade de processamento que faziam então a comunicação com o processador central que as tratava segundo a lógica programada e acionava os devidos atuadores. Estes sistemas, porém, sobrecarregavam a linha de comunicação. A tendência atual é da existência de sistemas totalmente isolados com capacidade de processamento, tornando assim o sistema num todo menos suscetível a falhas.

No entanto, estes sistemas devem possuir mecanismos de comunicação bem definidos para operação integrada, formando assim um sistema distribuído.

Nesta etapa, os sistemas passam a ser nós da rede doméstica, podendo ser acessados por quaisquer outros. Esta nova topologia permite, então, que a partir de qualquer nó, possa-se saber o estado de um outro e, no caso do uso de um terminal de operação/gerenciamento, este pode estar em qualquer local desde que possua conexão com a rede.

Seguindo esta tendência de distribuição, os sistemas devem ser o máximo abertos. Isto é, devem possuir condições de comunicarem-se e interagirem, em diferentes níveis, com outros sistemas ou componentes. Partindo deste princípio [LIM95] identifica duas características básicas destes sistemas: a utilização de Unidades Centrais de Processamento (UCP) padrões, produzidas em grande escala e polivalentes quanto a aplicações e de sistemas operacionais e ambientes de grande difusão que são reconhecidos como padrões no mercado informático.

Para melhor explorar as características dos sistemas domóticos separou-se o estudo destes em seus dois componentes básicos: os controladores e as estações de gerenciamento.

## **2.2.1 CONTROLADORES LOCAIS**

Os controladores locais são os “cérebros” dos sistemas de automação com inteligência distribuída, cabe a eles o perfeito funcionamento das máquinas sob seu controle.

Estes controladores, do tipo Controle Digital Direto (DDC), tem como elemento básico um microprocessador de 8 ou 16 bits, com resolução do conversor analógico/digital tipicamente de 8, 10 ou 12 bits, e em seu sistema operacional estão incluídas funções como:

- a) processamento de sinais de entrada e saída;
- b) controle dos comandos;
- c) controle automático;
- d) programas de racionalização de energia;
- e) software de comunicações.

O sistema operacional de gerenciamento das funções é normalmente multitarefa em tempo real e inclui software de diagnóstico.

O software de processamento dos sinais de entrada e saída através de amostragem em intervalos determinados, deve atualizar os valores e condições das variáveis analógicas e digitais. Este deve realizar ainda outras funções como: corrigir a não-linearidade dos sensores; memorizar valores mínimos e máximos das entradas analógicas e inibir alarmes por tempos programados.

## **2.2.2 ESTAÇÕES DE GERENCIAMENTO**

Baseada tipicamente em um microcomputador pessoal (PC), esta unidade tem como objetivo prover a interface entre o usuário e a instalação. É através desta que o operador poderá alterar parâmetros, modificar programas e emitir comandos a todo sistema. Esta unidade também é responsável por realizar os controles e ações que os controladores locais não estão aptos a realizar.

A escolha do sistema operacional deve seguir a regra acima citada, isto é, ser bem difundido. Como opções pode-se citar: MS-DOS, Windows, Unix, Xenix, Linux ou OS/2. Cabe ressaltar que o gargalo nesta estação é a comunicação em rede, devendo-se observar este detalhe na escolha do sistema operacional.

As facilidades de supervisão, controle e comando estão associadas ao programa aplicativo disponível na estação de gerenciamento. A utilização de sinópticos gráficos pode ser fator determinante na facilidade de entendimento e operação do sistema.

Basicamente este software deve responder as seguintes exigências:

- a) Simplicidade de uso;
- b) Modularidade;
- c) Configurabilidade;
- d) Flexibilidade;
- e) Conectibilidade.

Dentre outras possibilidades funcionais o software aplicativo deve:

- a) monitorar as variáveis de instalação;
- b) gerenciar alarmes e anomalias;
- c) comandar remotamente controladores ou unidades periféricas;
- d) gerenciamento simultâneo de controladores, mesmo que de tipos diferentes;

- e) livre estruturação gráfica do usuário;
- f) programa de análise de dados e de custos.

Retornando novamente ao enfoque global dos sistemas domóticos, deve-se considerar nesta nova topologia a questão da distribuição das bases de dados e funções, já que cada nó possui sua autonomia.

Atualmente os sistemas domóticos apresentam particularidades proprietárias nos controladores de processo e no software aplicativo por serem dependentes do fabricante, [LIM95].

### **2.2.3 REDES DOMÓTICAS**

A rede domótica é o elemento principal de um sistema domótico. A rede domótica é que permite a comunicação entre os diferentes equipamentos e é individualmente o instrumento essencial em que se baseia a domótica [ANG93].

A comunicação tanto entre computadores como entre dispositivos ou equipamentos compartilhados são proporcionados por redes locais. Uma rede local ou LAN (*Local Area Network*), possibilita a interconexão de uma variedade de dispositivos de comunicação de dados dentro de uma área, usualmente com extensões de até 10 km [ANG93].

Um dos maiores benefícios de uma LAN está relacionado a evolução do sistema. Com uma LAN é possível uma modificação gradual evitando e simplificando trabalhos complexos. Uma LAN permite a conexão de dispositivos, periféricos e computadores de diversos fabricantes, o que proporciona ao usuário maior flexibilidade e autonomia.

Segundo [LIM95], os elementos que compõem e caracterizam uma LAN são:

- a) A técnica de transmissão;
- b) Os meios de transmissão;
- c) A topologia;
- d) O método de acesso aos recursos do sistema.

Através da combinação destes elementos pode-se determinar os custos, serviços possíveis, a máxima distância de transmissão, a expansão e, assim, a perenidade de uma LAN [LIM95].

A técnica de transmissão pode ser escolhida entre dois tipos: analógica ou digital. Na técnica analógica, as informações são transmitidas de maneira contínua, utilizando-se sinais de corrente e tensão; a transmissão ocorre pela variação de um dos parâmetros, amplitude ou frequência do sinal. Como exemplo desta técnica pode-se citar a telefônica e televisiva.

Algumas características das LAN analógicas podem ser citadas [LIM95]:

- a) Num mesmo cabo podem coexistir simultaneamente vários serviços;
- b) Quanto mais ampla a faixa de transmissão, maior deve ser a qualidade do cabo;
- c) Quase todos os sistemas desse tipo utilizam amplificadores de sinal;
- d) Todos os pontos da rede precisam de um modem (modulador/demodulador) analógico.

Como meios de comunicação, as LAN podem utilizar-se basicamente dos modelos seguintes:

- a) Cabos sem blindagem: os mais utilizados e os mais econômicos, porém estão sujeitos a interferências elétricas e, por isso, permitem transmissões confiáveis somente em velocidades limitadas;
- b) Cabos blindados: podem ser de dois tipos – pares telefônicos e cabos coaxiais. Os pares conservam a simplicidade e economia dos cabos sem blindagem, permitindo transmissões em velocidade mais elevadas. Os cabos coaxiais são mais caros, porém permitem velocidades de transmissão ainda maiores.
- c) Fibras ópticas: apresentam-se insensíveis a interferências elétricas ou de radiofrequência, tendo reduzido pesos e dimensões. Permitem a realização de redes mais flexíveis, com baixas perdas e maior largura de banda. Como desvantagens pode-se citar o elevado custo e a instalação mais delicada.
- d) Rede elétrica (*Powerline*): em certos sistemas domóticos a rede elétrica é utilizada como meio de comunicação, sobre tudo por razões de simplicidade nas instalações já existentes. A modulação dos sinais faz-se em uma banda de 120 a 140 KHz (norma européia) utilizando-se de técnicas como: PCM (*Pulse Code Modulation*) segundo norma X-10 americana, e FSK (*Frequency Shift Keying*) segundo norma ESPRIT-HS européia.

Segundo [ANG93], existem basicamente duas classes básicas de LAN: ponto a ponto (*store and forward*) e multiponto (*broad cast*). A primeira é uma arquitetura simples, com

união bidirecional, onde um bloco completo de informação é recebido por um *buffer* de um nó ou estação intermediária antes de ser transmitida ao seu destino final. Em uma rede multiponto, uma mensagem transmitida por um nó é recebida por todos os nós que estão conectados a um meio de transmissão comum.

A topologia de uma rede é definida pelo modo em que seus nós estão interconectados, interferindo na confiabilidade, nos atrasos de transmissão e na reconfiguração. Atualmente não existe uma topologia única que seja apropriada para todas as aplicações [ANG93]. Os tipos básicos de topologia apresentados por [ANG93], são:

- a) Estrela: todos os aparelhos estão conectados separadamente a um nó central. As diferentes estrelas podem constituir as extremidades de outra estrela (estrela da estrela ou árvore).
- b) Bus (*binary unit system*): todos os aparelhos estão conectados em paralelo no suporte de comunicação *Bus* que percorre o conjunto do sistema.
- c) Anel: todos os aparelhos estão conectados em série em um circuito fechado. As informações são transmitidas de um aparelho para outro, em forma unidirecional.

Segundo [LIM95], os métodos de acesso ao meio em uma LAN podem ser por solicitação ou com acesso controlado, no primeiro o usuário solicita o acesso quando o necessita e, no segundo, uma regra comum deve ser seguida. O controle de transmissão pode ser centralizado ou distribuído. Da combinação das regras acima, obtêm-se diversos protocolos de acesso, os mais conhecidos entre eles são:

- o *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD);
- o *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA).

Diante desta variedade de possibilidades na construção de redes de comunicação, vê-se a necessidade de padronização dos sistemas, situação esta, ainda não conseguida [LIM95]. No mundo os principais comitês estudam padrões para que haja harmonia entre os diferentes dispositivos e tecnologias, com o objetivo de fazê-los compatíveis. Alguns padrões já definidos e seus estados no mercado atual são apresentados na Tabela 2.

**Tabela 2: Padrões Domóticos e suas regiões de utilização**

<b>Padrão</b>	<b>Tipo de Padrão</b>	<b>Região</b>	<b>Estado</b>
X-10	Fechado	USA	Partiu de Powerline
CEBUS	Aberto	USA	Especificações disponíveis em 1991
SMART HOUSE	Fechado	USA	Disponível em setembro de 1991
ECHELON	Fechado	USA	Disponível em meados de 1991
ESPRIT-HS	Aberto	Europa	Primeira Versão em Janeiro de 1991
D2B	Fechado	Europa	ISO para subsistema Audio/Vídeo
HBS	Aberto	Japão	Disponível em setembro de 1988
HES	Aberto	ISO/IEC	Disponível em 1993
KONNEX	Fechado	Europa	Disponível em 1996
BatiBUS	Aberto	Europa	Disponível em 1991
LonWorks	Aberto	USA	Disponível em 1992
HAVi	Aberto	USA	Lançado em 1999

## **3 DESENVOLVIMENTO DO PROTÓTIPO**

O protótipo desenvolvido para este trabalho é um software que tem por objetivo monitorar através de sensores, um ou vários ambientes de uma residência ou outro tipo de prédio, ao qual o sistema estiver conectado.

Neste capítulo serão demonstrados detalhes relevantes sobre as estruturas de dados e funcionamento do protótipo.

### **3.1 TECNOLOGIAS RELACIONADAS AO PROTÓTIPO**

Para a construção de um sistema de controle, pode-se escolher basicamente entre três modelos de atuadores: o pneumático, o hidráulico e o elétrico [JOH88]. Devido ao intuito básico deste trabalho ser a implementação de um sistema para uso doméstico, não serão explorados os controladores pneumáticos nem os hidráulicos, pois os mesmos se aplicam basicamente a processos industriais.

Nos próximos itens serão discutidos aspectos dos controladores elétricos, mais especificamente os eletrônicos. Devido a grande complexidade, os dispositivos eletrônicos de controle analógico não serão abordados, visto que estes não estão no escopo do trabalho aqui defendido, portanto é concentrada a atenção nos dispositivos eletrônicos digitais.

Os sistemas de controle digital normalmente possuem um módulo central de controle. Como o intuito deste trabalho é fornecer respostas ao usuário através da *Internet*, será utilizado um microcomputador que executará um software de controle. Neste capítulo serão vistos alguns aspectos deste software. Será tratada também, a forma com que o software, fará a comunicação com os sensores e atuadores presentes no sistema.

#### **3.1.1 SENSORES**

Para realização de qualquer controle realimentado necessita-se do uso de sensores devendo a precisão e confiabilidade dos mesmos serem considerados na sua escolha.

Sensores são dispositivos que mudam seu comportamento sob a ação de uma grandeza física, podendo fornecer diretamente ou indiretamente um sinal que indica esta grandeza. Quando operam diretamente, convertendo uma forma de energia neutra, são chamados

transdutores. Os de operação indireta alteram suas propriedades, como a resistência, a capacitância ou a indutância, sob ação de uma grandeza, de forma mais ou menos proporcional.

Os sensores podem ser classificados em passivos ou ativos. Na primeira classe estão os resistivos, capacitivos, indutivos e tem como principal característica a necessidade de tensão de alimentação. Já os segundos geram tensão, não necessitando de alimentação externa, nesta classe estão incluídos os diodos fotovoltaicos, termopares, cerâmicos e piezoelétricos [ANG93].

A Tabela 3 apresenta os diferentes tipos de sensores, segundo seu princípio físico de funcionamento, associada à medição de grandezas físico-químicas.

**Tabela 3: Elementos sensores e variáveis a medir mais comuns, [ANG93]**

	Temperatura	Luminosidade	Pressão	Força / Peso	Nível	Velocidade	Aceleração	Composição de Gás	Concentração Iônica	Umidade
<b>Resistivo</b>	X	X						X		X
<b>Capacitivo</b>			X		X					X
<b>Indutivo</b>						X	X			
<b>Piezoresistivo</b>			X	X			X			
<b>Fotovoltaico</b>	X	X								
<b>Eletromagnético</b>						X	X			
<b>Termoelétrico</b>	X	X								
<b>Piezoelétrico</b>	X			X		X	X			
<b>Eletroquímico</b>								X	X	
<b>Piroelétrico</b>	X	X								X

### 3.1.2 ATUADORES

O atuador é o componente responsável pela transformação do sinal de controle em uma ação no elemento de controle, [JOH88].

Existe grande variedade de atuadores, podendo atuar em áreas específicas ou em ampla escala. Como exemplo dos atuadores elétricos normalmente utilizados cita-se as solenóides, motores de corrente contínua/alternada e motores de passo.

Os elementos de saída dos controladores são constituídos normalmente por relés mecânicos ou de estado sólido e em casos específicos por transistores. A Tabela 4 faz um comparativo entre os dois tipos de relés. Vale ressaltar que relés de estado sólido são circuitos eletrônicos que tem como principal componente um *triac*.

**Tabela 4: Comparação entre relés mecânicos e eletrônicos, [OSR87].**

<b>Característica</b>	<b>Relé Eletrônico</b>	<b>Relé Mecânico</b>
Estabilidade a choques e vibrações	Excelente	Pouca
Estabilidade com a temperatura	Excelente	Boa
Compatibilidade com famílias lógicas	Sim	Sim
Múltiplos contatos	Não	Sim
Contatos inversores	Não	Sim
Isolação	Excelente	Excelente
Vida útil	Excelente	Boa
Dimensões	Boas	Boas
Capacidade de sobrecarga	Pouca	Boa
Centelhamento	Não	Sim
Operação silenciosa	Excelente	Pouca
Estabilidade de comutação	Boa	Boa
Corrente de fuga em repouso	Desprezível	Nenhuma
Tipos biestáveis (NA/NF)	Não	Sim
Queda de tensão de carga	Desprezível	Nenhuma
Capacidade de excitação	Boa	Excelente
Rebote dos contatos	Não	Sim
Proteção contra sobrecarga	Boa	Nenhuma

### 3.1.3 COMUNICAÇÃO SOFTWARE / HARDWARE

Para realizar a comunicação entre o software, sendo executado em um microcomputador, e o hardware de monitoramento e controle, será utilizada a porta paralela.

A porta paralela, conhecida também como a porta da impressora (LPT), pode ser utilizada como saída ou entrada de dados. É utilizada no interfaceamento entre o computador e um hardware específico (periférico). Atualmente vários dispositivos utilizam-se dela, tais como: *Zip Drivers*, *Scanners*, Câmeras e outros.

A Porta paralela é baseada na tecnologia TTL, isto é, trabalha com 0V ou 5V que significam respectivamente, nível baixo e nível alto, ou ainda desligado e ligado, [HOW02].

Em dispositivos de interface paralela são utilizados cabos paralelos, que numa ponta, a que fica conectada no microcomputador, fica um conector “fêmea” com 25 pinos (tipo DB25) e na outra ponta um conector “macho” de 36 pinos, também chamado de Centronics, conforme Figura 1.

Figura 1: Conectores de um cabo paralelo, [HOW02].



Cada um dos 25 pinos do conector DB25, tem uma função distinta, assim como os 36 pinos do Centronics 36, conforme Figura 2.

Figura 2: Relação dos sinais dos pinos dos conectores paralelos, [HOW02].

DB 25		Centronics 36	
Pin	Signal	Pin	Signal
1	Strobe	1	Strobe
2	data0	2	data0
3	data1	3	data1
4	data2	4	data2
5	data3	5	data3
6	data4	6	data4
7	data5	7	data5
8	data6	8	data6
9	data7	9	data7
10	Acknowledge	10	Acknowledge
11	Busy	11	Busy
12	Paper End	12	Paper End
13	Select	13	Select
14	Auto Feed	14	Auto Feed
15	Error	15	Error
16	Init	16	Init
17	Select In	17	Select In
18	GND	18	GND
19	GND	19	GND
20	GND	20	GND
21	GND	21	GND
22	GND	22	GND
23	GND	23	GND
24	GND	24	GND
25	GND	25	GND
		26	GND
		27	GND
		28	GND
		29	GND
		30	GND
		31	Init
		32	Error
		33	Ground
		34	NC
		35	NC
		36	Select In

Conectado a porta paralela, estará um circuito centralizador de todos os sensores e atuadores presentes no sistema. Este será responsável pelo agrupamento e direcionamento dos sinais e comandos, e também pela entrada e distribuição da tensão de alimentação.

### 3.1.4 SOFTWARE DE CONTROLE

Um sistema de controle sugere a utilização de um sistema em tempo real, assim, foram destinadas as próximas seções deste trabalho no esclarecimento de alguns detalhes destes sistemas. Ainda neste subcapítulo, dedicou-se uma seção para a apresentação de duas metodologias de projeto de sistemas.

### 3.1.4.1 SISTEMAS EM TEMPO REAL

“Uma propriedade fundamental de um programa em tempo real é que algumas, ou todas as suas entradas, chegam do mundo exterior de forma assíncrona, em relação a qualquer trabalho que o programa já esteja fazendo. O programa deve ser capaz de interromper sua atividade corrente imediatamente e então executar algum código predefinido para capturar ou responder a essa entrada, que é freqüentemente um sinal volátil, temporário”, [RIP93]. Após a entrada deste dado podem ser disparadas as execuções de um ou mais programas que estejam esperando pelo mesmo. Finalmente, o computador pode retornar a sua atividade original.

Citadas em [WAR85], as características normalmente encontradas em sistemas de tempo real são:

- a) Contém equipamentos que atuam no sensoriamento do sistema;
- b) Coletam dados não estruturados;
- c) Requerem processamento concorrente das múltiplas entradas;
- d) Exigem curtos tempos de resposta;
- e) Requerem maior precisão em relação a sistemas não em tempo real.

“Os sistemas em tempo real são classificados em *soft* e *hard*. Os sistemas *soft* são aqueles para os quais são exigidos tempos de resposta rápidos não existindo, entretanto, restrições temporais à execução do sistema. Os sistemas em tempo real *hard* (ou seja, com restrições severas de tempo) são aqueles para os quais a correção do sistema depende não somente do resultado lógico do processamento mas também dos instantes nos quais os resultados são produzidos”, [MAG90].

Os programas em tempo real são construídos de forma mais robusta, visando sua sobrevivência em virtude de perturbações que derrubariam programas que não são em tempo real, [RIP93].

Nas seções seguintes serão tratados aspectos da organização de um programa e do sistema operacional de tempo real.

#### 3.1.4.1.1 ORGANIZAÇÃO DE UM SISTEMA EM TEMPO REAL

Como já visto, uma das principais características destes sistemas é a necessidade de interrupção das atividades em execução visando atender outras com maior urgência, o que os

imprime uma característica particular. “Aplicações em tempo real devem ser escritas como uma série de programas componentes separados, que podem ser executados concorrentemente”, [RIP93]. Estes componentes são chamados tarefas (*tasks*) ou processos e a organização é chamada multitarefa (*multitasking*).

Cada tarefa é um programa completo e independente, possuindo seu próprio segmento de código, pilha e área de dados locais, permitindo assim armazenar parâmetros de chamada de procedimento, retornar elementos, dados temporários e variáveis similares que não são compartilhadas com outras tarefas. Além dos dados já citados, cada tarefa tem seu próprio contador de programa, indicador de pilha e outros registradores, caracterizando o contexto de execução da tarefa.

No projeto do software de monitoramento (módulo SISM-NET) foram integradas classes do tipo *TThread* para realizar alguns processos que necessitam ser executados sem interromper o processo principal de monitoramento, como por exemplo a rotina de envio de *e-mails* que, ocasionalmente, pode levar alguns minutos para concluir suas tarefas.

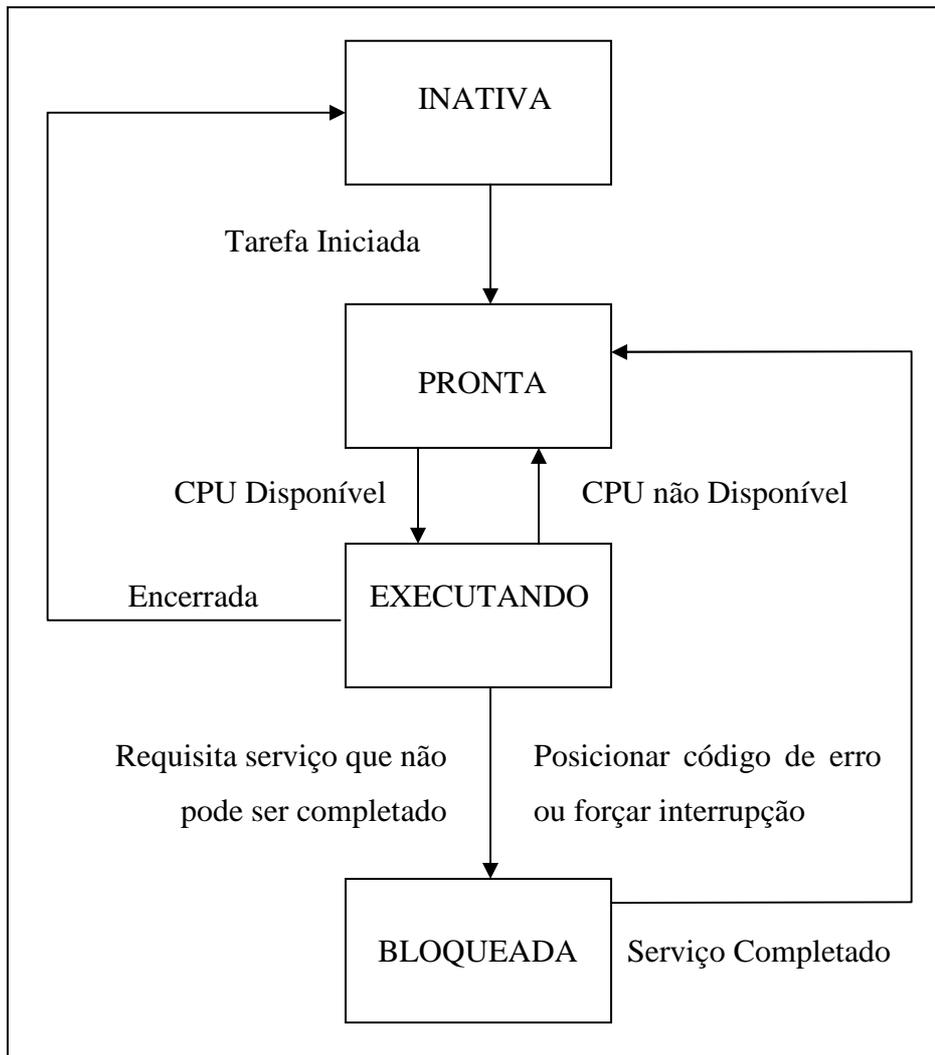
#### **3.1.4.1.2 SISTEMA OPERACIONAL DE TEMPO REAL**

A execução descontínua de uma tarefa, é transparente para a mesma, graças a existência do sistema operacional. Cabe ao sistema operacional decidir qual tarefa será executada pela CPU em um determinado instante além de se encarregar da devida troca de contexto. O sistema operacional ainda manuseia as interrupções geradas pelos equipamentos, ativando as tarefas necessárias aos respectivos tratamentos. De um modo geral, o sistema operacional organiza o trabalho do processador, [RIP93].

Graças ao papel centralizador do sistema operacional, em relação às tarefas e interrupções, a ele também são incorporados serviços de controle do hardware e software que é compartilhado por todas as tarefas. Este fato produz considerável ganho de eficiência e reduz sensivelmente a duplicação de código em cada tarefa. Portanto, “manutenção do tempo, entrada e saída periférica e alocação de memória são todos de domínio do sistema operacional” [RIP93].

Na Figura 3 está apresentado o fluxo de funcionamento de uma tarefa em tempo real, que é como as *Threads* contidas no software do projeto (módulo SISM-NET) funcionam.

**Figura 3: Estados e transições para uma tarefa em tempo real.**



Outro trabalho importante realizado pelo sistema operacional é o escalonamento das tarefas a serem executadas, isto é, a escolha da tarefa que poderá utilizar a UCP. Como regra na escolha da próxima tarefa a ser executada pode ser observado aspectos como: primeira a chegar, primeira a ser servida; fatias iguais para todas as tarefas e outros. Para maior facilidade de gerenciamento das tarefas é associada a cada uma a informação de sua prioridade corrente em relação ao uso do processador, e o estado atual em que ela se encontra, descritas por [RIP93] e apresentadas na Figura 3 os estados de uma tarefa são:

- a) EXECUTANDO, se encontra em execução no processador;
- b) PRONTA, pode ser executada logo que houver disponibilidade do processador;
- c) BLOQUEADA, requisitou um serviço que ainda não pode ser atendido;
- d) INATIVA, teve sua execução encerrada ou ainda não foi requisitada para execução.

## 3.2 MÉTODO DE CONSTRUÇÃO DO PROTÓTIPO

O método utilizado para a construção do sistema foi o *top-down* (de cima para baixo). Neste método, a completa especificação do sistema foi realizada sem o maior detalhamento das subfunções, somente numa etapa posterior foram especificados os detalhes das subfunções. Desta maneira, o sistema foi desenvolvido em partes de mais fácil implementação, com os limites e as propriedades das funções previamente especificadas para a melhor interação das mesmas em um todo.

No método *top-down*, o sistema é construído dos módulos mais altos para os mais baixos, dos mais genéricos aos mais específicos. O nível mais alto é chamado de Especificações do Sistema.

## 3.3 REQUISITOS DO SOFTWARE

O software irá obter informação das leituras realizadas nos sensores e informações das programações pré-estabelecidas pelo usuário. A resposta que o sistema gera ao usuário será em forma de arquivos de LOG, mensagens através de correio eletrônico e através do acionamento de dispositivos.

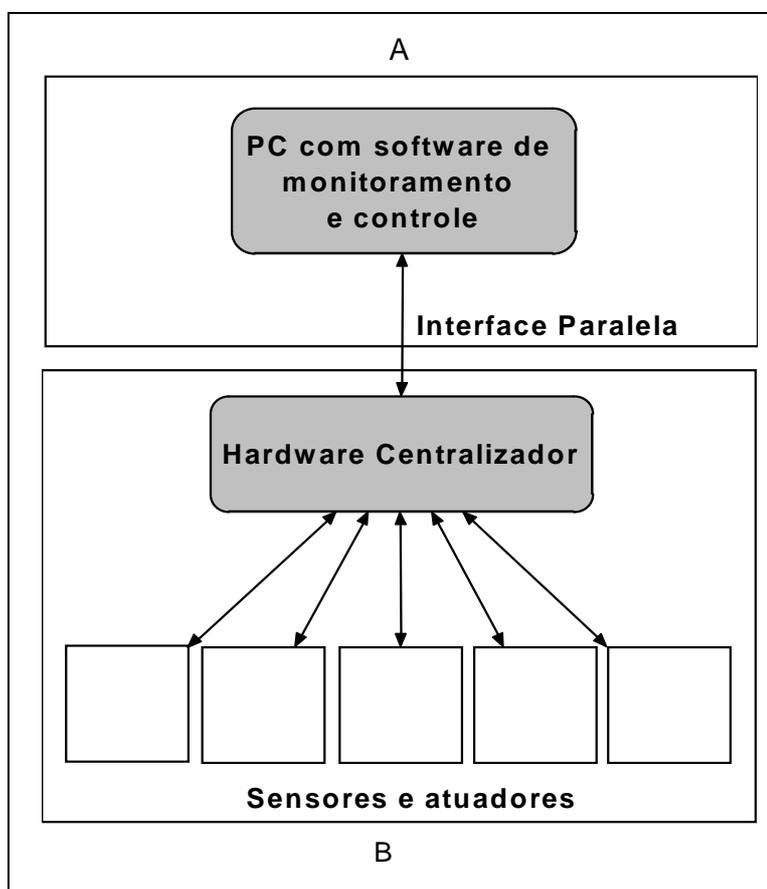
Como primeira etapa no desenvolvimento do sistema, foram levantados os requisitos do mesmo. Neste passo foi gerada uma lista que identifica exatamente as capacidades que o sistema final deverá executar para que o mesmo alcance o resultado final desejado:

- a) O protótipo deverá ser capaz de operar sobre o Microsoft Windows 95 / 98 / ME;
- b) Deverá ser capaz de utilizar uma conexão com a Internet, independente do tipo de conexão;
- c) Deverá ser capaz de utilizar uma porta de comunicação paralela;
- d) Deverá obter todos os sinais gerados pelos sensores;
- e) Deverá ser capaz de gravar em um arquivo de LOG todos os sinais obtidos;
- f) Deverá permitir ao usuário configurar as ações em relação ao sinal;
- g) Deverá permitir ao usuário configurar a ação em relação ao tempo;
- h) Deverá ser capaz de enviar por *e-mail* as mensagens e arquivos de LOG conforme configurações do usuário;

### 3.4 ESPECIFICAÇÕES DO SISTEMA

As especificações do sistema foram produzidas a partir dos requisitos levantados. Através deles, foram definidas as interações com o usuário (entradas e saídas), como também uma lista de funções que o sistema deverá executar para realizar as tarefas do usuário. Basicamente, os requisitos do sistema mostram o que o sistema deve fazer e, as especificações do sistema detalham como o sistema operará para executar estas tarefas. Conforme visto na Figura 4, a especificação do protótipo foi realizada visando a divisão do mesmo em dois módulos com funções bem distintas:

Figura 4: Visão geral do sistema



- a) Aplicação de monitorações e controle: parte do protótipo responsável pela leitura e tratamento dos sinais recebidos dos sensores e interface com o usuário. A mesma irá gravar os dados de configurações do usuário em tabelas. Para representar estas tabelas e os processos serão utilizados os modelos da especificação estruturada, conforme [POM02], usando a ferramenta Power Designer 6.1 para auxiliar;
- b) Hardware de monitoramento: parte do protótipo responsável por interagir com o

ambiente físico em que está instalado, deixando os dados disponíveis, para que a aplicação possa obtê-los.

A seguir será apresentada a primeiramente especificação da aplicação e posteriormente do hardware.

### 3.4.1 ESPECIFICAÇÃO DA APLICAÇÃO

Neste sistema toda capacidade de controle e decisão está centralizada na aplicação. O objetivo da aplicação é verificar o status dos sensores, gerar saídas para os atuadores, armazenar e aplicar das configurações do usuário e fornecer as respostas. A seguir estão relacionados os modelos que demonstram o *layout* da aplicação, e algumas características do seu funcionamento.

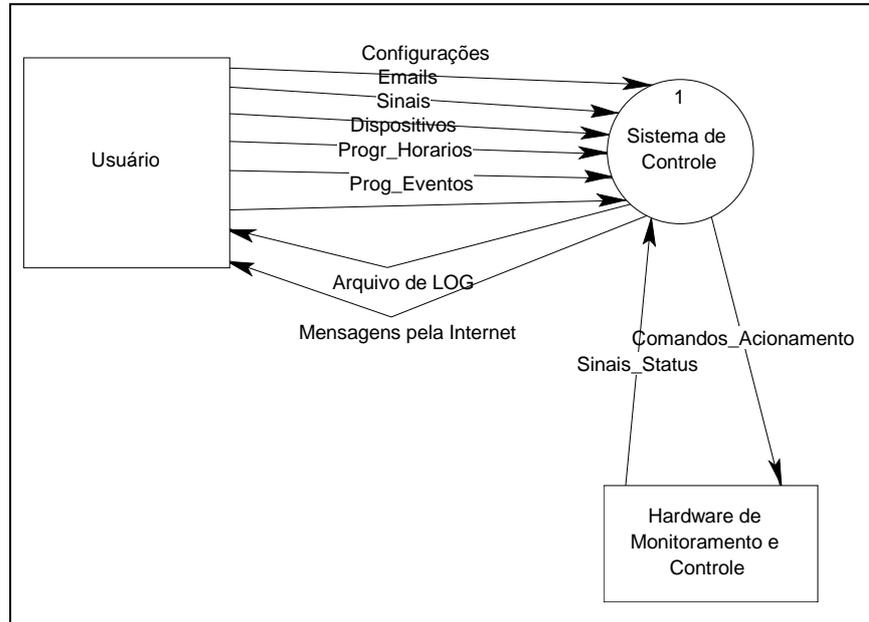
Na Tabela 5 está apresentada a lista de eventos da aplicação, que representam as funcionalidades que o software deverá possuir.

**Tabela 5: Lista de eventos da Aplicação**

1) Usuário informa configurações
2) Usuário informa lista de e-mails
3) Usuário informa sinais de status
4) Usuário informa dispositivos
5) Usuário informa programação por Horários
6) Usuário informa programação por Eventos
7) Sistema gera arquivo de LOG
8) Sistema envia mensagens pela internet
9) Sistema gera comandos de acionamento

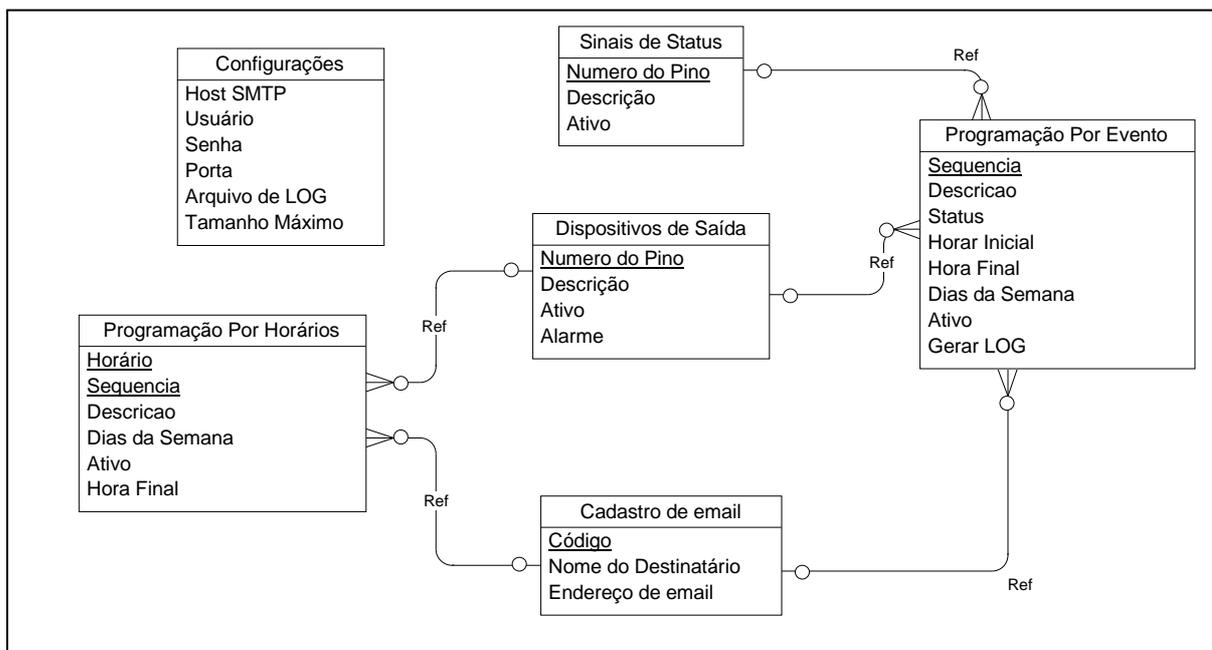
Apresentado na Figura 5, está o diagrama de contexto da aplicação, onde estão apresentados, em uma visão geral, os relacionamentos entre as entidades envolvidas.

**Figura 5: Diagrama de contexto**



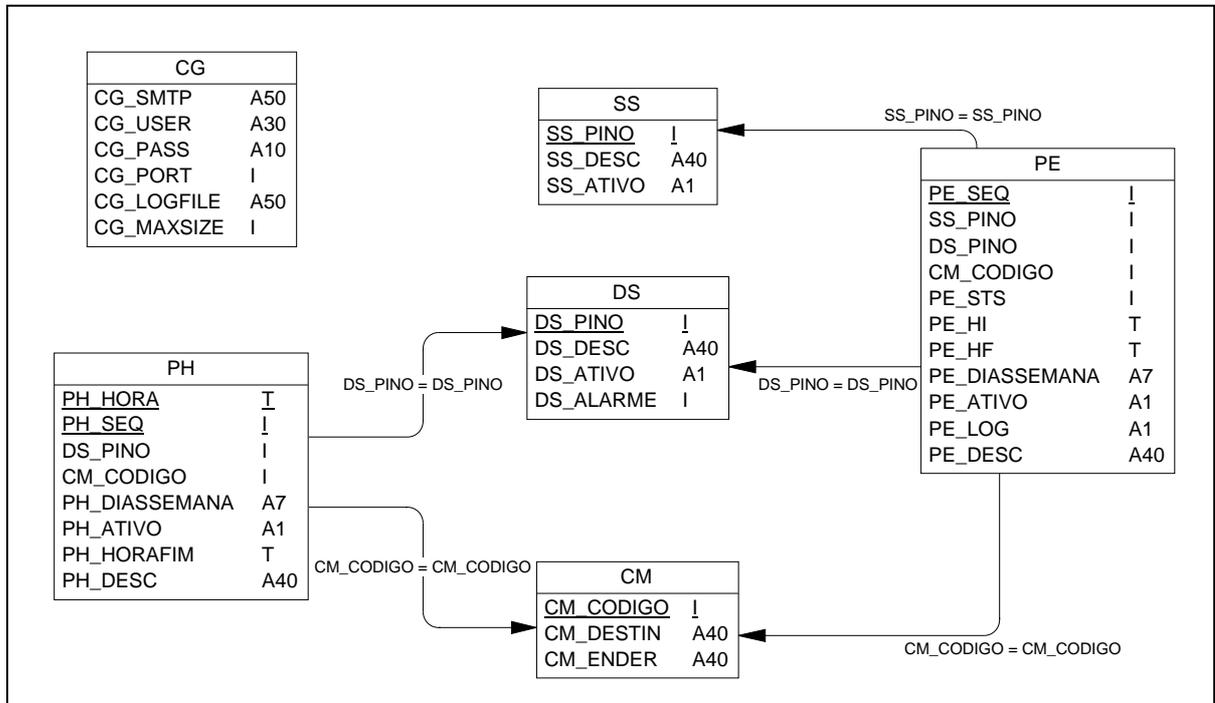
A Figura 6 apresenta como as tabelas da aplicação estão logicamente relacionadas.

**Figura 6: Modelo Entidade x Relacionamento Lógico**



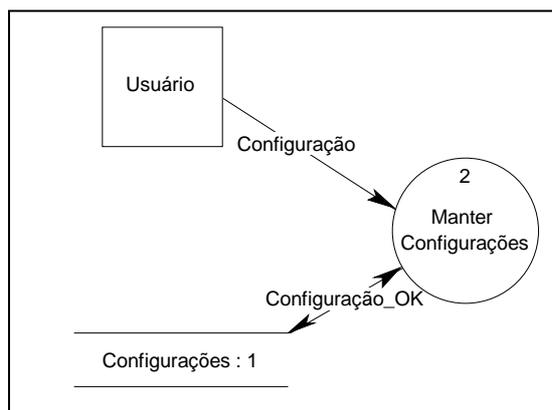
A Figura 7 apresenta como as tabelas da aplicação estão fisicamente relacionadas, e quais são seus atributos.

**Figura 7: Modelo Entidade x Relacionamento Físico**



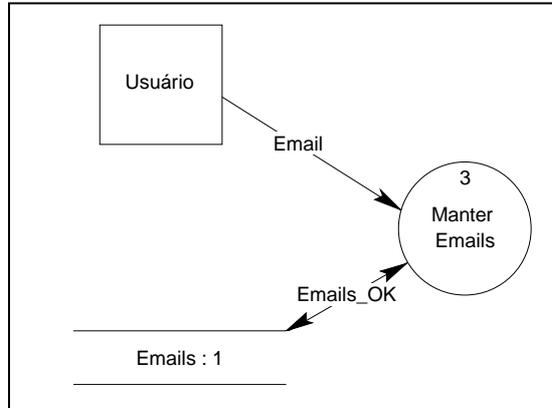
A Figura 8 apresenta o diagrama de fluxo de dados (DFD) do evento “Manter Configurações”, que representa o cadastro de configurações no sistema.

**Figura 8: Diagrama de fluxo de dados (DFD) do evento "Manter Configurações"**



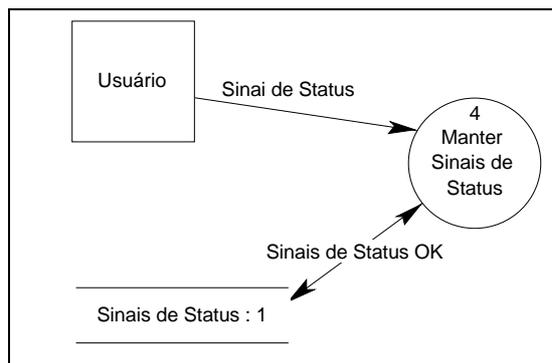
A Figura 9 apresenta o diagrama de fluxo de dados do evento “Manter emails”, que representa o cadastro de *e-mails* no sistema.

**Figura 9: DFD do evento "Manter Emails"**



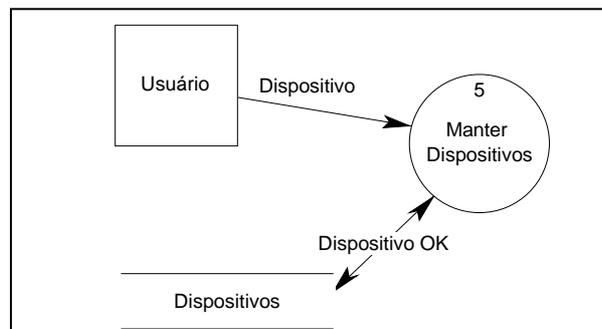
A Figura 10 apresenta o diagrama de fluxo de dados do evento “Manter sinais de status”, que representa o cadastro de sinais de status, ou sensores, no sistema.

**Figura 10: DFD do evento "Manter sinais de status"**



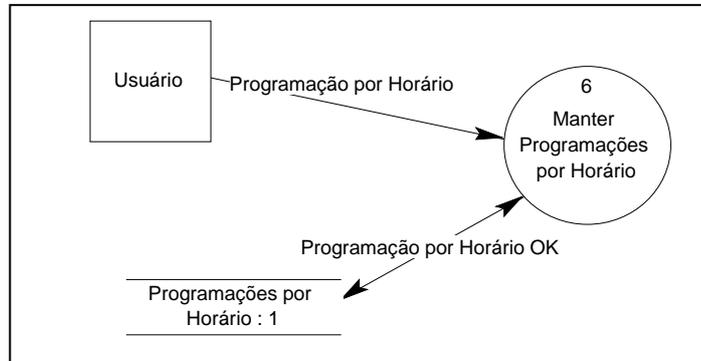
A Figura 11 apresenta o diagrama de fluxo de dados do evento “Manter dispositivos” que representa os dispositivos ou atuadores no sistema.

**Figura 11: DFD do evento "Manter dispositivos"**



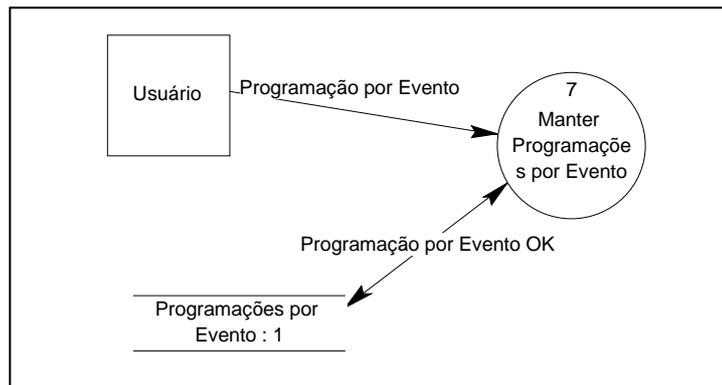
A Figura 12 apresenta o diagrama de fluxo de dados do evento “Manter programações por horário”, o mesmo representa o cadastro de programações por horários no sistema.

**Figura 12: DFD do evento "Manter programações por horário"**



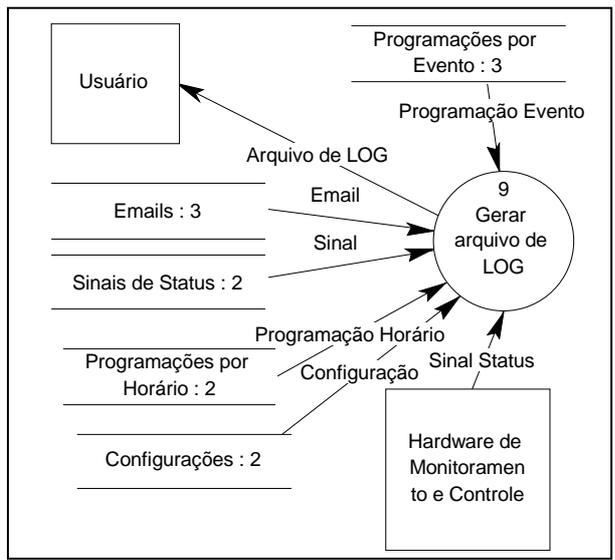
A Figura 13 apresenta o diagrama de fluxo de dados do evento “Manter programações por evento”, que representa o cadastro de programações por evento no sistema.

**Figura 13: DFD do evento "Manter programações por evento"**



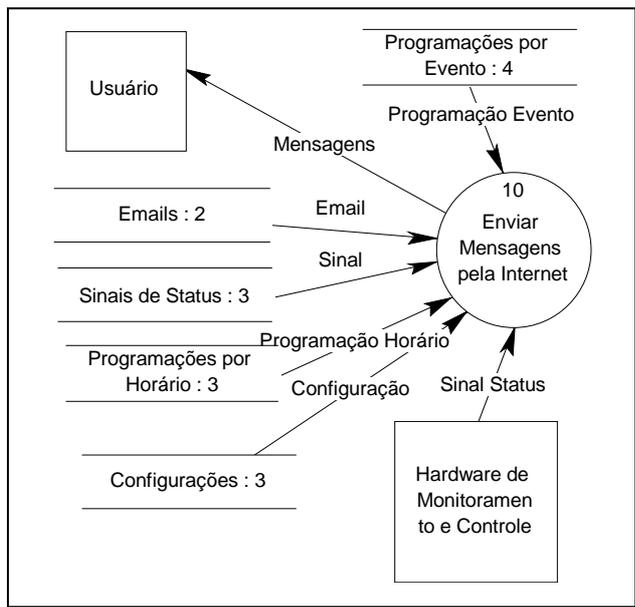
A Figura 14 apresenta o diagrama de fluxo de dados do evento “gerar arquivo de LOG”, que representa a rotina de geração do arquivo de LOG do sistema.

**Figura 14: DFD do evento "Gerar arquivo de LOG"**



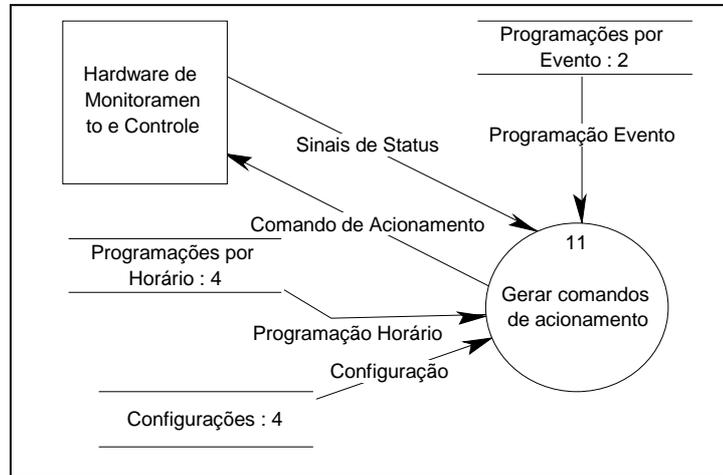
A Figura 15 apresenta o diagrama de fluxo de dados do evento “enviar mensagens pela internet”, que representa a rotina de envio de mensagens eletrônicas (*e-mails*) para o usuário.

**Figura 15: DFD do evento "Enviar mensagens pela internet"**



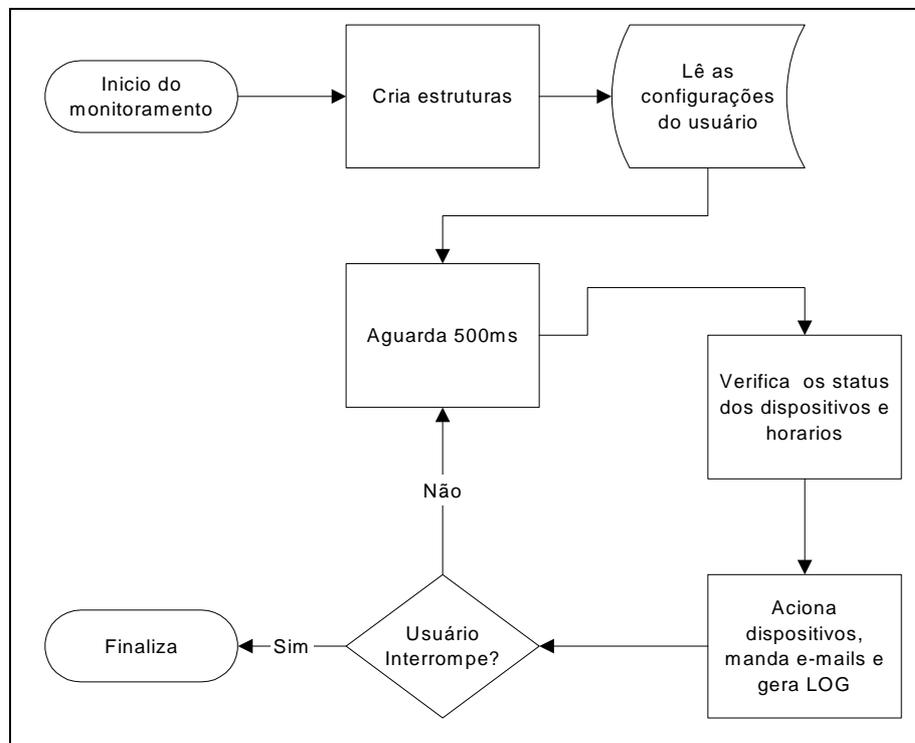
A Figura 16 apresenta o diagrama de fluxo de dados do evento “gerar comandos de acionamento”, que representa a rotina de controle dos atuadores do sistema.

**Figura 16: DFD do evento "Gerar comandos de acionamento"**



A Figura 17 apresenta um macro fluxo da rotina de monitoramento e controle. O seu código fonte está apresentado no Quadro 6 do Anexo I.

**Figura 17: Macro fluxo da rotina de monitoramento e controle.**



A Tabela 6 apresenta a lista de atributos de todas as tabelas contidas no sistema.

**Tabela 6: Lista de atributos das tabelas do sistema**

<b>Descrição do Campo</b>	<b>Nome</b>	<b>Tipo</b>
Alarme	DS_ALARME	I
Arquivo de LOG	CG_LOGFILE	A50
Ativo	PE_ATIVO	A1
Ativo	SS_ATIVO	A1
Ativo	PH_ATIVO	A1
Ativo	DS_ATIVO	A1
Código	CM_CODIGO	I
Descrição	PH_DESC	A40
Descrição	PE_DESC	A40
Descrição	SS_DESC	A40
Descrição	DS_DESC	A40
Dias da Semana	PH_DIASSEMANA	A7
Dias da Semana	PE_DIASSEMANA	A7
Dispositivo	PE_DS_PINO	BT1
Dispositivo	PH_DS_PINO	BT1
Email	PH_CM_CODIGO	BT2
Email	PE_CM_CODIGO	BT2
Endereço de email	CM_ENDER	A40
Gerar LOG	PE_LOG	A1
Hora Final	PH_HORAFIM	T
Hora Final	PE_HF	T
Horar Inicial	PE_HI	T
Horário	PH_HORA	T
Host SMTP	CG_SMTP	A50
Nome do Destinatário	CM_DESTIN	A40
Numero do Pino	DS_PINO	I
Numero do Pino	SS_PINO	I
Porta	CG_PORT	I
Senha	CG_PASS	A10
Seqüência	PH_SEQ	I
Seqüência	PE_SEQ	I
Sinal	PE_SS_PINO	BT1
Status	PE_STS	I
Tamanho Máximo	CG_MAXSIZE	I
Usuário	CG_USER	A30

A Tabela 7 apresenta uma lista de todos os fluxos de dados contidos nos diagramas de fluxo de dados anteriormente mostrados.

**Tabela 7: Dicionário dos fluxos de dados.**

<b>DFD</b>	<b>Fluxo</b>	<b>Campos</b>
Gerar arquivo de LOG	Arquivo de LOG	Todos
Gerar comandos de acionamento	Comando de Acionamento	DS PINO, CG PORT
Manter Configurações	Configuração	Todos
Manter Configurações	Configuração OK	Todos
Gerar Arquivo de LOG	Configurações	CG LOGFILE
Enviar Mensagem pela Internet	Configurações	CG SMTP, CG USER, CG PASS
Gerar comandos de acionamento	Configurações	CG PORT
Manter Dispositivo	Dispositivo	Todos
Manter Dispositivo	Dispositivo OK	Todos
Gerar Arquivo de LOG	Email	CM ENDER
Enviar Mensagem pela Internet	Email	CM ENDER
Manter emails	Emails	Todos
Manter emails	Emails OK	Todos
Enviar Mensagem pela Internet	Mensagens	Todos
Gerar Arquivo de LOG	Programação Evento	Todos
Enviar Mensagem pela Internet	Programação Evento	Todos
Gerar comandos de acionamento	Programação Evento	Todos
Gerar Arquivo de LOG	Programação Horário	Todos
Enviar Mensagem pela Internet	Programação Horário	Todos
Gerar comandos de acionamento	Programação Horário	Todos
Manter Programação por Eventos	Programação por Evento	Todos
Manter Programação por Eventos	Programação por Evento OK	Todos
Manter Programação por Horário	Programação por Horário	Todos
Manter Programação por Horário	Programação por Horário OK	Todos
Gerar comandos de acionamento	Sinais de Status	SS PINO, CG PORT
Manter Sinais de Status	Sinais de Status	Todos
Manter Sinais de Status	Sinais de Status OK	Todos
Gerar Arquivo de LOG	Sinal	Todos
Enviar Mensagem pela Internet	Sinal	Todos
Enviar Mensagem pela Internet	Sinal Status	SS PINO, CG PORT
Gerar Arquivo de LOG	Sinal Status	SS PINO, CG PORT

### 3.4.2 ESPECIFICAÇÃO DO HARDWARE

Neste trabalho o hardware construído tem a função de apenas facilitar a conexão dos sensores e atuadores à porta paralela. Por isso ele deve ser mais simples e de menor custo possível. A seguir estão os esquemáticos simplificados dos circuitos contidos na placa centralizadora de sinais. Os esquemáticos foram desenhados com auxílio do software Eagle Layout Editor 4.09. Maiores informações sobre o Eagle Layout Editor podem ser encontradas em CadSoft Online (2002).

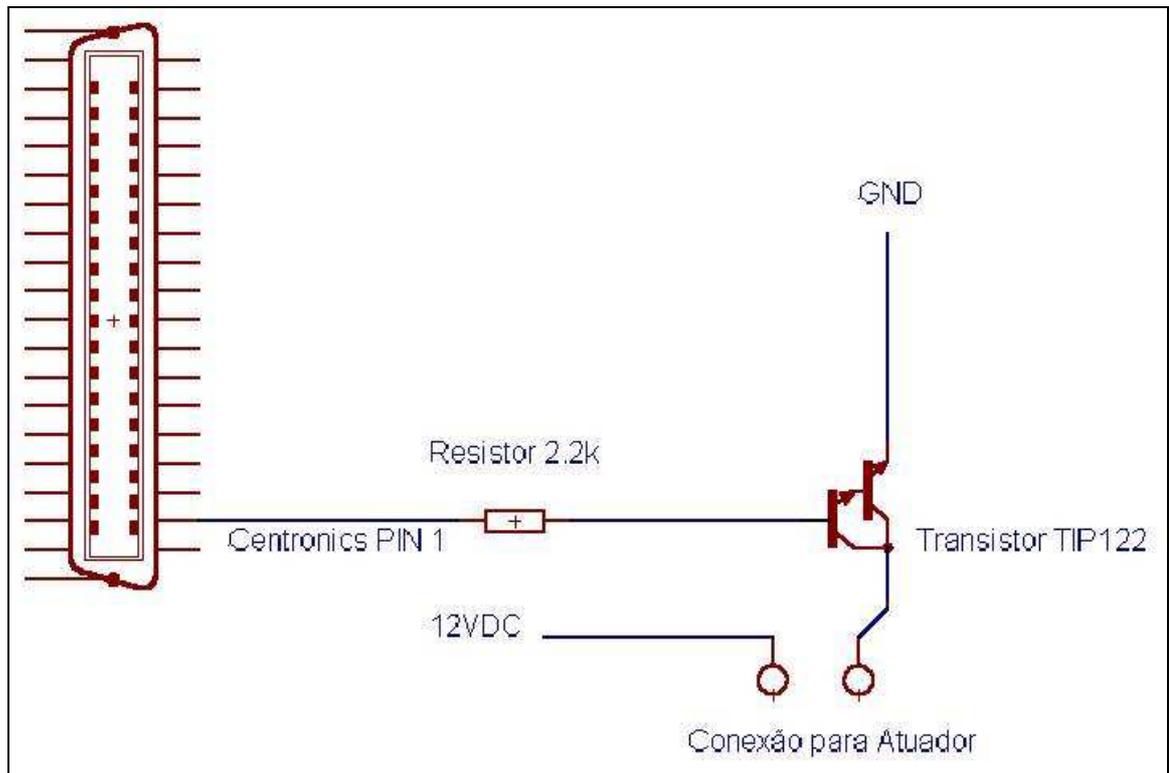
Na Tabela 8 está apresentada a lista de endereços da porta paralela no microcomputador, através desses endereços são acessados os dados na porta. Para acessar o byte de Status o endereço é o endereço base adicionando 1 (BASE+1). E para acessar o byte de Controle o endereço é o endereço base adicionando 2 (BASE+2).

**Tabela 8: Endereços da porta paralela no microcomputador.**

<b>PORTA</b>	<b>Endereço Base em Hexadecimal</b>	<b>Endereço base em Decimal</b>
LPT1	378h	888
LPT2	278h	632

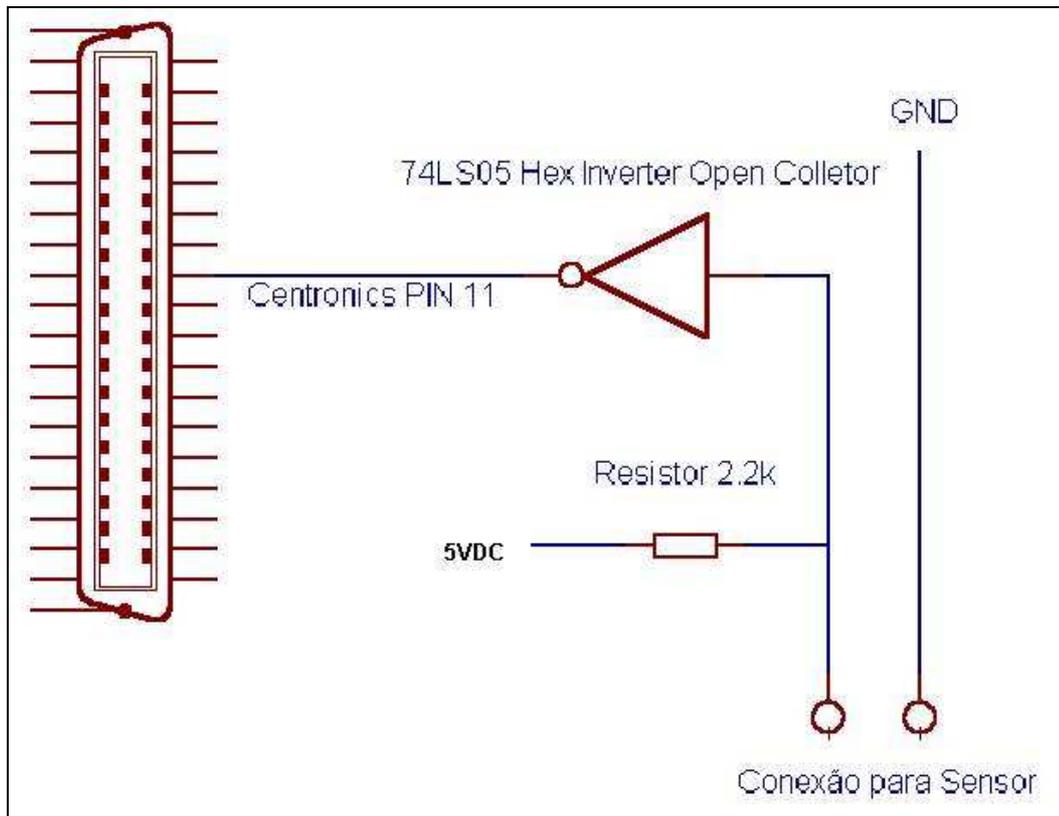
A Figura 18 apresenta os componentes e a disposição dos mesmos na placa centralizadora de sinais. O objetivo deste circuito é controlar o fornecimento de corrente para o atuador que estiver conectado em “Conexão para Atuador”. Para isso, quando o pino “Centronics PIN 2” for ligado por software, fornecerá um sinal para o transistor “TIP122”, que fará com que a corrente feche na “Conexão para Atuador”. O resistor contido no circuito serve apenas para controlar a corrente que o transistor receberá na base. Os 12 volts de corrente contínua “12VDC” são provenientes de uma fonte de alimentação conectada na placa e “GND” deve estar conectado no “terra” da fonte e do conector *centronics* simultaneamente.

Figura 18: Esquemático simplificado do circuito dos atuadores na placa centralizadora.



A Figura 19 apresenta de forma simplificada os componentes e as suas interconexões na placa centralizadora de sinais. Está sendo mostrada apenas uma conexão, as demais são de forma semelhante. O objetivo deste circuito é permitir que o estado de um sensor conectado em “Conexão para Sensor” possa ser lido por software. O componente 74LS05 é um inversor com coletor aberto, utilizado para os pinos 1, 14, 16, 17 do conector *centronics*, para os outros pinos (11, 12, 13,15), o esquemático fica com as mesmas conexões porém sem o componente 74LS05.

Figura 19: Esquemático simplificado do circuito dos sensores na placa centralizadora.



### 3.5 IMPLEMENTAÇÃO

A seguir serão apresentadas considerações sobre a implementação do protótipo, como as ferramentas que foram utilizadas na implementação, tanto da aplicação quanto do hardware. Também serão apresentadas as principais funções que compõem o protótipo e o funcionamento do mesmo.

#### 3.5.1 FERRAMENTAS UTILIZADAS NA IMPLEMENTAÇÃO

A implementação da aplicação de monitoramento e controle foi desenvolvida na linguagem *ObjectPascal* utilizando para tal implementação o ambiente de desenvolvimento Borland Delphi 5.0. Foi utilizada também a linguagem *Assembly* para realizar a comunicação com a porta paralela do microcomputador.

Para gravar e ler as configurações do usuário, armazenadas no banco de dados Paradox 5, foi utilizada a linguagem *Structured Query Language (SQL)*. A comunicação entre a

aplicação e o banco de dados é realizada a partir do suporte do *Borland Database Engine* (BDE) e de uma fonte de dados ODBC (Open Database Conectivity).

ODBC é uma interface de programação que permite que os aplicativos acessem dados em sistemas de gerenciamento de banco de dados que usam SQL como padrão de acesso a dados.

Para a confecção da placa de circuito impresso (PCB) responsável pela centralização dos sinais e na confecção das placas auxiliares, foi utilizado para o desenho o software Tango 2.0, que é específico para este fim.

### 3.5.2 PRINCIPAIS FUNÇÕES DA APLICAÇÃO

As funções mostradas nesta seção correspondem aos processos que realizam a comunicação via porta paralela com a placa centralizadora dos sinais, o processo que envia uma mensagem ao usuário via correio eletrônico e a rotina geradora do arquivo de LOG.

No Quadro 1 estão relacionadas as funções e procedimentos que fazem o acesso em baixo nível aos bits de dados, status e controle da porta paralela. Por ter sido utilizada a linguagem *Assembly*, o acesso é feito direto ao endereço da porta, com isso as versões do sistema operacional Windows baseadas na tecnologia NT, ou seja, Windows NT, Windows 2000 e Windows XP não permitirão o uso dessas funções, inviabilizando a execução do protótipo nestas versões. Para resolver este problema poderiam ser utilizadas funções da API de cada versão do Windows para o acesso à porta paralela. Portanto, este protótipo está apto a ser executado nos sistemas operacionais Windows 95, Windows 98 e Windows ME.

**Quadro 1: Funções de leitura e escrita na porta paralela.**

```
Interface

function LeByteBase(vPorta:Byte) : Byte;
function LeByteBaseM1(vPorta:Byte) : Byte;
function LeByteBaseM2(vPorta:Byte) : Byte;
procedure EscreveByteBase(vPorta, ByteOut:Byte);
procedure InicializaPorta(vPorta:Byte);

implementation

{Le Byte de dados }
function LeByteBase(vPorta:Byte) : Byte;
Var
  EndPorta : Word;
  RetByte : Byte;
begin
```

```

Case vPorta of
  0 : EndPorta := $378;   { Lpt1 }
  1 : EndPorta := $278;   { Lpt2 }
end;

asm
  mov dx,EndPorta
  in  al,dx
  mov RetByte,al
end;

Result := RetByte;
end;

{ Le Byte de Status }
function LeByteBaseM1(vPorta:Byte) : Byte;
Var
  EndPorta : Word;
  RetByte : Byte;
begin
  Case vPorta of
    0 : EndPorta := $378 + 1;   { Lpt1 }
    1 : EndPorta := $278 + 1;   { Lpt2 }
  end;

  asm
    mov dx,EndPorta
    in  al,dx
    mov RetByte,al
  end;

  Result := RetByte;
end;

{ Le Bytes de Controle }
function LeByteBaseM2(vPorta:Byte) : Byte;
Var
  EndPorta : Word;
  RetByte : Byte;
begin
  Case vPorta of
    0 : EndPorta := $378 + 2;   { Lpt1 }
    1 : EndPorta := $278 + 2;   { Lpt2 }
  end;

  asm
    mov dx,EndPorta
    in  al,dx
    mov RetByte,al
  end;

  Result := RetByte;
end;

{ Escreve byte na porta: bits de dados }
procedure EscreveByteBase(vPorta, ByteOut:Byte);
Var
  EndPorta : Word;
begin
  Case vPorta of
    0 : EndPorta := $378;   { Lpt1 }
    1 : EndPorta := $278;   { Lpt2 }
  end;

  asm

```

```

    mov dx,EndPorta
    mov ax,0
    mov al, ByteOut
    out dx,ax
end;
end;

{ Zera os bits do byte de dados }
procedure InicializaPorta(vPorta:Byte);
Var
    EndPorta : Word;
begin
    Case vPorta of
        0 : EndPorta := $378;    { Lpt1 }
        1 : EndPorta := $278;    { Lpt2 }
    end;

    asm
        mov dx,EndPorta        { Grava byte 0 na porta}
        mov ax,0
        mov al,0
        out dx,ax
    end;
end;
end;

```

No Quadro 2 estão listadas as funções utilizadas pela rotina de monitoramento e controle e fazem chamadas às rotinas de baixo nível anteriormente listadas do Quadro 1. Nestas rotinas estão os tratamentos lógicos para efetuar o acesso individual a cada bit da porta, que corresponde a um pino.

#### Quadro 2: Funções de controle da porta paralela.

```

Const
    BIT0 = $01; // 0000-0001
    BIT1 = $02; // 0000-0010
    BIT2 = $04; // 0000-0100
    BIT3 = $08; // 0000-1000
    BIT4 = $10; // 0001-0000
    BIT5 = $20; // 0010-0000
    BIT6 = $40; // 0100-0000
    BIT7 = $80; // 1000-0000

function TrataPinoBase(vPorta, Servico:Byte; Pino:Byte): Boolean;
function LePinoStatus(vPorta, Pino:Byte): Boolean;

implementation

{ Liga, desliga e le status dos bits de dados da porta }
function TrataPinoBase(vPorta, Servico:Byte; Pino:Byte): Boolean;
Var
    ByteLido:Byte;
    Bitx : Byte;
begin
    Bitx := 0;
    ByteLido := 0;

    Case Pino of
        1: Bitx := BIT0;
        2: Bitx := BIT1;

```

```

3: Bitx := BIT2;
4: Bitx := BIT3;
5: Bitx := BIT4;
6: Bitx := BIT5;
7: Bitx := BIT6;
8: Bitx := BIT7;
end;

Case Servico of
1: begin // Liga Pino
    ByteLido := LeByteBase(vPorta);
    ByteLido := ByteLido or Bitx;
    EscreveByteBase(vPorta, ByteLido);
    end;
2: begin // Desliga pino
    ByteLido := LeByteBase(vPorta);
    ByteLido := ByteLido and (Bitx xor 255);
    EscreveByteBase(vPorta, ByteLido);
    end;
3: begin // Apenas retorna o status do pino
    ByteLido := LeByteBase(vPorta);
    end;
end;

Result := (ByteLido and Bitx) <> 0;
end;

{ Le o status dos bits de status da porta paralela}
function LePinoStatus(vPorta, Pino:Byte):Boolean;
Var
    ByteLido: Byte;
    Bitx : Byte;
begin
    Bitx := 0;

    Case Pino of
        1: Bitx := BIT7;
        2: Bitx := BIT5;
        3: Bitx := BIT4;
        4: Bitx := BIT6;
    end;
    ByteLido := LeByteBaseM1(vPorta);
    Result := (ByteLido and Bitx) = 0;

    If Pino = 1 then // Bit invertido
        Result := not Result;

    If Pino in [5,6,7,8] then
        Result := false;
    end;
end.

```

O Quadro 3 apresenta a rotina responsável pelo envio de mensagens por *e-mail*. Por ser uma operação que pode levar algum tempo para ser concluída, levando em conta que é aberta uma operação de comunicação ponto a ponto com um servidor de envio de mensagens (SMTP *server*), a rotina foi desenvolvida para ser executada em paralelo, não podendo prejudicar, então, a performance da rotina de monitoramento e controle.

### Quadro 3: Rotina de envio de e-mail utilizando Thread.

```
TMandaMail = class(TThread)
private
protected
    procedure Execute; override;
public
    PosMail : Byte;
    Tipo : Byte; // 1 - PH, 2 - PE
end;

{ Cria, Prepara e dispara a Thread }
Procedure TFPanCE.MandaEmailPrg(pos, tip:Byte);
Var
    MandaMail : TMandaMail;
begin
    MandaMail := TMandaMail.Create(True);
    MandaMail.PosMail := pos;
    MandaMail.Tipo := tip;
    MandaMail.FreeOnTerminate := True;
    MandaMail.Resume;
end;

{ Método Execute da Thread MandaMail }
Procedure TMandaMail.Execute;
Var
    Email : TNMSmtp;
    QMail, QCfg : TQuery;
begin
    Try
        QCfg := TQuery.Create(Application);
        QMail := TQuery.Create(Application);
        Try
            QCfg.DatabaseName := 'SISM';
            QCfg.SQL.Text := 'SELECT * FROM CG';
            QCfg.Open;

            QMail.DatabaseName := 'SISM';
            QMail.SQL.Text := 'SELECT * FROM CM WHERE CM_CODIGO = :cm';
            If Tipo = 1 then // PH
                QMail.ParamByName('cm').AsInteger := ProHor[PosMail].CodMai
            else // PE
                QMail.ParamByName('cm').AsInteger := ProEve[PosMail].CodMai;
            QMail.Open;

            GravaLog('Email sendo enviado para:'+QMail.FieldByName('CM_ENDER').AsString);

            Email := TNMSmtp.Create(Application);
            Try

                // Seta os parametros para o envio do email

                EMail.Host := QCfg.FieldByName('CG_SMTP').AsString;
                EMail.UserID := QCfg.FieldByName('CG_USER').AsString;
                EMail.PostMessage.FromAddress := 'erasmo@inf.furb.br';
                EMail.PostMessage.FromName := 'SISM-NET / Erasmo Kruger';

                EMail.PostMessage.ToAddress.Clear;
                EMail.PostMessage.ToAddress.Add(QMail.FieldByName('CM_ENDER').AsString);
                if Tipo = 1 then // PH
                    EMail.PostMessage.Subject := 'SISM Status: '+
                        String(ProHor[PosMail].DesPro)
                else // PE
```

```

Email.PostMessage.Subject := 'SISM Status Alerta: '+
                             String(ProEve[PosMail].DesPro);

Email.PostMessage.Body.Clear;
Email.PostMessage.Body.Add('Mensagem enviada automaticamente'+
                             ' pelo sistema SISM-NET');
Email.PostMessage.Body.Add('TCC de Erasmo Kruger');
Email.PostMessage.Body.Add('Orientador: Antonio Carlos Tavares');
if Tipo = 1 then // PH
    Email.PostMessage.Body.Add('Evento Ocorrido: Acionamento de '+
                                'Dispositivo por Horário')
else // PE
    Email.PostMessage.Body.Add('Evento Ocorrido: Alerta Detectado '+
                                'no sensor: '+String(ProEve[PosMail].DesPro));
Email.PostMessage.Body.Add('Arquivo de LOG Anexado. ');

Email.PostMessage.Attachments.Clear;
if FileExists(QCfg.FieldByName('CG_LOGFILE').AsString) then
    Email.PostMessage.Attachments.Add(QCfg.FieldByName('CG_LOGFILE').AsString);

// Conecta ao host e envia o email

Email.Connect;
Email.SendMail;
Email.Disconnect;
Finally
    Email.Free;
end;
Finally
    QCfg.Free;
    QMail.Free;
End;

except
    // Gera msg no log
    On E:Exception do
        GravaLog('Não foi possível enviar e-mail. Mensagem original: '+E.Message);
    end;
end;
end;

```

Para executar este processo em paralelo foi utilizada a classe TThread, e para enviar mensagens via *e-mails* foi utilizada a classe TNMSmtp, ambas disponíveis no Delphi 5.

A classe TNMSmtp é derivada da classe TPowerSock, que implementa um *Socket* de comunicação entre *hosts* da internet.

A rotina listada no Quadro 4 cria o arquivo de LOG e adiciona as linhas no mesmo. Esta rotina centraliza esta funcionalidade e é chamada de vários pontos do processo de monitoramento e controle, apresentado no Quadro 6 no Anexo I.

#### Quadro 4: Rotina de geração de LOG.

```
// Grava nova ocorrencia no arquivo de LOG
Procedure GravaLog(Msg:String);
Var
  Log: TextFile;

begin
  AssignFile(Log, gLogFile);
  If FileExists(gLogFile) then
    Append(Log)
  else
    begin
      Rewrite(Log);
      Writeln(Log, 'SISM-NET - Arquivo de LOG');
      Writeln(Log, 'Aplicativo desenvolvido por Erasmo Kruger para TCC - 2002/2');
      Writeln(Log, 'Orientador: Prof. Antonio Carlos Tavares');
      Writeln(Log, 'Data de Criação do arquivo: '+DateTimeToStr(Now));
      Writeln(Log, ' ');
    end;

  Writeln(Log, DateTimeToStr(Now)+' : '+Msg);
  CloseFile(Log);
end;
```

### 3.5.3 PRINCIPAIS FUNÇÕES DO HARDWARE

Como já visto anteriormente, a aplicação utilizará a porta paralela para enviar os comandos e receber os status dos dispositivos. Como uma porta paralela possui um número limitado de sinais, a sua utilização ficou distribuída conforme Tabela 9. Cada sinal será identificado no sistema com um nome “amigável” e ficará transparente para o usuário que pino da porta paralela será acessado.

Para facilitar a conexão dos sensores e atuadores conforme a Tabela 9 foi criada a placa centralizadora dos sinais que possui os seguintes itens:

- a) 1 conector paralelo do tipo *Centronics*;
- b) 1 entrada para alimentação de 12 volts de corrente contínua (DC);
- c) 8 conexões com entradas para fios terra individuais, para ligação dos sensores;
- d) 8 conexões com entradas para fios terra individuais, para ligação dos atuadores, que receberão uma corrente de 12 volts, caso acionados.

Os endereços citados na Tabela 9 como “Base”, corresponde a 378H em hexadecimal para a LPT1 e 278H em hexadecimal para a LPT2, como já visto anteriormente na Tabela 8.

**Tabela 9: Utilização dos pinos da porta paralela.**

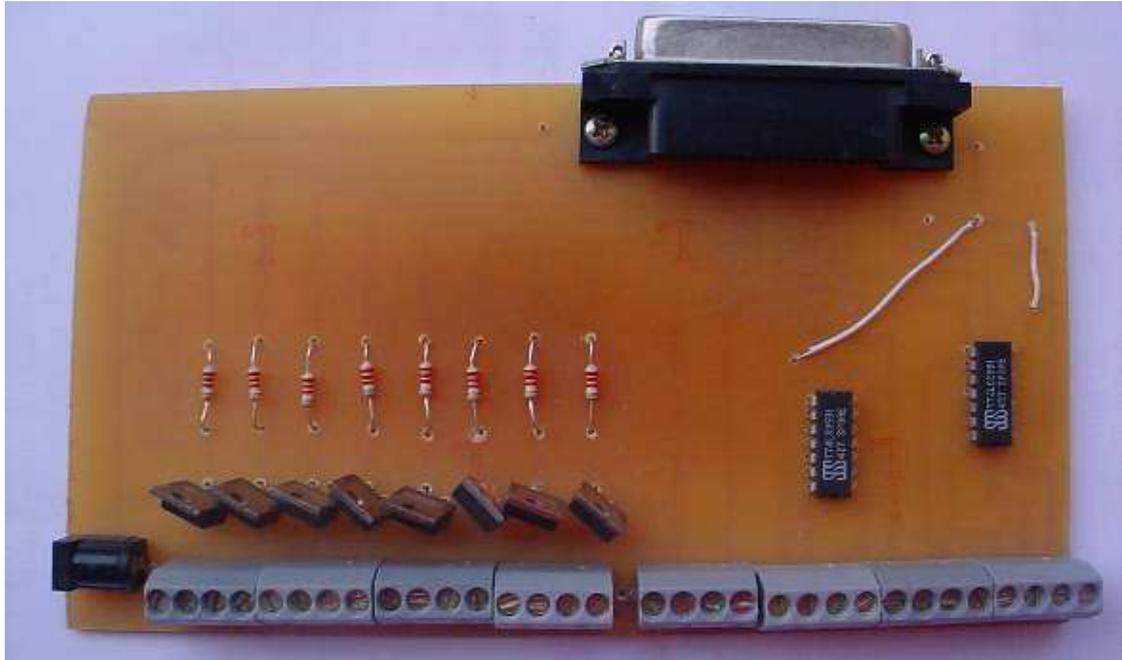
<b>Pino</b>	<b>Significado DB25</b>	<b>Finalidade</b>	<b>Observação</b>	<b>Endereço/Tipo</b>
1	Strobe	Pino Sensor 1	Bit Invertido	Base+2/Controle
2	Data 0	Pino Atuador 1		Base/Dados
3	Data 1	Pino Atuador 2		Base/Dados
4	Data 2	Pino Atuador 3		Base/Dados
5	Data 3	Pino Atuador 4		Base/Dados
6	Data 4	Pino Atuador 5		Base/Dados
7	Data 5	Pino Atuador 6		Base/Dados
8	Data 6	Pino Atuador 7		Base/Dados
9	Data 7	Pino Atuador 8		Base/Dados
11	Busy	Pino Sensor 2	Bit Invertido	Base+1/Status
12	Paper End	Pino Sensor 3		Base+1/Status
13	Select	Pino Sensor 4		Base+1/Status
14	Auto Feed	Pino Sensor 5	Bit Invertido	Base+2/Controle
15	Error	Pino Sensor 6	Bit Invertido	Base+1/Status
16	Init	Pino Sensor 7		Base+2/Controle
17	Select In	Pino Sensor 8	Bit Invertido	Base+2/Controle

As funções desta placa, conectada ao microcomputador através do conector *Centronics*, são:

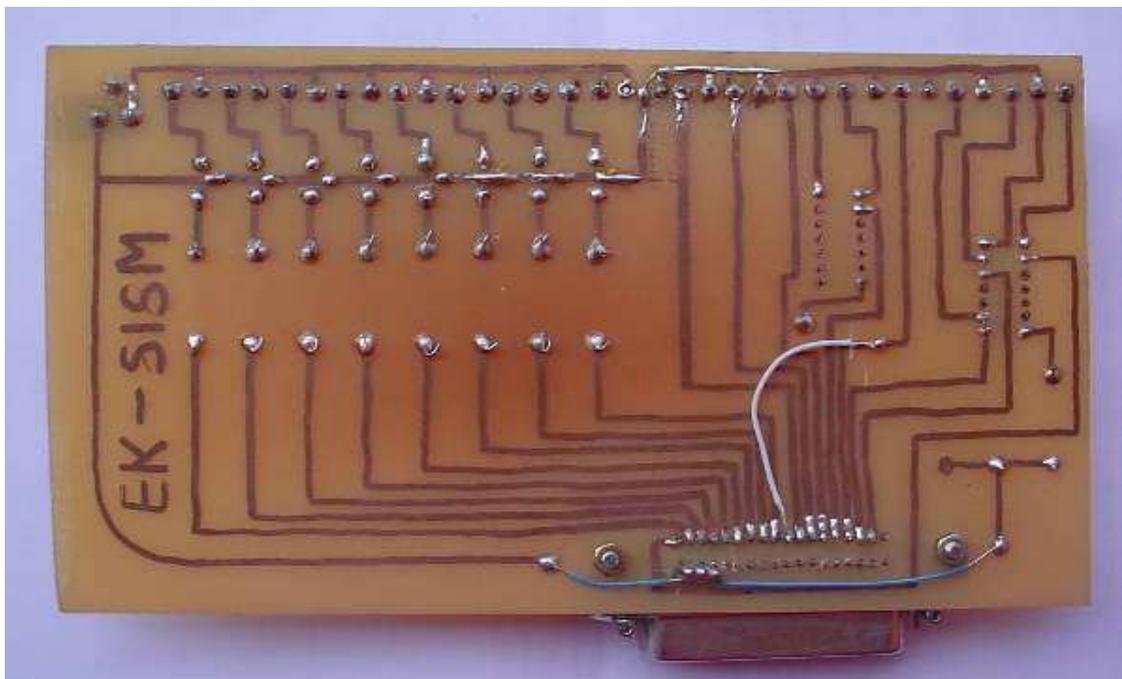
- a) Interligar os conectores dos sensores aos pinos do conector *Centronics* correspondente;
- b) Ativar e desativar o fornecimento de corrente elétrica aos conectores dos atuadores;
- c) Permitir a inserção, retirada e troca de posição dos dispositivos sem a necessidade de soldas ou outros procedimentos especiais.
- d) Permitir a utilização de um cabo paralelo comum, semelhante ao utilizado na maioria das impressoras.

Nas figuras a seguir (Figura 20 e Figura 21) estão apresentadas imagens da placa confeccionada para o protótipo.

**Figura 20: Foto da placa centralizadora (Componentes).**



**Figura 21: Foto da placa centralizadora (trilhas).**



### 3.5.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO

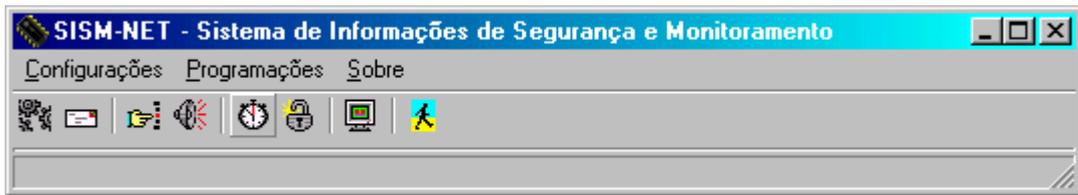
Nesta seção é apresentado o funcionamento do protótipo. Para o funcionamento correto do sistema deverão ser observados os seguintes itens:

- a) Deverá haver um *alias* para o banco de dados Paradox chamado “SISM”, que poderá ser criado no gerenciador de fontes ODBC ou no *Database Desktop* do Delphi 5. Este alias deverá estar direcionado para a pasta onde estiverem as tabelas do sistema;
- b) O cabo paralelo deverá estar corretamente conectado na porta paralela do microcomputador e no conector *Centronics* da placa centralizadora de sinais;
- c) A fonte de alimentação deve estar corretamente conectada no local especificado na placa centralizadora de sinais. Esta fonte deve fornecer tensão de 12 volts, corrente contínua (12 VDC 1A).
- d) Os sensores e atuadores devem estar conectados corretamente conforme necessidades do usuário.

A Figura 22 apresenta a tela principal do aplicativo que possui um menu com as seguinte opções:

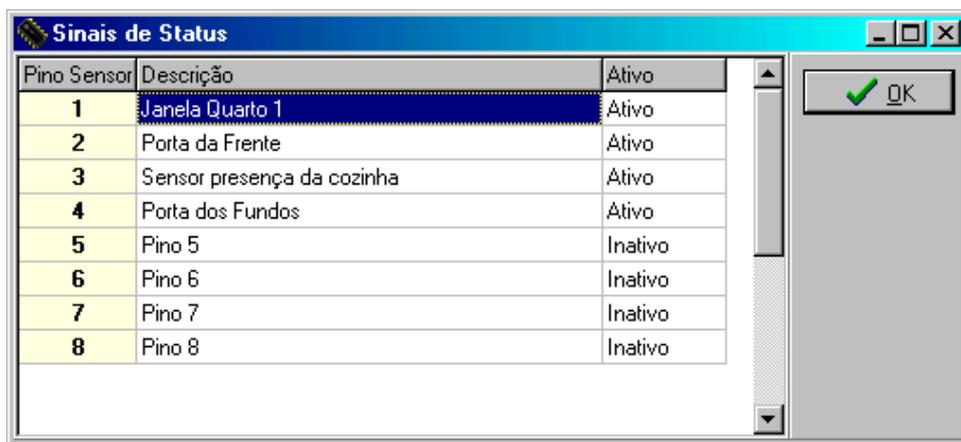
- a) configurações / sinais de status: é um cadastro onde são relacionados os sensores;
- b) configurações / dispositivos: é um cadastro onde são relacionados os atuadores;
- c) configurações / lista de e-mails: é um catálogo de endereços de e-mail;
- d) configurações / gerais: algumas informações gerais do aplicativo;
- e) programações / por horários: cadastro de ações a serem realizadas, baseadas em um determinado horário;
- f) programações / por eventos: cadastro de ações a serem realizadas, baseadas em um determinado sinal de um sensor;
- g) programações / painel de operação: abre uma tela de status e dispara a rotina de monitoramento e controle, baseada nas configurações anteriormente cadastradas;
- h) programações / painel de testes: abre uma tela onde são visualizados os status dos sensores e podem ser acionados individualmente cada um dos atuadores;
- i) sobre / sobre o SISM-NET: exibe algumas informações sobre o aplicativo;
- j) sobre / sair: encerra a execução do aplicativo.

Figura 22: Tela principal do aplicativo.



A Figura 23 apresenta o cadastro de sensores, onde são cadastrados nomes “amigáveis” para cada pino dos sinais de status da porta paralela. Os sensores que estiverem marcados como inativos não serão considerados no monitoramento.

Figura 23: Tela de cadastro dos sensores.



A Figura 24 apresenta o cadastro dos atuadores, onde são cadastrados os nomes “amigáveis” para os pinos relacionados aos atuadores da porta paralela. Os dispositivos que estiverem marcados como alarme terão um tratamento especial no monitoramento e ficarão ligados por menos tempo.

Figura 24: Tela de cadastro dos dispositivos atuadores.



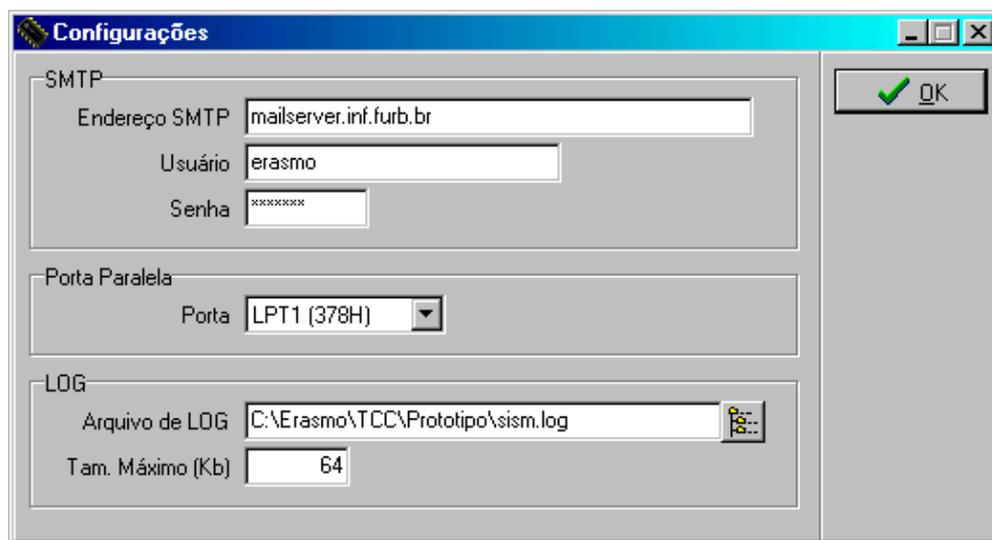
A Figura 25 apresenta o cadastro de endereços de e-mail, que será utilizado como catálogo nas telas de programação por horário e programação por evento.

**Figura 25: Catálogo de endereços de e-mail.**



A Figura 26 apresenta a tela de configurações gerais do aplicativo. Os dados da caixa SMTP são utilizados para estabelecer conexão com o servidor de mensagens enviadas. No campo "porta" é indicada qual porta paralela será utilizada na operação. E na caixa LOG são informados a pasta e o nome do arquivo de LOG e o tamanho máximo que o arquivo poderá ter.

**Figura 26: Tela de configurações gerais.**



Após feitas as configurações iniciais acima, poderão então, serem feitos os cadastros das programações a seguir, lembrando que todas as configurações podem ser ajustadas e adicionadas posteriormente.

A Figura 27 apresenta a tela que contém a relação de programações por horários cadastradas e permite a inclusão, alteração e exclusão de programações.

**Figura 27: Tela de cadastro de programações por horário.**



Programação	Hora Inicial	Hora Final	Dias da Semana	Dispositivo	E-mail
Controlar Luz Quarto	17:00:00	23:45:00	Dom, Seg, Ter, Qua, Qui, Sex	Luz Quarto	Erasmus Krüger
Controlar Luz da Sacada	20:00:00	06:00:00	Dom, Seg, Ter, Qua, Qui, Sex	Luz Sacada	Erasmus (Comercial)
Teste Semanal da Sirene	20:16:00	20:16:15	Dom	Sirene	Tavares
Controlar Luz do Jardim	21:00:00	23:00:00	Qui, Sex	Luz Jardim	Erasmus Krüger

A Figura 29 apresenta a tela de cadastro e alteração de programações por horário. Os dados desta tela possuem as seguintes funções:

- descrição: é dada uma descrição para a programação, para fins de visualização;
- horário: determina as horas inicial e final que o atuador ficará ligado;
- dias da semana: determina em quais dias da semana a programação será ativada;
- dispositivo: indica qual atuador será acionado;
- e-mail: indica para qual endereço de e-mail será enviada uma mensagem com as informações atuais e com o arquivo de LOG anexado;
- programação ativa: indica se esta programação está sendo usada atualmente.

**Figura 28: Tela de cadastro e alteração de uma programação por horário.**

Programação por Horário

Descrição  
Controlar Luz Quarto

Horário  
Hora Inicial: 17:00:00  
Hora Final: 23:45:00

Dias da Semana  
 Domingo  Quarta  Sábado  
 Segunda  Quinta  
 Terça  Sexta

Parâmetros  
Dispositivo: Luz Quarto  
E-Mail: Erasmus Krüger

Programação Ativa

OK Cancelar

A Figura 29 apresenta a tela que contém a lista de programações por eventos cadastradas e permite a inclusão, alteração e exclusão de programações.

**Figura 29: Tela de cadastro das programações por evento.**

Programação	Hora Inicial	Hora Final	Sensor	Dispositivo	E-Mail
Monitoramento Porta dos Fundos	08:00:00	23:00:00	Porta dos Fundos	Sirene	Erasmus Krüger
Verificação Janela do Quarto	02:00:00	18:41:00	Janela Quarto 1	Sirene	

Incluir Alterar Excluir OK

A Figura 30 apresenta a tela de cadastro e alterações das programações por evento. Os dados desta tela apresentam as seguintes funções:

- descrição: é dada uma descrição para a programação, para fins de visualização no painel de operação;
- sensor: indica qual sensor é monitorado para esta programação;
- considerar sinal invertido: indica que o sinal será considerado ativo quando o valor do sensor estiver desligado;

- d) horário: determina as horas inicial e final que o atuador ficará ligado;
- e) dias da semana: determina em quais dias da semana a programação será ativada;
- f) dispositivo: indica qual atuador será acionado;
- g) e-mail: indica para qual endereço de e-mail será enviada uma mensagem com as informações atuais e com o arquivo de LOG anexado;
- h) gerar *log*: indica se a mudança de status do sensor desta programação irá gerar ou não um ocorrência no arquivo de LOG;
- i) programação ativa: indica se esta programação está sendo usada atualmente.

**Figura 30: Tela de cadastro e alteração de uma programação por evento.**

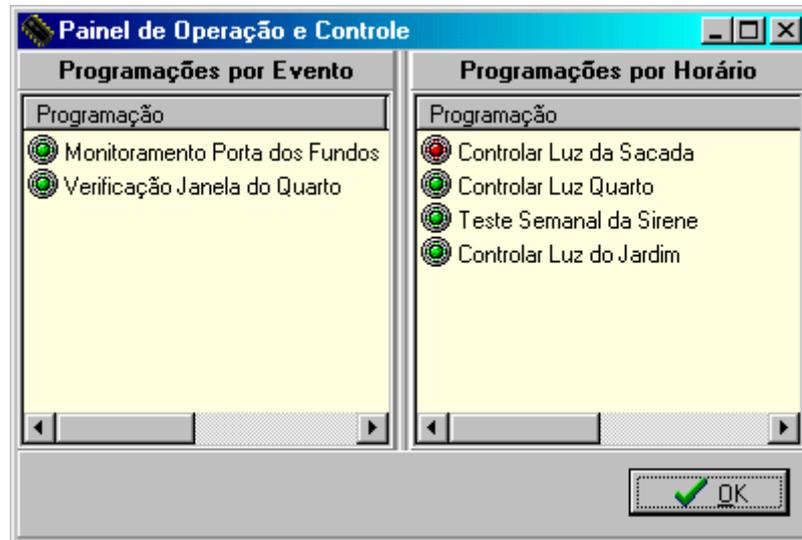
The image shows a software dialog box titled "Programação por Evento". It contains the following fields and options:

- Descrição:** A text box containing "Monitoramento Porta dos Fundos".
- Sensor:** A dropdown menu with "Porta dos Fundos" selected. Below it is a checkbox labeled "Considerar Sinal Invertido" which is unchecked.
- Horário:** Two text boxes for "Inicial" (08:00:00) and "Final" (23:00:00).
- Dias da Semana:** A grid of checkboxes for each day of the week: Domingo (checked), Segunda (unchecked), Terça (unchecked), Quarta (unchecked), Quinta (unchecked), Sexta (unchecked), and Sábado (checked).
- Parâmetros:** A dropdown for "Dispositivo" (Sirene), a dropdown for "E-Mail" (ErasmO (Celular)), and a checked checkbox for "Gerar LOG".
- Programação Ativa:** A checked checkbox at the bottom of the main area.
- Buttons:** "OK" and "Cancelar" buttons at the bottom right.

A Figura 31 apresenta a tela de *status* da rotina de monitoramento e controle do aplicativo. Enquanto esta tela estiver ativa os sensores estarão sendo lidos a cada 500ms, ou seja, duas vezes por segundo, a fim de perceber alguma alteração nos sinais. Caso alguma alteração seja detectada e houver alguma programação aguardando esta alteração a rotina irá proceder conforme configurações da programação. A rotina também estará verificando neste

intervalo se está no horário de efetuar algum acionamento conforme as programações por horário.

**Figura 31: Tela de status da rotina de monitoramento e controle.**



A rotina desta tela também é responsável por disparar os processos paralelos que enviam as mensagens por *e-mail* e gerar as ocorrências no arquivo de LOG.

Esta tela exhibe ao lado de cada programação um ícone, que pode ser na cor verde ou vermelho, que significam os estados “aguardando” ou “ativo”, respectivamente.

A Figura 32 apresenta a tela de painel de testes, que tem a finalidade de testar visualmente os sensores e atuadores. Para os sensores que estiverem ativos a cor irá mudar de azul para vermelho. Para ligar ou desligar um atuador é só clicar com o *mouse* na caixa do atuador correspondente.

Figura 32: Tela do painel de testes.



### 3.6 TESTE E VALIDAÇÃO

Para avaliação do sistema foram realizados alguns testes simulando situações reais. Os módulos do sistema foram todos conectados a um microcomputador Pentium MMX 233MHz. Foi utilizado um cabo paralelo comum para conectar a placa centralizadora de sinais ao microcomputador e fios de telefone para conectar os sensores e atuadores à placa centralizadora. Foi também conectada à placa centralizadora uma fonte de alimentação de 12 volts (12VDC 1A, Marca Hayonik, modelo FTP1201).

Foram utilizados os seguintes sensores e atuadores no teste:

- Um sensor de presença infra-vermelho passivo (Marca PPA, modelo Sensit A26016) com as seguintes características: Ajuste de sensibilidade, alcance de 13 metros, tempo de estabilização de 60 segundos e tempo de acionamento de 2 segundos;
- Um sensor magnético do tipo Reed Switch;
- Dois atuadores, baseados no relé mecânico marca Chansin, modelo JQC-3F-12VDC-1ZS (28VDC, 5A – 250VAC, 10A – 125VAC);
- Uma sirene pequena marca Digilectron, modelo Sonalarme (SA 6 a 30VDC, 0.1A)

Primeiramente todos os sensores e atuadores foram testados individualmente no painel de testes, conforme visto na Figura 32, para verificar o funcionamento conforme o previsto e preparado pelo software de controle. Todos os sensores e atuadores funcionaram sem nenhum



### Quadro 5: Exemplo de LOG gerado pelo sistema.

```
SISM-NET - Arquivo de LOG
Aplicativo desenvolvido por Erasmo Kruger para TCC - 2002/2
Orientador: Prof. Antonio Carlos Tavares
Data de Criação do arquivo: 07/11/02 23:19:48

07/11/02 23:19:48: Monitoramento Iniciado.
08/11/02 00:00:05: Programação por horário foi acionada: Alimentar Sensor de
Presença
08/11/02 06:58:33: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 07:11:47: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 07:18:41: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 07:32:51: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 07:43:28: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 12:09:25: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 12:58:58: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 13:45:51: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 14:07:45: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 14:27:48: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 14:57:08: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 15:10:14: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 15:16:26: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 15:27:54: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 15:56:12: Evento detectado no sensor: Monitoramento Quarto (Presença)
08/11/02 15:56:39: Monitoramento Finalizado.
```

Estes testes apresentam apenas algumas das inúmeras opções de configurações que podem ser realizadas pelo usuário do sistema. Eles validam a implementação do protótipo especificado e comprovam a eficácia do protótipo em:

- a) monitoramento de portas, janelas e de presença;
- b) comunicação com o microcomputador através da porta paralela;
- c) ligar e desligar dispositivos através de um interruptor eletrônico, acionado pela porta paralela;
- d) suportar flexibilidade nas configurações do sistema, permitindo várias formas de atuação do sistema no meio;
- e) fornecer respostas adequadas através de LOG e mensagens eletrônicas através da internet.

### 3.6.1 FUNÇÕES ESPECIAIS

Visto que o sistema possui uma grande flexibilidade nas suas configurações, ele permite que o usuário possa utilizá-las para algumas funções especiais. Dentre elas pode-se destacar:

- a) simulador de presença: ligando e desligando luzes e outros dispositivos no interior do prédio, através de programações por horário, de maneira diferenciada para cada

dia da semana, cria a ilusão de que existem pessoas dentro do prédio, inibindo ações de assaltantes;

- b) mensagens no telefone celular: como atualmente já está disponível o serviço de recebimento de e-mail através de telefones celulares, o usuário poderá receber informações estando em qualquer lugar dentro da área de cobertura de sua operadora de telefonia celular;
- c) marcador de ponto oculto: através das informações do sensor de presença contidas no LOG, o usuário poderá concluir, por exemplo, se o vigia fez a ronda nos horários previstos ou ainda se o escritório foi aberto no horário correto.

## 4 CONCLUSÕES

Pode-se verificar que o sistema atendeu às especificações estabelecidas inicialmente. Entretanto, observou-se alguns aspectos que restringiram o seu funcionamento, como a paralisação do sistema em todos os seus módulos no caso de queda de energia. Isso pode ser resolvido com a utilização de um *nobreak* tanto para o microcomputador quanto para a fonte de alimentação da placa centralizadora. Outra restrição observada é o número limitado de sensores e atuadores interligados ao sistema, que dependendo do prédio onde o mesmo estiver instalado pode ser insuficiente. Mesmo diante das restrições apresentadas, verificou-se que é possível a implementação deste tipo de sistema.

A realização deste trabalho foi válida principalmente pelo conhecimento adquirido através do contato com tecnologias não exploradas durante o decorrer regular do curso. Devido a isso, certas dificuldades foram encontradas, podendo-se citar a comunicação em baixo nível com a porta paralela utilizando a linguagem *Assembly*, e a construção dos módulos eletrônicos.

### 4.1 EXTENSÕES

Como sugestões para futuras melhorias e extensões do sistema aqui desenvolvido pode-se citar:

- a) a utilização de câmeras digitais do tipo *webcam* para capturar imagens do ambiente monitorado, que seriam enviadas em anexo nos *e-mails*;
- b) permitir a alteração das configurações e enviar comandos remotamente, através da internet;
- c) utilizar sensores e atuadores *wireless* (sem fio);
- d) utilização de controle remoto para alguns comandos básicos no sistema, como ativação e desativação do sistema de monitoramento;
- e) a preparação do software para poder ser executado nos sistemas operacionais Microsoft Windows com tecnologia NT, ou seja, Windows NT, Windows 2000 e Windows XP;
- f) melhoria da interface com o usuário, usando ícones maiores e com significados mais específicos para cada tipo de atuador ou sensor;
- g) Construir um módulo centralizador com maior número de entradas e saídas.

# ANEXO I

Quadro 6: Rotina principal do módulo de monitoramento e controle.

```
procedure TFPanCE.TimerAtualizaTimer(Sender: TObject);
Var
  K : Byte;
  Hora : TDateTime;
  Sts : Boolean;

begin
  TimerAtualiza.Enabled := False;
  AtualizaDados;

  Hora := Time;

  { Verifica acionar de acordo com a hora / dia da semana }
  For k := 1 to NroProHor do
  begin
    if (((Hora >= ProHor[k].HorIni) and (Hora <= ProHor[k].HorFim) and
      (ProHor[k].HorIni < ProHor[k].HorFim)) Or
      (not((Hora >= ProHor[k].HorFim) and (Hora <= ProHor[k].HorIni)) and
      (ProHor[k].HorIni > ProHor[k].HorFim)))
      and (Pos(IntToStr(DayOfWeek(Now)), String(ProHor[k].DiaSem)) <> 0) then
    begin
      TrataPinoBase(gPorta, 1, ProHor[k].CodDis);

      if (ProHor[k].Status = 0) then
        GravaLog('Programação por horário foi acionada: '+
          String(ProHor[k].DesPro));

      if (ProHor[k].CodMai <> 0) and (ProHor[k].Status = 0) then
        MandaEmailPrg(k, 1);

      ProHor[k].Status := 1;
    end
    else
    begin
      if (ProHor[k].Status = 1) then
      begin
        GravaLog('Programação por horário foi finalizada: '+
          String(ProHor[k].DesPro));
        ProHor[k].Status := 0;
        TrataPinoBase(gPorta, 2, ProHor[k].CodDis);
      end;
    end;
  end;

  { Ativa sensores apenas depois do tempodalay }
  if TempoDelay > 0 then
  begin
    dec(TempoDelay);
  end
  else
  { Verifica acionar de acordo com a hora / dia da semana / Sensor }
  For k := 1 to NroProEve do
  begin
    if (((Hora >= ProEve[k].HorIni) and (Hora <= ProEve[k].HorFim) and
      (ProEve[k].HorIni < ProEve[k].HorFim)) Or
      (not((Hora >= ProEve[k].HorFim) and (Hora <= ProEve[k].HorIni)) and
      (ProEve[k].HorIni > ProEve[k].HorFim)))
```

```

    and (Pos(IntToStr(DayOfWeek(Now)), String(ProEve[k].DiaSem)) <> 0) then
begin
    Sts := LePinoStatus(gPorta, ProEve[k].CodSin);
    if (ProEve[k].SinInv = 1) then
        Sts := not Sts;

    if (sts) then
begin
    if ProEve[k].CodDis <> 0 then
begin
    TrataPinoBase(gPorta, 1, ProEve[k].CodDis);

    If (ProEve[k].GerLog = 1) and (ProEve[k].Status = 0) then
        GravaLog('Dispositivo acionado por Evento: '+String(ProEve[k].DesPro));

    ProEve[k].TimSir := Hora + 0.0007; // Aprox. 1 minuto
end else
begin
    If (ProEve[k].GerLog = 1) and (ProEve[k].Status = 0) then
        GravaLog('Evento detectado no sensor: '+String(ProEve[k].DesPro));

    ProEve[k].TimSir := Hora + 0.0021; // Aprox. 3 minutos
end;

    if (ProEve[k].CodMai <> 0) and (ProEve[k].Status = 0) then
        MandaEmailPrg(k, 2);

    ProEve[k].Status := 1;
end;
end
else
begin
    if (ProEve[k].Status = 1) then
begin
    if (ProEve[k].CodDis <> 0) then
begin
    If ProEve[k].GerLog = 1 then
        GravaLog('Dispositivo foi desligado. Evento: '+
            String(ProEve[k].DesPro));
        TrataPinoBase(gPorta, 2, ProEve[k].CodDis);
    end;
    ProEve[k].Status := 0;
end;
end;
end;
end;

For k := 1 to NroProEve do
begin
    if (ProEve[k].DisSir in [0,2]) and (ProEve[k].Status = 1) and
        (Hora >= ProEve[k].TimSir) then
begin
    if (ProEve[k].CodDis <> 0) then
begin
    TrataPinoBase(gPorta, 2, ProEve[k].CodDis);
    If ProEve[k].GerLog = 1 then
        GravaLog('Dispositivo do tipo sirene foi desligado: '+
            String(ProEve[k].DesPro));

    end;
    ProEve[k].Status := 0;
end;
end;
end;

    TimerAtualiza.Enabled := True;
end;

```

## REFERÊNCIAS BIBLIOGRÁFICAS

ANGEL, Patrícia Marta; FRAIGI, Liliana Beatriz, **Introduccion a la Domotica**, Tomo I, Embasse: EBAI, 1993. 172 p.

ARAÚJO, Kleitor Franklin Correa de. **Delphi 5 para internet com banco de dados: o guia**. São Paulo: Érica, 2000. 324 p.

BERGSMAN, Paul. **Controlling the world with your PC**. Solana Beach: HightText, 1994. 257 p.

CADSOFT ONLINE. **Home of the EAGLE layout editor**, [GmbH], set. [2002]. Disponível em: <http://www.cadsoftusa.com>. Acesso em: 5 out. 2002.

CALABUIG, José González. **Circuitos impressos: teoria, diseño y montaje**. Madrid: Paraninfo, 1997. 183 p.

CANTU, Marco. **Dominando o Delphi 5: a bíblia**. Sao Paulo: Makron Books, 2000. 860 p.

DOMINGUES, Alvaro Alipio Lopes. **A Linguagem Assembly**. São Paulo: Icone Ed, 1986. 203 p.

HORN, Delton . **Eletronic alarm and security systems – a technician’s guide**. Blue Ridge Summit: TAB Books, 1995. 256 p.

HOWSTUFFWORKS. **How parallel port works**, [s.l.], jun. [1998]. Disponível em: <http://www.howstuffworks.com/parallel-port.htm>. Acesso em: 10 ago. 2002.

JOHNSON, Curtis D., **Process control instrumentation technology**, New Jersey: Pentice Hall, 1988. 573 p.

LIMA, Norberto de Paula. Automação Predial. **Eletricidade moderna**, São Paulo, n.254, maio de 1995.

MAGALHÃES, Mauricio Ferreira, **Sistemas em tempo real**, SBA, 8 Congresso Brasileiro de Automática, 10 a 14 de setembro de 1990, Belém.

Os Relés Eletrônicos. **Elektor Eletrônica**, São Paulo, n.10, maio 1987.

POMPILHO, S., **Análise essencial**: guia prático de análise de sistemas. Rio de Janeiro: Ciência Moderna, 2002. 269 p.

RIPPS, David L., **Guia de implementação para programação em tempo real** / David L. Ripps; tradução de Caetano Loiola, Rio de Janeiro: Campus, 1993. 314 p.

WARD, Paul T.; MELLOR, Stephen J., **Structured development for real-time systems**, Englewoods-Cliffs: Prentice Hall, 1985, v1.