

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

**SISTEMA PARA AUTOMAÇÃO E CONTROLE RESIDENCIAL
VIA E-MAIL**

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

ANGELA CENSI

BLUMENAU, JUNHO/2001

2001/1-08

SISTEMA PARA AUTOMAÇÃO E CONTROLE RESIDENCIAL VIA E-MAIL

ANGELA CENSI

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. Miguel Alexandre Wisintainer — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Miguel Alexandre Wisintainer

Prof. Antônio Carlos Tavares

Prof. Sérgio Stringari

DEDICATÓRIA

Aos meus pais Vendelino e Marlene, minha irmã Karina e meu noivo Ricardo pelo carinho, compreensão e incentivo sem os quais este trabalho não seria realidade.

SUMÁRIO

| | |
|--|-----|
| LISTA DE FIGURAS | VII |
| LISTA DE QUADROS | IX |
| LISTA DE TABELAS | 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 |
| 2.1.1.1 FUNÇÃO DE GESTÃO | 4 |
| 2.1.1.2 FUNÇÃO DE CONTROLE-COMANDO | 5 |
| 2.1.1.3 FUNÇÃO DE COMUNICAÇÃO | 7 |
| 2.2 APLICAÇÕES DAS NOVAS TECNOLOGIAS | 7 |
| 3 SENSORES E ATUADORES | 10 |
| 3.1 SENSORES | 11 |
| 3.1.1 SENSORES DE TEMPERATURA | 12 |
| 3.1.2 SENSORES DE LUMINOSIDADE | 13 |
| 3.1.3 SENSORES DE ULTRA-SOM | 14 |
| 3.1.4 SENSORES DE UMIDADE E FOGO | 15 |
| 3.2 ATUADORES | 16 |
| 3.2.1 RELÉS | 16 |
| 3.2.2 VÁLVULAS | 17 |

| | | |
|-----------|---|----|
| 4 | TCP/IP - TRANSMISSION CONTROL PROTOCOL / INTERNET PROTOCOL..... | 19 |
| 4.1 | VISÃO GERAL DA ARQUITETURA TCP/IP | 20 |
| 4.1.1 | MODELO DE COMUNICAÇÃO DE DADOS | 21 |
| 4.1.2 | MODELO DE REFERÊNCIA OSI | 21 |
| 4.1.3 | COMPARAÇÃO ENTRE O MODELO OSI E O TCP/IP | 22 |
| 4.2 | FUNÇÕES DAS CAMADAS DA ARQUITETURA TCP/IP..... | 23 |
| 4.2.1 | CAMADA FÍSICA | 24 |
| 4.2.2 | CAMADA DE INTERNET OU REDE..... | 24 |
| 4.2.2.1 | Protocolo de transporte de dados: IP - Internet Protocol | 25 |
| 4.2.3 | CAMADA TRANSPORTE | 26 |
| 4.2.3.1 | Socket | 28 |
| 4.2.4 | CAMADA APLICAÇÃO | 29 |
| 4.2.4.1 | SMTP - Simple Mail Transfer Protocol..... | 29 |
| 4.2.4.1.1 | Formato das Mensagens..... | 31 |
| 4.2.4.2 | POP – POST OFFICE PROTOCOL | 36 |
| 4.2.4.2.1 | Formato das Mensagens..... | 39 |
| 5 | KIT DE DESENVOLVIMENTO RABBIT 2000 TCP/IP | 40 |
| 6 | DESENVOLVIMENTO DO SISTEMA | 45 |
| 6.1 | ESPECIFICAÇÃO | 45 |
| 6.2 | IMPLEMENTAÇÃO | 49 |
| 6.2.1 | HARDWARE..... | 49 |
| 6.2.2 | SOFTWARE | 50 |
| 6.2.2.1 | Módulo de recepção dos e-mails | 50 |
| 6.2.2.2 | Módulo de envio de e-mails | 52 |
| 6.2.2.3 | Módulo de execução das tarefas agendadas | 52 |

| | |
|--|----|
| 6.3 OPERACIONALIDADE DA IMPLEMENTAÇÃO..... | 54 |
| 7 CONCLUSÕES | 56 |
| 7.1 EXTENSÕES | 56 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 57 |
| ANEXOS | 59 |

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 2.1 – GERENCIAMENTO DAS ÁREAS DE CONTROLE RESIDENCIAL..... | 8 |
| FIGURA 2.2 - – ARQUITETURA DE UM SISTEMA DE AUTOMAÇÃO RESIDENCIAL | 8 |
| FIGURA 3.1 – ESTRUTURA GERAL DE UM SISTEMA DE MEDIÇÃO | 10 |
| FIGURA 3.2 – SENSOR DE TEMPERATURA | 13 |
| FIGURA 3.3 – CONTROLADOR DE LUMINOSIDADE | 14 |
| FIGURA 3.4 – SENSOR DE PRESENÇA COM ULTRA-SOM..... | 15 |
| FIGURA 3.5 – SENSOR DE UMIDADE..... | 15 |
| FIGURA 3.6 – SENSOR DE FUMAÇA | 16 |
| FIGURA 3.7 – ESQUEMA DO RELÉ ELETROMECAÂNICO | 17 |
| FIGURA 3.8 – VÁLVULA | 18 |
| FIGURA 4.1 – DIVISÃO EM CAMADAS DO MODELO OSI | 22 |
| FIGURA 4.2 – COMPARAÇÃO ENTRE O MODELO OSI E O TCP/IP | 22 |
| FIGURA 4.3 – POSICIONAMENTO GERAL DE DIVERSOS PROTOCOLOS NAS ARQUITETURAS OSI, TCP/IP E NOVELL NETWARE | 23 |
| FIGURA 4.4 – DIVISÃO EM CAMADAS DA ARQUITETURA TCP/IP..... | 23 |
| FIGURA 4.5 – VISÃO DETALHADA DE CADA CAMADA DA ARQUITETURA TCP/IP | 24 |
| FIGURA 4.6 – CLASSIFICAÇÃO DOS ENDEREÇOS IP..... | 26 |
| FIGURA 4.7 – PROCESSO DE TROCA DE MENSAGENS | 31 |
| FIGURA 4.8 – FORMATO DA MENSAGEM | 32 |
| FIGURA 4.9 – EXEMPLO DE FUNCIONAMENTO DO PROTOCOLO SMTP..... | 35 |
| FIGURA 4.10 – EXEMPLO DE UMA SESSÃO POP..... | 39 |
| FIGURA 5.1 – KIT DE DESENVOLVIMENTO RABBIT 2000 TCP/IP | 40 |

| | |
|--|----|
| FIGURA 5.2 – PLACA PROGRAMÁVEL DO KIT RABBIT 2000 TCP/IP..... | 43 |
| FIGURA 5.3 - PORTAS SERIAIS E ENTRADAS E SAÍDAS DIGITAIS | 44 |
| FIGURA 6.1 – FORMATO DOS COMANDOS..... | 45 |
| FIGURA 6.2 – HARDWARE DO SISTEMA | 49 |
| FIGURA 6.3 – <i>E-MAIL</i> COM TAREFA A SER AGENDADA..... | 54 |
| FIGURA 6.4 – SISTEMA BUSCANDO MENSAGENS DE <i>E-MAIL</i> | 54 |
| FIGURA 6.5 – PRÓXIMA TAREFA A SER EXECUTADA PELO SISTEMA DE CONTROLE..... | 55 |
| FIGURA 6.6 – SISTEMA INDICANDO QUE NÃO HÁ MAIS TAREFAS NA LISTA..... | 55 |

LISTA DE QUADROS

| | |
|--|----|
| QUADRO 4.1 – SINTAXE PARA TROCA DE MENSAGENS..... | 33 |
| QUADRO 4.2 – CÓDIGO NUMÉRICO DAS RESPOSTAS RETORNADAS PELO PROTOCOLO SMTP..... | 36 |
| QUADRO 4.3 – SINTAXE DOS ARGUMENTOS NO PROTOCOLO POP..... | 38 |
| QUADRO 6.1 – CONSTANTES DO SISTEMA..... | 50 |
| QUADRO 6.2 – IMPLEMENTAÇÃO DA RECEPÇÃO DE <i>E-MAILS</i> | 50 |
| QUADRO 6.3 – SEPARAÇÃO DOS ATRIBUTOS DA MENSAGEM..... | 51 |
| QUADRO 6.4 – IMPLEMENTAÇÃO DO ENVIO DE <i>E-MAILS</i> | 52 |
| QUADRO 6.5 – IMPLEMENTAÇÃO DA FUNÇÃO DE EXECUÇÃO DE TAREFAS..... | 53 |

LISTA DE TABELAS

| | |
|--|----|
| TABELA 3.1 – SENSORES E VARIÁVEIS A MEDIR MAIS COMUNS..... | 11 |
| TABELA 3.2 - SENSORES TÍPICOS PARA O SETOR RESIDENCIAL E SUAS APLICAÇÕES..... | 12 |
| TABELA 4.1 - COMPARAÇÃO TCP X UDP..... | 28 |
| TABELA 4.2 – SEMÂNTICA DOS COMANDOS PARA TROCA DE MENSAGENS..... | 32 |
| TABELA 4.3 – DESCRIÇÃO DOS COMANDOS PARA TROCA DE MENSAGENS UTILIZANDO O PROTOCOLO SMTP..... | 33 |
| TABELA 4.4 – COMANDOS UTILIZADOS NO PROTOCOLO POP..... | 37 |
| TABELA 4.5 – SEMÂNTICA DOS COMANDOS POP..... | 38 |
| TABELA 5.1 – ESPECIFICAÇÕES DA PLACA PROGRAMÁVEL..... | 41 |

RESUMO

O presente trabalho apresenta considerações sobre a área de automação residencial ou domótica, conceitos sobre a arquitetura TCP/IP, principalmente os protocolos SMTP e POP que foram utilizados para construção do sistema de controle residencial via *e-mail*. Apresenta também detalhes da especificação e desenvolvimento (*hardware / software*) do sistema.

ABSTRACT

This work presents concepts forward especificly to residential automation area or domotic. Presents to TCP/IP architetur concepts, mainly the SMTP and POP protocols that were used for the residential system's control building by e-mail. Contains specification and developing detail (hardware and software).

1 INTRODUÇÃO

Hoje, graças à evolução tecnológica, principalmente observada na área de computação, utiliza-se quase que constantemente em casas, às vezes sem perceber, objetos que possuem algum tipo de automação, como máquinas de lavar, forno de microondas, *vídeo-cassete*, televisores e outros (Besen, 1996).

Até o final dos anos 60, a exigência de qualidade mínima dos espaços era o maior problema habitacional. A partir dos anos 70, o conceito de habitação foi se transformando e as condições de qualidade compreendidas como uma necessidade social. É nesta época que grandes esforços se dirigiram aos meios e as condições da construção, visando a redução econômica (custo global) e técnica (processos construtivos) (Angel, 1993).

No início dos anos 80, principalmente na Europa, surge a exigência de maior economia de consumo energético. E assim se iniciam a produção de equipamentos e sistemas com maior eficiência energética, que além de reduzidos consumos de energia, também ofereciam ao usuário maiores condições de conforto (Angel, 1993).

Na década de 90 a tendência foi de integrar os diferentes sistemas (segurança, condicionamento ambiental, etc.) que compõem uma residência, com o objetivo de melhorar a produtividade e o conforto das pessoas, onde todos os elementos começam a se inter-relacionar tanto internamente como com o exterior (Gomez, 1998).

A partir desta nova situação, criaram-se novas necessidades tecnológicas, tendo início então um novo campo de aplicação tecnológica, a domótica. Nesta situação, a integração do espaço arquitetônico, da informática e das telecomunicações torna-se a nova necessidade objetivando a maior qualidade de vida, a redução do trabalho doméstico, o aumento do bem estar e da segurança, e a racionalização dos consumos (Angel, 1993).

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um sistema de automação residencial que execute suas tarefas através de comandos recebidos por *e-mail*, bem como comandos digitados localmente.

Os objetivos específicos do trabalho são:

- a) Desenvolver o software responsável pelo gerenciamento do sistema;
- b) Desenvolver o hardware de interface entre o microcontrolador e o sistema a ser controlado, como iluminação, ar condicionado, alarme e eletrodomésticos.

1.2 ESTRUTURA DO TRABALHO

O primeiro capítulo é destinado a introdução, objetivos e estrutura do trabalho, ou seja, é o presente capítulo.

O segundo capítulo é dedicado ao estudo da domótica. Nele são apresentadas as funções domóticas através dos serviços de gestão, controle e comunicação. Também é apresentado um estudo sobre as tendências para o futuro na área de domótica.

O terceiro capítulo foi reservado ao estudo dos sensores e atuadores que são elementos fundamentais para um sistema domótico. São apresentadas as principais características e aplicações destes equipamentos.

O quarto capítulo é dedicado ao estudo da arquitetura TCP/IP. São apresentados conceitos, características, principais protocolos e um estudo mais detalhado sobre os protocolos SMTP e POP3, que formam a base para o desenvolvimento deste trabalho.

O quinto capítulo faz uma breve descrição sobre o kit de desenvolvimento Rabbit 2000 TCP/IP, composto de placa programável e acessórios, utilizado para a construção do sistema de controle residencial.

O sexto capítulo enfoca a especificação e implementação do sistema, são fornecidos detalhes das rotinas como por exemplo envio e recebimento de *e-mails* e execução das tarefas.

No capítulo sete são relatadas as conclusões obtidas do trabalho, dificuldades encontradas e propostas para futuras implementações.

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 (Angel, 1993). 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 1977 nos EUA quando da construção do World Trade Center em Nova York. A utilização da palavra *smart* lembrava "estar na moda, ser elegante, inteligente etc", porém, tinha outro significado no sentido de "doloroso, que faz mal", o que gerava confusões e os faziam lembrar da *smart bomb* utilizada na segunda guerra mundial. Por esta razão foi substituído o termo *smart building* pelo *intelligent building* consensuado e utilizado em todos os EUA e também no Japão. Segundo Angel (1993), 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 estações do ano através de dispositivos reguladores que visam a redução do consumo de energia;
- 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**, 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**, estar protegido de intrusos garantindo os bens e as pessoas (conforto moral);
- f) **conforto de atividade**, assegurar múltiplas tarefas domésticas: limpeza e conservação dos locais, armazenamento dos alimentos etc;
- g) **conforto higiênico-saudável**, assegurar atividades do corpo: comer, dormir, lavar-se, manter a forma;
- h) **distrair-se**;

- i) **comunicar-se**, tanto com o interior como com o exterior da casa;
- j) **mantimento e otimização dos gastos**, gerenciar o conjunto: prover-se dos materiais e fluidos necessários;
- k) eventualmente **trabalhar** e estudar, todo ou parte do tempo em casa.

2.1.1 FUNÇÕES DOMÓTICAS

Ainda de acordo com (Angel, 1993), as funções domóticas permitem satisfazer um número considerável das necessidades acima relacionadas. Segundo o tipo de "serviço", estas funções se dirigem a 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.

2.1.1.1 FUNÇÃO DE GESTÃO

Esta função tem por 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 primeiros elementos voltados ao conforto adequando o ambiente segundo a necessidade de cada um segundo a idade, capacidades físico-motoras, uso dos espaços ao longo do dia, repercussão sobre a ocupação dos espaços etc. 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 desta sub-função pode-se citar a temporização, a variação da intensidade o acendimento/apagamento automático; programado ou comandado a distância. Como um objetivo futuro desta 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 encontram-se em estado bem avançado. Como necessidade de conforto voltada 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 á distância, passagem automática do regime "conforto" para o regime "redução" no caso de ausência das pessoas; desativamento do ar condicionado ou ventilação em caso de janelas abertas, dentre outras;

c) **gestão da qualidade do ar**, através desta sub-função a domótica pode prover controle total do ar no ambiente, não somente da temperatura acima citada e da umidade, 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 as 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 pisos e tetos flexíveis, etc.

As exigências acima citadas vêm do fato de que o edifício é um conjunto flexível que integra distintas tecnologias, abrangendo sistemas de gestão, sistemas de controle energético, sistemas de vigilância e manutenção, sistemas de segurança, sistemas de comunicação interna e externa, como também sistemas mecânicos, elétricos e eletrônicos, lembrando que os mesmos devem conectar-se entre si.

Os requisitos voltados á 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.

2.1.1.2 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 nesta unidade 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 equipamentos e eletrodomésticos, das diferentes redes de alimentação, 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 indivíduos da casa. Como exemplos de controle técnico pode-se citar: visualização das informações em um monitor de TV ou outro indicador de controle; desligamento seletivo de cargas para evitar sobrecargas no sistema; informações de consumo de água, de gás, de eletricidade, custos dos mesmos e horários de sua aplicação; comandos únicos que atuam sobre diversos equipamentos, como um comando de saída de férias que desligaria as lâmpadas, ativaria o sistema de alarme, cortaria o suprimento de água e luz.

b) **segurança e tele-transmissão**, a esta sub-função está associada a segurança dos bens materiais e das pessoas, o que inclui a prevenção de intrusos, incêndios e acidentes domésticos tele-transmitindo alarmes em caso da ocorrência de um deles. Como fatores importantes desta sub-função, cita-se a confiabilidade do sistema, evitando, por exemplo, falsos alarmes, e a 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 de meio; a detecção de eventuais acidentes como incêndios, fuga de gás e água; a detecção de intrusos, e a 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.

2.1.1.3 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 obedecem a uma certa padronização.

A função de comunicação pode ser classificada segundo dois ângulos: a comunicação sem significado, 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 seguir são analisadas as três sub-funções da função de comunicação:

- a) **comunicação - controle**, com a utilização da Rede Digital de Serviços Integrados (RDSI), que apresenta 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 possibilitará também a troca de comandos entre os vários equipamentos e o operador.
- b) **comunicação - 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 a realização destes serviços coletivos tem-se a necessidade de inter-conectar 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 desse serviço estão incluídos. o tele-trabalho, a tele-educação, o tele-mantimento, a tele-vigilância-assistência, os tele-serviços, etc.

2.2 APLICAÇÕES DAS NOVAS TECNOLOGIAS

De acordo com Neto (2001), com as mais novas tecnologias disponíveis no mercado, um bom projeto e a correta previsão da infra-estrutura para as instalações, será possível contar com uma rede única de comunicação, que propiciará à residência todas as facilidades, conforme figura 2.1:

FIGURA 2.1 – GERENCIAMENTO DAS ÁREAS DE CONTROLE RESIDENCIAL

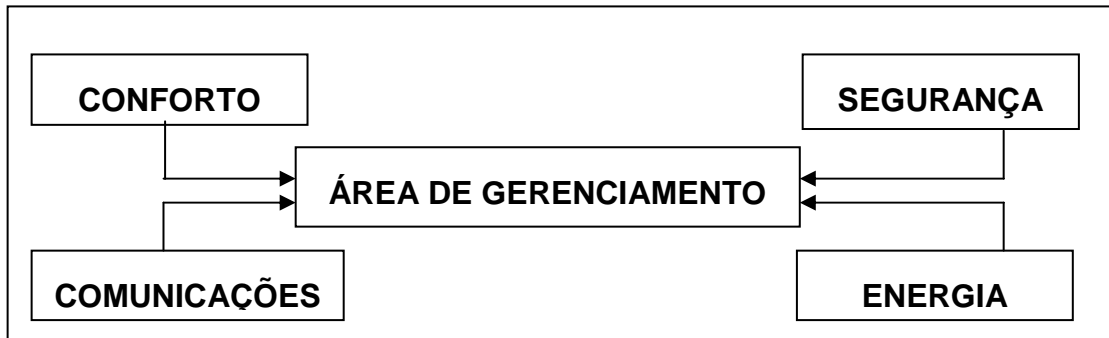
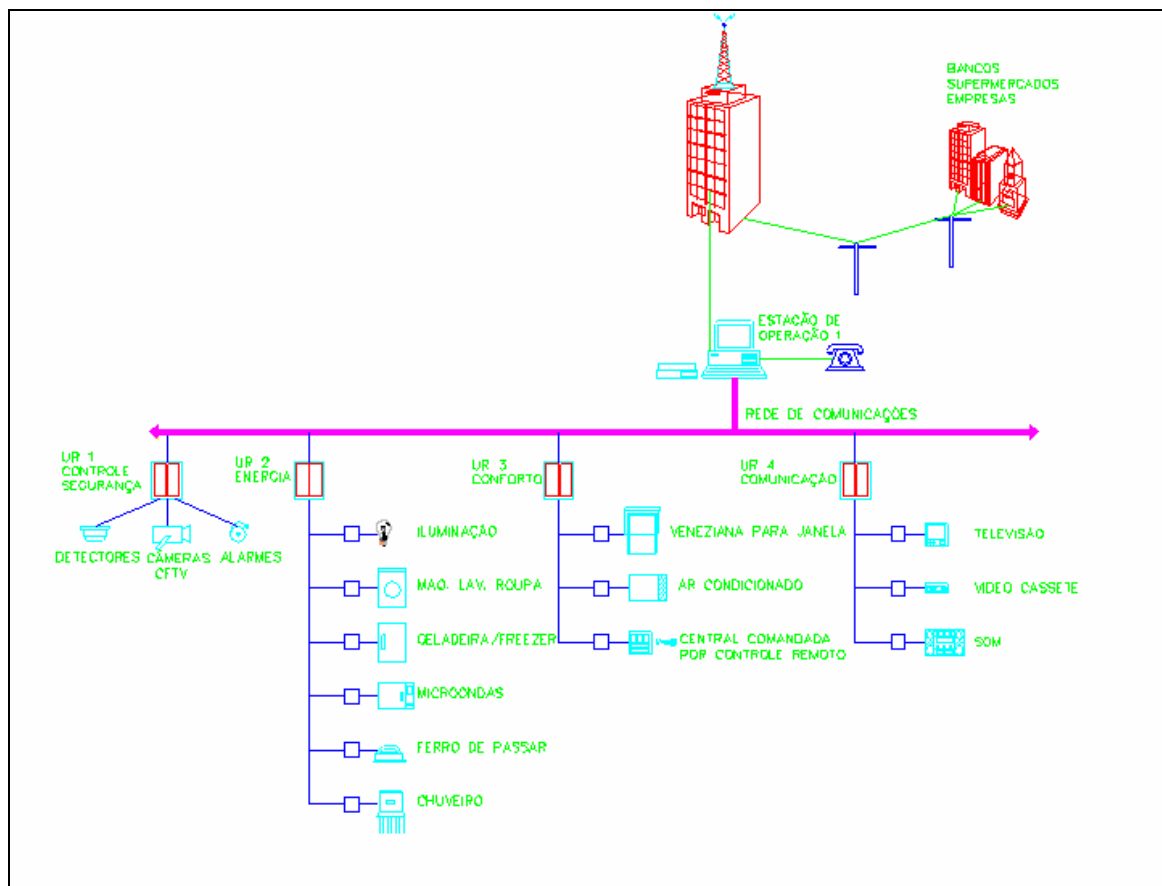


FIGURA 2.2 -- ARQUITETURA DE UM SISTEMA DE AUTOMAÇÃO RESIDENCIAL



Através desta integração e, contando com os recursos da Rede Mundial de Computadores (*World Wide Web*) a *Internet*, as possibilidades de comunicação, através da rede telefônica ou da recepção de sinais pelos cabos de TV e antenas, os condôminos das residências que contam com as novas tecnologias podem controlar todas as utilidades domésticas, tanto interna como remotamente.

Estas instalações estarão preparadas, inclusive, para funções previstas para um futuro próximo como a interconexão de eletrodomésticos à *Internet*, uma vez que estas possibilidades encontram-se em fases de teste pelas indústrias, o que possibilitará consultar uma receita na rede através do próprio micro ondas ou receber um telefonema da assistência técnica do refrigerador, informando que foram detectados indícios de mau funcionamento do mesmo, pela verificação de rotina feita, também, através da *Internet*.

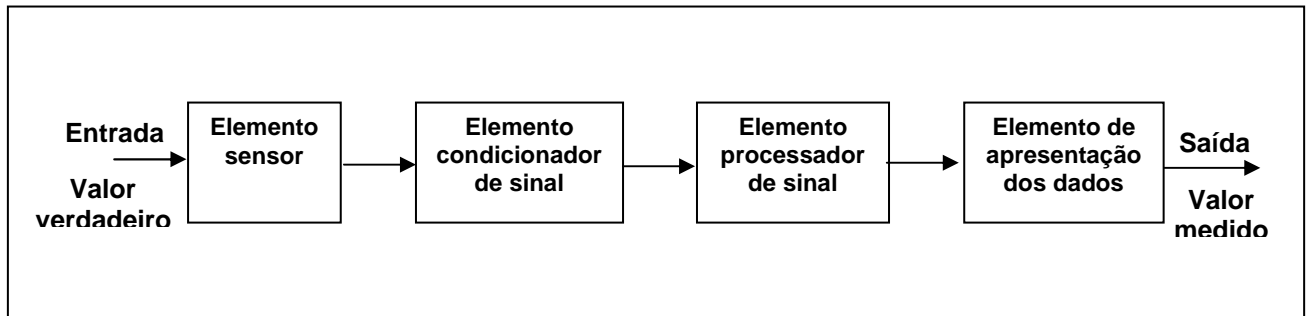
Desta maneira, é possível a este morador receber, em seu escritório ou em qualquer computador ligado à *Internet*, alarmes de intrusão ou de mau funcionamento de algum equipamento da residência, programar à distância tarefas de ligar ou desligar eletrodomésticos, receber imagens de seus filhos brincando em seus quartos, entre outras facilidades. Da mesma forma, as pessoas idosas podem, tranqüilamente, viver sozinhas em seus apartamentos, uma vez que, em caso de emergência, poderão rapidamente comunicar-se com parentes, clínicas ou com os serviços do próprio condomínio.

Tudo isto sem contar com a comodidade de efetuar o comando de grande parte das instalações da unidade, através de controle remoto.

3 SENSORES E ATUADORES

Segundo Angel (1993), um sistema de medição, em geral, consiste de quatro tipos de elementos: sensor, condicionador de sinal, processador de sinal e de apresentação dos dados. A figura 3.1 apresenta a estrutura geral de um sistema de medição.

FIGURA 3.1 – ESTRUTURA GERAL DE UM SISTEMA DE MEDIÇÃO



O elemento sensor é o que está direto em contato com a magnitude física-química a medir.

O elemento condicionador de sinal pega a saída do sensor e converte em uma forma mais adequada para processá-la posteriormente, geralmente em sinais de tensão contínua, corrente contínua ou frequência.

O elemento processador de sinal pega a saída condicionada e converte a uma forma conveniente para apresentação. Por exemplo: um conversor analógico/digital converte a tensão a uma forma digital adequada para ser processada por um computador.

O elemento de apresentação dos dados apresenta o valor medido em uma forma que pode ser facilmente reconhecida pelo observador. Por exemplo: um *display* alfanumérico ou um gráfico.

Outro elemento importante em um sistema de medição é o transdutor, este elemento é geralmente utilizado em conexões com medição e instrumentação. É um dispositivo que traduz em saída elétrica variáveis de entrada físicas ou químicas. Um transdutor incorpora um sensor e um condicionador de sinais. O condicionador de sinais converte o sinal obtido no sensor em uma saída normalizada (0-10V, 4-20mA). (Angel, 1993)

3.1 SENSORES

Os sensores podem ser classificados em passivos e ativos. Na primeira classe estão incluídos os sensores resistivos, capacitivos, indutivos e tem por característica básica a necessidade de tensão de alimentação externa. Já os segundos geram tensão, não necessitando de alimentação externa, nesta classe estão incluídos os diodos fotovoltaicos, termopares, cerâmicos piezoelétricos (Angel, 1993).

A tabela 3.1 apresenta os diferentes tipos de sensores, segundo seu princípio físico de funcionamento, associada á medição de magnitudes físico-químicas.

TABELA 3.1 – SENSORES E VARIÁVEIS A MEDIR MAIS COMUNS

| | Temperatura | Luminosidade | Pressão | Força / Peso | Nível | Deformação | Velocidade | Aceleração | Composição de gás | Concentração iônica | Umidade |
|------------------------|-------------|--------------|---------|--------------|-------|------------|------------|------------|-------------------|---------------------|---------|
| Resistivo | ■ | ■ | | | | ■ | | | ■ | | ■ |
| Capacitivo | | | ■ | | ■ | ■ | | | | | ■ |
| Indutivo | | | | | | ■ | ■ | ■ | | | |
| Piezo-resistivo | | | ■ | ■ | | ■ | | ■ | | | |
| Fotovoltaico | ■ | ■ | | | | | | | | | |
| Eletromagnético | | | | | | | ■ | ■ | | | |
| Termoelétrico | ■ | ■ | | | | | | | | | |
| Piezoelétrico | ■ | | | ■ | | | ■ | ■ | | | |
| Eletroquímico | | | | | | | | | ■ | ■ | |
| Piroelétrico | ■ | ■ | | | | | | | | | ■ |

A tabela 3.2 demonstra alguns sensores típicos para o setor residencial e suas aplicações para várias funções nas áreas de energia, segurança e conforto.

TABELA 3.2 - SENSORES TÍPICOS PARA O SETOR RESIDENCIAL E SUAS APLICAÇÕES

| Funções Sensores | Energia | | | | Segurança | | | | Conforto | | | |
|---------------------|-------------|--------|----------------|---------|-----------|--------|------------|------------|------------|-----------|------------------|----------------|
| | Água quente | Cargas | Perfil energia | Leitura | Intrusos | Acesso | Emergência | Vigilância | Iluminação | Persianas | Eletrodomésticos | Entretenimento |
| Movimento | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Temperatura | ■ | | ■ | | | | ■ | | ■ | ■ | ■ | |
| Portas / janelas | | | ■ | | | | ■ | ■ | | | | |
| Fogo | | | | | | | ■ | | | | | |
| Água | | | | | | | ■ | | | | | |
| Gás | | | | | | | ■ | | | | | |
| Vento | | | | | | | | | | ■ | | |
| Corrente elétrica | | ■ | ■ | ■ | | | | | | | ■ | |

Os itens a seguir serão tratados com maiores detalhes, visto que são a base fundamental para a medição de um conjunto de variáveis físicas essenciais à domótica.

3.1.1 SENSORES DE TEMPERATURA

Como visto na tabela anterior, um dos elementos empregados na medição de temperatura são os termoresistivos, chamados normalmente de termoresistências. A característica de funcionamento destes sensores é baseada na propriedade que certos metais possuem de variar sua resistência de acordo com a temperatura. A faixa de temperatura pode variar entre -100 a 800°C . (Angel, 1993)

O modelo normalmente empregado nas indústrias é, segundo Angel (1993), as termoresistências de platina, apesar de seu alto custo. Estes modelos apresentam uma resposta de resistência/temperatura linear e repetida, podendo ser utilizada em uma faixa de -200 a $+800^{\circ}\text{C}$. Um sensor de platina típico tem uma resistência de 100 ohms a 0°C com uma variação de $3,91$ mohms/ $^{\circ}\text{C}$.

Os materiais semicondutores também apresentam resistências que variam de acordo com a temperatura. Os termistores são componentes baseados nesta propriedade. Geralmente se utilizam de óxidos de metais como: cromo, magnésio, cobalto e níquel. A resistência destes materiais decresce de uma forma não linear em relação à temperatura. Um termistor típico tem uma resistência de $12k\Omega$ a 25°C , diminuindo a $0,95k\Omega$ a 100°C . O efeito de auto-aquecimento é de 1°C a cada 7mW de potência elétrica. (Angel, 1993)

FIGURA 3.2 – SENSOR DE TEMPERATURA



3.1.2 SENSORES DE LUMINOSIDADE

Os sensores fotoelétricos consistem basicamente de uma fonte emissora de luz e um receptor sensível à luz.

Os detectores/receptores de luz se baseiam em dois efeitos: o de fotocondução e a fotogeração. (Angel, 1993) Os detectores fotocondutivos podem ser divididos em dois grupos; células a fotodiodos ou fototransistores.

As células fotoresistivas ou fotoresistores, atualmente as células mais utilizadas são as de sulfato de cádmio, apresentam grande variação de resistência quando expostas à luz. São utilizadas em muitas aplicações como brilho e contraste em receptores de televisão, detecção de fogo, de presença e posição.

Os fotodiodos e fototransistores, são normalmente construídos a base de silício. Quando a luz de determinado comprimento de onda incide sobre os mesmos, produzem um fluxo de corrente proporcional à variação de radiação sobre o mesmo.

Os detectores de fotogeração ou fotovoltaicos, não necessitam de alimentação externa, sendo sua tensão de saída gerada por si próprio e em função de iluminação sobre o mesmo.

Materiais como silício, selênio e germânio são usados com frequência neste tipo de células. Sua construção é baseada na união de dois materiais diferentes, como silício-germânio e, o potencial produzido por cada célula é pequeno, tipicamente na ordem de 0,45V por unidade, mas várias células podem ser ligadas em série.

Vale mencionar também três recursos que tem melhorado notadamente a performance dos sensores fotoelétricos: os diodos emissores de infra-vermelho (LED), o uso de luz pulsante e as fibras óticas. As fibras óticas têm permitido superar limitações dos sensores fotoelétricos para operar em altas temperaturas e ambientes de condições extremas, evitando que o sensor esteja em zonas de alta temperatura e desta forma aumentado sua vida útil. (Angel, 1993)

FIGURA 3.3 – CONTROLADOR DE LUMINOSIDADE



3.1.3 SENSORES DE ULTRA-SOM

Os sensores de ultra-som estão enquadrados dentro dos sensores acústicos, já que basicamente detectam ondas sonoras, pode-se utilizar destas ondas sonoras para medir posteriormente outro tipo de magnitude. A função mais importante para este tipo de sensor é a de detecção de obstáculos e objetos. Também são utilizados para medição de distância e velocidade. (Angel, 1993)

Existem basicamente dois tipos de sensores de ultra-som: os capacitivos e os pizeoelétricos.

FIGURA 3.4 – SENSOR DE PRESENÇA COM ULTRA-SOM



3.1.4 SENSORES DE UMIDADE E FOGO

Existem dois tipos básicos de detectores de umidade: fotoelétricos e câmara de ionização. As características técnicas permitem que se faça uso tanto de um quanto de outro, as diferenças em tempo de resposta não são consideradas críticas para a maioria das aplicações residenciais.

O principal sinal de fogo em um detector de umidade residencial é a presença de partículas em suspensão. O processo de combustão libera na atmosfera um grande número de partículas sólidas e líquidas, de diferentes tamanhos (0,001 a 10 μ m). De acordo com o tamanho das partículas resultantes do fogo podemos considerar dois sinais diferentes: os provenientes das partículas menores de 0,3 μ m são classificados como invisíveis e aqueles maiores que são classificadas como visíveis. Os produtos resultantes da combustão são considerados partículas invisíveis, e a umidade é considerada como partículas visíveis. As partículas invisíveis são as que primeiro aparecem como sinal de fogo. (Angel, 1993)

FIGURA 3.5 – SENSOR DE UMIDADE



FIGURA 3.6 – SENSOR DE FUMAÇA



3.2 ATUADORES

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

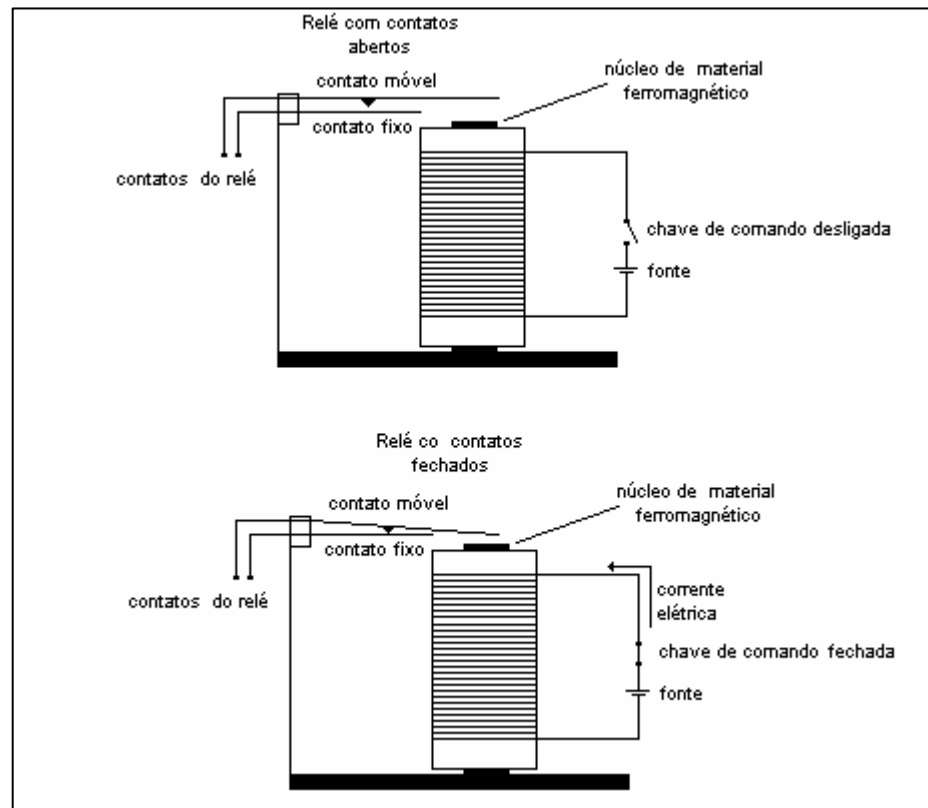
Existe grande variedade de atuadores, podendo atuar em áreas específicas ou em ampla escala. Como exemplo de atuadores elétricos normalmente utilizados cita-se as solenóides, motores de corrente contínua/alternada e motores de passo.

Os elementos de saída dos controladores são constituídos normalmente por relés eletromecânicos ou de estado sólido e em casos específicos transistores.

3.2.1 RELÉS

Os relés são componentes de grande importância, com eles pode-se controlar grandezas elétricas como a passagem de uma corrente elétrica ou a ligação de um circuito qualquer a partir de um sinal elétrico de pequena intensidade. Em relés eletromecânicos, o sinal de pequena intensidade (tensão e corrente) é aplicado aos terminais da bobina do relé, a corrente elétrica que circula por esta bobina cria um campo magnético em seu entorno que atrai um ou mais contatos elétricos do relé que servirão para desligar ou ligar um circuito ou equipamento. A vantagem do uso de relés está na isolação elétrica entre o circuito de comando e o circuito comandado, que geralmente apresenta magnitudes de grandezas envolvidas diferentes e a possibilidade de se controlar grandes equipamentos ou sistemas a partir de sinais elétricos de pequena intensidade. A figura 3.7 mostra em detalhes o esquema do relé eletromecânico.

FIGURA 3.7 – ESQUEMA DO RELÉ ELETROMECAÂNICO

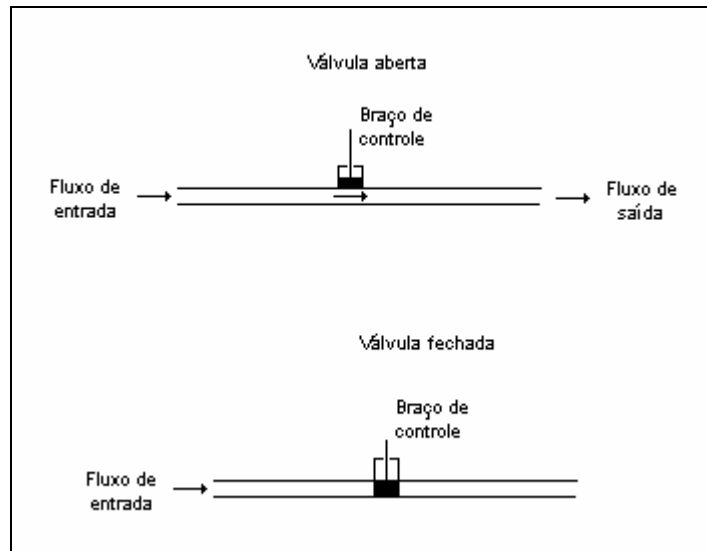


O relé eletrônico possui a mesma função de um relé eletromecânico com a diferença de que o chaveamento dos terminais é realizado através de semicondutores, também conhecidos como componentes discretos.

3.2.2 VÁLVULAS

A válvula é um dispositivo capaz de interromper o fluxo de um fluido qualquer, como por exemplo, água, gás, óleo, ar, etc. Existem diversos tipos construtivos de válvulas de acordo com a aplicação e a tecnologia disponível. Na figura 3.8 é mostrado um exemplo de uma válvula simples, que quando aberta, permite a passagem de um fluido. A válvula pode ser fechada através do braço de controle. Após o fechamento, a circulação do fluido é interrompida. (Sihuieri, 1973)

FIGURA 3.8 – VÁLVULA



A atuação pode ser feita manualmente, através de um mecanismo hidráulico, motor elétrico ou através de um atuador eletromecânico. No último caso, o braço atuador é conectado em uma bobina conhecida como solenóide. O solenóide possui o mesmo princípio de funcionamento do relé, ou seja, é passada uma corrente elétrica por uma bobina, esta corrente gera um campo magnético que atrai um braço mecânico que então se movimenta.

4 TCP/IP - TRANSMISSION CONTROL PROTOCOL / INTERNET PROTOCOL

A arquitetura TCP/IP (*Transmission Control Protocol / Internet Protocol*) segundo Chiozzoto (1999) é o resultado do desenvolvimento de uma filosofia de interligação de redes de computadores cuja característica mais relevante é a total transparência, aos seus usuários, dos detalhes relativos às tecnologias e à forma com a qual essa interligação é feita.

O primeiro evento que pode ser relacionado à origem do TCP/IP e conseqüentemente da *Internet* data de 1957, a partir da criação de uma agência de pesquisa ligada ao Departamento de Defesa Norte-Americano, a *Advanced Research and Projects Agency* (ARPA).

Os pesquisadores ligados a essa agência publicaram em 1961, por Leonard Kleinrock o documento *Information Flow in Large Communication Nets*, no qual se apresentou uma nova técnica de transferência de informações entre computadores, baseada na comutação de pacotes. Essa técnica foi a base para todas as tecnologias de redes que se desenvolveram desde então, até os dias atuais.

Em 1966 foi elaborado um projeto que viria a ser a ARPANET, a princípio uma rede de pacotes ligando alguns centros de pesquisa. Essa ligação se concretizou em 1969, interligando algumas localidades. Em cada uma delas foi instalado um tipo de equipamento especial conhecido como *Interface Message Processor* (IMP). Essa pequena rede foi o ponto de partida da *Internet*.

A esses nós iniciais juntarem-se outros. Nessa fase, surgiram as primeiras aplicações (*e-mail*, *telnet*), que usavam o protocolo *Network Control Protocol* (NCP) para transporte por meio dos equipamentos. Esse protocolo passou a apresentar limitações, à medida que o número de equipamentos e aplicações cresciam.

Na primeira metade dos anos 70, um grupo de trabalho liderado por Vint Cerf passou a elaborar um protocolo de comunicação alternativo para uso na ARPANET. O resultado dessa pesquisa culminou com a publicação do documento *A Protocol for Packet Network Interconnection*, que continha a especificação do protocolo TCP. A esse protocolo caberia a

função de fornecer um meio de transporte entre os programas de aplicações por intermédio da ARPANET, de forma transparente às conexões entre os MIPs.

A operação com este protocolo iniciou em 1975, e a experiência mostrou a necessidade de dividir as funções delegadas a esse protocolo em dois grupos: as funções de controle de fluxo e as de conectividade entre os equipamentos. As do primeiro grupo continuaram com o protocolo TCP e para as do segundo foi criado um segundo protocolo, o IP.

Em 1983 todos os equipamentos da ARPANET passaram a usar exclusivamente o TCP/IP para comunicação entre os programas, sendo o NCP desativado em definitivo.

Com isso, a ARPANET, já desde 1982 oficialmente denominada *Internet*, teve crescimento explosivo. Passando assim a ser a espinha dorsal à qual todas as demais redes se ligavam, ou seja, caracterizando-se como o *backbone* da *Internet*.

A arquitetura TCP/IP, que propiciou a infra-estrutura de comunicação que viabilizou esse crescimento, teve utilização proporcionalmente difundida, de forma que, hoje, qualquer tipo de equipamento ou plataforma de software e serviços voltados a operar em um ambiente de rede tem o TCP/IP implementado como uma das arquiteturas.

4.1 VISÃO GERAL DA ARQUITETURA TCP/IP

Segundo Gasparini (1993), o princípio básico da arquitetura TCP/IP é oferecer aos usuários e suas aplicações, um conjunto de ferramentas que possibilite a comunicação entre redes e equipamentos diversos, escondendo os detalhes físicos (hardware e protocolos de nível inferior) dos mesmos, viabilizando o que se denomina “Redes Corporativas”. Desta forma, quando se implementa TCP/IP em uma ou mais redes interconectadas, está-se apenas adicionando a estas um conjunto de protocolos (conjunto de regras e convenções que rege a troca de informações entre computadores) e aplicações padronizadas, mantendo as características anteriores do hardware já existente.

Os protocolos da arquitetura TCP/IP podem ser utilizados sobre qualquer estrutura de rede, seja ela simples como uma ligação ponto-a-ponto ou uma rede de pacotes complexa. Como exemplo, pode-se empregar estruturas de rede como Ethernet, Token-Ring, FDDI, PPP,

ATM, X.25, Frame-Relay, barramentos SCSI, enlaces de satélite, ligações telefônicas discadas e várias outras como meio de comunicação do protocolo TCP/IP. (Werner, 1999)

4.1.1 MODELO DE COMUNICAÇÃO DE DADOS

Quando modernas redes de computadores são projetadas, muitos fatores influenciam no seu projeto. Para facilitar o desenvolvimento e a manutenção, o sistema de rede é dividido em várias camadas. Cada camada tem função de oferecer certos serviços às camadas superiores, abstraindo essas camadas dos detalhes de como os serviços oferecidos são de fato implementados. (Tanenbaum, 1994)

A camada n em uma máquina estabelece uma conversação com a camada n em outra máquina. As regras de convenções utilizadas nesta conversação são chamadas coletivamente de protocolo da camada n .

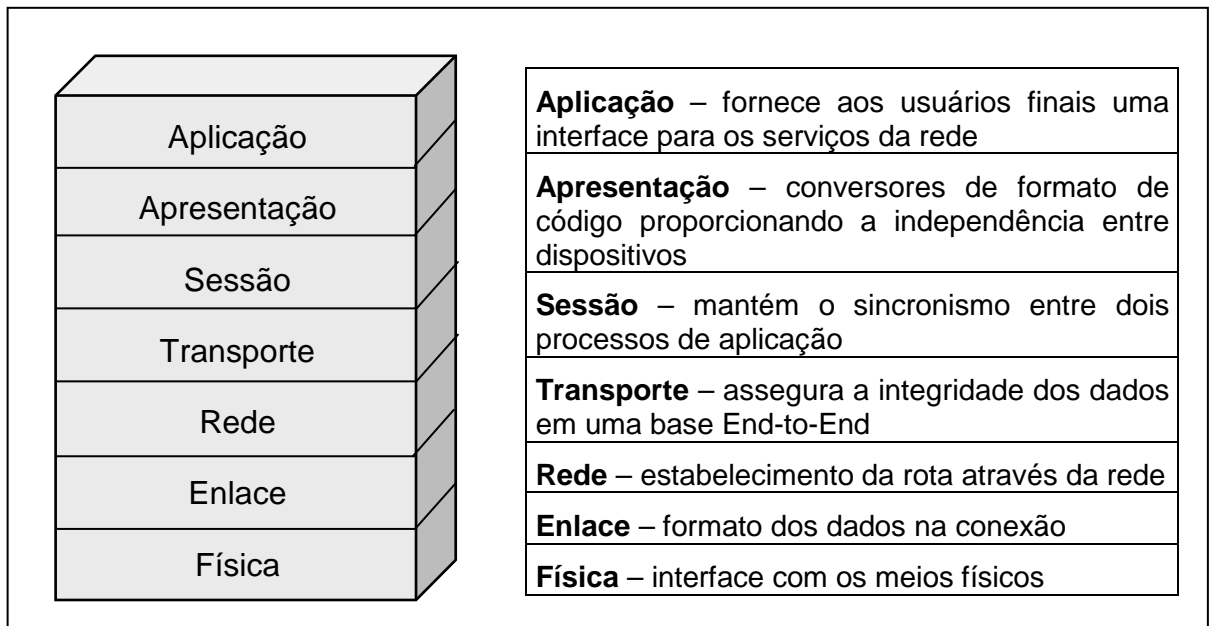
4.1.2 MODELO DE REFERÊNCIA OSI

A divisão do modelo de referência OSI em camadas é um meio bastante conveniente para ajudar a entender os conceitos de rede. É preciso uma abordagem do tipo “dividir e conquistar” ao problema com um todo, usando uma arquitetura de camadas para dividir a funcionalidade de uma rede. Uma noção básica do modelo OSI é útil para entender vários conceitos diferentes de rede. (Bockenski, 1995)

Esta arquitetura foi idealizada e padronizada por uma instituição responsável por padronizações na área de informática sendo, portanto, independente de fabricantes. A arquitetura OSI (*Open Systems Interconnection*) da ISO (*International Standards Organization*) definiu um modelo de sete camadas. (Gasparini, 1993)

As camadas que formam o modelo OSI são: física, enlace de dados, rede, transporte, sessão, apresentação e aplicação. As funções de cada camada são descritas na figura 4.1:

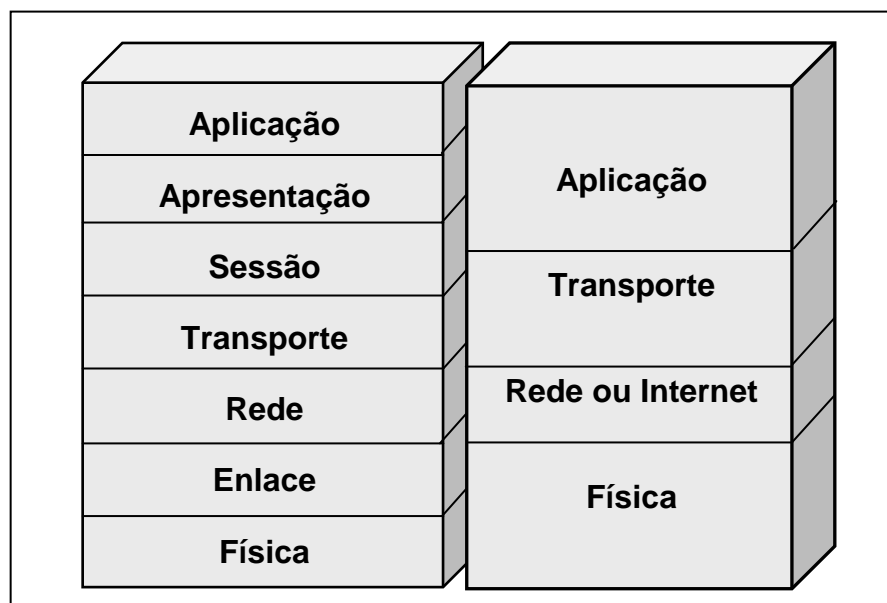
FIGURA 4.1 – DIVISÃO EM CAMADAS DO MODELO OSI



4.1.3 COMPARAÇÃO ENTRE O MODELO OSI E O TCP/IP

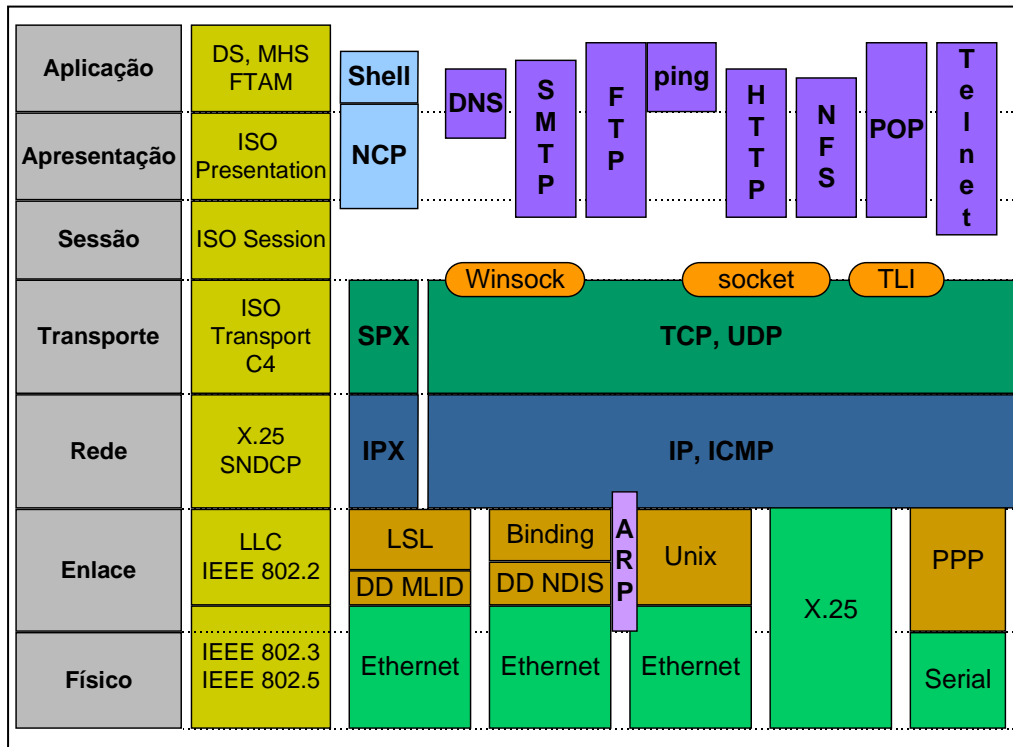
A comparação entre estas arquiteturas tem um intuito construtivo e não pejorativo, pois podemos fazer uso de ambas em uma grande rede, de forma a obter o melhor desempenho. (Gasparini, 1993)

FIGURA 4.2 – COMPARAÇÃO ENTRE O MODELO OSI E O TCP/IP



A figura 4.3 ilustra um posicionamento geral de diversos protocolos nas arquiteturas OSI, TCP/IP e Novell Netware.

FIGURA 4.3 - POSICIONAMENTO GERAL DE DIVERSOS PROTOCOLOS NAS ARQUITETURAS OSI, TCP/IP E NOVELL NETWARE



4.2 FUNÇÕES DAS CAMADAS DA ARQUITETURA TCP/IP

A arquitetura TCP/IP estabelece uma estrutura em quatro camadas ou níveis conforme é mostrado na figura 4.4:

FIGURA 4.4 – DIVISÃO EM CAMADAS DA ARQUITETURA TCP/IP

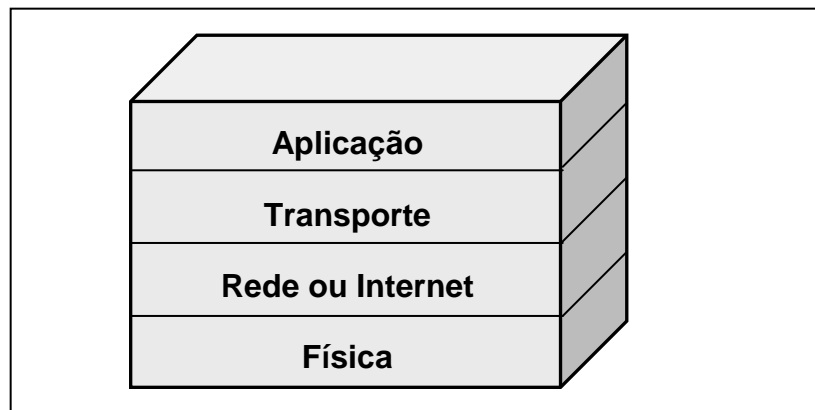
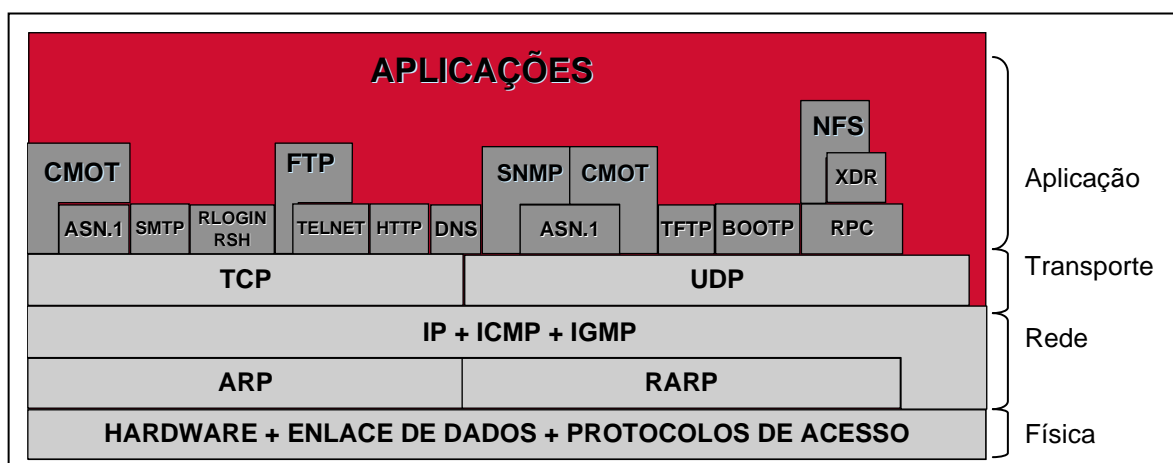


FIGURA 4.5 – VISÃO DETALHADA DE CADA CAMADA DA ARQUITETURA TCP/IP



4.2.1 CAMADA FÍSICA

Também chamada camada de abstração de hardware, tem como função principal a interface do modelo TCP/IP com os diversos tipos de redes (X.25, ATM, FDDI, Ethernet, Token Ring, Frame Relay, sistema de conexão ponto-a-ponto SLIP, etc.). Como há uma grande variedade de tecnologias de rede, que utilizam diferentes velocidades, protocolos, meios transmissão, etc., esta camada não é normatizada, o que provê uma das grandes virtudes do modelo TCP/IP: a possibilidade de interconexão e interoperação de redes heterogêneas. (Aguiar, 2000)

4.2.2 CAMADA DE INTERNET OU REDE

É o conjunto mais importante do TCP/IP, tendo como funções principais: transmitir dados entre a Camada Física e a Camada de Transporte, permitir a interconexão de diversas redes com seus diferentes meios de transmissão e a definição, montagem e roteamento dos datagramas. (Gasparini, 1993)

Suas principais funções são:

- definir para que computador (e por meio de qual interface de rede, no caso de computadores ligados a mais de uma rede) as informações oriundas da camada de transporte deverão ser encaminhadas;

- b) definir o destino das informações recebidas por intermédio das interfaces de rede, que podem ser: o reenvio para outra interface, o envio para uma das entidades da camada de transporte, ou simplesmente o seu descarte;
- c) agrupar (ou reagrupar) essas informações em unidades de transmissão conhecidas como datagramas, de tamanho compatível com a tecnologia da rede que será usada para transmissão;
- d) providenciar a sinalização, para camada de transporte, de condições de erro detectadas no processo de transmissão dos datagramas. (Chiozzoto, 1999)

Os protocolos existentes nesta camada são:

- a) protocolo de transporte de dados: IP - *Internet Protocol*;
- b) protocolo de controle e erro: ICMP - *Internet Control Message Protocol*;
- c) protocolo de controle de grupo de endereços: IGMP - *Internet Group Management Protocol*.

4.2.2.1 PROTOCOLO DE TRANSPORTE DE DADOS: IP - INTERNET PROTOCOL

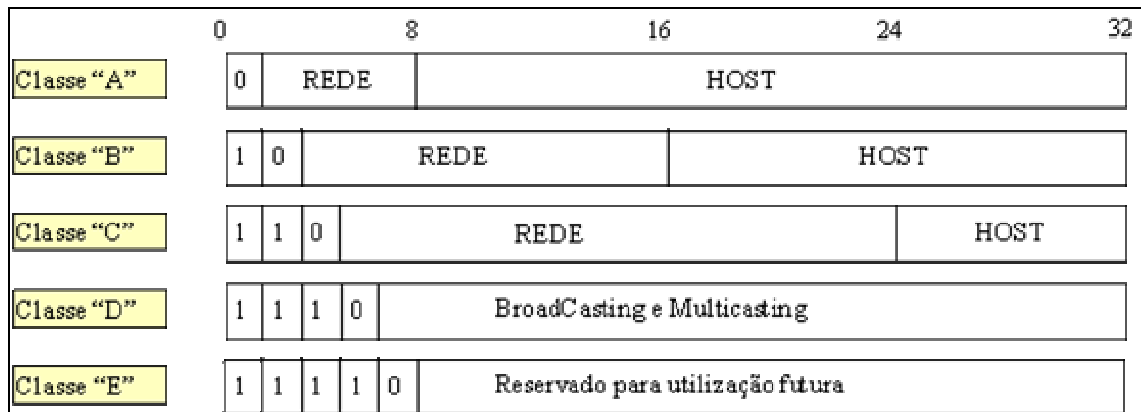
É responsável pelo encaminhamento de pacotes de dados, entre diversas sub-redes, da origem até o seu destino, sendo um protocolo do tipo datagrama (pacote a ser roteado) que opera no modo não orientado à conexão, ou seja, como um exemplo pode-se criar uma analogia com um sistema postal de entrega de cartas. Cada carta (análoga a um datagrama IP) contém o endereço de seu destinatário, as cartas podem ser reenviadas ou extraviadas. Não existe conexão entre as duas partes (o remetente e o destinatário). Segundo Aguiar (2000) a função básica do protocolo IP é o transporte dos blocos de dados por entre as sub-redes até chegar ao destinatário. Segundo Tanenbaum (1994), o IP funciona da seguinte maneira: a camada de transporte usa mensagens e as divide em datagramas de até 64 kbytes cada um. Cada datagrama é transmitido podendo ser fragmentado em unidades menores (múltiplos de 8 bits) conforme avança. Quando chegam finalmente ao equipamento de destino, todas as peças são remontadas pela camada de transporte, a fim de reconstituir a mensagem original.

Endereçamento IP

Em uma rede, a função de roteamento é feita através de *gateways* com base na interpretação dos endereços IP, constantes no cabeçalho do datagrama.

Segundo Aguiar (2000), o endereço IP é formado por um número de 32 bits no formato **nnn.nnn.nnn.nnn** onde cada **nnn** pode variar de 0 até 255 (1 octeto = 8 bits). Os endereços possuem uma classificação que varia de acordo com o número de sub-redes e de *hosts*. Tal classificação tem por finalidade otimizar o roteamento de mensagens na rede. A figura 4.6 ilustra a classificação dos endereços IP.

FIGURA 4.6 – CLASSIFICAÇÃO DOS ENDEREÇOS IP



Os endereços são fornecidos por uma entidade central: NIC (*Network Information Center*), e devem ser únicos para cada estação (*host*).

Roteamento

Roteamento é a escolha do caminho ao qual uma mensagem (que pode ser dividida em vários pacotes) percorre até chegar ao seu destino.

O endereço IP contém um campo com a identificação da sub-rede física, a análise deste campo permite ao emissor (estação ou *gateway*), determinar se o destinatário pertence ou não a sua sub-rede. Desta forma existem basicamente dois tipos de roteamento:

- a) roteamento direto, se a sub-rede do emissor for a mesma do destinatário;
- b) roteamento indireto, ocorre quando o destinatário não está diretamente conectado à mesma sub-rede física do emissor.

4.2.3 CAMADA TRANSPORTE

Sua tarefa é prover o transporte confiável e econômico de dados do equipamento de origem para o destino, independentemente da rede física ou das redes atualmente em uso

(Tanenbaum, 1994). Possui mecanismos de segurança, controle de fluxo e sequenciamento. Como esta camada serve de transporte a várias aplicações simultaneamente, ela deve controlar vários canais lógicos distintamente. Os protocolos da arquitetura TCP/IP que implementam estas funções são o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*), sendo o primeiro muito mais sofisticado e eficiente. (Gasparini, 1993)

UDP – o protocolo UDP foi desenvolvido para a utilização de aplicações que não devem gerar volume muito alto de tráfego na *Internet*.

Em comparação ao TCP, o UDP é inferior, porém com *overhead* muito menor. O UDP é considerado um protocolo “não confiável”, uma vez que não possui técnicas implementadas para detecção e correção de erros, assim sendo o UDP só é utilizado em aplicações que necessitem de maior velocidade e menor grau de confiabilidade, ou para aplicações que possuam implementadas seu próprio controle de erros. (Gasparini, 1993)

Segundo Chiozzoto (1999) o UDP fornece um serviço não orientado a conexão, isto é, não é necessário o estabelecimento de uma conexão entre os programas que desejam se comunicar.

TCP – este protocolo, ao contrário do UDP, possui mecanismos para detectar erros de comunicação. Se houver algum problema na transmissão, automaticamente o TCP solicita a retransmissão de dados até que estes cheguem íntegros ao seu destino. (Gilbert, 1995)

Segundo Chiozzoto (1999) o TCP fornece um serviço orientado a conexão, ou seja, para que a troca de informações entre programas ocorra, é necessário antes o estabelecimento de um circuito entre eles, a partir do que a transmissão dar-se-á de forma ordenada, isto é, os dados enviados por cada programa serão recebidos, pelo outro programa, na mesma ordem em que foram transmitidos.

A tabela 4.1 mostra uma comparação entre os protocolos UDP e TCP.

TABELA 4.1 - COMPARAÇÃO TCP X UDP

| Característica | TCP | UDP |
|-----------------------------------|----------------------------|-----------------------|
| Serviço Fornecido | Connection-oriented | Connectionless |
| Unidade de Transmissão | Bytes | Mensagens |
| Endereçamento de Aplicação | Por conexão | Por destino |
| Integridade da Informação | Sim | Opcional |
| Recuperação de erros | Sim | Não |
| Transmissão Ordenada | Sim | Não |

Para otimizar os recursos de comunicação, tanto o TCP como o UDP utilizam o conceito de multiplexação sobre o IP. Ou seja, mensagens a transmitir são aceitas simultaneamente de várias aplicações e passadas para o IP no sistema de origem, ocorrendo posteriormente a demultiplexação no sistema destino, onde cada mensagem é entregue à aplicação correspondente.

Para permitir identificar uma aplicação em um dado sistema, é empregado o conceito de *ports* (portas) na camada de transporte, que é representada por um número inteiro associado à aplicação a partir da negociação com o sistema operacional. Aplicações padronizadas possuem números de portas, atribuídos pelo IAB (*Internet Activities Board*), que são universalmente conhecidos dentro de uma rede *Internet*, não importando em que sistema estejam sendo processadas. Como exemplo pode-se citar: o correio eletrônico implementado pelo SMTP (*Simple Mail Transfer Protocol*), que utiliza a porta 25; e o FTP (*File Transfer Protocol*) que utiliza duas portas, sendo a 20 dedicada a conexão de dados e a porta 21 à conexão de controle.

Em seguida, são apresentados os conceitos e terminologias relacionados à *socket*, que é o mecanismo utilizado para comunicação das aplicações com a camada de transporte (TCP e UDP).

4.2.3.1 SOCKET

Um *socket* pode ser considerado um ponto de referência para o qual as mensagens podem ser enviadas e a partir do qual as mensagens são recebidas. Qualquer processo pode criar um *socket* para se comunicar com outro processo, mas os dois processos devem criar seus próprios *sockets*, visto que os dois *sockets* são usados como um par. (Bockenski, 1995)

O *socket* é a associação do endereço *Internet* de determinada máquina com um endereço de *port* (porta), atribuído por determinada aplicação. *Ports* são endereços associados às aplicações operando em um sistema, e são especificados pela aplicação em seu domínio de abrangência. Para aplicações globais, como é o caso do correio eletrônico, da transferência de arquivos e de outras, existe um conjunto de endereços *ports* reservados, possibilitando sempre o mesmo valor de porta, qualquer que seja o sistema onde estiver sendo processado. (Carvalho, 1994)

Segundo Carvalho (1994), as interfaces *sockets* acessam diretamente a camada de transporte, TCP ou UDP, e escondem das aplicações todas as complexidades existentes nos respectivos protocolos. Elas definem padrões para:

- a) determinação dos pontos finais de comunicação entre dois sistemas;
- b) estabelecimento das conexões;
- c) troca de dados entre sistemas.

4.2.4 CAMADA APLICAÇÃO

Segundo Gasparini (1993) é o nível mais alto desta arquitetura e por função principal efetuar a interface com as aplicações que desejam utilizar os meios de comunicação de dados. A arquitetura provê aplicações padronizadas para as principais tarefas do segmento de comunicação de dados. Estas aplicações são:

- a) transferência de arquivos (FTP – *File Transfer Protocol*);
- b) transferência de arquivos, documentos e aplicações (HTTP – *Hyper Text Transfer Protocol*);
- c) emulações de terminal (telnet);
- d) envio e recebimento de correio eletrônico (SMTP – *Simple Mail Transfer Protocol*);
- e) leitura de correio eletrônico (POP – *Post Office Protocol*).

4.2.4.1 SMTP - SIMPLE MAIL TRANSFER PROTOCOL

Segundo Gasparini (1993), diversos ambientes computacionais implementam facilidades de correio eletrônico através de redes de teleprocessamento. As aplicações de correio eletrônico são largamente difundidas e utilizadas, visto oferecerem um serviço simples

e rápido de transferência de mensagens, minimizando o uso do telefone e papel dentro de uma corporação, de uma empresa, de um departamento ou mesmo de um escritório. Esta facilidade pode ser aplicada para pequenas mensagens ou até extensos memorandos. Na verdade é mais comum encontrar usuários que transmitem arquivos via correio eletrônico do que os que utilizam softwares de transferência de arquivos. Diferente dos demais programas de comunicação (FTP, TELNET, etc.), na utilização de um correio eletrônico, o transmissor não precisa aguardar que determinada máquina destinatária esteja disponível, como também não precisa que a operação de transmissão de mensagens seja interrompida por falhas na rede ou na máquina receptora.

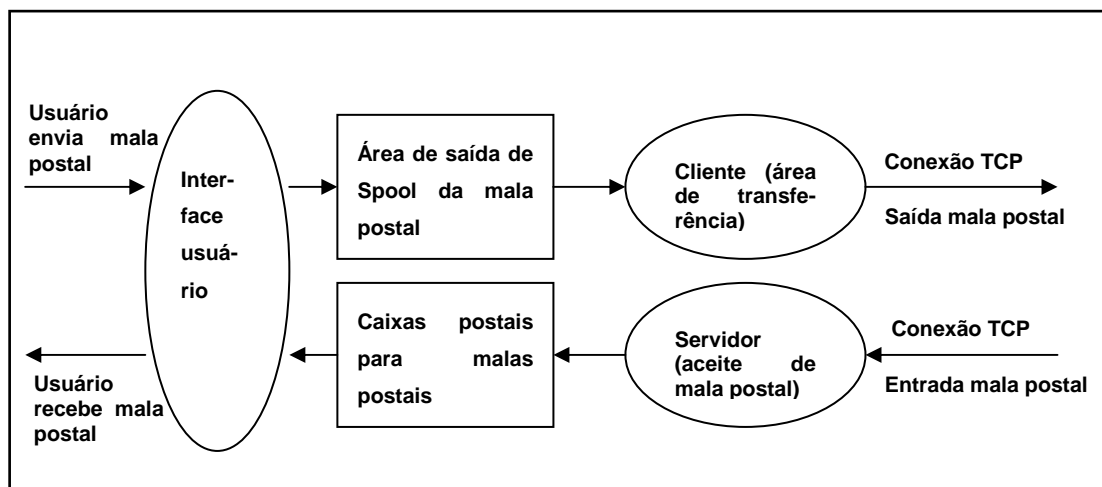
Enquanto os demais protocolos da camada de aplicação da arquitetura TCP/IP efetuam comunicações de forma *on line*, verificando o destino se ativo ou inativo, implementando métodos de detecção de erros, *time outs*, retransmissões, etc., o protocolo de correio eletrônico envia sua mensagem a uma máquina *server* (servidor) que deve armazenar a mesma em uma área denominada *spool*, e após isto, efetuar a entrega da mensagem ao respectivo destinatário. Esta técnica conhecida como *spooling*, baseia-se também na filosofia “cliente-servidor”, onde o transmissor é um cliente, que envia sua mensagem a um servidor. O servidor recebe a mensagem e coloca a mesma em uma área de armazenamento denominada *spool*, junto com a identificação do transmissor (remetente) da máquina destinatária e o tempo de armazenamento. (Gasparini 1993).

O servidor então inicia uma conexão com a máquina destino, efetuando a transferência da mensagem, sendo que após a conclusão desta etapa, efetua a confirmação de entrega da mesma ao cliente, limpando o *spool*. Desta forma, o cliente envia sua mensagem rapidamente, ficando livre para executar outras tarefas.

Quando o servidor inicia um processo de conexão, o mesmo torna-se um cliente do processador remoto, efetuando uma conexão TCP, mapeando o endereço IP da máquina destino. Quando a conexão TCP é estabelecida, o agora cliente, envia uma cópia da mensagem ao servidor de mensagens remoto, o qual copia a mesma para a área de *spool* do sistema remoto. Assim que a mensagem torna-se disponível na área de *spool* correspondente ao destinatário, o mesmo retira a mensagem com um processo cliente. (Gasparini 1993).

A figura 4.7 ilustra o processo de troca de mensagens.

FIGURA 4.7 – PROCESSO DE TROCA DE MENSAGENS



Quando a conexão TCP não é possível (por indisponibilidade da rede, do servidor remoto, etc.), o processo de transferência registra o horário da tentativa e aguarda um tempo para nova conexão. Na verdade, existe um processo periódico de verificação da área de *spool* para transmissões pendentes. O processo de transferência tenta transmitir estas mensagens continuamente em intervalos de tempo predeterminados até expirar um número de tentativas ou um período de tempo configurado (este método é o mais utilizado), como uma hora, um dia, ou uma semana. (Gasparini 1993).

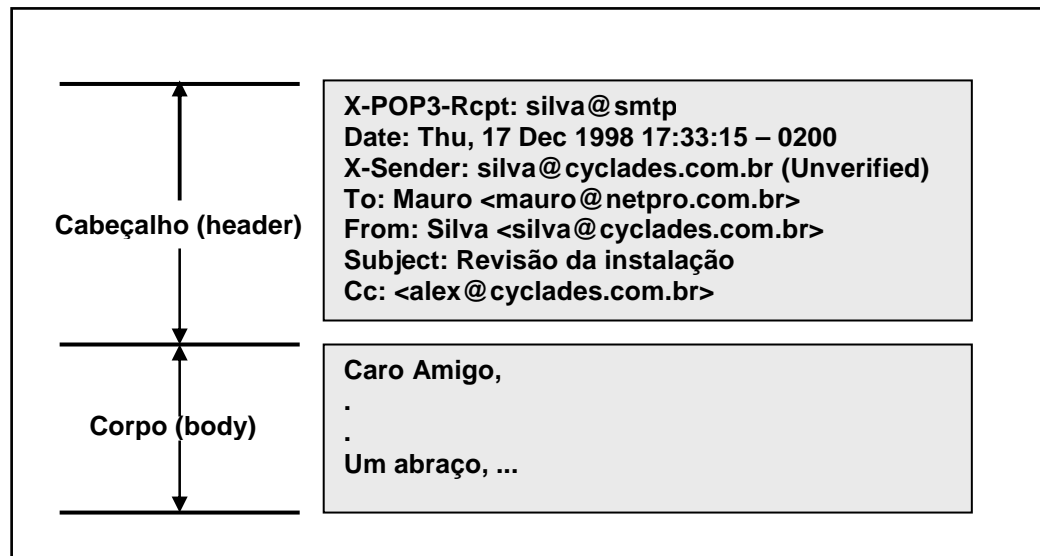
4.2.4.1.1 FORMATO DAS MENSAGENS

As mensagens de correio eletrônico possuem, além do texto fornecido pelo usuário, informações adicionais necessárias para o funcionamento do processo de transmissão. Essas informações vão desde as mais básicas, tais como: os endereços de origem e destino da mensagem, até informações sobre o caminho percorrido pela mensagem até atingir o seu destino. (Chiozzotto, 1999)

A mensagem, portanto, é composta de duas partes: um cabeçalho, em que constam essas informações acima, e um “corpo” com o texto da mensagem fornecido pelo usuário emissor da mensagem, separados por uma linha em branco. (Chiozzotto, 1999)

De acordo com a RFC 822 (1982), uma mensagem tem o formato como descrito na figura 4.8.

FIGURA 4.8 – FORMATO DA MENSAGEM



A tabela 4.2 mostra a semântica dos comandos. (Bellomo Filho, 2000)

TABELA 4.2 – SEMÂNTICA DOS COMANDOS PARA TROCA DE MENSAGENS

| Comandos |
|---|
| HELO <SP> <domain> <CRLF>; |
| MAIL <SP> FROM:<reverse-path> <CRLF> |
| RCPT <SP> TO:<forward-path> <CRLF> |
| DATA <CRLF> |
| RSET <CRLF> |
| SEND <SP> FROM:<reverse-path> <CRLF> |
| SOML <SP> FROM:<reverse-path> <CRLF> |
| SAML <SP> FROM:<reverse-path> <CRLF> |
| VRFY <SP> <string> <CRLF> |
| EXPN <SP> <string> <CRLF> |
| HELP [<SP> <string>] <CRLF> |
| NOOP <CRLF> |
| QUIT <CRLF> |
| TURN <CRLF> |

O quadro 4.1 mostra a sintaxe dos argumentos. (Bellomo Filho, 2000)

QUADRO 4.1 – SINTAXE PARA TROCA DE MENSAGENS

| |
|--|
| <code><reverse-path> ::= <path></code> |
| <code><forward-path> ::= <path></code> |
| <code><path> ::= "<" [<a-d-l> ":"] <mailbox> ">"</code> |
| <code><at-domain> ::= "@" <domain></code> |
| <code><domain> ::= <element> <element> "." <domain></code> |
| <code><element> ::= <name> "#" <number> "[" <dotnum> "]"</code> |
| <code><mailbox> ::= <local-part> "@" <domain></code> |
| <code><local-part> ::= <dot-string> <quoted-string></code> |
| <code><name> ::= <a> <ldh-str> <let-dig></code> |
| <code><ldh-str> ::= <let-dig-hyp> <let-dig-hyp> <ldh-str></code> |
| <code><let-dig> ::= <a> <d></code> |
| <code><let-dig-hyp> ::= <a> <d> "-"</code> |
| <code><dot-string> ::= <string> <string> "." <dot-string></code> |
| <code><string> ::= <char> <char> <string></code> |
| <code><quoted-string> ::= "\"" <qtext> "\""</code> |
| <code><qtext> ::= "\" <x> "\" <x> <qtext> <q> <q> <qtext></code> |
| <code><char> ::= <c> "\" <x></code> |
| <code><dotnum> ::= <snum> "." <snum> "." <snum> "." <snum></code> |
| <code><number> ::= <d> <d> <number></code> |
| <code><CRLF> ::= <CR> <LF></code> |
| <code><CR> ::= o caracter Carriage Return (código ASCII 13)</code> |
| <code><LF> ::= o caracter Line Feed (código ASCII 10)</code> |
| <code><SP> ::= caracter de espaço (código ASCII 32)</code> |
| <code><snum> ::= um, dois ou três dígitos representando um valor decimal inteiro entre 0 e 255</code> |
| <code><a> ::= qualquer um dos caracteres alfabéticos, maiúsculo ou minúsculo</code> |
| <code><c> ::= qualquer um dos 128 caracteres ASCII exceto <special> ou <SP></code> |
| <code><d> ::= qualquer dígito de 0 a 9</code> |
| <code><q> ::= qualquer um dos 128 caracteres ASCII exceto <CR>, <LF>, aspas (") ou barra invertida (\)</code> |
| <code><x> ::= qualquer um dos 128 caracteres ASCII (sem exceções)</code> |
| <code><special> ::= "<" ">" "(" ")" "[" "]" "\" "." "," ";" ":" "@" "" caracteres de controle códigos ASCII entre 0 e 31, inclusive, e 127)</code> |

A tabela 4.3 contém a descrição dos comandos para envio de mensagens. (Bellomo Filho, 2000)

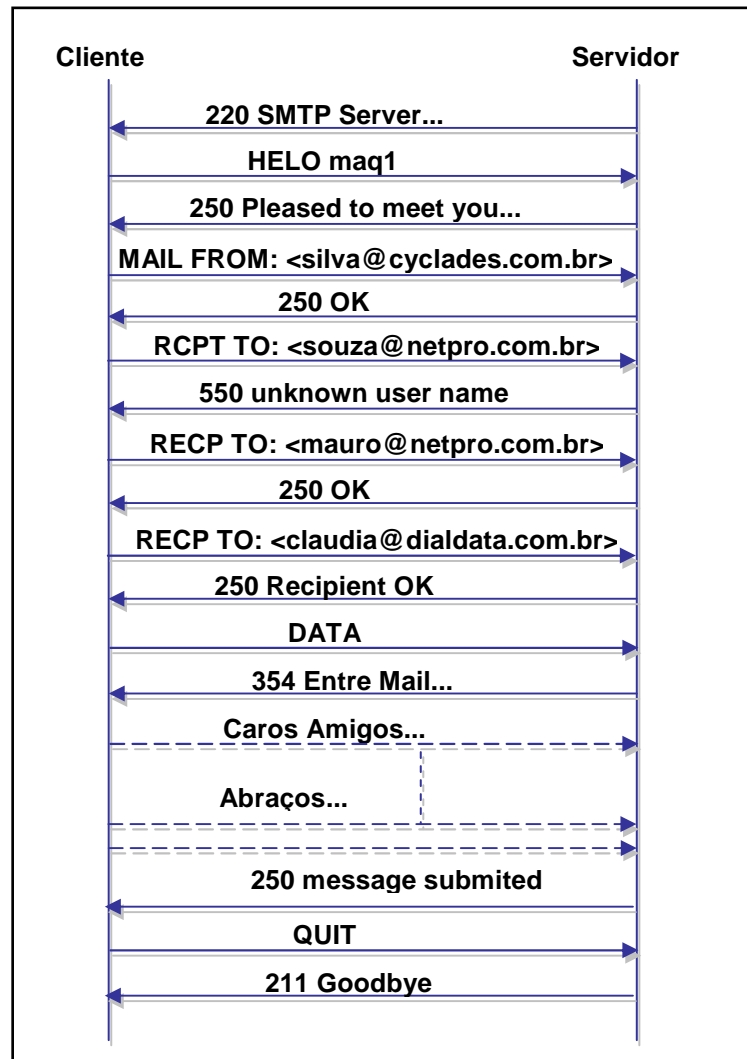
TABELA 4.3 – DESCRIÇÃO DOS COMANDOS PARA TROCA DE MENSAGENS UTILIZANDO O PROTOCOLO SMTP

| Comando | Significado |
|------------------|--|
| HELO | Identifica o emissor da mensagem para o receptor. |
| MAIL FROM | Este comando inicializa uma transação de mail na qual uma mensagem é enviada a uma ou mais caixa de mensagens (mailbox). |
| RCPT TO | Este comando identifica o destinatário da mensagem; múltiplos destinatários são definidos por múltiplos usos desse comando. |
| DATA | Inicializa a transmissão da mensagem, após seu uso é transmitido o |

| | |
|-------|--|
| | conteúdo da mensagem, que pode conter qualquer um dos 128 caracteres ASCII. O seu término é especificado por uma sequência "<CRLF>.<CRLF>". |
| RSET | Este comando determina que a operação atual de <i>mail</i> deverá ser abortada. Todos os dados referentes são descartados. |
| SEND | É usado para inicializar uma transação de <i>mail</i> na qual uma mensagem é enviada para um ou mais terminais onde estejam os destinatários e não para os seus <i>mailboxes</i> . É um comando alternativo ao comando MAIL. |
| SOML | É usado para inicializar uma transação de <i>mail</i> na qual uma mensagem é enviada para um ou mais terminais onde estejam os destinatários ou a seus <i>mailboxes</i> . A mensagem é direcionada aos terminais dos destinatários ativos no momento (e aceitando mensagens) caso contrário é direcionada aos seus <i>mailboxes</i> . É alternativo ao comando MAIL. |
| SAML | É usado para inicializar uma transação de <i>mail</i> na qual uma mensagem é enviada para um ou mais terminais dos destinatários e aos seus <i>mailboxes</i> . A mensagem é direcionada aos terminais dos destinatários ativos no momento (e aceitando mensagens) e a todos os <i>mailboxes</i> . |
| VERFY | Solicita ao Receptor-SMTP a confirmação de que o argumento identifica um usuário conhecido. Se for identificado é retornado o nome completo do usuário (se este possuir) e seu <i>mailbox</i> completo. |
| EXPN | Solicita ao Receptor-SMTP a confirmação de que o argumento identifica uma lista de usuários de <i>mail</i> (<i>mailing list</i>). Se for identificada serão retornados os membros desta lista no mesmo formato retornado pelo comando VRFY. |
| HELP | Faz com que o Receptor-SMTP envie informação de ajuda ao Emissor-SMTP. |
| NOOP | Este comando não possui efeitos nem parâmetros. Apenas faz com que o receptor envie um OK. |
| TURN | Este comando faz com que o Receptor e o Emissor troquem de papéis, o Receptor fica como Emissor e o Emissor como Receptor. |
| QUIT | Este comando determina que o Receptor-SMTP envie um OK e então feche o canal de comunicação com o Emissor-SMTP. |

Para demonstrar o funcionamento do protocolo SMTP, será usado o exemplo da figura 4.9.

FIGURA 4.9 – EXEMPLO DE FUNCIONAMENTO DO PROTOCOLO SMTP



O exemplo corresponde à transferência de uma mensagem do usuário silva do domínio (especifica o equipamento em que reside a caixa postal do usuário) cyclades.com.br aos usuários mauro e souza do domínio netpro.com.br e a usuária claudia do domínio dialdata.com.br, sendo que o usuário souza não tem caixa postal.

Inicialmente o cliente estabelece uma conexão com o servidor via TCP, e aguarda que o servidor SMTP envie uma mensagem de aceitando a conexão: “220 SMTP SERVER”.

Quando o cliente recebe a mensagem “220”, envia o comando “HELO” (abreviação de HELLO). Ao receber a mensagem “HELO”, o servidor envia outra com sua identificação, após isto, o transmissor pode enviar suas mensagens. O receptor deve confirmar positiva ou negativamente a recepção de todas as mensagens. A mensagem positiva de recepção é: “250 OK”, caso haja algum problema e o transmissor não consiga acesso ao receptor, a mensagem SMTP será: “550 UNKNOWN USER NAME”. (Gasparini, 1993)

O quadro 4.2 mostra o código numérico das respostas retornadas pelo protocolo SMTP. (Bellomo Filho, 2000)

QUADRO 4.2 – CÓDIGO NUMÉRICO DAS RESPOSTAS RETORNADAS PELO PROTOCOLO SMTP

| |
|--|
| 211 - System status, or system help reply |
| 214 - Help message (Informação de como usar o Receptor-SMTP ou algum comando não padronizado) |
| 220 - <domain> Service ready |
| 221 - <domain> Service closing transmission channel |
| 250 - Requested mail action okay, completed |
| 251 - User not local; will forward to <forward-path> |
| 354 - Start mail input; end with <CRLF>.<CRLF> |
| |
| 421 - <domain> Service not available, closing transmission channel (É uma resposta que pode ser dada a qualquer comando; indica que a conexão foi desfeita) |
| 450 - Requested mail action not taken: mailbox unavailable (Ex.: mailbox está em uso) |
| 451 - Requested action aborted: local error in processing |
| 452 - Requested action not taken: insufficient system storage |
| |
| 500 - Syntax error, command unrecognized (Usado também para casos tal como linha muito longa) |
| 501 - Syntax error in parameters or arguments |
| 502 - Command not implemented |
| 503 - Bad sequence of commands |
| 504 - Command parameter not implemented |
| 550 - Requested action not taken: mailbox unavailable (ex.: mailbox não encontrado, sem acesso) |
| 551 - User not local; please try <forward-path> |
| 552 - Requested mail action aborted: exceeded storage allocation |
| 553 - Requested action not taken: mailbox name not allowed (ex.:sintaxe do mailbox errada) |
| 554 - Transaction failed |

4.2.4.2 POP – POST OFFICE PROTOCOL

Segundo a RFC1939 (1996), este protocolo é usado para leitura de correio eletrônico.

Inicialmente, o servidor inicializa o serviço POP3. Quando o cliente desejar fazer uso deste serviço, ele estabelece uma conexão TCP (através da porta 110) com o servidor. A partir do momento que a conexão está estabelecida, o cliente e o servidor irão trocar comandos e respostas até que a conexão seja fechada ou abortada. (RFC1939, 1996)

Os comandos no POP3 consistem em palavras, possivelmente podem ser formados por um ou mais argumentos. Todos os comandos são terminados por um <CRLF>. Palavras e argumentos são formados por caracteres ASCII, e são separados por um caractere de espaço. As palavras são formadas por três ou quatro caracteres. Cada argumento pode ter no máximo 40 caracteres. (RFC1939, 1996)

As respostas no POP3 consistem em um indicador de *status* e uma palavra, possivelmente pode conter informações adicionais. Todas as respostas são terminadas por um <CRLF> (verificar em quadro 4.3). As respostas devem ter no máximo 512 caracteres incluindo o <CRLF>. O *status* indica se a operação foi bem sucedida (“+OK”) ou não (“-ERR”). (RFC1939, 1996)

Uma sessão POP3 contém vários estados. Inicialmente quando o cliente estabelece uma conexão TCP com o servidor, a sessão entra em estado de AUTORIZAÇÃO. Neste estado o cliente deve identificar-se ao servidor enviando as informações do usuário e senha do mesmo. Se esta operação for bem sucedida então a sessão passa para o estado de TRANSAÇÃO. Neste estado o cliente requer ações por parte do servidor POP3. Quando o cliente enviar o comando “QUIT”, a sessão entra em estado de ATUALIZAÇÃO. Neste estado o servidor POP3 analisa as operações adquiridas durante o estado de TRANSAÇÃO. Terminada esta operação a conexão TCP é fechada. (RFC1939, 1996)

A tabela 4.4 mostra a descrição dos comandos utilizados segundo a RFC1939 (1996).

TABELA 4.4 – COMANDOS UTILIZADOS NO PROTOCOLO POP

| Comando | Significado |
|-------------|--|
| USER | Identifica a caixa postal (<i>mailbox</i>). |
| PASS | Senha da caixa postal. |
| STAT | Verifica o estado da transação. |
| LIST | Mostra as mensagens ou uma mensagem em específico se for indicada qual mensagem deve ser listada. |

| | |
|-------------|---|
| RETR | Mostra uma mensagem. Deve ser indicada a mensagem. |
| DELE | Exclui a mensagem. |
| NOOP | Este comando não possui efeitos nem parâmetros. Apenas faz com que o receptor envie um OK. |
| RSET | Este comando determina que a operação atual de <i>mail</i> deverá ser abortada. Todos os dados referentes são descartados. |
| QUIT | Fecha a conexão. |
| APOP | Identifica o usuário e senha. |
| TOP | Coloca uma linha em branco entre o cabeçalho e o corpo da mensagem. |
| UIDL | Estabelece um identificador único para cada mensagem. |

A tabela 4.5 mostra a semântica dos comandos.

TABELA 4.5 – SEMÂNTICA DOS COMANDOS POP

| Comandos |
|--|
| USER <SP> <string> <CRLF> |
| PASS <SP> <string> <CRLF> |
| STAT <CRLF> |
| LIST [<number>] <CRLF> |
| RETR <number> <CRLF> |
| DELE <number> <CRLF> |
| NOOP <CRLF> |
| RSET <CRLF> |
| QUIT <CRLF> |

O quadro 4.3 mostra a sintaxe dos argumentos.

QUADRO 4.3 – SINTAXE DOS ARGUMENTOS NO PROTOCOLO POP

| |
|--|
| <string> ::= <char> <char> <string> |
| <char> ::= <c> "\" <x> |
| <number> ::= <d> <d> <number> |
| <CRLF> ::= <CR> <LF> |
| <CR> ::= o caractere Carriage Return (código ASCII 13) |
| <LF> ::= o caractere Line Feed (código ASCII 10) |
| <SP> ::= caractere de espaço (código ASCII 32) |
| <c> ::= qualquer um dos 128 caracteres ASCII exceto <especial> ou <SP> |
| <d> ::= qualquer dígito de 0 a 9 |
| <x> ::= qualquer um dos 128 caracteres ASCII (sem exceções) |
| <especial> ::= "<" ">" "(" ")" "[" "]" "\" "." ", ";" ":" "@" "" caracteres de controle códigos ASCII entre 0 e 31, inclusive, e 127) |

A figura 4.10 exemplifica uma sessão POP3:

FIGURA 4.10 – EXEMPLO DE UMA SESSÃO POP

```

S: <wait for connection on TCP port 110>
C: <open connection>
S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends message 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <the POP3 server sends message 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <wait for next connection>

```

S – servidor

C - cliente

4.2.4.2.1 FORMATO DAS MENSAGENS

Todas as mensagens têm formato de acordo com a RFC822, este formato está especificado no item 4.2.4.1.1 - SMTP - *Simple Mail Transfer Protocol*.

5 KIT DE DESENVOLVIMENTO RABBIT 2000 TCP/IP

Para o desenvolvimento do protótipo foi utilizado o *kit* de desenvolvimento Rabbit 2000 TCP/IP. Este *kit* é composto de uma placa programável (processador Rabbit 2000, memória *flash* e SRAM, interface Ethernet, portas seriais e entradas e saídas digitais), uma fonte de alimentação 12V e cabo serial para programação. Como mostra a figura 5.1. (Rabbit, 2000)

FIGURA 5.1 – KIT DE DESENVOLVIMENTO RABBIT 2000 TCP/IP



Esta placa foi especialmente desenvolvida para aplicações que utilizam protocolos da arquitetura TCP/IP. Entre estes protocolos estão: HTTP (*Hypertext Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), POP (*Post Office Protocol*), FTP (*File Transfer Protocol*) e TFTP (*Trivial File Transfer Protocol*).

A placa possui também saída para *display* digital e entrada para teclado.

A linguagem para programação é o Dynamic C, desenvolvida pela Z-World e fornecida juntamente com o kit Rabbit 2000 TCP/IP. O software para desenvolvimento inclui um editor, compilador, depurador e bibliotecas de funções. Algumas destas bibliotecas são:

- a) *costate.lib* – inclui funções para gerenciamento multitarefa;

- b) math.lib – funções matemáticas;
- c) rs232.lib – funções para transferência de dados;
- d) rtclock.lib – funções para controle de tempo;
- e) string.lib – funções para tratamento de caracteres;
- f) xmem – funções para controle de acesso à memória.

As bibliotecas abaixo são para uso específico com TCP/IP:

- a) arp.lib – funções para resolução de endereços IP;
- b) bootp.lib – funções para protocolo *bootstrap*;
- c) drctcp.lib – funções TCP/IP;
- d) ftp_client.lib – funções FTP para implementação de aplicações cliente;
- e) ftp_server.lib - funções FTP para implementação de aplicações servidor;
- f) http.lib – funções para o protocolo HTTP;
- g) icmp.lib – funções para o protocolo ICMP;
- h) pktdrv.lib – funções para *packet driver*;
- i) pop3.lib – funções para o protocolo POP3;
- j) smtp.lib – funções para o protocolo SMTP;
- k) vserial.lib – funções para Telnet.

A tabela 5.1 mostra com mais detalhes as especificações da placa programável.

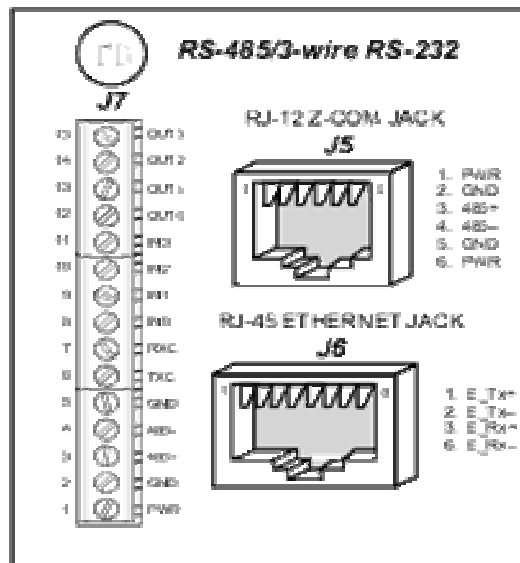
TABELA 5.1 – ESPECIFICAÇÕES DA PLACA PROGRAMÁVEL

| Componentes | Especificação |
|--|---|
| Tamanho da placa (com bateria de backup) | 4.30" x 4.71" x 0.79" (109 mm x 120 mm x 20 mm) |
| Conectores | 15 terminais, 1 RJ-12, e 1 RJ-45 |
| Temperatura de operação | -20°C to +70°C |
| Umidade | 5% to 95% |
| Tensão de entrada | 9 V to 40 V DC |
| Corrente | 100 mA - 12 VDC |
| Interface Ethernet | Conexão direta 10BaseT Ethernet via RJ-45 |
| Entradas digitais | 4 protegidas, 0 V to 5 V DC (proteção de -36 V até + 36 VDC max.) |
| Saídas Digitais | 4 open collector, sinking (200 mA, 40 V DC max.) |

| | |
|----------------------------|--|
| Microprocessador | Rabbit 2000™ compatível com Z180 |
| Clock | 18.432 MHz |
| SRAM | 128K (suporta 32K-512K) |
| Flash EPROM | 256K para programas e dados, 256K para armazenar arquivos (suporta 128K-512K) |
| Timers | 7 |
| Portas Seriais | <ul style="list-style-type: none">• 1 RS-232 (3-fios), 1 RS-485, e 1 RS-232 para programação• RS-232 (3-fios) e RS-485 pode ser reconfigurada para 1 RS-232 (5-fios) ou 2 RS-232 (3-fios) |
| Serial Rate | Velocidade máxima assíncrona 115.200 bps para ambas as portas seriais |
| Watchdog/Supervisor | Sim |
| Time/Date Clock | Sim |

A figura 5.2 mostra a placa em tamanho real.

FIGURA 5.3 - PORTAS SERIAIS E ENTRADAS E SAÍDAS DIGITAIS



6 DESENVOLVIMENTO DO SISTEMA

6.1 ESPECIFICAÇÃO

O sistema deve executar as seguintes tarefas: ligar e desligar eletrodomésticos, iluminação, ar condicionado e alarme, obedecendo a comandos (palavras chaves como: aparelho - hora - liga/desliga) enviados pelo usuário através do texto contido no *e-mail*.

O *e-mail* é enviado pelo usuário a uma conta associada ao sistema de automação, sendo que este lê o *e-mail* em períodos definidos de tempo, uma vez lido, o mesmo é interpretado. A interpretação do *e-mail* é dividida em duas fases: a primeira verifica se o usuário é válido, ou seja, se ele possui autorização para agendar tarefas. A segunda consiste em verificar se o comando enviado no *e-mail* é válido. Após passar pelas duas fases o comando é armazenado em uma lista de tarefas.

A cada *e-mail* recebido o sistema enviará uma resposta ao remetente. As respostas podem ser:

- a) usuário inválido;
- b) comando inválido;
- c) tarefa agendada com sucesso.

O formato dos comandos deve seguir a regra especificada conforme figura 6.1.

FIGURA 6.1 – FORMATO DOS COMANDOS

| | aparelho | horário | operação |
|-----|----------|---------|----------|
| Ex: | alarme | 15:30 | ligar |

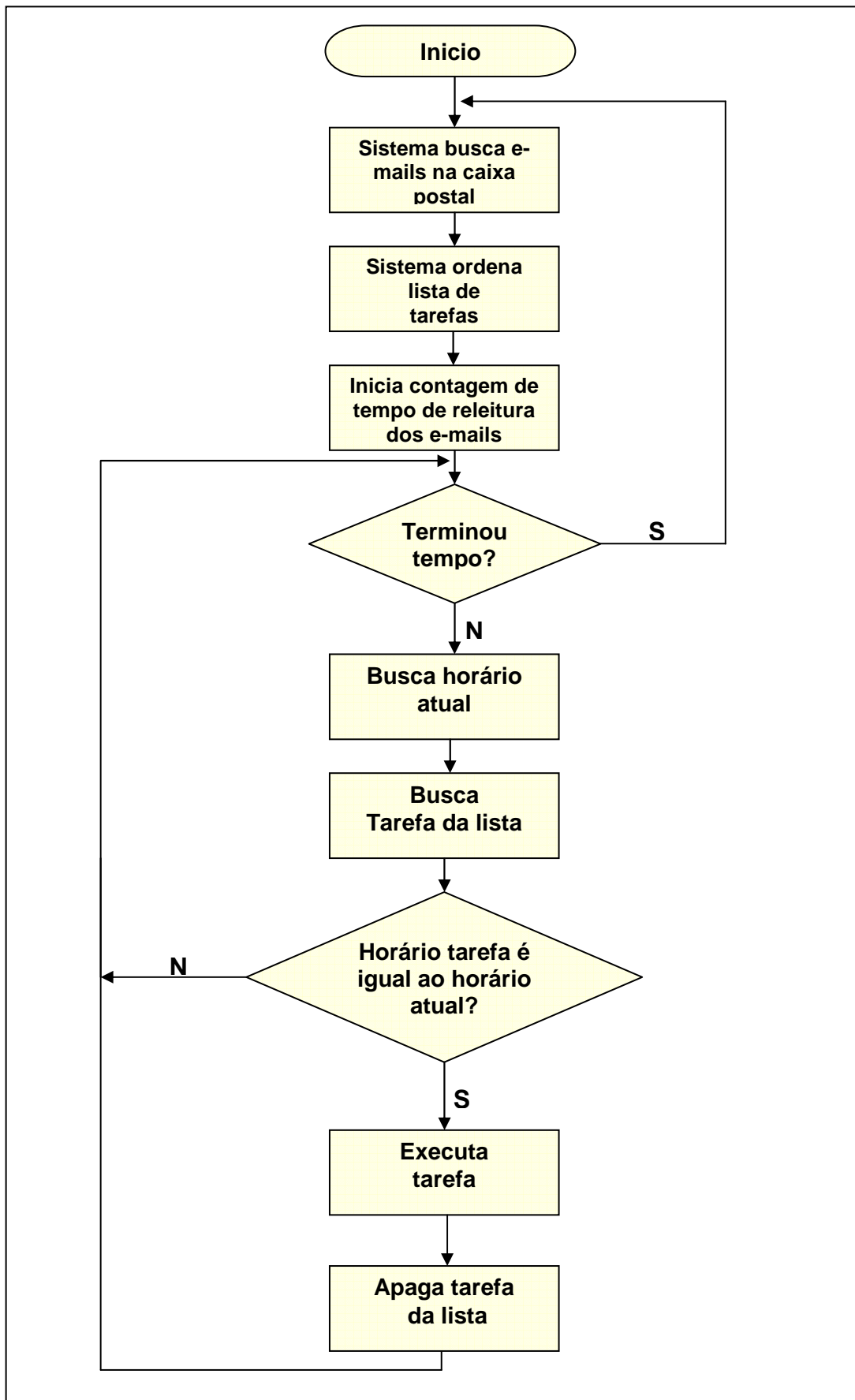
O horário deve ser no formato 24 horas.

O sistema armazena as tarefas ordenando-as de acordo com o horário que deverão ser executadas.

Para a especificação será usada a técnica de fluxogramas.

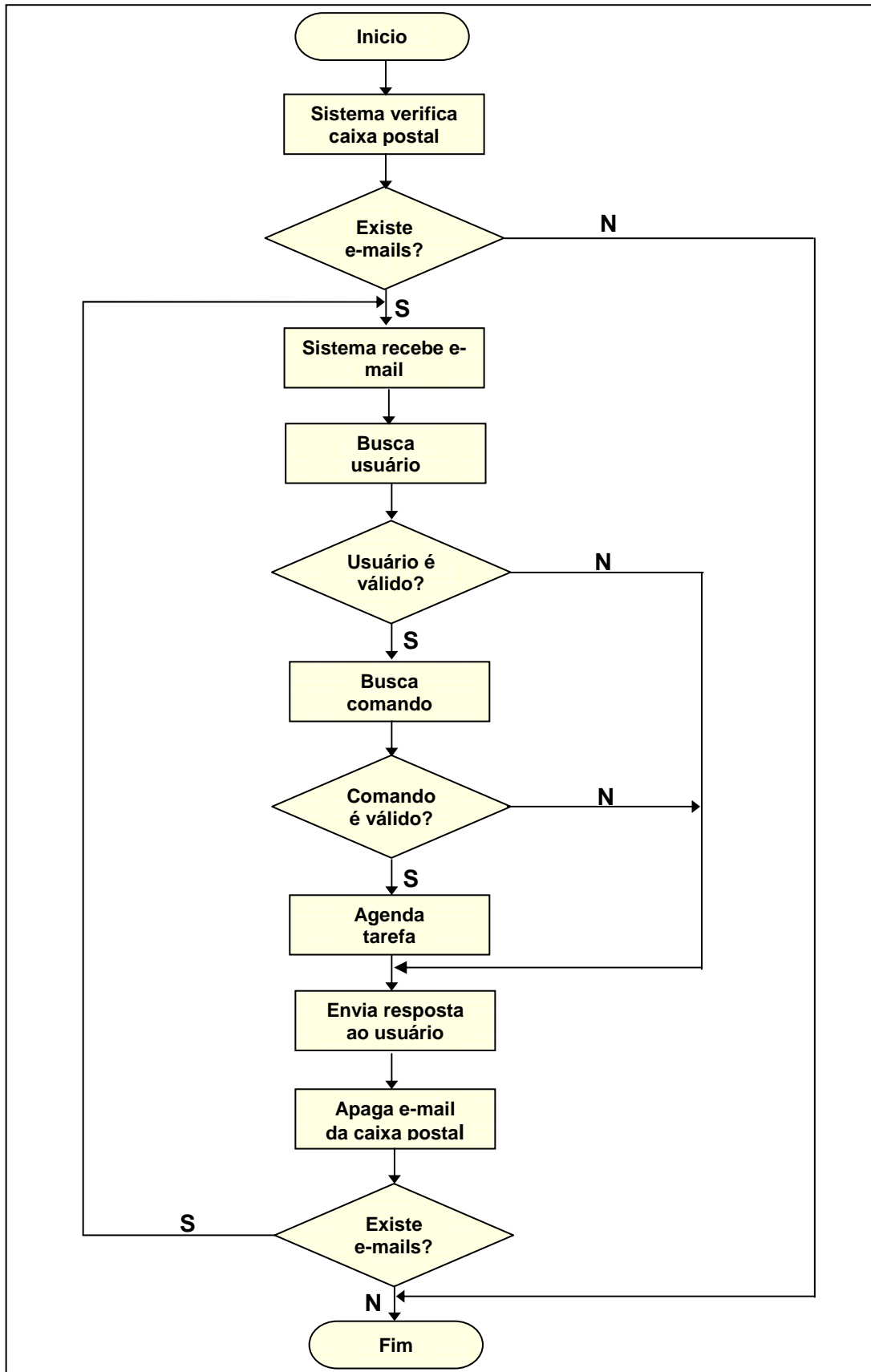
O fluxograma 6.1 proporciona uma visão geral do sistema.

FLUXOGRAMA 6.1 – VISÃO GERAL DO SISTEMA



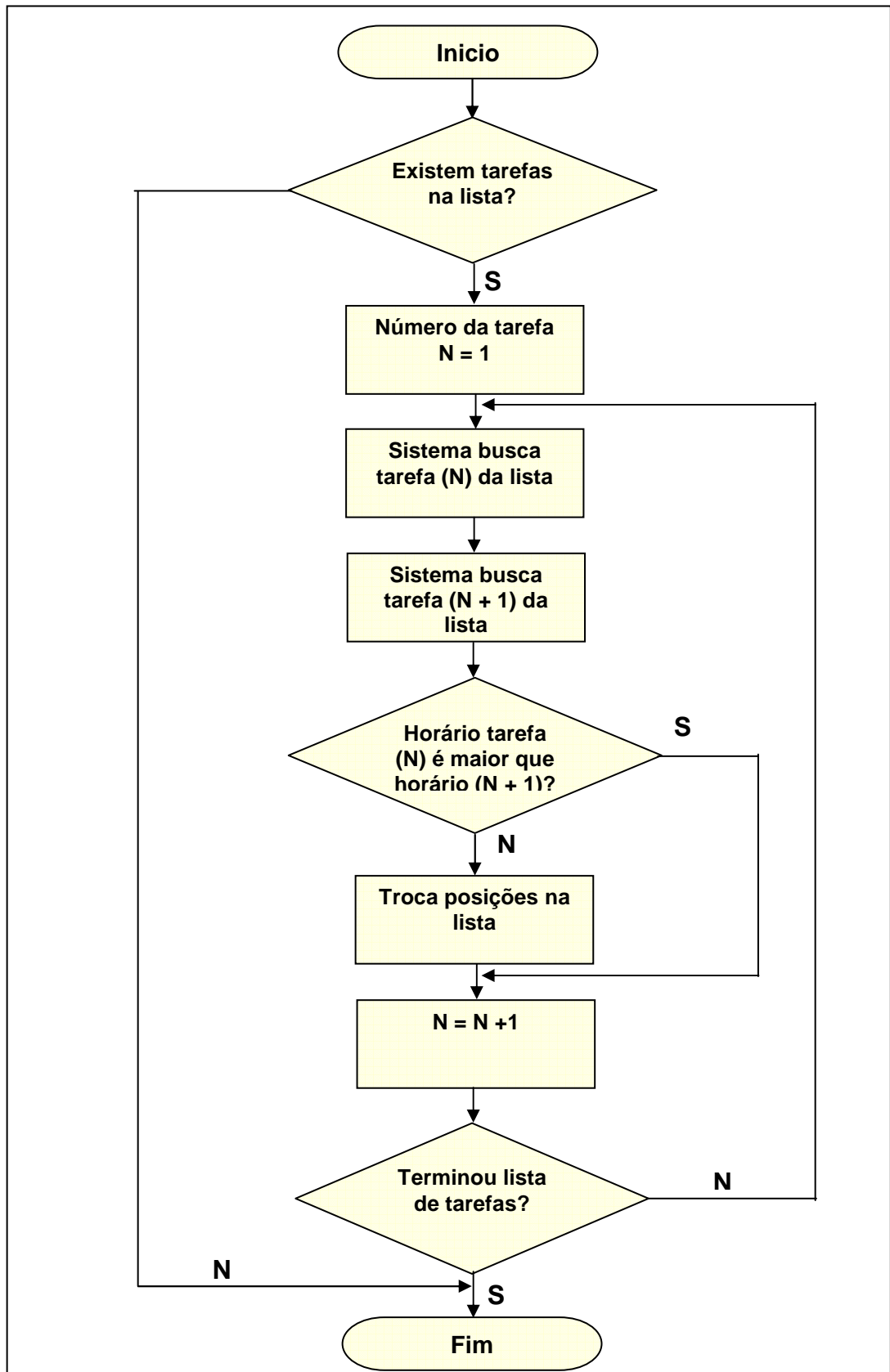
O fluxograma 6.2 demonstra o processo em que “sistema busca os *e-mails* na caixa postal”.

FLUXOGRAMA 6.2 – BUSCA DOS E-MAILS NA CAIXA POSTAL



O fluxograma 6.3 demonstra como o sistema ordena a lista de tarefas a serem executadas pelo sistema.

FLUXOGRAMA 6.3 – ORDENAÇÃO DA LISTA DE TAREFAS



6.2 IMPLEMENTAÇÃO

6.2.1 HARDWARE

O sistema é composto de placa programável (*kit Rabbit 2000 TCP/IP*), *display* de cristal líquido (LCD), placa de interface, fontes de alimentação e duas tomadas.

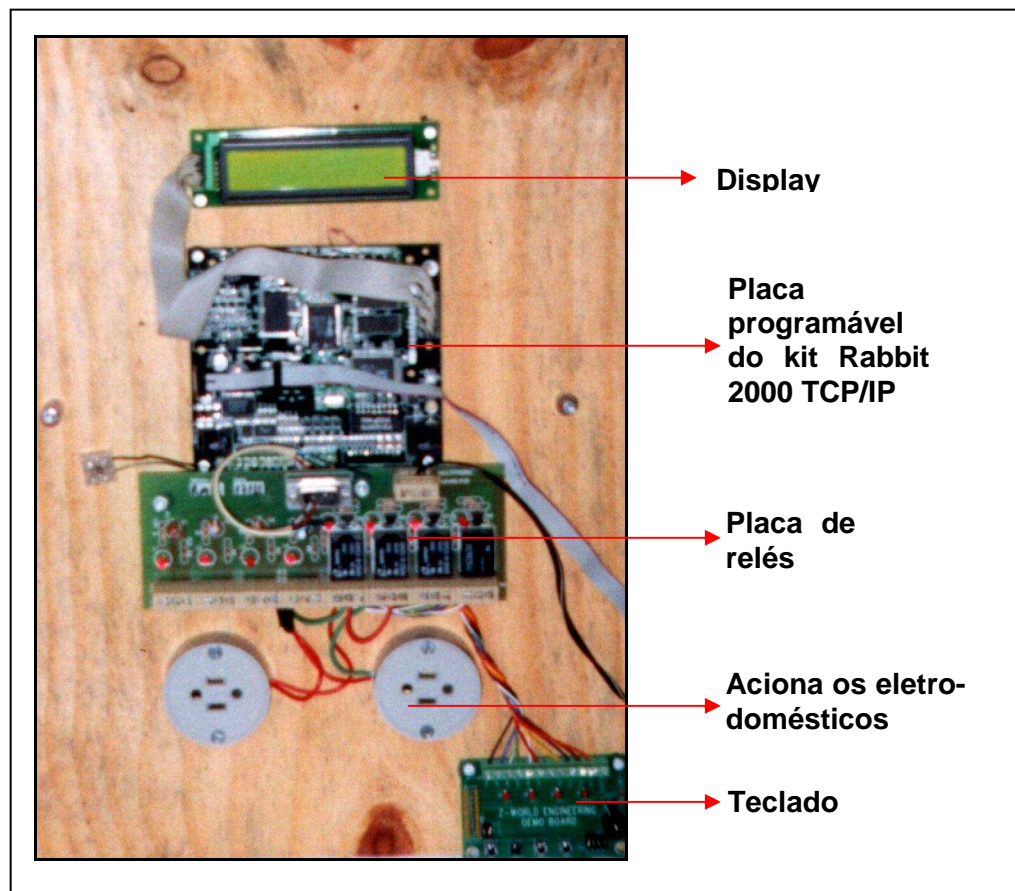
A placa programável é responsável pela recepção e envio de *e-mails*. Outra função é acionar as entradas e saídas digitais de acordo com o comando recebido através do *e-mail*.

O *display* é utilizado para visualização da próxima tarefa a ser executada pelo sistema, horário atual e andamento da execução do sistema.

A placa de interface é utilizada para transformar o sinal das saídas digitais em ações, ou seja, abrir ou fechar os contatos do relé.

As tomadas simulam os diversos aparelhos que poderão ser controlados pelo sistema.

FIGURA 6.2 – HARDWARE DO SISTEMA



6.2.2 SOFTWARE

Para o desenvolvimento do software de controle residencial foram utilizadas bibliotecas de funções fornecidas pela Z-World juntamente com a linguagem de desenvolvimento Dynamic C.

Para que o sistema possa funcionar de maneira adequada é necessário inicializar algumas constantes do sistema como: endereço IP do sistema, máscara de rede, *gateway*, e endereço do servidor onde se encontra a caixa postal de mensagens.

O quadro 6.1 demonstra como estas constantes foram definidas no sistema.

QUADRO 6.1 – CONSTANTES DO SISTEMA

```
#define MY_IP_ADDRESS    "200.135.24.67"    // endereço IP do sistema
#define MY_NETMASK      "255.255.255.0"    // máscara de rede
#define MY_GATEWAY      "200.135.24.8"     // gateway
#define MY_NAMESERVER   "200.135.24.65"    // servidor de e-mails
```

6.2.2.1 MÓDULO DE RECEPÇÃO DOS E-MAILS

Para recepção dos *e-mails* foi utilizada a biblioteca de funções POP3.LIB. Para receber um *e-mail* são necessárias três funções, estas devem estar na seqüência como descrito abaixo:

- a) `pop3_init(storemsg);`
- b) `pop3_getmail(POP_USER, POP_PASS, address);`
- c) `pop3_tick().`

O quadro 6.2 demonstra como estas funções foram utilizadas no sistema.

QUADRO 6.2 – IMPLEMENTAÇÃO DA RECEPÇÃO DE *E-MAILS*

```
sock_init();

pop3_init(storemsg);

printf("Resolvendo nome...\n");
address = resolve(POP_HOST);
printf("Chamando pop3_getmail()...\n");
pop3_getmail(POP_USER, POP_PASS, address);

while((retval = pop3_tick()) == POP_PENDING)
    continue;

if(retval == POP_SUCCESS)
    printf("POP teve sucesso!\n");
if(retval == POP_TIME)
    printf("POP timed out!\n");
if(retval == POP_ERROR)
    printf("POP retornou um erro!\n");
```

Inicialmente utiliza-se o comando **sock_init()** para iniciar uma conexão com o servidor.

O comando **pop3_init(storemsg)** inicializa o serviço de *e-mail*.

O comando **pop3_getmail(POP_USER, POP_PASS, address)** é utilizado para receber os *e-mails*, os parâmetros **POP_USER**, **POP_PASS** e **address** são respectivamente, o usuário, a senha do usuário e o endereço IP do servidor de *e-mail*.

O comando **pop3_tick()** retorna o estado da recepção. Estes estados são:

- a) POP_PENDING – a recepção dos *e-mails* ainda não terminou;
- b) POP_SUCCESS – a recepção de *e-mails* finalizou com sucesso;
- c) POP_ERROR – um erro ocorreu durante a recepção dos *e-mails*;
- d) POP_TIME – o tempo para buscar os *e-mails* terminou.

Para separar os atributos da mensagem foi criada a função **storemsg** descrita no quadro 6.3, esta função deve servir como parâmetro para a função **pop3_init()**. A função **storemsg** possui seis parâmetros que são:

- a) *num* – especifica o número do *e-mail*;
- b) *to* – indica quem é o destinatário do *e-mail*;
- c) *from* – indica quem é o emissor do *e-mail*;
- d) *subject* – especifica o assunto do *e-mail*;
- e) *body* – especifica o conteúdo do *e-mail*;
- f) *len* – indica o tamanho do *e-mail*.

QUADRO 6.3 – SEPARAÇÃO DOS ATRIBUTOS DA MENSAGEM

```
int storemsg(int num, char *to, char *from, char *subject,
char *body, int len)
{
#GLOBAL_INIT
    {
        n = -1;
    }
    if(n != num)
    {
        n = num;
        printf("RECEBENDO MENSAGEM <%d>\n", n);
        printf("\tFrom: %s\n", from);
        printf("\tTo: %s\n", to);
        printf("\tSubject: %s\n", subject);
        printf("MSG > '%s'\n", body);
        VerificaTarefa(body, from);
    }
    return 0;
}
```

6.2.2.2 MÓDULO DE ENVIO DE E-MAILS

Para envio dos *e-mails* de resposta ao usuário foi utilizada a biblioteca de funções SMTP.LIB. Para enviar um *e-mail* são necessárias três funções, estas devem estar na seqüência como descrito abaixo:

- a) `smtp_sendmail(FROM, TO, SUBJECT, BODY);`
- b) `smtp_maintick();`
- c) `smtp_status().`

O comando `smtp_sendmail(FROM, TO, SUBJECT, BODY)` é utilizado para enviar *e-mails*, os parâmetros FROM, TO, SUBJECT, BODY significam respectivamente: o remetente da mensagem, o destinatário, o assunto e o corpo da mensagem.

O comando `smtp_maintick()` retorna o estado durante o envio dos *e-mails*. Estes estados podem ser:

- a) SMTP_SUCCESS – o *e-mail* foi enviado com sucesso;
- b) SMTP_PENDING – o *e-mail* ainda não foi enviado por completo;
- c) SMTP_TIME – terminou o tempo para envio do *e-mail*;
- d) SMTP_UNEXPECTED – quando o servidor retorna uma resposta inválida.

O comando `smtp_status()` retorna o estado do último *e-mail* que foi enviado.

O quadro 6.4 demonstra como estas funções foram utilizadas no sistema.

QUADRO 6.4 – IMPLEMENTAÇÃO DO ENVIO DE E-MAILS

```

smtp_sendmail(FROM, TO, SUBJECT, BODY);

while(smtp_maintick()==SMTP_PENDING)
    continue;

if(smtp_status()==SMTP_SUCCESS)
    printf("Mensagem enviada!\n");
else
    printf("Erro enviando mensagem!\n");

```

6.2.2.3 MÓDULO DE EXECUÇÃO DAS TAREFAS AGENDADAS

Este módulo é responsável por varrer uma lista de tarefas verificando se existe alguma tarefa para ser executada. O primeiro passo executado pelo sistema é buscar a hora atual, em

seguida o sistema começa a percorrer a lista de tarefas buscando o horário das mesmas e comparando com o horário atual. Se os horários forem iguais, a tarefa é executada e retirada da lista.

Para a implementação do sistema foram utilizadas duas palavras chaves: **lâmpada** e **alarme** que representam os equipamentos que são acionados (ligados/desligados) pelo sistema.

O quadro 6.5 demonstra a função de execução de tarefas.

QUADRO 6.5 – IMPLEMENTAÇÃO DA FUNÇÃO DE EXECUÇÃO DE TAREFAS

```

void ExecutaTarefa()
{
    struct tm  relógio;
    int  i;
    char *aux, *endptr;
    unsigned int hora, minuto;

    // lê o horário da máquina
    tm_rd(&relógio);

    dispGoto(15, 1);
    dispPrintf("%02d:%02d", relógio.tm_hour, relógio.tm_min);

    // verifica se existem tarefas para o horário
    for (i = 0; i < num_tarefa; i++)
    {
        // busca o horário da lista
        aux = agenda[i].horario;
        hora = (unsigned int)strtod(aux, &endptr);
        aux = endptr + 1;
        minuto = (unsigned int)strtod(aux, &endptr);

        if (hora == relógio.tm_hour && minuto <= relógio.tm_min)
        {
            // executa tarefa
            if (strcmp(agenda[i].aparelho, "LAMPADA") == 0)
            {
                if (strcmp(agenda[i].operacao, "LIGAR") == 0)
                    BitWrPortI(PDDR, &PDDRShadow, 0xFF, 0);
                else
                    BitWrPortI(PDDR, &PDDRShadow, 0x00, 0);
            }
            else
            {
                if (strcmp(agenda[i].operacao, "LIGAR") == 0)
                    BitWrPortI(PDDR, &PDDRShadow, 0xFF, 1);
                else
                    BitWrPortI(PDDR, &PDDRShadow, 0x00, 1);
            }
            num_tarefa--;
        }
        // mostra a tarefa a ser executada
        MostraTarefa();
    }
}

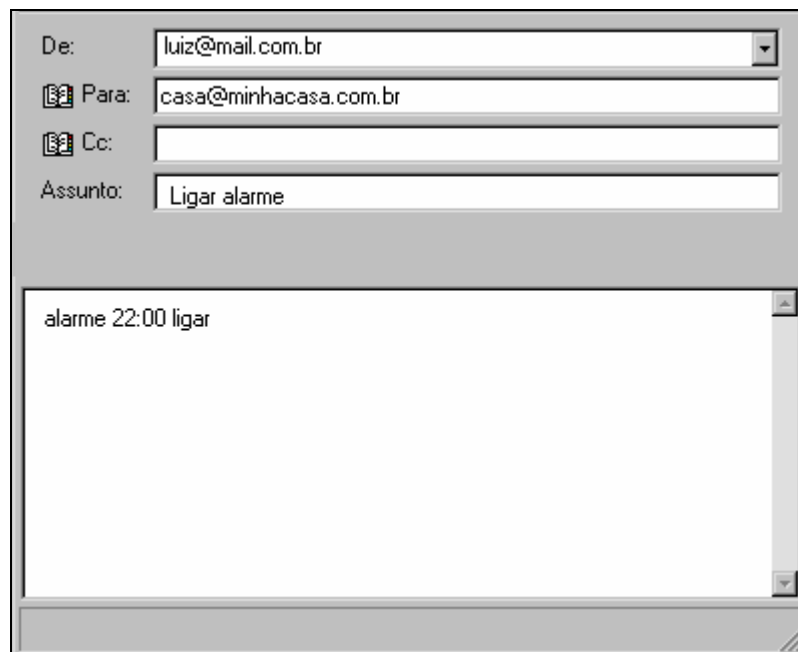
```

6.3 OPERACIONALIDADE DA IMPLEMENTAÇÃO

O exemplo a seguir tem como objetivo demonstrar o funcionamento do sistema de controle residencial.

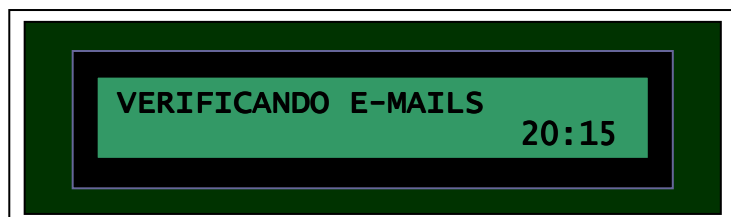
Supondo que se deseje ligar o alarme de uma casa às 22:00h. O primeiro passo a ser feito é enviar um *e-mail* agendando a tarefa. O endereço de *e-mail* “casa@minhacasa.com.br” é a caixa postal onde o sistema de controle buscará as mensagens com os comandos. A figura 6.3 ilustra o *e-mail* a ser enviado para o sistema de controle residencial.

FIGURA 6.3 – E-MAIL COM TAREFA A SER AGENDADA



O sistema de controle irá buscar as mensagens na caixa postal a cada intervalo de tempo pré-determinado. A figura 6.4 ilustra o sistema de controle buscando as mensagens de *e-mail*.

FIGURA 6.4 – SISTEMA BUSCANDO MENSAGENS DE E-MAIL



Após buscar as mensagens e verificar se o conteúdo do *e-mail* é válido, o sistema irá mostrar no *display* qual a próxima tarefa a ser executada. Estas tarefas estão ordenadas por horário, ou seja, a próxima tarefa será aquela que estiver mais próxima do horário atual. O horário atual pode ser visualizado no canto inferior direito do *display*. Também será enviada ao usuário uma resposta indicando se a tarefa foi agendada com sucesso ou se houveram erros na mensagem. A figura 6.5 mostra a próxima tarefa a ser executada.

FIGURA 6.5 – PRÓXIMA TAREFA A SER EXECUTADA PELO SISTEMA DE CONTROLE



Quando não houver mais tarefas o sistema mostrará uma mensagem como indica a figura 6.6.

FIGURA 6.6 – SISTEMA INDICANDO QUE NÃO HÁ MAIS TAREFAS NA LISTA



7 CONCLUSÕES

Além do aprendizado conceitual e tecnológico envolvidos nos estudos descritos neste trabalho, verificou-se também que o trabalho atingiu o objetivo proposto. Entre os principais resultados alcançados destacam-se a integração de duas áreas distintas: a domótica e a *internet*. Esta integração fez com que este trabalho fosse considerado destaque na Coninfo – Feira e Congresso de Informática e Telecomunicações do Cone Sul realizada de 15 a 19 de maio de 2001 (verificar reportagem em anexo), obtendo grande apoio do público presente. Outro resultado importante obtido foi o fato de se poder construir um sistema como este com um custo acessível, o que poderá torná-lo futuramente um produto comercial.

As ferramentas (*software/hardware*), utilizadas para implementação do sistema mostraram-se adequadas e eficientes possibilitando que novas expansões possam ser feitas futuramente, como por exemplo, adição de novos comandos e equipamentos a serem acionados pelo sistema.

Algumas dificuldades foram encontradas no decorrer deste trabalho como por exemplo a função para ler *e-mails*, onde não era possível saber qual das mensagens já havia sido lida pelo sistema. A solução encontrada para este problema foi apagar as mensagens da caixa postal após estas terem sido lidas pelo sistema.

A maior limitação deste sistema encontra-se no fato de que as tarefas somente podem ser agendadas diariamente, ou seja, não se pode especificar qual o dia que cada tarefa deve ser executada. Esta limitação pode ser eliminada através da implementação de novas funções.

7.1 EXTENSÕES

Como sugestões para melhorias neste trabalho pode-se citar:

- a) adição de novos comandos;
- b) implementação de uma página *html* para visualização do estado dos equipamentos;
- c) implementação de rotinas de segurança mais complexas para acesso dos usuários;
- d) implementação de funções para agendar tarefas em datas específicas;
- e) comunicação com equipamentos que utilizam IR (infra-vermelho).

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, André Luiz Santos. **Arquitetura TCP/IP**. São Paulo, [2000]. Disponível em: <<http://www.siteflow.com/hvb2e/stats.cgi?andrelsatcpip>>. Acesso em: 27 nov. 2000.

ANGEL, Patrícia Marta; FRAIGI, Liliana Beatriz. **Introducion a la domotica**. Córdoba: Embasse: EBAI, 1993. 171 p.

BELLOMO FILHO, João Avelino. **SMTP – Simple Mail Transfer Protocol**, Santa Maria, [2000]. Disponível em: < <http://penta.ufrgs.br/rc952/trab1/smtp-sum.html>>. Acesso em: 30 abr. 2001.

BESEN, Nelson. **Sistema domótico para automação e controle de um cômodo residencial**. 1996. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BOCKENSKI, Bárbara. **Implementando sistemas cliente/servidor de qualidade**. São Paulo: Makron Books, 1995.

CARVALHO, Tereza Cristina Melo de Brito. **Arquitetura de redes locais de computadores OSI e TCP/IP**. Rio de Janeiro: Makron Books, 1994.

CHIOZZOTO, Mauro; SILVA, Luís Antonio Pinto. **TCP/IP: tecnologia e implementação**. São Paulo: Érica, 1999.

GASPARINI, Anteu Fabiano L.; BARRELA, Francisco Eugenio. **TCP/IP: solução para conectividade**. São Paulo: Érica, 1993.

GILBERT, H. **Introduction to TCP/IP**. PC Lube and Tune, 1995.

GOMEZ, Luiz Alberto; ALVES, João Bosco. **Controladores multiagentes em edifícios inteligentes**, Florianópolis, dez. 1998. Disponível em: <<http://www.inf.ufsc.br/~jbosco/adriana/Luis/artigoagentes.html>>. Acesso em: 24 out. 2000.

NETO, Jayme Spinola Castro. **Edifícios residenciais de alta tecnologia**. São Paulo, maio 2001. Disponível em: <<http://www.camaradearquitetos.com.br/artigos/Artigodomotica.doc>>. Acesso em: 28 maio 2001.

RABBIT. **Rabbit 2000: TCP/IP development kit**. California: Rabbit Semiconductor, 2000.

RFC822 - Standard for the format of ARPA-Internet text messages. California: University of Delaware, 1982.

RFC1939 - Post Office Protocol - version 3. Pittsburgh: Carnegie-Mellon University, 1996.

SIHUIERI, Luciano; NISHINARI, Akiyoshi. **Controle automático de processos industriais**. São Paulo: Blücher, 1973.

TANENBAUM, Andrew S.. **Redes de computadores**. Rio de Janeiro: Campus, 1994.

WERNER, José Alberto Vasi. **Internet e arquitetura TCP/IP**. Rio de Janeiro: PUC-Rio, 1999.

ANEXOS

Reportagem extraída do jornal A Notícia de 16 de maio de 2001.