

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

**PROTÓTIPO DE SISTEMA PARA CONSULTA DE NOTAS DE
ALUNOS ATRAVÉS DE CHAMADAS TELEFÔNICAS**

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

MARLON RICHARD TRETTIN

BLUMENAU, JUNHO DE 2001

2001/1-53

PROTÓTIPO DE SISTEMA PARA CONSULTA DE NOTAS DE ALUNOS ATRAVÉS DE CHAMADAS TELEFÔNICAS

MARLON RICHARD TRETTIN

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. Sérgio Stringari

Prof. Antônio Carlos Tavares

Aos meus pais Rolf e Mariane e a minha irmã
Giselle, por tudo de bom que fizeram por mim.

AGRADECIMENTOS

Primeiramente eu gostaria de agradecer a todos os professores que durante minha vida acadêmica, de alguma maneira, contribuíram para a minha formação.

Agradeço aos meus amigos Luiz Ângelo Heizen, Fabrício Olbrisch, Rafael Innig, Jorge Lucas de Mello, Marlo Alexandre Back, Anderson Luiz Ferrari, Fabrício Jailson Barth, Giovani Endrigo Cani, André Trettin Coelho, Alexandro Siqueira da Mata, Adriano Luís da Silva e a todos os outros não citados, pela amizade, companheirismo e por todas as alegrias e dificuldades que passamos.

Agradeço especialmente ao meu orientador, Miguel Alexandre Wisintainer por ter acreditado em mim e me ajudado a concretizar este trabalho.

Finalmente, agradeço aos meus pais e a minha irmã, pela paciência, e por terem me dado todo apoio de que necessitei nos momentos difíceis, e de uma forma ou de outra, sempre estiveram presentes durante a realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS	IX
LISTA DE TABELAS	XI
LISTA DE QUADROS	XI
LISTA DE SIGLAS E ABREVIATURAS	XII
RESUMO	XVI
ABSTRACT	XVII
1 INTRODUÇÃO	1
1.1 ÁREA	1
1.2 PROBLEMA	2
1.3 JUSTIFICATIVAS	2
1.4 OBJETIVO	3
1.5 ESTRUTURA DO TRABALHO	3
2 SISTEMAS DE COMUNICAÇÃO	4
2.1 SISTEMA TELEFÔNICO	5
2.1.1 O APARELHO TELEFÔNICO	6
2.1.1.1 Discagem por pulsos decádicos	7
2.1.1.2 Discagem por tons multifrequenciais	7
2.1.2 LINHA TELEFÔNICA	8
2.1.2.1 Sinal de voz	8
2.1.3 CENTRAIS DE COMUTAÇÃO	10
2.2 COMUNICAÇÃO DE DADOS	11
2.2.1 TIPOS DE TRANSMISSÃO	11
2.2.1.1 Transmissão serial	11

2.2.1.2 Transmissão paralela	12
2.2.2 SENTIDO DA TRANSMISSÃO.....	12
2.2.2.1 Simplex	12
2.2.2.2 Half-duplex	13
2.2.2.3 Full-duplex.....	13
2.2.3 CANAL DE COMUNICAÇÃO (MEIO FÍSICO).....	14
2.2.3.1 Par metálico trançado	14
2.2.3.2 Cabo coaxial	14
2.2.3.3 Fibra ótica	15
2.2.3.4 Radiodifusão	16
2.2.3.5 Microondas	16
2.2.3.6 Satélite	16
2.2.4 MODOS DE TRANSMISSÃO.....	17
2.2.4.1 Transmissão assíncrona	17
2.2.4.2 Transmissão síncrona.....	18
2.3 MODEM.....	19
2.3.1 BREVE HISTÓRICO	20
2.3.2 CONCEITO DE MODEM.....	20
2.3.3 CLASSIFICAÇÕES DE MODEM.....	21
2.3.3.1 Modems analógicos	22
2.3.3.2 Modems digitais.....	23
2.3.4 FUNCIONAMENTO DO MODEM.....	24
2.3.5 MODEMS “TRÊS-EM-UM”	25
2.3.6 INTERFACE PADRÃO RS-232	26
2.3.6.1 Descrição dos sinais da interface	27

2.3.7	UART	28
2.4	PROTOCOLO HAYES	30
2.4.1	SINTAXE DOS COMANDOS	31
2.4.2	ESTADOS DE OPERAÇÃO	32
2.4.3	REGISTRADORES DO MODEM	34
2.4.4	COMANDOS VOICE AT	35
3	INTEGRAÇÃO ENTRE O COMPUTADOR E O TELEFONE	37
3.1	TECNOLOGIA CTI	37
3.1.1	TIPOS DE APLICAÇÕES DE CTI	38
3.1.1.1	Screen pop	38
3.1.1.2	Priorização automática de chamados	38
3.1.1.3	Caixa de mensagens integrada	38
3.1.1.4	Resposta interativa de voz	39
3.1.1.5	Roteamento avançado de chamadas	39
3.1.2	REQUISITOS PARA CTI	39
3.2	TAPI	40
3.2.1	INTRODUÇÃO	41
3.2.2	CONEXÃO BASEADA EM DESKTOP	41
3.2.3	CONEXÃO BASEADA EM REDES LOCAIS	42
3.3	ASYNC PRO 32	43
3.3.1	COMPONENTE PARA CONTROLE DE PORTAS SERIAIS	43
3.3.2	COMPONENTE TAPI	44
4	DESENVOLVIMENTO DO TRABALHO	46
4.1	ESPECIFICAÇÃO	46
4.2	IMPLEMENTAÇÃO	49

4.2.1 TÉCNICAS E FERRAMENTAS UTILIZADAS.....	49
4.2.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	54
4.2.2.1 Manutenção dos dados do sistema.....	55
4.2.2.2 Controle e acompanhamento da chamada	57
4.2.2.3 Telas auxiliares	58
5 CONCLUSÕES	59
5.1 CONSIDERAÇÕES FINAIS	59
5.2 EXTENSÕES	60
REFERÊNCIAS BIBLIOGRÁFICAS	61

LISTA DE FIGURAS

FIGURA 1 – SISTEMA DE COMUNICAÇÃO.....	4
FIGURA 2 – SISTEMA TELEFÔNICO	5
FIGURA 3 – ESTABELECIMENTO DE UMA LIGAÇÃO TELEFÔNICA.....	6
FIGURA 4 – DISCAGEM POR PULSOS DECÁDICOS.....	7
FIGURA 5 – TECLADO DO MULTIFREQÜENCIAL	8
FIGURA 6 – DENSIDADE ESPECTRAL DO SINAL DE VOZ.....	9
FIGURA 7 – TOPOLOGIA PONTO A PONTO X COMUTAÇÃO.....	10
FIGURA 8 – TRANSMISSÃO SERIAL.....	11
FIGURA 9 – TRANSMISSÃO PARALELA	12
FIGURA 10 – TRANSMISSÃO SIMPLEX.....	13
FIGURA 11 – TRANSMISSÃO HALF-DUPLEX	13
FIGURA 12 – TRANSMISSÃO FULL-DUPLEX.....	13
FIGURA 13 – CABO DE PAR TRANÇADO.....	14
FIGURA 14 – CABO COAXIAL	15
FIGURA 15 – CABO DE FIBRA ÓTICA.....	15
FIGURA 16 – ANTENA DE RÁDIO.....	16
FIGURA 17 – TRANSMISSÃO POR SATÉLITE.....	17
FIGURA 18 – BYTES EM TRANSMISSÃO ASSÍNCRONA.....	18
FIGURA 19 – TRANSMISSÃO SÍNCRONA.....	18
FIGURA 20 – DISTORÇÃO DO SINAL DIGITAL.....	19
FIGURA 21 – USO DO MODEM	21
FIGURA 22 – MODULAÇÕES FSK E DPSK	23
FIGURA 23 – CODIFICAÇÕES EM BANDA-BASE	24

FIGURA 24 – O PAPEL DA INTERFACE RS-232C	26
FIGURA 25 – CONECTOR PADRÃO DB-25P MACHO	26
FIGURA 26 – DIAGRAMA DA UART.....	29
FIGURA 27 – FORMATO SERIAL ASSÍNCRONO	29
FIGURA 28 – DIAGRAMA DE ESTADOS DO SMARTMODEM.....	31
FIGURA 29 – COMANDO HAYES	32
FIGURA 30 – EQUIPAMENTO DE CTI.....	37
FIGURA 31 – FLUXO DE FUNCIONAMENTO DO SISTEMA.....	47
FIGURA 32 – MODELO DE ENTIDADES E RELACIONAMENTOS	48
FIGURA 33 – TELA PRINCIPAL DO SISTEMA	54
FIGURA 34 – BOTÕES DE CADASTROS E ROTINAS.....	55
FIGURA 35 – TELA DE CADASTROS	55
FIGURA 36 – GRAVADOR DE SOM.....	56
FIGURA 37 – CONTROLE DE VOLUME.....	56
FIGURA 38 – TELA DE ROTINAS	57
FIGURA 39 – CONTROLE E ACOMPANHAMENTO DA CHAMADA.....	58
FIGURA 40 – BOTÕES DAS TELAS AUXILIARES	58

LISTA DE TABELAS

TABELA 1 – PRINCIPAIS SINAIS DA RS-232.....	28
TABELA 2 – COMANDOS BÁSICOS DO PROTOCOLO HAYES	33
TABELA 3 – REGISTRADORES “S”	35
TABELA 4 – DESCRIÇÃO DOS COMANDOS VOICE AT	36
TABELA 5 – TIPOS DE CONEXÃO BASEADA EM DESKTOP	42
TABELA 6 – TIPOS DE CONEXÃO BASEADA EM REDES LOCAIS	43
TABELA 7 – COMPONENTE TAPI.....	45

LISTA DE QUADROS

QUADRO 1 – COMANDOS PARA O ESTADO “COMUNICAÇÃO”	33
QUADRO 2 – DETECÇÃO DE PORTADORA	34
QUADRO 3 – CONDIÇÕES PARA O MODEM VOLTAR AO ESTADO LOCAL	34
QUADRO 4 – TAREFAS EXECUTADAS NO ATENDIMENTO DE UMA CHAMADA	46
QUADRO 5 – DICIONÁRIO DE DADOS DAS TABELAS	48
QUADRO 6 – LAYOUT DOS ARQUIVOS DE IMPORTAÇÃO DE DADOS	49
QUADRO 7 – SELECIONANDO O DISPOSITIVO TAPI.....	50
QUADRO 8 – DEIXANDO O SISTEMA ESPERANDO POR LIGAÇÃO.....	51
QUADRO 9 – RECONHECENDO TECLAS DIGITADAS NO TELEFONE	51
QUADRO 10 – VALIDANDO A IDENTIFICAÇÃO DO ALUNO	52
QUADRO 11 – PROCEDIMENTO DE TOCAR ARQUIVO DE VOZ.....	53

LISTA DE SIGLAS E ABREVIATURAS

ADPCM	– <i>Adaptative Differential Pulse Code Modulation</i> . Técnica de digitalização de voz através da codificação binária das amostras tomadas periodicamente
AM	– <i>Amplitude Modulation</i> . Modulação em amplitude
ANSI	– <i>American National Standard Institute</i>
API	– <i>Application Program Interface</i>
ASCII	– <i>American Standard Conde for Information Interchange</i>
BBS	– <i>Bulletin Board System</i> . Sistema computadorizado de quadro de avisos. As BBSs são os ancestrais diretos dos atuais provedores de Internet.
BELL 212A	– Padrão de comunicação para <i>modem</i> síncrono / assíncrono, 1.200 bps, <i>full-duplex</i> a 2 fios
BIT	– Contração das palavras <i>binary digit</i> . Representa a menor unidade de informação num sistema binário. Pode assumir os valores ‘0’ ou ‘1’
bps	– <i>bits</i> por segundo
BYTE	– Conjunto de <i>bits</i> – normalmente oito
CCITT	– <i>Comitê Consultatif Internationale de Telegraphie et Telephonie</i>
CTI	– <i>Computer Telephony Integration</i>
dB	– Decibéis
DBA	– <i>Data Base Administrator</i> . Administrador de dados
DBMS	– <i>Data Base Management System</i> . Sistema de gerenciamento de banco de dados
DCD	– <i>Data Carrier Detected</i>
DOS	– <i>Disk Operation System</i>
DPSK	– <i>Differential Phase Shift Keying</i> . Modulação por deslocamento diferencial de fase
DRX	– Dados Recebidos

ETD	– Equipamento terminal de dados
DTMF	– <i>Dial Tone Multi Frequential</i> . Tom de discagem multifrequencial
DTX	– Dados transmitidos
EEPROM	– <i>Electric Erasable Programable Read Only Memory</i>
EIA	– <i>Electronic Industries Association</i>
FAX	– Acronismo de fac-símile. Aparelho de transmissão de imagens por linha telefônica, através de uma leitura ótica e transmissão via <i>modem</i> embutido.
FM	– <i>Frequency Modulation</i> . Modulação em frequência
FSK	– <i>Frequency Shift Keying</i> . Modulação por deslocamento de frequência
FURB	– Universidade Regional de Blumenau.
G721	– Recomendação CCITT que especifica o algoritmo para implementar o codificador de voz ADPCM
GSM	– Protocolo de codificação de voz
Hz	– Hertz.
I/O	– <i>Input / Output</i> . Entrada / saída
ISDN	– <i>Integrated Services Digital Network</i> .
ISO	– <i>International Standard Organization</i> .
ITU-T	– <i>International Telecommunications Union – Telecom</i> .
Kbps	– <i>Kilo bits</i> por segundo
LAN	– <i>Local Area Network</i> .
mA	– Mili Ampere.
MNP1	– Protocolo para <i>modem</i> , <i>half-duplex</i> , assíncrono, orientado a <i>byte</i> , com uma eficiência aproximada de 70%
MNP2	– Protocolo para <i>modem</i> , <i>full-duplex</i> , assíncrono, orientado a <i>byte</i> , com uma eficiência aproximada de 80%

MNP3	– Protocolo para <i>modem, full-duplex</i> , converte os dados assíncronos em síncronos, orientado a <i>bit</i> , transmite blocos de 1 a 64 caracteres. A eficiência fica em torno de 100%
MNP4	– Protocolo similar ao MNP3, porém, com alocação dinâmica do comprimento do bloco. A eficiência fica em torno de 120%
MNP5	– Protocolo para <i>modem, full-duplex</i> , com compressão de dados. Inclui as classes 3 e 4. A eficiência típica fica em torno de 200%, variando conforme o tipo dos dados
OAI	– <i>Open Architecture Interfaces</i>
PABX	– <i>Private Automatic Branch Exchange</i>
PC	– <i>Personal Computer</i>
PCI	– <i>Peripheral Component Interconnect</i>
PL/SQL	– <i>Procedural Language / Structured Query Language</i>
PM	– <i>Phase Modulation</i> . Modulação em fase
QAM	– <i>Quadrature Amplitude Modulation</i> . Modulação em amplitude por quadratura
RAS	– <i>Remote Access Service</i> . Protocolo para acesso remoto ao computador, normalmente utilizado em BBS.
TAPI	– <i>Telephony Application Programming Interface</i>
TSAPI	– <i>Telephone Services Application Programming Interface</i>
UART	– <i>Universal Assynchronous Receiver Transmitter</i>
V	– Volt
V.17	– Recomendação CCITT para <i>modem</i> síncrono, 14400 bps, <i>half-duplex</i> a 2 fios, específico para transmissão de <i>fax</i>
V.22	– Recomendação CCITT para <i>modem</i> síncrono / assíncrono, 1200 bps, <i>full-duplex</i> a 2 fios

- V.24 – Recomendação CCITT para definição dos sinais de interface entre o *modem* e o ETD
- V.28 – Recomendação CCITT para definição das características elétricas dos circuitos de interface entre *modem* e ETD
- V.29 – Recomendação CCITT para *modem* síncrono, 9600 bps, *full-duplex* a 4 fios, a ser conectado em linhas privadas
- V.34 – Recomendação CCITT para *modem* síncrono / assíncrono, 28800 bps, *full-duplex* a 2 fios, utilizando a técnica de cancelamento de eco
- V.42 – Recomendação CCITT para correção de erro em *modems*. *Modems* que atendem a V.42 são compatíveis com aqueles que atendem a MNP4
- V.42bis – Recomendação CCITT para compressão de dados, que pode aumentar a taxa real de transmissão por um fator de 4 vezes
- WAN – *Wide Area Network*.

RESUMO

O presente trabalho propõe o desenvolvimento de um protótipo de sistema com a principal função de acesso remoto a um banco de dados acadêmico via linha telefônica. O banco de dados deste protótipo contém informações de notas de alunos em disciplinas. O acesso, via telefone, a determinado item do banco de dados é realizado através do teclado telefônico, e o resultado da consulta é devolvido ao usuário através de voz digitalizada.

ABSTRACT

The present work proposes the development of a system prototype with the main function of remote access to an academic database through phone line. The database of this prototype contains information of students' notes in disciplines. The access to certain item of the database, via telephone, is accomplished through the phone keyboard, and the result of the consultation is returned to the user through digitized voice.

1 INTRODUÇÃO

Hoje em dia, as empresas precisam estreitar o relacionamento com seus clientes para se manterem competitivas. Estreitar esse relacionamento significa tanto a empresa conhecer bem o perfil do seu cliente e mantê-lo bem informado sobre seus produtos, quanto facilitar os canais de comunicação do cliente para com a empresa. Para isso, existe uma gama de ferramentas à disposição das empresas, como centrais de atendimento telefônico, *fax*, página na Internet e *e-mail*, servindo como canal de comunicação entre a empresa e o cliente. Este último, por sua vez, procura sempre a maneira mais ágil e descomplicada de obter a informação que lhe interesse da empresa, e na maioria das vezes essa informação pode ser obtida através do telefone (Shiozawa, 1993).

Porém, para a empresa disponibilizar um telefone a um menor custo, é necessário integrar os conceitos de telefonia ao computador. A integração entre telefone e computador, dá-se o nome técnico de *Computer Telephony Integration* (CTI). Sistemas de CTI auxiliam na automação de escritórios, criando a infra-estrutura necessária para que a informação possa fluir de maneira rápida e eficaz, permitindo acesso de qualquer ponto, dentro ou fora da empresa, utilizando-se de apenas uma linha telefônica. Uma solução completa de CTI compreende diversos módulos, como o de auto-atendimento, *voice mail* (correio de voz), *telemarketing*, classificados de voz, *fax on demand* e *fax broadcast*, entre outros.

Neste trabalho será abordado somente o módulo de auto-atendimento, que realiza as funções de um “atendente eletrônico”, para possibilitar ao usuário realizar consulta ao banco de dados através de uma ligação telefônica.

1.1 ÁREA

A área da computação na qual o trabalho está inserido é a de comunicação de dados, sendo que a sub-área refere-se à integração entre o computador e o telefone, denominada CTI. Esta tecnologia é utilizada na implementação de projetos de centrais de atendimento telefônico, como centrais de banco ou de cartão de crédito, por exemplo; porém, sua área de abrangência não se restringe somente a isto. CTI é usada para uma série de aplicações, como caixa de mensagens integrada, roteador automático de chamadas, entre outras (este tema é mais amplamente abordado no capítulo 4).

1.2 PROBLEMA

Atualmente, os acadêmicos da FURB tem acesso as suas notas através da Internet, quiosques de auto-atendimento espalhados pela universidade, ou dirigindo-se até a Divisão de Registros Acadêmicos e solicitando seu Histórico Escolar. Os três meios de acesso atuais possuem suas vantagens e desvantagens. Como desvantagem comum aos dois últimos, tem-se a obrigação do aluno estar fisicamente nas instalações da FURB, além de invariavelmente ter que enfrentar filas para acessar suas notas. O acesso via Internet tem a desvantagem do aluno necessitar de um computador para consultar suas notas; já que, apesar de um grande número de alunos ter acesso a computadores com Internet, estes computadores não estão disponíveis em qualquer lugar, e a qualquer hora.

Com o desenvolvimento deste trabalho, pretende-se ampliar as opções de meios para consulta de informações acadêmicas de alunos, neste caso, consulta de notas em disciplinas. A grande vantagem do sistema de auto-atendimento telefônico, é que o aluno pode ter acesso às suas notas 24 horas por dia, 7 dias por semana, e de uma forma bastante simples e amplamente difundida que é o telefone.

1.3 JUSTIFICATIVAS

A tecnologia CTI, apesar de não ser muito nova, merece ser estudada, pois há uma grande diversidade de produtos que ela incorpora. A principal justificativa para seu estudo é que esta tecnologia trata da integração das duas principais ferramentas de um negócio atualmente, o computador e o telefone. Só este fato basta para se imaginar as inúmeras possibilidades de aplicação da mesma.

Outro fato que justifica a realização deste trabalho é que a implementação de sistemas de comunicação, normalmente, é encarada com certo receio por parte dos desenvolvedores. Este trabalho visa desmistificar o assunto, provendo uma implementação de fácil compreensão para que outros desenvolvedores possam incrementar funcionalidades ao protótipo implementado, tornando-o uma referência para o estudo de CTI.

1.4 OBJETIVO

Este trabalho tem como objetivo desenvolver um protótipo de sistema de auto-atendimento que possibilite a consulta de notas de alunos em disciplinas, via linha telefônica.

1.5 ESTRUTURA DO TRABALHO

A introdução do trabalho é apresentada no capítulo 1.

No capítulo 2 são discutidos alguns conceitos e principais componentes de um sistema de comunicação, sistema telefônico, comunicação de dados, *modem* e protocolo Hayes.

O capítulo 3 aborda a integração entre o computador e o telefone, tecnologia envolvida (CTI), padrão (TAPI) e ferramenta (ASYNC PRO 32).

No capítulo 4 encontram-se a especificação e a implementação do trabalho.

A conclusão e considerações finais estão dispostas no capítulo 5.

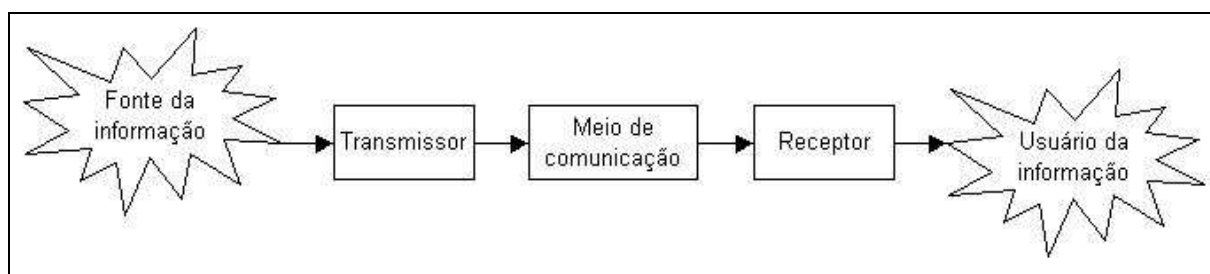
2 SISTEMAS DE COMUNICAÇÃO

Segundo Montoro (1995), um sistema de comunicação tem a finalidade de transportar uma certa informação de um ponto chamado “fonte” para outro chamado “usuário”, através de um meio de comunicação. A informação pode ser de qualquer natureza e ter qualquer forma e conteúdo. Desde que o sistema seja capaz de transportá-la de um ponto a outro, mantendo sua inteligibilidade, fica caracterizada a comunicação.

A FIGURA 1 ilustra os elementos básicos de uma comunicação, são eles:

- a) **fonte de informação:** quem gera a informação a ser transportada para o outro ponto;
- b) **transmissor:** quem tem a tarefa de transmitir, de forma adequada, através do meio de comunicação disponível;
- c) **meio de comunicação:** meio físico por onde deve passar a informação;
- d) **receptor:** quem recebe a informação transmitida pelo transmissor e a traduz para a forma desejável ao usuário;
- e) **usuário:** quem utiliza a informação.

FIGURA 1 – SISTEMA DE COMUNICAÇÃO

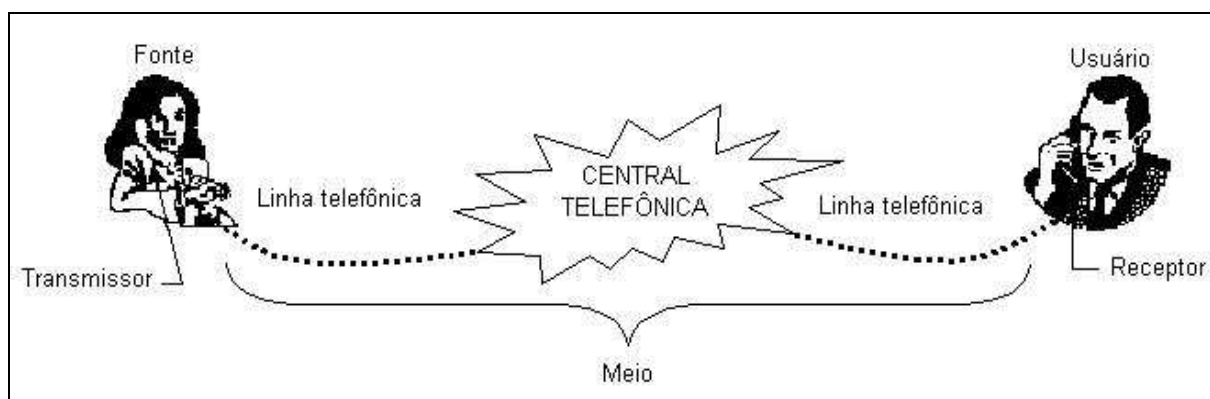


Fonte: Montoro (1995).

Um bom exemplo de sistema de comunicação, que propicia o fácil entendimento da distinção entre os elementos básicos de uma comunicação, é o sistema telefônico. A FIGURA 2 ilustra o sistema telefônico de maneira simplificada.

O telefone, aparelho inventado pelo escocês naturalizado americano, Alexander Graham Bell, é um dos elementos de um sistema telefônico. Além do telefone, o sistema telefônico é constituído basicamente pelas linhas telefônicas e pelas centrais de comutação (ambas serão melhor descritas posteriormente).

FIGURA 2 – SISTEMA TELEFÔNICO



Fonte: Montoro (1995).

Atualmente, o sistema telefônico, que inicialmente foi desenvolvido para transmitir voz, é largamente utilizado para a comunicação de dados. A seguir, uma breve abordagem ao sistema telefônico, e depois alguns conceitos de comunicação de dados.

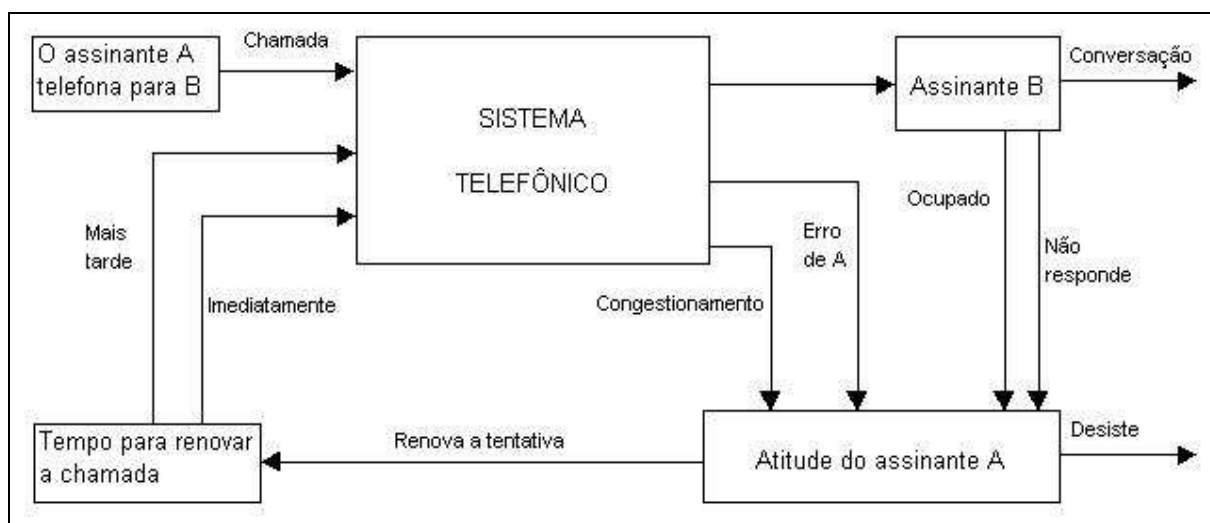
2.1 SISTEMA TELEFÔNICO

O sistema telefônico é basicamente composto pelos aparelhos telefônicos (terminais), as centrais de comutação (que permitem interligar dois aparelhos) e as linhas telefônicas (Montoro, 1995).

O fundamento da telefonia é o estabelecimento da ligação telefônica. Para tanto, além do telefone e do sistema telefônico, é necessária a existência do assinante (seguindo a linha do que foi discutido até aqui: fonte e usuário).

O processo inicia com o desejo de um determinado assinante 'A' conversar com outro assinante 'B'. O assinante 'A' começa então uma chamada por meio do sistema telefônico. Essa chamada pode ser atendida por 'B', ou sofrer alguma interrupção por conta de congestionamento no sistema, erro na discagem, telefone ocupado ou ausência de resposta por parte do assinante chamado. Em função do sinal recebido, o assinante 'A' pode tomar a atitude de desistir, ou renovar a tentativa. Essa atitude pode ser imediata, ou após algum tempo. A nova tentativa irá, dessa forma, sobrecarregar o sistema telefônico (Alencar, 1998). Esse processo está ilustrado na FIGURA 3.

FIGURA 3 – ESTABELECIMENTO DE UMA LIGAÇÃO TELEFÔNICA



Fonte: Alencar (1998).

Alencar (1998), afirma que uma das idéias mais festejadas da década é a integração de voz e dados, em sistemas telefônicos. No entanto, o sistema telefônico foi projetado e instalado com a finalidade de transportar a voz humana, sob a forma de sinal elétrico, entre vários pontos. Segundo Montoro (1995), esse sistema, que tem as características apropriadas para a transmissão do sinal de voz, tornou-se um meio atrativo para a transmissão de dados, pelo fato de já estar instalado numa extensão geográfica muito grande em quase todos os países.

2.1.1 O APARELHO TELEFÔNICO

O aparelho telefônico é o responsável pela origem e recepção das ligações. Apesar do seu aspecto simples, ele desempenha um grande número de operações (Alencar, 1998). Suas funções incluem:

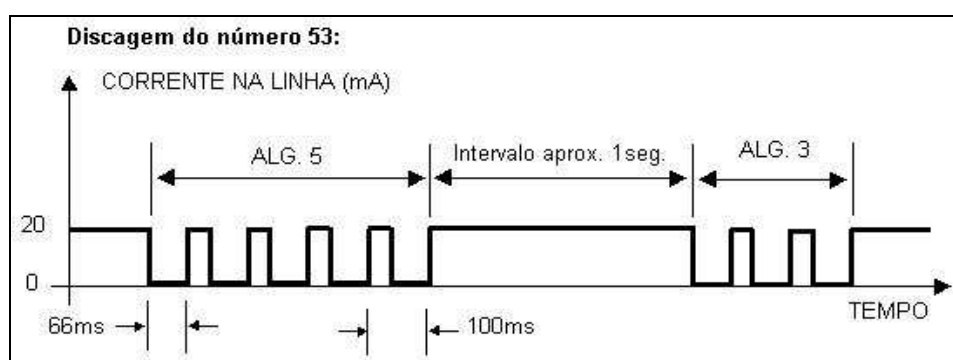
- Solicitar o uso do sistema telefônico, quando o telefone é tirado do gancho;
- Indicar que o sistema está pronto para uso, por meio de recepção de tom de discagem;
- Enviar o número do telefone chamado ao sistema;
- Indicar o estado da ligação, por meio de sinalização acústica;
- Acusar o recebimento de uma ligação, com o toque da campainha;
- Converter voz para sinais elétricos, para transmissão;
- Sinalizar o término de uma ligação ao sistema.

O aparelho telefônico, dependendo da sua central de comutação, permite dois tipos de discagem: por pulsos decádicos, ou por tons multifrequenciais.

2.1.1.1 DISCAGEM POR PULSOS DECÁDICOS

No tipo de discagem por pulsos decádicos, os dígitos são transmitidos por seqüências de pulsos. Cada algarismo discado faz com que o disco do telefone provoque, no seu retorno, a interrupção de uma chave tantas vezes quanto for o valor desse algarismo, com exceção do algarismo “0”, que provoca dez interrupções. As interrupções duram cerca de 66ms e são feitas a cada 100ms, conforme demonstra a FIGURA 4 (Montoro, 1995).

FIGURA 4 – DISCAGEM POR PULSOS DECÁDICOS

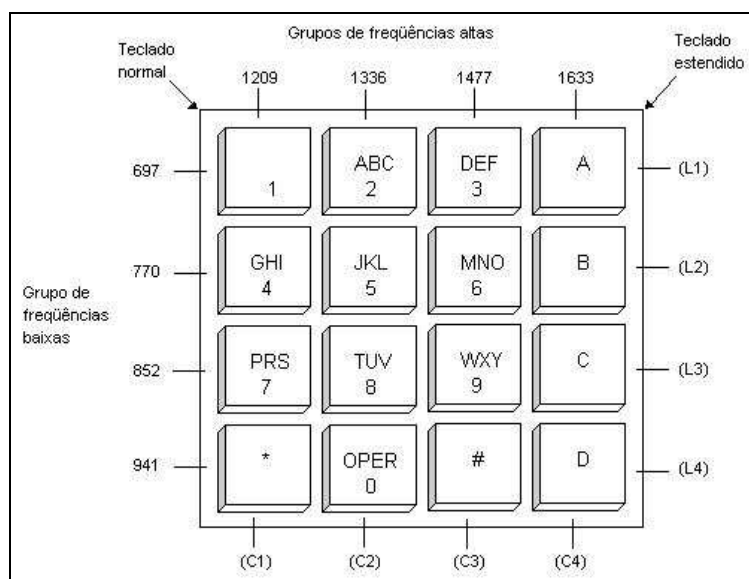


Fonte: Montoro (1995).

2.1.1.2 DISCAGEM POR TONS MULTIFREQÜENCIAIS

A discagem por tons multifrequenciais (do inglês DTMF – *Dial Tone Multi Freqüential*), ao contrário da discagem por pulsos decádicos, não é aceita por todas as centrais de comutação. Neste tipo de discagem, os dígitos são transmitidos por combinações de freqüências, com o par de freqüências associados a cada tecla, conforme demonstra a FIGURA 5, numa versão do teclado de um telefone multifreqüencial.

FIGURA 5 – TECLADO DO MULTIFREQÜENCIAL



Fonte: Alencar (1998).

2.1.2 LINHA TELEFÔNICA

Chama-se linha telefônica, todo circuito que une dois aparelhos telefônicos, permitindo a comunicação por voz. É o meio de comunicação mais utilizado em todo mundo.

É importante esclarecer que uma linha telefônica inclui todos os elementos necessários para completar a comunicação, podendo ter em sua trajetória, pares físicos (fios), amplificadores, equalizadores, multiplexadores etc. Entretanto, estes componentes não serão abordados neste trabalho, para maiores informações consulte Montoro (1995).

2.1.2.1 SINAL DE VOZ

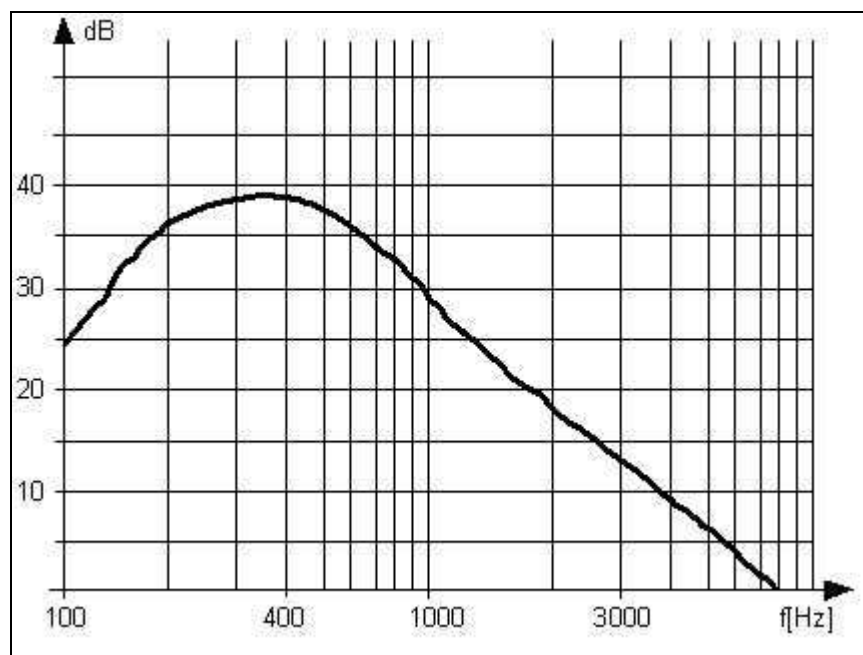
O sinal de voz transmitido pela linha telefônica é um sinal elétrico e como tal possui uma densidade espectral. A densidade espectral do sinal de voz modifica-se continuamente durante todo o tempo em que a pessoa fala. Isto é perfeitamente compreensível, pois a fala possui, por exemplo, pausas para a respiração, além de uma infinidade de sons diferentes: o som da palavra “esse” possui componentes de freqüências mais altas do que o da palavra “um”, por exemplo, por causa do “ss” (Montoro, 1995).

Apesar da constante variação da densidade espectral do sinal de voz dificultar sua caracterização, foram feitos levantamentos da densidade espectral média, ao longo de um

certo intervalo de tempo. A FIGURA 6 mostra esse resultado e nele podemos observar dois pontos interessantes (Montoro, 1995):

- O espectro da voz humana cobre uma faixa de frequência que vai de 100Hz a 8000Hz.
- A maior parte da energia se concentra na região de baixa frequência, em torno de 400Hz.

FIGURA 6 – DENSIDADE ESPECTRAL DO SINAL DE VOZ



Fonte: Montoro (1995).

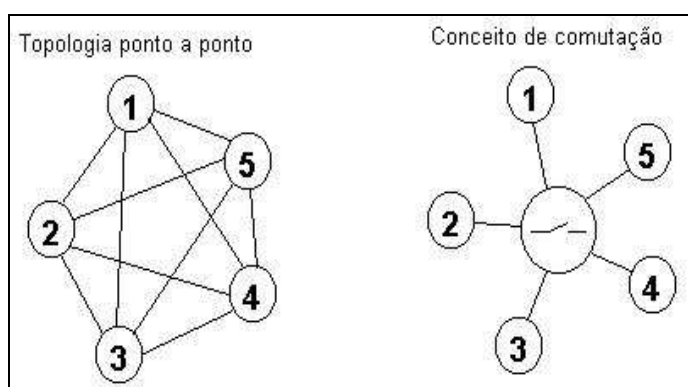
Na linha telefônica esse sinal de voz é submetido a um filtro de frequências, que só deixa passar suas componentes que estão entre 300Hz e 3400Hz. Naturalmente o sinal obtido na saída do filtro continua sendo um sinal de voz, porém sem suas componentes abaixo de 300Hz e acima de 3400Hz (fora da faixa permitida pelo filtro) (Montoro, 1995).

Apesar da energia do sinal de voz se concentrar nas baixas frequências, a importância para manter a legibilidade está entre 700Hz e 3000Hz. A faixa de 300 a 3400Hz foi adotada como “faixa de voz”, simplesmente porque esta faixa dos espectro do sinal de voz é suficiente para garantir a sua inteligibilidade. A faixa de voz é, portanto, um canal de 3100Hz, comumente referido como canal de voz (Montoro, 1995).

2.1.3 CENTRAIS DE COMUTAÇÃO

Com a introdução do telefone como meio de comunicação, bem como com o crescimento desta necessidade, houve como fator econômico o surgimento do conceito de comutação telefônica. Uma central de comutação, serve para centralizar o serviço de interligação entre terminais telefônicos. Com o conceito de comutação, essa interligação torna-se mais manutenível e racional, além de ser economicamente mais viável do que interligar cada terminal “ponto a ponto”. A FIGURA 7 traz o comparativo entre 5 terminais telefônicos interligados numa topologia ponto a ponto, e interligados através de uma central de comutação (Soares, 1993).

FIGURA 7 – TOPOLOGIA PONTO A PONTO X COMUTAÇÃO



Adaptado de: Soares (1993).

Dentro da evolução natural dos conceitos, existem 3 tipos básicos de centrais de comutação:

- a) **Centrais manuais:** a interligação entre dois assinantes é feita manualmente, por um operador humano;
- b) **Centrais analógicas:** a interligação entre dois assinantes é feita de maneira automática, porém o equipamento da central funciona analogicamente, através de relês;
- c) **Centrais digitais:** a interligação entre dois assinantes é feita de maneira automática, o equipamento da central é digital e programável por *software*.

2.2 COMUNICAÇÃO DE DADOS

Comunicação de dados, segundo Nunes (1989), é um sistema de comunicação que trata da transferência de dados entre computadores e componentes de um sistema de computação.

Os conceitos vistos no início deste capítulo, sobre os elementos de um sistema de comunicação, como fonte de informação, transmissor, meio de comunicação, receptor e usuário da informação, também se aplicam à comunicação de dados. Há, porém, na comunicação de dados outras relevantes características, quanto ao tipos, sentidos, canais e modos de transmissão de dados.

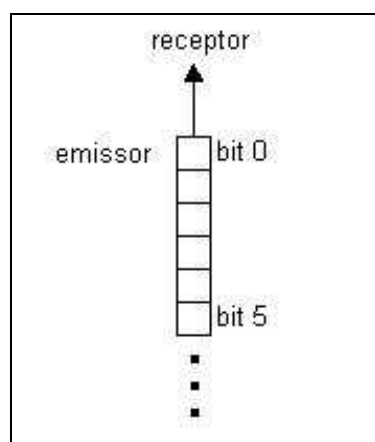
2.2.1 TIPOS DE TRANSMISSÃO

A transferência, ou transmissão, de dados entre dois pontos (fonte e destino) pode ser realizada, basicamente de duas formas: transmissão serial ou transmissão paralela.

2.2.1.1 TRANSMISSÃO SERIAL

Os *bits* que representam uma informação (*byte*) são transmitidos seqüencialmente, um a um, por um único suporte físico. Esta forma de transmissão é a mais utilizada na comunicação de dados entre computadores (Tafner, 1996). A FIGURA 8 ilustra o processo de transmissão serial.

FIGURA 8 – TRANSMISSÃO SERIAL

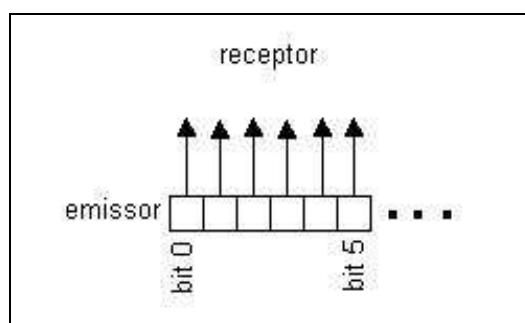


Fonte: Tafner (1996).

2.2.1.2 TRANSMISSÃO PARALELA

Os *bits* que representam uma informação (*byte*) são transmitidos simultaneamente, através de diversos suportes físicos em paralelo. Esta forma de transmissão é utilizada para curtas distâncias entre computadores, ou ainda, para operações internas em um computador e na comunicação do computador com os periféricos (por exemplo, uma impressora) (Tafner, 1996). A FIGURA 9 ilustra o processo de transmissão paralela.

FIGURA 9 – TRANSMISSÃO PARALELA



Fonte: Tafner (1996).

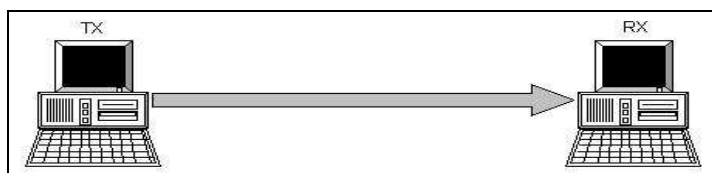
2.2.2 SENTIDO DA TRANSMISSÃO

Na comunicação de dados, considerando a conexão entre dois computadores, a transferência dos dados é classificada de acordo com o sentido da transmissão. O sentido da transmissão, especificado particularmente para cada conexão, determina a figura do transmissor e do receptor. Quanto ao sentido, a transmissão de dados pode ser *simplex*, *half-duplex* ou *full-duplex* (Tafner, 1996).

2.2.2.1 SIMPLEX

Caracteriza a transmissão de dados num único sentido, da origem (transmissor) para o destino (receptor), durante o processo de comunicação. Este tipo de comunicação é pouco utilizado, principalmente por não possibilitar uma resposta do receptor, confirmando se a transmissão foi ou não recebida. Na prática, o computador configurado para ser o transmissor das informações permanecerá assim até que se configure o contrário, o que em alguns casos não é possível devido a características do *hardware* ou do *software* instalado. O mesmo ocorre com o receptor (Tafner, 1996). A FIGURA 10 ilustra uma transmissão *simplex*.

FIGURA 10 – TRANSMISSÃO SIMPLEX

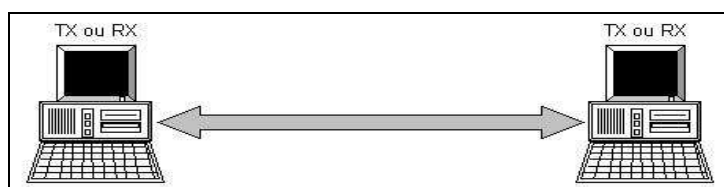


Adaptado de: Tafner (1996).

2.2.2.2 HALF-DUPLEX

A transmissão de dados *half-duplex* pode ser realizada nos dois sentidos, porém não simultaneamente. Ou seja, a figura do transmissor e do receptor se alternam durante o processo de comunicação, quando este for o caso. Atualmente, em relação ao sentido da transmissão, *half-duplex* é o mecanismo mais implementado nas aplicações de comunicações de dados (Tafner, 1996). A FIGURA 11 ilustra uma transmissão *half-duplex*.

FIGURA 11 – TRANSMISSÃO HALF-DUPLEX

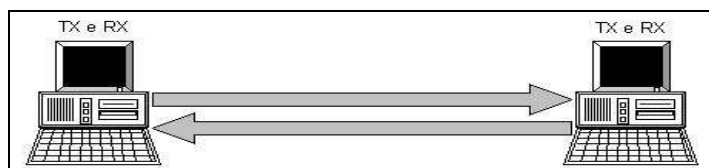


Adaptado de: Tafner (1996).

2.2.2.3 FULL-DUPLEX

A transmissão de dados *full-duplex*, é realizada nos dois sentidos simultaneamente. Porém, para implementar esta característica na comunicação de dados, é necessário a disponibilidade de dois caminhos distintos, um canal (meio de transmissão) para cada um dos sentidos. Consequentemente, as figuras do transmissor e do receptor estão presentes nos dois computadores envolvidos no processo de comunicação (Tafner, 1996). A FIGURA 12 ilustra uma transmissão *full-duplex*.

FIGURA 12 – TRANSMISSÃO FULL-DUPLEX



Adaptado de: Tafner (1996).

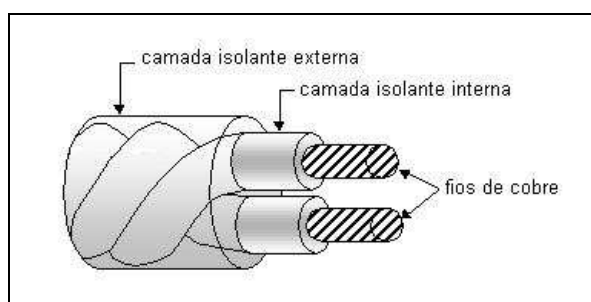
2.2.3 CANAL DE COMUNICAÇÃO (MEIO FÍSICO)

Para Tafner (1996), o canal é, basicamente, um meio físico por onde as informações (*bits*) trafegam, durante o processo de comunicação. Segundo Nunes (1989), os meios físicos podem ser classificados em **meios de condução** e **meios de irradiação**. Os meios de condução englobam as várias formas de transmissão por cabos, tais como: par metálico trançado, cabo coaxial e fibra ótica. Os meios de irradiação compreendem as transmissões por radiodifusão, microondas e transmissão por satélite. Nas próximas seções será discutido cada um destes canais de transmissão.

2.2.3.1 PAR METÁLICO TRANÇADO

Normalmente, é composto por dois fios de cobre enrolados em espiral. Este tipo de fio é usado em linhas telefônicas, em que as taxas mais elevadas de comunicação atingem alguns *megabits* por segundo. O par trançado, sofre influências do meio externo, além do fato de que a atenuação do seu sinal é diretamente proporcional à distância. Sua principal vantagem, em relação aos outros meios, é o seu baixo custo, além do fato de utilizar-se de materiais de menor custo como suporte para a transmissão (Tafner, 1996). A FIGURA 13 mostra um cabo de par trançado.

FIGURA 13 – CABO DE PAR TRANÇADO



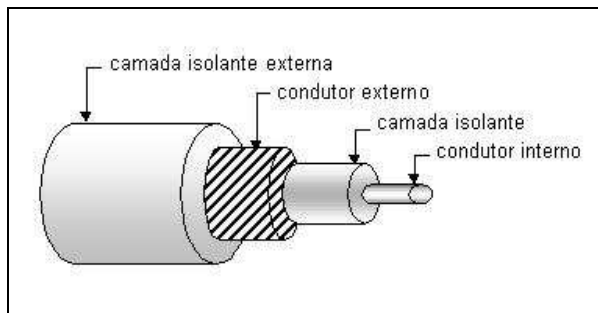
Adaptado de: Tafner (1996).

2.2.3.2 CABO COAXIAL

Este tipo de cabo é formado por um condutor cilíndrico, dentro de um tubo metálico que serve também como retorno, além de possuir uma blindagem eletrostática. O cabo coaxial é considerado um meio de transmissão bastante seguro, caracterizado pelo alto grau de imunidade a ruídos externos. Esta característica fez do cabo coaxial um meio de transmissão

muito utilizado na comunicação de dados e nas redes locais de computadores (Tafner, 1996). A FIGURA 14 apresenta a estrutura de um cabo coaxial.

FIGURA 14 – CABO COAXIAL



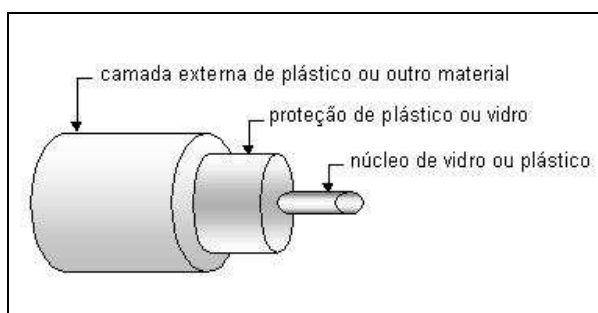
Fonte: Nunes (1989).

2.2.3.3 FIBRA ÓTICA

As fibras óticas transmitem sinais luminosos ao invés de sinais elétricos. Um cabo de fibra ótica consiste de um filamento de vidro ou plástico de aproximadamente 0,05mm de diâmetro, e um revestimento de vidro ou plástico de aproximadamente 0,12mm de diâmetro, para evitar perdas na transmissão, e uma camada externa de plástico ou de outro material de aproximadamente 0,22 mm de diâmetro (Nunes, 1989).

A velocidade de transmissão é da ordem de 500 Mbps, podendo chegar a 1 Gbps, livre de interferências eletromagnéticas. Seu custo é maior do que o dos cabos coaxiais, e é mais difícil a conexão de novas estações. A FIGURA 15 mostra a anatomia de uma fibra ótica.

FIGURA 15 – CABO DE FIBRA ÓTICA

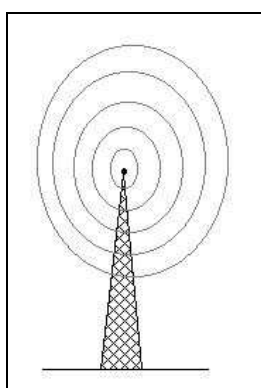


Fonte: Nunes (1989).

2.2.3.4 RADIODIFUSÃO

A radiodifusão utiliza frequências de rádio típicas, como AM e FM (tipos de modulação são melhor detalhados no próximo capítulo), e também ondas curtas. Este meio de transmissão não tem sido utilizado em grande escala na comunicação de dados. Para um pequeno número de estações, é um método eficiente; no entanto, quando este número aumenta, a ocorrência de colisões (interferência) também aumenta, diminuindo o desempenho. A FIGURA 16 mostra uma antena de rádio.

FIGURA 16 – ANTENA DE RÁDIO



Fonte: Tafner (1996).

2.2.3.5 MICROONDAS

Na transmissão por microondas os sinais são transmitidos em linha reta, em consequência disto, não pode haver qualquer tipo de obstáculo físico entre o transmissor e o receptor. Devido à curvatura da Terra, as estações de microondas deve estar distantes umas das outras a menos de 48 quilômetros.

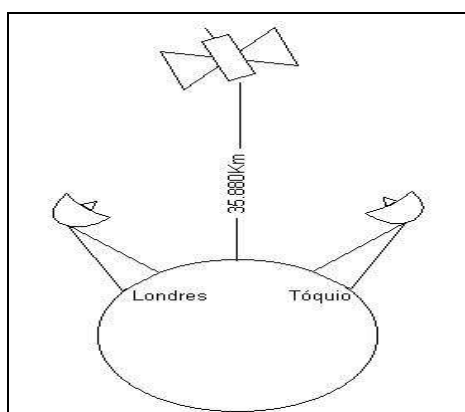
A transmissão por microondas pode chegar a taxa 45 Mbps, e o seu custo é razoável, sendo de fácil implementação. Por outro lado, apresenta problemas de interferência por outras ondas de rádio e é um tanto insegura, uma vez que as informações transmitidas podem ser interceptadas por qualquer receptor na linha de transmissão (Nunes, 1989).

2.2.3.6 SATÉLITE

Da mesma forma que a transmissão por microondas, a transmissão por satélite usa ondas de altíssima frequência. As diferenças básicas entre estas duas formas de comunicação

ser referem à localização das estações e ao método de transmissão do sinal. No caso de microondas, somente são usadas estações terrestres e em órbita. Em relação à transmissão, as estações devem estar ajustadas numa frequência para enviar e receber mensagens, uma vez que a transmissão de estações terrestres é feita em uma frequência e de estações em órbita, em frequência diferente. Sua velocidade é da ordem de 50 Mbps. A possibilidade de interceptação da transmissão por qualquer receptor é um dos problemas neste caso (Nunes, 1989). A FIGURA 17 ilustra a transmissão por satélite.

FIGURA 17 – TRANSMISSÃO POR SATÉLITE



Fonte: Nunes (1989).

2.2.4 MODOS DE TRANSMISSÃO

O modo de transmissão, na comunicação de dados, pode ser síncrono ou assíncrono. É esta a característica que determina como será “ritmada” a transmissão. Isto é, de acordo com o modo pelo qual são transmitidos, os dados recebem *bits* adicionais para controle.

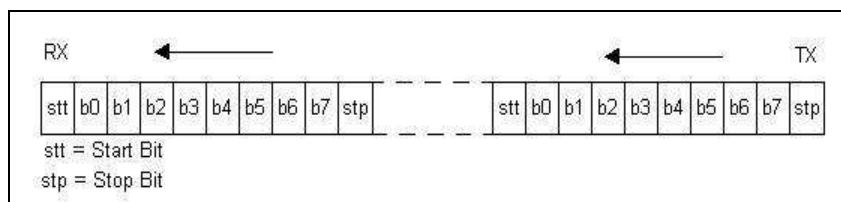
2.2.4.1 TRANSMISSÃO ASSÍNCRONA

A principal característica da transmissão assíncrona é a de poder ser iniciada em qualquer tempo, sem limite de tamanho de mensagem. Neste modo de transmissão, cada caracter recebe *bits* adicionais, que indicarão o início e o fim dos mesmos. Desta forma não há necessidade de uma pré-fixação de cadência entre o transmissor e o receptor.

Os *bits* adicionais são *start bit* e *stop bit*, início e fim, respectivamente. O transmissor e o receptor possuem *clocks* autônomos, sendo o *clock* do receptor sincronizado pelo *bit* de *start* recebido e devendo manter o sincronismo até o *bit* de *stop*. Uma ressalva: na transmissão

assíncrona o transmissor e o receptor devem ser inicializados com a mesma configuração, ou seja, a mesma velocidade, e o número de *bits* em cada *byte* (Tafner, 1996). A FIGURA 18 ilustra uma *string* de *bits* em forma assíncrona.

FIGURA 18 – BYTES EM TRANSMISSÃO ASSÍNCRONA

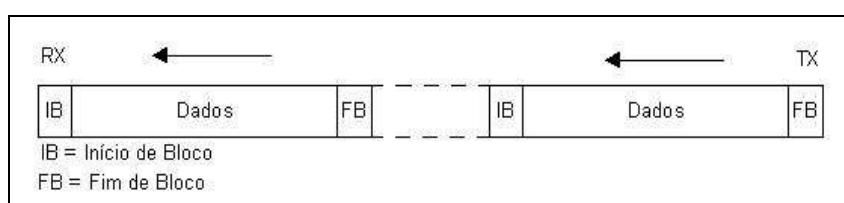


Fonte: Tafner (1996).

2.2.4.2 TRANSMISSÃO SÍNCRONA

Sua principal característica é a possibilidade de transmitir blocos de dados com a adição de controles apenas no começo e no fim deste bloco, isto é, o caractere de controle “IB – Início de Bloco” e “FB – Fim de Bloco”. Este modo de transmissão é possível, pois se baseia no estabelecimento de uma cadência (*clock*) fixa para a transmissão dos *bits*. Esta cadência deve ser compatibilizada pelos dois elementos no processo, o transmissor e o receptor. A FIGURA 19 ilustra uma transmissão serial síncrona.

FIGURA 19 – TRANSMISSÃO SÍNCRONA



Fonte: Tafner (1996).

A transmissão assíncrona exige que o *clock* de transmissão e o de recepção tenham exatamente a mesma frequência. O requisito pode ser satisfeito de duas maneiras:

- **Equipamentos próximos:** O transmissor e o receptor utilizam o mesmo *clock*, que é gerado por um deles, e transmitido ao outro através de uma linha auxiliar de sincronismo;
- **Equipamentos remotos (*modems*):** O receptor gera o seu *clock* com base no próprio sinal recebido.

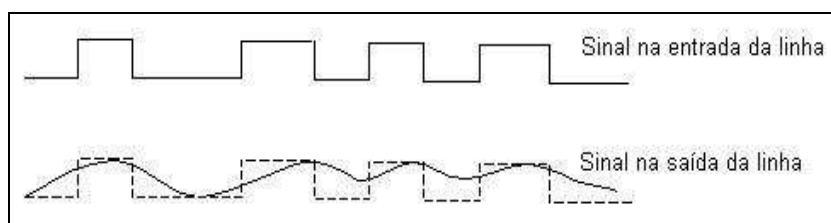
2.3 MODEM

Os meios de comunicação mais utilizados no momento são as linhas telefônicas, sistemas de rádio e microondas. Tais meios foram projetados para transmitir frequências da voz humana, sendo que estes sinais são analógicos, pois a voz é um sinal complexo e a sua energia está distribuída de um modo não uniforme em uma faixa de frequência de 15 Hz a 15000 Hz aproximadamente. Conforme visto no capítulo anterior, foi escolhida a faixa de voz entre 300 e 3400 Hz, para a construção das linhas telefônicas, o que garante 85% de inteligibilidade (palavras bem compreendidas numa conversação). A largura de banda do equipamento de telecomunicações é de 4000 Hz, sendo que as extremidades (0 a 300 Hz e 3400 a 4000 Hz) são utilizadas como banda de guarda para evitar a interferência entre os sinais que fluem em canais vizinhos (Silveira, 1991).

Se injetarmos em uma linha telefônica os sinais binários oriundos de um equipamento terminal de dados (ETD), obteremos na outra ponta um sinal distorcido, no qual as transições se mostram bem atenuadas, que poderá ser mal interpretado pelo equipamento de recepção, que terá eventualmente dificuldade de distinguir entre os níveis 0 e 1. A distorção será maior quanto mais estreita for a largura de banda da linha, pois as componentes de altíssima frequência serão brutalmente atenuadas ao passarem pela linha, sendo virtualmente inexistentes na saída (Silveira, 1991).

A FIGURA 20 mostra a distorção do sinal digital em uma linha telefônica:

FIGURA 20 – DISTORÇÃO DO SINAL DIGITAL



Fonte: Silveira (1991).

A princípio, poderia se pensar em aumentar a largura de banda das linhas telefônicas, entretanto, isto não é economicamente viável, já que as linhas instaladas atendem à finalidade para a qual foram projetadas.

A solução é adaptar o sinal digital aos tipos de degradação inerentes aos meios de transmissão. Para isto, foram desenvolvidos dispositivos capazes de transformar o sinal digital do computador em uma forma possível de ser transmitida pelo meio sem que ocorram danos graves. Estes dispositivos são chamados MODEMS (Silveira, 1991).

2.3.1 BREVE HISTÓRICO

O desenvolvimento do primeiro *modem* começou em 1919 e o primeiro *modem* da história é o *Telecom 110bps Tipo 4*. A partir deste modelo, os desenvolvedores de *modem* puderam aperfeiçoar a arquitetura e velocidade dos *modems* para satisfazer propósitos militares e comerciais. No final da década de 60, *modems* de 9600bps chegavam ao mercado, seguidos dos de 14.4Kbps e 19.2Kbps em 1980 (TCI, 1996).

Depois da ITU-T ter anunciado o padrão do protocolo de transmissão V.34, as velocidades dos *modems* aumentaram para 28.8Kbps e 33.6Kbps ou superiores. Atualmente, uma grande variedade de *modems* de alta velocidade está disponível no mercado de microcomputadores (TCI, 1996).

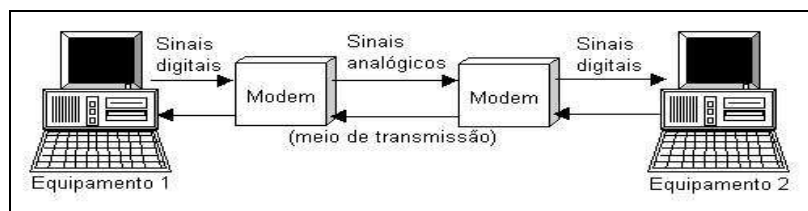
2.3.2 CONCEITO DE MODEM

O *modem*, cujo nome é formado pela contração das palavras modulador e demodulador, é um equipamento bidirecional que, instalado nas duas extremidades de um canal de comunicação de dados, tem por função adequar um sinal binário oriundo de um computador às características da linha (Silveira, 1991).

Este equipamento executa uma transformação, por modulação (*modem* analógico) ou codificação (*modem* digital), dos sinais digitais emitidos pelo computador, gerando sinais analógicos adequados à transmissão sobre uma linha telefônica. No destino, um equipamento igual a este demodula (*modem* analógico) ou decodifica (*modem* digital) a informação, entregando o sinal digital restaurado ao equipamento terminal a ele associado (Silveira, 1991).

A FIGURA 21 ilustra o comportamento de um *modem* genérico:

FIGURA 21 – USO DO MODEM



Fonte: Silveira (1991).

Na prática, os sinais, no seu formato digital normal, podem ser transmitidos por cabo comum a uma distância de no máximo 15 metros. Além deste limite, o índice de erros pode se tornar extremamente elevado, exigindo o uso de *modems* para resolver o problema (Silveira, 1991).

2.3.3 CLASSIFICAÇÕES DE MODEM

Segundo TCI (1996), os *modems* podem ser classificados da seguinte maneira:

- a) De acordo com a velocidade de transmissão
 - **Modem de baixa velocidade:** suporta uma velocidade de transmissão inferior a 2400bps;
 - **Modem de média velocidade:** suporta uma velocidade de transmissão entre 4600bps e 9600bps;
 - **Modem de alta velocidade:** suporta uma velocidade de transmissão superior a 14.4Kbps;
- b) De acordo com o meio de transmissão
 - **Modem de canal de voz:** utiliza o par trançado para conexão e é instalado nas linhas telefônicas tradicionais;
 - **Modem de fibra óptica:** transforma sinais digitais em pulsos de luz e usa cabos de fibra óptica para conexão;
 - **Modem de rádio frequência:** transmite dados através de ondas VHF. É mais suscetível a ruído e o alcance de sua transmissão é limitado.
- c) De acordo com a conexão no computador:
 - **Modem interno:** também chamado de placa *modem*. Esta placa é instalada no sistema inserindo-a num *slot* PCI;
 - **Modem externo:** é um dispositivo independente que é conectado ao PC através de uma porta serial ou paralela. Requer fornecimento de energia separado;
 - **Modem onboard:** refere-se a um *chip* de *modem* integrado à placa mãe.

Além das classificações acima, os *modems* se dividem ainda, em duas classes: *modems* analógicos, que realizam modulação e demodulação de uma onda portadora; e *modems* digitais (ou banda-base), que realizam codificação e decodificação de um sinal digital.

2.3.3.1 MODEMS ANALÓGICOS

São equipamentos que realizam o processo de modulação e demodulação para que os sinais digitais possam trafegar pelo meio telefônico (Silveira, 1991).

Modulação é um processo pelo qual são modificadas uma ou mais características de uma onda denominada portadora, segundo um sinal modulante (informação que se deseja transportar pelo meio; no caso de comunicação de dados, é o sinal digital binário). A modulação pode ser feita variando amplitude, frequência ou fase da onda portadora, isoladamente ou conjuntamente. A informação impõe o modo como vai ser modificado a portadora. Ao se analisar, na recepção, as modificações sofridas pela portadora, pode-se recuperar a informação digital. É por isso que se diz que a portadora transporta a informação (Silveira, 1991).

De acordo com Montoro (1995), os métodos de modulação mais comuns são:

- Modulação em amplitude – AM;
- Modulação em frequência – FM;
- Modulação em fase – PM.

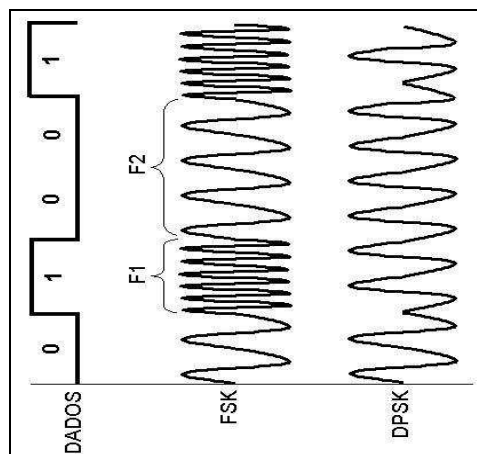
Outros três métodos mais específicos para a transmissão de dados, são:

- Deslocamento de frequência – FSK;
- Deslocamento diferencial de fase – DPSK;
- Deslocamento de fase e amplitude – QAM.

Modular em FSK é alterar a frequência da portadora para F_1 (ver FIGURA 22) quando o *bit* é de dados for “1” e para F_2 (ver FIGURA 22) quando ele for “0”. Modular em DPSK é alterar a fase da portadora, em graus diferentes, conforme o *bit* de dados seja “0” ou “1”, mantendo a amplitude constante. Modular em QAM é alterar a frequência, a fase e a amplitude da portadora. QAM não representa *bit* a *bit* na portadora, mas sim *strings* de *bits*, que podem ser de 4 *bits* por exemplo. É importante ressaltar que este é o método de

modulação mais utilizado nos *modems* atuais. A FIGURA 22 ilustra a diferença entre as modulações FSK e DPSK.

FIGURA 22 – MODULAÇÕES FSK E DPSK



Adaptado de: Montoro (1995).

2.3.3.2 MODEMS DIGITAIS

São equipamentos que realizam uma codificação no sinal digital visando adequá-lo à transmissão em uma linha física. A codificação é uma mudança na representação do sinal digital oriundo do ETD em um outro sinal mais adequado às condições da linha (Silveira, 1991).

Diz-se que um sinal é banda-base quando seu espectro de frequência não sofre translação, ou seja, ele não está modulando nenhuma portadora (Montoro, 1995).

Rigorosamente, esse tipo de equipamento não deveria ser chamado de *modem*, uma vez que não realiza modulação / demodulação do sinal digital. Os *modems* digitais são também chamados de MODEM DE BANDA BASE ou DATA SET (Silveira, 1991).

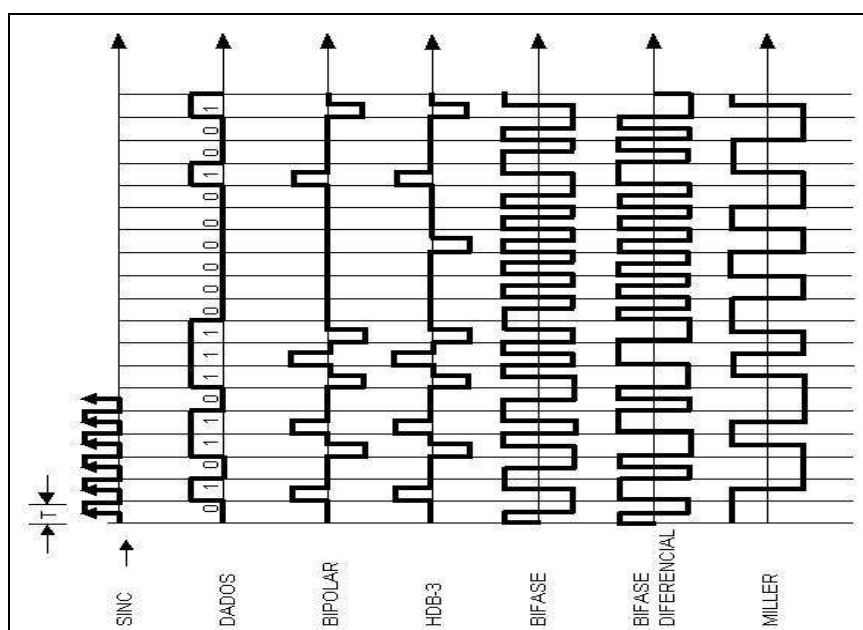
Considerando que a faixa de frequência disponível nos meios de transmissão geralmente é limitada, esse sinal digital codificado sofre bastante distorção ao se propagar pelo meio. Isso obriga ao uso desses *modems* digitais apenas em distâncias curtas (alguns quilômetros) e com linhas de boa qualidade (Silveira, 1991).

Para Montoro (1995), os *modems* digitais codificam o sinal de dados a fim de alcançar os seguintes objetivos:

- Concentrar o espectro de transmissão dentro de uma faixa de frequência que possua pouca atenuação;
- O sinal codificado deve conter boa informação de sincronismo a fim de facilitar sua recuperação no *modem* receptor;
- O sinal codificado deve ter boa imunidade ao ruído;
- A complexidade deve ser a menor possível, a fim de aumentar a confiabilidade e diminuir o custo.

A FIGURA 23, mostra um comparativo entre as diversas codificações dos *modems* digitais.

FIGURA 23 – CODIFICAÇÕES EM BANDA-BASE



Fonte: Montoro (1995).

2.3.4 FUNCIONAMENTO DO MODEM

Pode-se conectar um *modem* a um PC através da porta serial RS-232, conforme descrito mais adiante. Entretanto, devido a certas limitações de *hardware* no ETD, os *modems* de portas seriais não suportam altas velocidades de transmissão. Por isso, foram desenvolvidos *modems* para porta paralela do micro, afim de melhorar a velocidade de

transmissão do ETD para mais de 57600bps e 115200bps. Os *modems* pioneiros para porta paralela foram da AT&T e da ZyXEL (TCI, 1996).

Modems permitem que se transmita ambos sinais analógicos e digitais via linha telefônica, transformando sinais digitais em analógicos (modulador), ou vice versa (demodulador). Durante a transmissão de dados, esses dispositivos podem detectar e corrigir erros de dados através dos protocolos de correção de erros MNP1 à MNP4 e V42, assim como fazer compressão de dados através dos protocolos de compressão MNP5 e V42bis (TCI, 1996).

A funcionalidade de compressão de dados reduz o tempo de transmissão e seu custo. Por exemplo: usando o protocolo V.42bis com uma taxa de compressão de 1:4, o *modem* pode reduzir da velocidade de 57.6Kbps do ETD para 14.4Kbps (TCI, 1996).

2.3.5 MODEMS “TRÊS-EM-UM”

Outras funções como *fax* e voz também podem ser integradas ao *modem*. Este tipo de *modem* é chamado de *modem* “três-em-um” (TCI, 1996).

A função *fax* do *modem* “três-em-um” suporta uma velocidade maior do que se comparada a uma máquina de *fax* tradicional. Essa característica é possível graças a ITU-T V.17 e V.29, e os protocolos da EIA *Fax* Classe 1, Classe 2 e Classe 2.0 (TCI, 1996).

O protocolo *Voice View* que permite transmissão simultânea de voz e dados em uma única linha foi introduzido pela *Radish Communications System, Inc.* *Modems* transformam o sinal de voz em sinal digital, enviam o sinal digital ao PC e o salvam em arquivos. Através de técnicas de compressão de voz tal como ADPCM, o tamanho do arquivo de voz recebido pode ser reduzido para ½ ou 1/3 do tamanho do arquivo original (TCI, 1996).

O arquivo de voz gravado pode ser tocado pelo usuário usando aplicações de comunicação, que dependem do conjunto de comandos suportados pelo *modem*. Essas aplicações permitem que se use o conjunto de comandos *Voice AT* (estes comandos são abordados no próximo capítulo), que são usados para a transmissão de voz. Diferentemente dos comandos de *fax* e dados, os comandos *Voice AT* variam para cada aplicação (TCI, 1996).

2.3.6 INTERFACE PADRÃO RS-232

A interface RS-232, elaborada pela EIA (*Electronic Industries Association*), estabelece, atualmente, uma norma para o interfaceamento serial entre dispositivos. Ao longo dos anos, conforme a norma RS-232 sofria revisão, ganhava uma letra adicional em seu nome, seguindo a seqüência alfabética. A revisão “C”, ou seja, RS-232C, é a que ficou de fato conhecida. Em janeiro de 1987, a norma sofreu mais uma revisão e passou a ser denominada de EIA-232D. Mas apesar das mudanças, a norma EIA-232D continua sendo chamada de RS-232 (Montoro, 1995). Ela define:

- características elétricas dos sinais;
- características mecânicas dos sinais;
- características lógicas (pinagem do conector);
- características funcionais (função de cada sinal).

No Brasil, os *modems* devem atender ao Padrão Telebrás 225-540-730 de 1986, baseado nas normas EIA-232C, CCITT V28 e CCITT V24. Na FIGURA 24, está demonstrado o papel da interface RS-232.

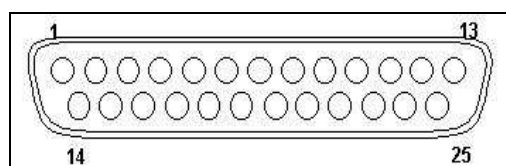
FIGURA 24 – O PAPEL DA INTERFACE RS-232C



Fonte: Tafner (1996).

Os sinais trocados entre o ETD e o *modem* estão especificados em um conector de 25 ou 9 pinos tipo “D” (DB-25P e DB-9P). A conexão mecânica é padronizada pela ISO (*International Standard Organization*), através da norma ISO 2593, e realizada através de um conector-padrão denominado de DB-25P (Tafner, 1996). A FIGURA 25 demonstra a pinagem de um conector DB-25P macho.

FIGURA 25 – CONECTOR PADRÃO DB-25P MACHO



Fonte: Tafner (1996).

Segundo Tafner (1996), uma comunicação de dados, entre o computador e o *modem*, pode ser dividida em 4 fases distintas:

- a) **Estabelecimento:** essa fase representa a continuidade entre os dois pontos, ocorrendo apenas para circuito não dedicado;
- b) **Inicialização:** é nesta fase que os sinais de controle são trocados entre as partes envolvidas para estabelecer a comunicação. Esses sinais de controle têm o objetivo de preparar, coordenar e sincronizar as partes envolvidas na transmissão / recepção;
- c) **Transmissão:** representa, efetivamente, a transmissão e recepção dos dados após estabelecida a fase de inicialização;
- d) **Corte:** é o “desligamento” do circuito por iniciativa de um dos lados para uma ligação *full-duplex*, ou então inversão do sentido para o caso de ligação *half-duplex*.

2.3.6.1 DESCRIÇÃO DOS SINAIS DA INTERFACE

A norma EIA-232D tem uma denominação para cada sinal da interface, e a recomendação CCITT V24 tem outra, mas na prática, os sinais são conhecidos por siglas mnemônicas. Todos os sinais tem seus níveis referenciados ao pino 7 (no caso do conector DB-25P), que corresponde a 0 Volt. Dos 25 sinais (pinos) especificados no padrão RS-232, os mais utilizados estão descritos na TABELA 1:

TABELA 1 – PRINCIPAIS SINAIS DA RS-232

Categoria	Pino conector	Circuito (CCITT)	Denominação CCITT-V.24	Origem	Função dos sinais
-	01	101	Protective Ground (PG)	-	Terra de proteção (não faz parte do sistema de sinalização)
	07	102	Signal Ground (SG)	-	Este é o sinal referencial de zero volts. Todos os outros sinais são positivos ou negativos em relação a este sinal.
Dados	02	103	Transmitted Data (TXD)	Terminal	Este é o fio por onde os dados são transmitidos. Quando não há dados sendo transmitidos, o circuito fica em condição de “1” lógico.
	03	104	Received Data (RXD)	modem	Este é o fio por onde os dados são recebidos. Quando não há dados sendo transmitidos, o circuito fica em condição de “1” lógico.
Controle	04	105	Request To Send (RTS)	terminal	O terminal faz solicitação ao modem para transmitir. O modem responde com um CTS se estiver ok.
	05	106	Clear To Send (CTS)	modem	O modem responde para um RTS do terminal dizendo que está ok para enviar dados pela linha de dados.
	06	107	Data Set Ready (DTR)	modem	Este sinal ligado sinaliza ao terminal que o modem está conectado ao canal de comunicação e pronto para transmitir.
	08	109	Data Carrier Detected (DCD) ou Received Line Signal Detector (RLSD)	modem	Este sinal do modem informa ao terminal que recebeu algum sinal de dado (portadora).
	11	126	(MOD)	terminal	Seleção do modo.
	20	108	Data Terminal Ready (DTR)	terminal	Este sinal do terminal indica ao modem que o equipamento está pronto para comunicação.
	22	125	Ring Indicator (RING)	modem	Este sinal indica ao terminal que o modem está recebendo um sinal de tom (sinal de chamada telefônica).
	23	111	(VEL)	terminal	Seleção de velocidade.

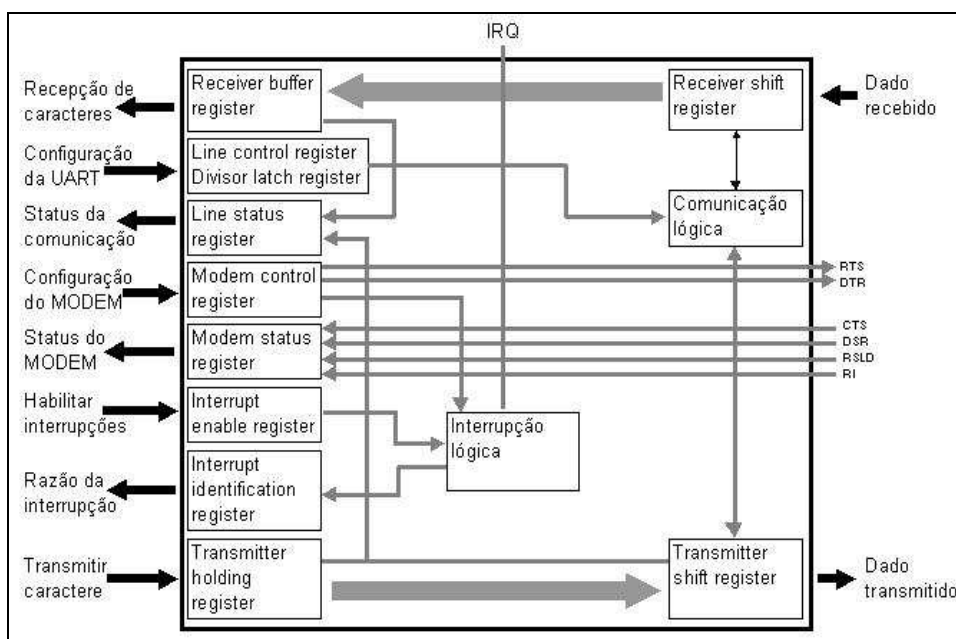
Adaptado de: Montoro (1995) e Tafner (1996).

2.3.7 UART

A linha de computadores IBM-PC, utiliza em sua arquitetura, o *chip* 16550 para a realização da comunicação. Este *chip*, também chamado de UART (*Universal Asynchronous Receiver Transmitter*), é usado para a conversão do protocolo serial (Tafner, 1996).

O dispositivo, associado a uma porta serial, recebe os dados do barramento do micro, na forma paralela (8 *bits* de cada vez). Um circuito converte de paralelo para serial e, no caso do *modem* interno, transmite os dados pela linha telefônica (Montoro, 1995). A FIGURA 26, ilustra o funcionamento dos registradores da UART.

FIGURA 26 – DIAGRAMA DA UART

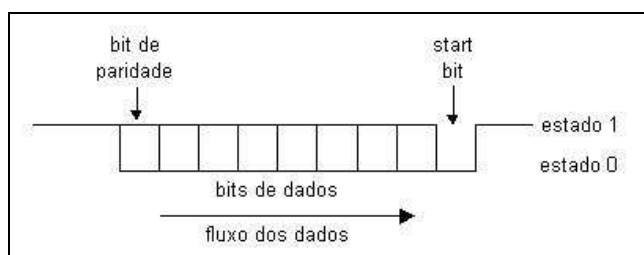


Fonte: Tafner (1996).

A UART poupa ao programa a tarefa de realizar, através do processador, a transmissão e a recepção de caracteres (*bytes*) pelo canal serial. Para receber e enviar dados, o programa, simplesmente, lê e escreve *bytes* na UART, que se mostra aos “olhos” do processador, apenas como uma região de memória, ou portas de I/O (endereços: 2F8, 3F8, etc).

Esse *chip* utiliza, a nível de *bit*, o protocolo serial assíncrono para a comunicação. O principal conceito deste protocolo é que todos os dados e informações de controle, necessários para transmitir e receber um caractere de informação, precisam ser movidos sobre uma linha simples, um *bit* de cada vez. (Tafner, 1996). A FIGURA 27 demonstra o formato serial assíncrono.

FIGURA 27 – FORMATO SERIAL ASSÍNCRONO



Fonte: Tafner (1996).

Quando não estão sendo transmitidos dados sobre a linha, o estado lógico da linha é 1, ou *marking state* (*mark*). Quando é transmitido um caractere de dado, o primeiro *bit* a ser transmitido é o *start bit*. O *start bit* é representado pelo estado lógico 0 da linha, ou *spacing state* (*space*). O receptor sabe que está esperando um caractere de dado quando o estado da linha muda de *mark* para *space*, de 1 para 0.

Os *bits* de dados são transmitidos imediatamente após o *start bit*. O número de *bits* de dado pode variar de 5 a 8 *bits*. Após os *bits* de dados, segue, opcionalmente, o *bit* de paridade. Este *bit* deve ser constante durante toda a transmissão. Por exemplo, se a paridade escolhida for par, então o número de *bits* lógicos de dados precisa ser par até o final da transmissão. Este tipo de paridade pode permitir ao receptor detectar possíveis erros que venham a ocorrer durante a transmissão dos dados, porém não possui 100% de confiabilidade.

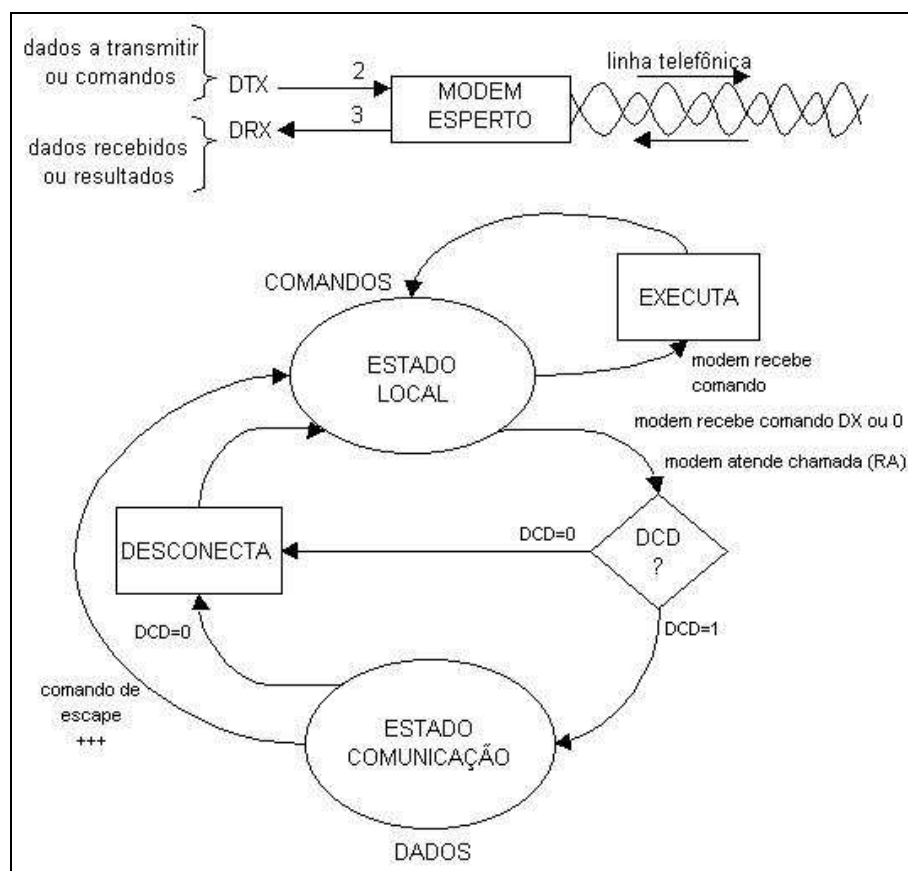
Após o *bit* de paridade, segue 1, 1.5 ou 2 *bits* de *mark* (1 lógico). Estes *bits* são chamados de *stop bits*. O *stop bit* representa o tempo mínimo que a linha precisa para voltar à condição de *mark* antes do próximo *space* aparecer. O número de *stop bits* também precisa ser constante durante toda a transmissão (Tafner, 1996).

2.4 PROTOCOLO HAYES

Uma das funções do *modem* é transmitir e receber dados digitais seriais do usuário, através dos pinos 2 (transmissão – DTX) e 3 (recepção – DRX) da interface, respectivamente.

No entanto, os *modems* “espertos”, ou mais conhecidos como *smartmodems*, aceitam, também, comandos especiais pelo pino 2 e devolvem mensagens de resultado pelo pino 3. O *Hayes* é um protocolo de *smartmodem*, desenvolvido pela *Hayes Microcomputer Products Inc*, da Geórgia, nos Estados Unidos, que possui um repertório de comandos e um conjunto de regras que o define (Montoro, 1995). A FIGURA 28, mostra um diagrama dos estados do *smartmodem*.

FIGURA 28 – DIAGRAMA DE ESTADOS DO SMARTMODEM



Fonte: Montoro (1995).

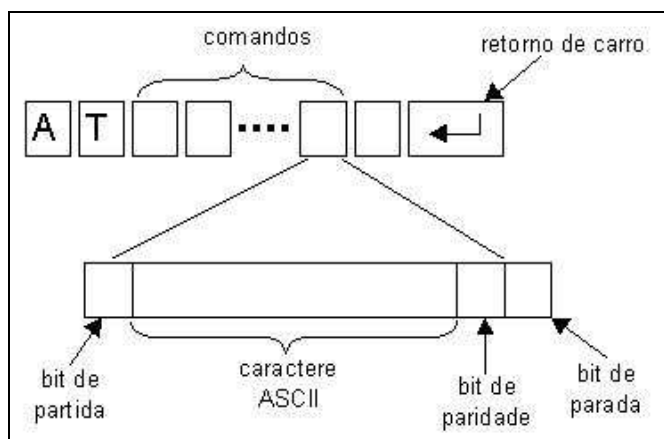
A seguir será abordadas a sintaxe dos comandos *Hayes*, os estados de operação do *modem*, alguns de seus registradores e os comandos *Voice AT* utilizados no protótipo.

2.4.1 SINTAXE DOS COMANDOS

Os comandos do protocolo *Hayes* devem ser enviados ao *modem* pelo pino 2 da interface, podendo ser, por exemplo, através de caracteres de 10 *bits*, sendo 1 *start bit*, 7 de dados, 1 de paridade e 1 *stop bit*. Os *bits* de dados devem seguir o código ASCII. Os comandos são enviados ao *modem* em linhas que podem conter mais de um comando. Toda linha de comandos deve iniciar com os caracteres AT e terminar com um retorno de carro (“Return” ou “Enter”, código ASCII 13), de acordo com Montoro (1995).

A FIGURA 29 ilustra a composição de um comando do protocolo *Hayes*.

FIGURA 29 – COMANDO HAYES



Fonte: Montoro (1995).

2.4.2 ESTADOS DE OPERAÇÃO

O *modem* sabe se as informações no pino 2 são dados a transmitir ou comandos a executar, pelo estado de operação em que ele se encontra. Os *smartmodems* possuem dois estados de operação:

- **Estado LOCAL:** O *modem* supõe que as informações no pino 2 são comandos a executar;
- **Estado COMUNICAÇÃO:** O *modem* supõe que as informações no pino 2 são dados e portanto, as transmite ao *modem* remoto.

Quando o *modem* é ligado, ele entra no estado LOCAL e fica pronto para aceitar comandos. O repertório de comandos varia conforme o fabricante do *modem*. A TABELA 2 mostra os mais comuns (Montoro, 1995).

TABELA 2 – COMANDOS BÁSICOS DO PROTOCOLO HAYES

COMANDO	DESCRIÇÃO DA FUNÇÃO
ATBO	Operar no modo CCITT V22
ATB1	Operar no modo BELL 212A
ATDn	Disca o número 'n' especificado. Antes de 'n' é inserido 'P' para discagem por pulso, ou 'T' para discagem por tom multifrequencial
ATH	Desconectar
ATMO	Desligar o alto-falante
ATM1	Ligar o alto-falante após a discagem
ATNn?	Ler posição "n" da agenda de telefones
ATO	Passar para o estado COMUNICAÇÃO
ATSn=x	Carregar o registrador "n" com o valor "x"
ATSn?	Ler o registrador "n"
AT&F	Carregar a configuração de fábrica
AT&Zn=x	Carregar a posição "n" da agenda com "x"
+++	Escape para estado LOCAL

Fonte: Montoro (1995).

O *modem* passa para o estado COMUNICAÇÃO se receber um dos comandos do descritos no QUADRO 1, abaixo:

QUADRO 1 – COMANDOS PARA O ESTADO “COMUNICAÇÃO”

<p>Dx : O modem deve se posicionar no modo origem, ocupar a linha telefônica, aguardar um certo tempo e discar o número telefônico. O parâmetro "x" é uma cadeia de algarismos que define o número telefônico. Alguns modems aceitam outros caracteres com funções específicas, neste campo, por exemplo:</p> <p>W : O modem deve aguardar o tom de discar</p> <p>P : O modem deve fazer a discagem por pulsos</p> <p>T : O modem deve fazer a discagem por tons</p> <p>Nn : O modem deve discar o número armazenado na posição "n" de sua agenda interna.</p>
--

Fonte: Montoro (1995).

Após a discagem o *modem* aguarda a portadora do remoto (DCD=1) e passa para o estado COMUNICAÇÃO, conforme demonstrado no QUADRO 2. Se o *modem* não receber a portadora, ele se desconecta e continua no estado LOCAL (Montoro, 1995).

QUADRO 2 – DETECÇÃO DE PORTADORA

O:	O modem passa para o estado COMUNICAÇÃO (<i>on-line</i>) se estiver recebendo portadora (DCD=1).
----	--

Fonte: Montoro (1995).

O *modem* pode passar para o estado COMUNICAÇÃO, também, se estiver programado para resposta automática e receber uma chamada. Ele pode sair do estado COMUNICAÇÃO em duas condições, conforme descrito no QUADRO 3:

QUADRO 3 – CONDIÇÕES PARA O MODEM VOLTAR AO ESTADO LOCAL

DCD=0:	Se a portadora do modem remoto cair
--------	-------------------------------------

+++ :	Se receber um comando de escape. Este é o único comando que o modem aceita quando está no estado COMUNICAÇÃO.
-------	---

Fonte: Montoro (1995).

No protocolo *Hayes*, o comando de *escape* consiste no *modem* receber três caracteres “+” precedidos e seguidos por um tempo de silêncio de um segundo. Note que, se o *modem* estiver no estado COMUNICAÇÃO, o usuário pode passá-lo ao estado LOCAL, com um comando de *escape*, enviar outros comandos, e voltar ao estado COMUNICAÇÃO enviando o comando ATO, tudo sem perder a conexão (Montoro, 1995).

2.4.3 REGISTRADORES DO MODEM

Registradores são posições de memória internas do *modem* que armazenam parâmetros regraváveis através de comandos. Basicamente, os comandos servem para programar parâmetros internos do *modem* e providenciar conexão com uma estação remota. Os parâmetros do *modem*, que podem ser alterados pelos comandos *Hayes*, ficam armazenados em registradores internos denominados registradores “S”, numa EEPROM. Então, para alterar um parâmetro, basta alterar o conteúdo do registrador correspondente (Montoro, 1995). Na TABELA 3 estão alguns registradores como exemplo:

TABELA 3 – REGISTRADORES “S”

REGISTRADOR	DESCRIÇÃO DA FUNÇÃO
S0	Quantidade de toques para o modem atender uma chamada telefônica
S2	Valor do código ASCII correspondente ao caractere de escape. O modem vem de fábrica com 43, em decimal, que corresponde ao “+”
S3	Valor do código ASCII correspondente ao caractere de retorno de carro. O modem vem de fábrica com 13, em decimal, que corresponde ao “Return”
S4	Avanço de linha (semelhante a S2 e S3)
S5	Retrocesso (semelhante a S2 e S3)
S6	Define o tempo, em segundos, que o modem deve esperar, antes de iniciar a discagem e após ocupar a linha.
S7	Define o tempo, em segundos, que o modem deve esperar pela portadora do remoto após uma discagem.

Fonte: Montoro (1995).

2.4.4 COMANDOS VOICE AT

Neste item, encontram-se os principais comandos *Voice AT*, destinados a configurar, inicializar, discar, tocar ou gravar arquivos de voz e reconhecer tons de discagem (DTMF); de *modems* que tem suporte a tais comandos.

Os comandos vistos aqui destinam-se especificamente ao *modem US Robotics Sporster Voice 33600* (U.S. Robotics, 1996), que é o modelo de *modem* utilizado para a implementação do protótipo deste trabalho. Porém, nada impede que os mesmos comandos sirvam para manipular *modems* compatíveis com o modelo mencionado. Apenas para salientar, conforme visto no capítulo anterior, os comandos *Voice AT* não são padronizados, segundo TCI (1996).

A TABELA 4 descreve os principais comandos *Voice AT*, e suas funções.

TABELA 4 – DESCRIÇÃO DOS COMANDOS VOICE AT

Descrição da função	Comandos	
Selecionar o protocolo e o método de compressão com que o modem irá trabalhar.	Comando:	AT#VSM=
	Considerações:	Valores Método compressão Taxa 128,8000 – GSM USR - 19200 bps 129,8000 – G721 ADPCM USR - 38400 bps 130,8000 – IMA ADPCM USR - 38400 bps Obs.: o modem US Robotics Sporster Voice 33600 opera corretamente se for utilizado: AT#VSM=128,8000.
String para inicializar o modem no modo Voice.	Comando:	AT&F1 M1 L3 S40=2 #CLS=8 #VSM=129,8000 #VLS=2
	Considerações:	&F – resseta os parâmetros do modem M1 – liga o autofalante L3 – ajusta o volume S40=2 – define que as tentativas de discagem serão duas #CLS=8 – seta o modem para Voice #VSM=129,8000 – toca arquivos no método de compressão G721 #VLS=2 – envia a voz ao telefone conectado ao modem Obs.: Após este comando o modem aguarda por: ‘OK’
String de discagem.	Comando:	ATDT3456789
	Considerações:	Com comando o modem disca para o número definido, no exemplo: 3456789 Obs.: Após este comando o modem aguarda por: ‘OK’
String para conectar o modem no modo Voice.	Comando:	AT#CLS=8 #VSM=129,8000 #VLS=2
	Considerações:	#CLS=8 – seta o modem para Voice #VSM=129,8000 – toca arquivos no método de compressão G721 #VLS=2 – envia a voz ao telefone conectado ao modem Obs.: Após este comando o modem aguarda pela string: ‘VCON’. Ao receber ‘VCON’, o modem já está apto a reconhecer DTMF, e além disso ele pode gravar ou tocar arquivos de voz.
Tocar um arquivo de voz. * - Normalmente este arquivo contém a extensão WAV, porém isso depende dos padrões reconhecidos pelo modem. O formato WAV carrega todas as características da onda sonora, sem perda de qualidade.	Comando:	AT#VTX <Enter> Arq_exemplo_play.wav
	Considerações:	Estando no status VCON, pode-se passar o comando #VTX , e o modem aguardará por ‘CONNECT’. A partir daí, cada caractere digitado é considerado parte do nome do arquivo de voz que será tocado, no exemplo: arq_exemplo_play.wav Somente quando receber mais um <Enter>, o modem passa a tocar o arquivo. Ao terminar de tocar o arquivo, o modem pode receber mais nomes de arquivos, ou receber os caracteres <DLE> <ETX>, e voltar para o status ‘VCON’. Obs.: a extensão do arquivo dependerá do método selecionado em #VSM=?.
Gravar um arquivo de voz.	Comando:	AT#VRX <Enter> Arq_exemplo_gravacao.wav
	Considerações:	Estando no status VCON, pode-se passar o comando #VRX , e o modem aguardará por ‘CONNECT’. A partir daí cada caractere digitado é considerado parte do nome do arquivo onde a voz será gravada, no exemplo: arq_exemplo_gravacao.wav Somente quando receber mais um <Enter>, o modem passa a gravar os sons no arquivo. Para terminar a gravação, deve ser passado o caractere <DLE> para o modem, e ele retornará ao status ‘VCON’. Obs.: a extensão do arquivo dependerá do método selecionado em #VSM=?.
String de desconexão.	Comando:	ATH0
	Considerações:	Esta continua a mesma, tanto para um modem normal, quanto um modem Voice. Após dado o comando, o modem espera por: ‘OK’.

Adaptado de: Giorgetti (1998).

3 INTEGRAÇÃO ENTRE O COMPUTADOR E O TELEFONE

A tecnologia de integração entre o computador e o telefone, ou *Computer Telephony Integration* (CTI), em inglês, compreende uma gama de padrões e ferramentas. Dentre os quais estão o TAPI, padrão tecnológico para integração entre computador e telefonia, e a biblioteca de componentes ASYNC PRO 32, ferramenta utilizada para construção de aplicações CTI. Primeiramente, será conceituada a tecnologia CTI, para então serem abordados o padrão TAPI e a biblioteca ASYNC PRO 32.

3.1 TECNOLOGIA CTI

Em Softpac (1999), tem-se que CTI, na essência, consiste em incrementar o profissionalismo em atender clientes, ter mais conhecimento ao receber suas ligações e reduzir custos. Fazer com que o telefone e o computador trabalhem juntos para sincronizar ligações telefônicas através de aplicações de computador. Com CTI, o computador pode fazer as ligações externas e ajudar a atender as chamadas recebidas com mais eficiência, enquanto provê informações necessárias a melhorar as relações com o cliente e os serviços.

A FIGURA 30 ilustra um terminal de computador equipado com dispositivos de CTI.

FIGURA 30 – EQUIPAMENTO DE CTI



Fonte: Inter-tel (1998).

Segundo Inter-tel (1998), CTI não é uma tecnologia nova. No início da década de 80, estava reservada à apenas grandes centrais de atendimento corporativas através de equipamentos proprietários, de alto custo e *softwares* “feitos em casa”.

3.1.1 TIPOS DE APLICAÇÕES DE CTI

CTI não é um produto específico, mas uma tecnologia usada para conectar dois instrumentos essenciais de trabalho, o telefone e o computador. Facilitando inteligentes interações entre o computador e os dispositivos de telefonia, CTI possibilita um grande número de aplicações, como: Acesso aos Registros do Cliente ou “*Screen Pop*”, Priorização Automática de Chamados, Caixa de Mensagens Integrada, Resposta Interativa de Voz, e Roteamento Avançado de Chamadas (Inter-tel, 1998).

3.1.1.1 SCREEN POP

Uma das habilidades de CTI é automaticamente buscar informações relativas ao cliente que está ligando, de um banco de dados com os registros do mesmo. Utilizando um identificador de chamadas, as informações sobre o cliente aparecem na tela (que é o chamado “screen pop”) antes mesmo de se atender o telefone (Inter-tel, 1998).

3.1.1.2 PRIORIZAÇÃO AUTOMÁTICA DE CHAMADOS

A tecnologia CTI também pode implementar priorização automática de chamados. Através dos produtos de telefonia, pode-se implementar no sistema a facilidade de automaticamente priorizar chamados recebidos, com base nos parâmetros estabelecidos. Pode-se, por exemplo, usar esta funcionalidade para programar o sistema a aceitar uma ligação, ou encaminhá-la para a caixa de mensagens, dependendo do número do telefone de quem está ligando (Inter-tel, 1998).

3.1.1.3 CAIXA DE MENSAGENS INTEGRADA

Outro benefício da tecnologia CTI, permite acesso instantâneo às informações de voz e dados, e rápida transmissão de correspondências críticas. Pode-se acessar o correio eletrônico, correio de voz e faxes diretamente de um PC ou através de uma ligação telefônica. Faxes e correios eletrônicos são convertidos para voz utilizando-se a tecnologia *text-to-speech* (síntese de voz), segundo Inter-tel (1998).

3.1.1.4 RESPOSTA INTERATIVA DE VOZ

Aplicações que usam resposta interativa de voz são úteis em centrais de atendimento e outros serviços de atendimento ao consumidor. Através de resposta interativa de voz, pode-se, por exemplo, automatizar uma central de atendimento bancária. Neste caso o sistema interage com o cliente que fez a chamada solicitando uma identificação, que pode ser o número de sua conta, e depois oferece menus audíveis dos serviços disponíveis. Fazendo com que o cliente passe antes pelo “atendente automático”, as organizações reduzem o tempo de cada ligação, e mantêm o foco em chamadas que necessitam de uma intervenção humana. Além disso, acesso automatizado à informação 24 horas por dia, é mais conveniente ao cliente e aumenta a sua satisfação (Inter-tel, 1998).

3.1.1.5 ROTEAMENTO AVANÇADO DE CHAMADAS

A tecnologia CTI possibilita também o roteamento avançado de chamadas. Pode-se programar o sistema para rotear chamadas baseado numa variedade de especificações. Por exemplo, o chamado de certo cliente pode ser roteado a um vendedor específico, um consultor ou ao supervisor (Inter-tel, 1998).

3.1.2 REQUISITOS PARA CTI

O primeiro e mais importante passo para utilizar a tecnologia CTI é ter uma plataforma de telefonia “aberta”. OAI (*Open Architecture Interfaces*) provê as conexões necessárias para ligar os sistemas telefônico e de computador. OAI possibilita também que desenvolvedores independentes de *software* criem produtos aplicativos com incrementada capacidade de telecomunicações.

O segundo importante passo para CTI é uma plataforma de telecomunicações que adere aos padrões da indústria de CTI (Inter-tel, 1998).

O que o sistema operacional *Microsoft Windows* fez pelas indústrias de impressoras, por exemplo, os padrões de CTI estão fazendo pela indústria de telecomunicações. Com aplicações para o sistema operacional MS-DOS, os desenvolvedores tinham que criar seus *softwares* de interface (*drivers*) para cada tipo diferente de impressora. Consequentemente, os desenvolvedores tinham dificuldade criando novas aplicações que funcionassem com todos os

tipos de impressora. O *Microsoft Windows* definiu um padrão na indústria, permitindo que uma aplicação trabalhe com todos os tipos de impressora. Seguindo o mesmo princípio, plataformas de comunicação que suportam CTI executarão qualquer *software* escrito para os padrões CTI (Inter-tel, 1998).

Para os negócios, tais padrões abrem as portas para uma gama de soluções de telecomunicações, que precisa ser escrita apenas uma vez e depois pode ser reaproveitada.

No desenvolvimento de interfaces padronizadas de CTI, existem duas facções. A *Novell* se aproximando mais do conceito de “conexão em redes”, ao ligar o PABX ao servidor de telecomunicações. E a *Microsoft* e *Intel*, que mantiveram primeiramente seu foco nas aplicações para pequenas empresas, as chamadas de “baseadas em *desktop*”. Mas também já oferecem sua solução para grandes organizações, seguindo o conceito de “conexão em redes” (Inter-tel, 1998).

A *Novell* criou o padrão TSAPI (*Telephony Server Application Programming Interface*), que possibilita aos usuários de redes *Novell* acessar os serviços do servidor de telecomunicações através da rede local.

A *Microsoft* e *Intel* co-desenvolveram o padrão TAPI (*Telephony Application Programming Interface*), que utiliza o popular formato das aplicações do sistema operacional *Microsoft Windows*. Este padrão está mais detalhado no próximo capítulo.

3.2 TAPI

Para melhor conceituar TAPI (*Telephony Application Programming Interface*), é relevante discorrer um pouco sobre API (*Application Program Interface*), pois pode-se dizer que TAPI vem da junção das palavras “*Telephony + API*”. Portanto, API é um conjunto de funções predefinidas do sistema operacional *Microsoft Windows*, usadas para controlar a aparência e comportamento de todos elementos do sistema operacional (desde a aparência das janelas à alocação de memória para um novo processo). Toda ação do usuário causa a execução de uma ou várias funções API, informando ao sistema operacional o que está acontecendo (Billy & George, 2000). Por consequência, a biblioteca TAPI é a que contém funções API de suporte aos dispositivos de telefonia.

3.2.1 INTRODUÇÃO

Co-desenvolvido pela *Intel* e *Microsoft*, TAPI (*Telephony Application Programming Interface*) está proporcionando um método para implementações totalmente abrangentes de CTI. TAPI isola ambos, usuários de PC e desenvolvedores de aplicações, das especificações de princípios básicos da conexão de equipamentos telefônicos, abrindo as portas à nova geração de *softwares* de automação de escritório (Softpac, 1999).

Em 1993 era lançada a primeira versão do TAPI (Versão 1.x), com o objetivo de ligar o PC ao telefone em nível de *desktop*. TAPI (Versão 2.x), lançada em 1996 com a versão 4.0 do *Windows NT*, foi o primeiro passo em ligar o PC diretamente ao PABX. Esta tecnologia está sendo mais aperfeiçoada para a versão 3.x do TAPI que está incorporada na versão 5.0 do *Windows NT* (Inter-tel, 1998).

O padrão TAPI (a nível de *desktop*) incorpora a tecnologia CTI usando o popular formato *Microsoft Windows*. O mesmo está disponível como um upgrade para o *Windows 3.1/3.11* e embutido no *Windows 95* e *98*. O padrão TAPI (a nível de sistema) é suportado no *Windows NT 4.0* (Inter-tel, 1998).

TAPI pode estabelecer dois diferentes métodos de conexão:

- Conexão baseada em desktop (individual ou de pequenos grupos);
- Conexão baseada em redes locais (para grande grupos).

Além disso, ambos métodos de conexão tem dois níveis de capacidade:

- Somente controle de chamada (iniciando, manipulando e terminando mídia de chamados via TAPI);
- Controle de mídia e de chamada (inclui a habilidade de acessar e manipular o conteúdo da chamada).

3.2.2 CONEXÃO BASEADA EM DESKTOP

Este método de integração é tipicamente aplicado a usuários domésticos, pequenos escritórios, e indivíduos ou pequenos grupos numa corporação. Cada PC possui seu *hardware* de comunicação individual, o que torna este método o melhor custo benefício para um pequeno número de PCs (Inter-tel, 1998).

Existem duas variações da conexão baseada em desktop: conexão serial do PC ao aparelho telefônico e PC conectado diretamente ao sistema de comunicação. A TABELA 5 mostra um comparativo dos dois tipos de conexão.

TABELA 5 – TIPOS DE CONEXÃO BASEADA EM DESKTOP

Conexão serial do PC ao aparelho telefônico	PC conectado diretamente ao sistema de comunicação
<ul style="list-style-type: none"> - A conexão é feita entre o aparelho telefônico digital e o PC; - Permite ao PC realizar controle de chamadas, como notificação de chamadas recebidas, fazer ligações, transferir chamadas, colocar chamada em espera, criar conferências, e terminar chamadas, se o <i>hardware</i> e o método de conexão suportarem este tipo de funções; - Permite ao PC controlar somente a chamada propriamente dita e não o seu conteúdo. 	<ul style="list-style-type: none"> - Conexão do PC diretamente ao sistema de comunicação; - Permite ao PC executar as funções do controle de chamadas e acessar e manipular o conteúdo de uma chamada; - Suporta ambas conexões analógicas (usando <i>voice/data/fax modems</i> analógicos como estes usados em PCs domésticos) e digitais (usando serviços proprietários ou ISDN).

Adaptado de: Inter-tel, (1998).

3.2.3 CONEXÃO BASEADA EM REDES LOCAIS

A conexão baseada em redes locais é o método de integração normalmente aplicado a grandes organizações. O *hardware* de comunicação não fica localizado em cada PC. Ao invés disso, ele é centralizado e compartilhado por todos os membros da organização (Inter-tel, 1998).

Existem duas variações da integração baseada em redes locais: *LAN/server-based call control* e *LAN-based communication server*. A TABELA 6 mostra um comparativo destes dois tipos de integração.

TABELA 6 – TIPOS DE CONEXÃO BASEADA EM REDES LOCAIS

<i>LAN/server-based call control</i>	<i>LAN-based communication server</i>
<ul style="list-style-type: none"> - Permite ao PC executar as funções de controle de chamadas se o <i>hardware</i> e o método de conexão suportarem estas funções; - Requer que os PCs rodem o sistema o <i>Windows</i> conectados a um servidor <i>NT</i> sobre uma rede local; - Neste método não existe nenhuma conexão entre o PC o aparelho telefônico. Quando o telefone toca, informações são enviadas através da rede local ao PC informando-o da chamada, quem está ligando, e outras informações relativas. Outrora, o PC pode enviar comandos pela rede local requisitando que o telefone seja atendido, colocado em espera, ou transferido para outro ramal, por exemplo; - Não permite que o PC manipule o conteúdo de uma ligação. 	<ul style="list-style-type: none"> - Esta segunda variação de integração baseada em redes locais permite que o PC execute funções de controle de chamadas usando TAPI; - O PC pode também acessar e manipular o conteúdo de uma ligação, porque a ligação é feita através do PC pela rede de comunicação; - Permite que toda comunicação ocorra sobre a mesma rede local. Não sendo necessárias redes de voz (telefônicas) e de dados (redes locais) separadas. Com isso, o servidor de comunicação fica localizado numa rede local corporativa, e atende todas as necessidades de comunicação da organização.

Adaptado de: Inter-tel, (1998).

3.3 ASYNC PRO 32

A biblioteca de componentes ASYNC PRO 32, foi desenvolvida pela *TurboPower Software Company* (TurboPower, 1999), para o ambiente de programação *Delphi 5* (Cantú, 2000). Esta biblioteca, possui uma grande variedade de componentes, que abrangem a praticamente todas as necessidades de aplicações de CTI.

Os componentes oferecidos pela biblioteca estão divididos em grupos, de acordo com a sua aplicação: *fax*, *FTP*, *modem*, *pager*, portas seriais, protocolo RAS, *script*, luzes de *status*, TAPI, emulador de terminal e *winsock*. Dentre os componentes citados, dois deles foram especialmente importantes para este trabalho: o que provê controle de portas seriais e o TAPI.

3.3.1 COMPONENTE PARA CONTROLE DE PORTAS SERIAIS

Uma aplicação utiliza o componente *TapdComPort* para controlar o *hardware* da porta serial. Todas as operações de I/O da porta serial são executadas através da chamada de métodos do *TapdComPort*, e pelos tratadores de evento que respondem aos eventos da porta. Ações de comunicação de alto nível como discagem de números pelo *modem* ou transferência de arquivos usam o *TapdComPort* para interagir com o *hardware* (TurboPower, 1999).

O *Async Professional* utiliza o termo “*trigger*” (gatilho) para denominar qualquer ação na porta serial que possa fazer que o gerenciador de comunicação gere um evento do *Delphi*. Existem 4 tipos de *triggers*:

- ***Data available***: dado recebido está disponível
- ***Data match trigger***: um caractere ou cadeia de caracteres em especial foi recebida.
- ***Status trigger***: um evento de *status* ocorreu.
- ***Timer trigger***: o tempo expirou.

O componente *TApdComPort* contém uma variedade de rotinas para gerenciar *triggers*. As *triggers* podem ser adicionadas, ativadas, modificadas, e desativadas. Mais detalhes sobre este componente e seu funcionamento são discutidos no próximo item, junto com o componente TAPI.

3.3.2 COMPONENTE TAPI

O componente *TApdTapiDevice* provê discagem do *modem*, atendimento de chamadas e serviços de configuração usando o padrão TAPI embutido no *Microsoft Windows*.

Sem TAPI, o componente *TApdComPort* abre diretamente a porta serial usando a chamada apropriada da API do *Windows*, que retorna um tratador para a porta de comunicação física. O componente *TApdComPort*, então, utiliza este tratador para enviar e receber dados e, controlar a porta serial. Configurar, discar números ou atender chamadas pelo *modem* requer que se envie comandos explícitos ao *modem* (comandos “AT”), se interprete as respostas, e que se trate de milhares de *modems* diferentes que estão disponíveis no mercado (TurboPower, 1999).

Com TAPI, ainda é necessário o uso do componente *TApdComPort*, mas ele não está inicialmente envolvido em estabelecer uma conexão com o *modem*, o componente *TApdTapiDevice* é usado para este propósito. A aplicação chama os métodos *TApdTapiDevice.Dial* para fazer uma ligação, ou a *TApdTapiDevice.AutoAnswer* para aguardar chamadas. TAPI envia os comandos “AT” apropriados, interpreta as respostas e estabelece a conexão do *modem*.

Se o *modem* usado dispor da capacidade de voz, pode-se detectar tons de discagem (DTMF) com o evento *OnTapiDTMF*, e enviar estes mesmos tons com método *SendTone*. Pode-se também tocar arquivos de voz (“.WAV”) através do *modem* ou pela placa de som, com o método *PlayWaveFile*, e gravar conversas com os métodos *StartWaveRecord* e *StopWaveRecord* (TurboPower, 1999).

Quando a conexão do *modem* é cancelada, o *TapdTapiDevice* automaticamente fecha o *TapdComPort* associado. O *TapdComPort* não pode ser usado para I/O a não ser que a conexão do *modem* seja restabelecida pelo dispositivo TAPI, ou então, se o programa descartar o TAPI e abrir a porta usando diretamente o *TapdComPort*. A propriedade *TapiMode* do *TapdComPort* é que determina se o controle da porta fica sob a responsabilidade do *TapdTapiDevice* ou do *TapdComPort*.

Junto com o *TapdTapiDevice* também foram utilizados os componentes *TapdTapiStatus* e *TapdTapiLog*. Estes servem para acompanhar o *status* atual da chamada e para gravar um histórico dos acontecimentos (*log*) do componente TAPI.

A TABELA 7 mostra as principais propriedades, eventos e métodos do componente *TapdTapiDevice*, que foram utilizados na implementação do protótipo.

TABELA 7 – COMPONENTE TAPI

Componente	Propriedades	Métodos	Eventos
TapdTapiDevice	<ul style="list-style-type: none"> - ComPort: nome do componente <i>TapdComPort</i> correspondente. - EnableVoice: ativa as características de voz do modem. Recebe os valores ‘True’ ou ‘False’. - SelectedDevice: nome do dispositivo selecionado compatível com o padrão TAPI. - UseSoundCard: ‘True’ indica que a saída do som será a placa de som; ‘False’ indica que a saída do som será o modem. - InterruptWave: ‘False’ indica que o arquivo que está sendo tocado não pode ser interrompido por um evento <i>OnTapiDTMF</i>; ‘True’ permite que se interrompa o arquivo. 	<ul style="list-style-type: none"> - Dial: disca o número do telefone desejado. - AutoAnswer: coloca o modem no estado de espera, para atender chamadas telefônicas. - SelectDevice: apresenta uma lista de possíveis dispositivos compatíveis com TAPI, para que o usuário selecione um deles. - PlayWaveFile: toca um arquivo de som do formato WAVE na saída selecionada (de acordo com a propriedade <i>UseSoundCard</i>). - StartWaveRecord: inicia a gravação de um arquivo de som no formato wave. - StopWaveRecord: termina a gravação de um arquivo wave. - CancelCall: termina uma chamada telefônica. 	<ul style="list-style-type: none"> - OnTapiConnect: evento gerado quando o modem recebe uma chamada telefônica. - OnTapiDTMF: evento gerado quando o modem detecta o recebimento de tons multifrequenciais (DTMF). Estes tons são tratados como dígitos de ‘0..9’ mais os caracteres ‘*’ e ‘#’. - OnTapiWaveNotify: indica o status do arquivo wave que está sendo tocado no momento. - OnTapiFail: evento gerado quando a chamada telefônica é encerrada pelo usuário.

Adaptado de: TurboPower (1999).

4 DESENVOLVIMENTO DO TRABALHO

De acordo com o que foi discutido na introdução, e apresentado como objetivo deste trabalho, foi desenvolvido um protótipo de sistema de atendimento a chamadas telefônicas que possibilite a consulta de notas de alunos em disciplinas. O presente capítulo trata da especificação e implementação deste protótipo.

4.1 ESPECIFICAÇÃO

O propósito deste protótipo é servir como um sistema atendente de chamadas telefônicas, que possibilite consultar notas de alunos em disciplinas. Para consultar suas notas o aluno fará uma ligação telefônica e será atendido pelo sistema através de voz digitalizada. Ao ser atendido pelo sistema, o aluno deverá informar, através do teclado do telefone, a sua identificação (código e senha). Então lhe serão apresentadas as opções disponíveis de cursos e semestres estudados, através de menus audíveis. A seleção das alternativas também será efetuada através do teclado do telefone, por tom multi-frequencial (DTMF), e o resultado da consulta às notas será devolvido em forma de voz digitalizada ao aluno. Estes arquivos de voz digitalizada deverão estar previamente gravados numa pasta do sistema, em formato *wave* (.WAV).

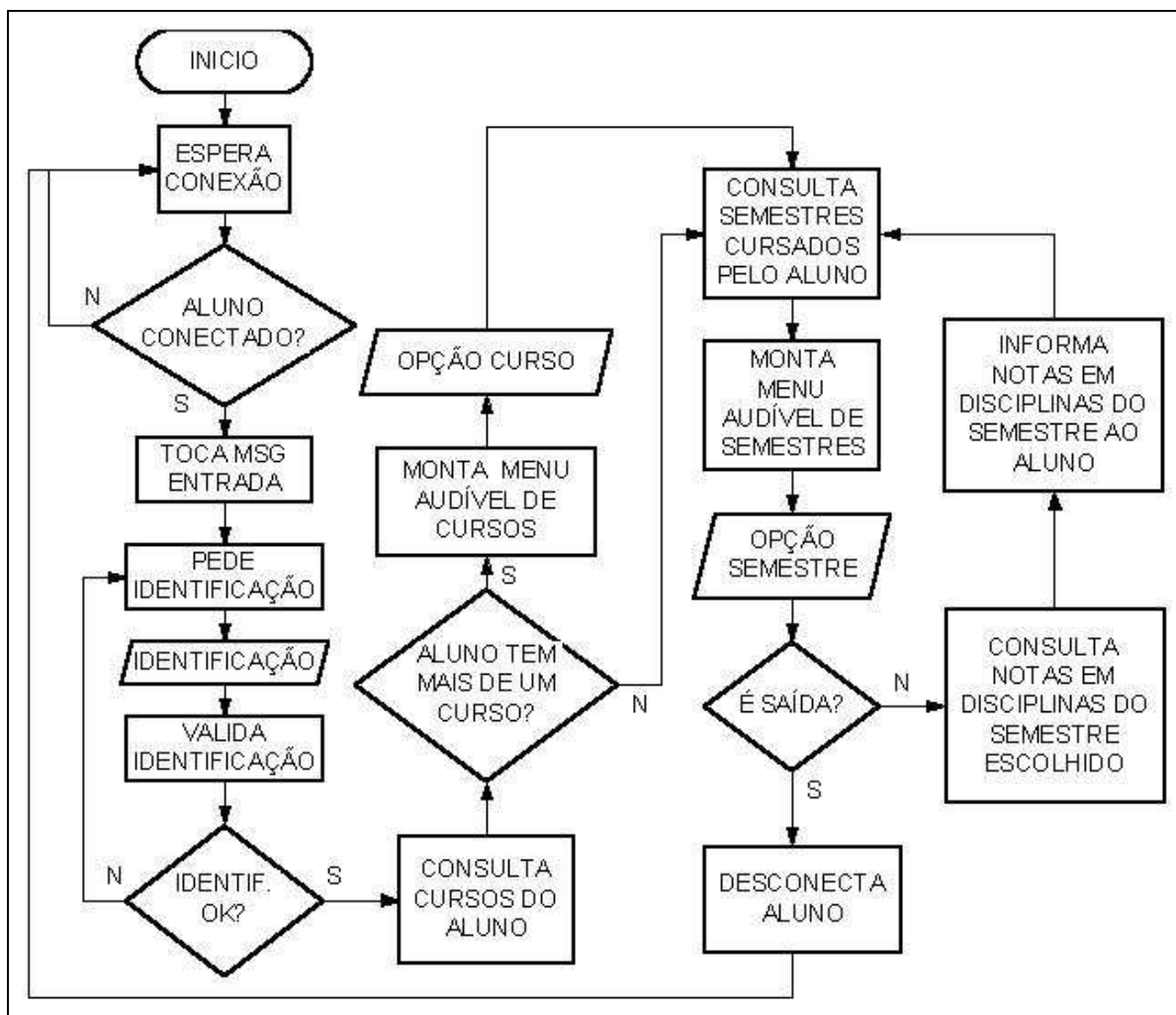
O QUADRO 4 , mostra numa linguagem descritiva, as tarefas que o sistema deve executar ao receber a ligação de um aluno que queira consultar a sua nota.

QUADRO 4 – TAREFAS EXECUTADAS NO ATENDIMENTO DE UMA CHAMADA

- Sistema espera chamada telefônica;
- Sistema atende telefone e solicita identificação do aluno;
- Aluno informa identificação;
- Sistema valida identificação;
- Sistema oferece menu audível de cursos, caso o aluno curse mais de um;
- Aluno informa curso que deseja consultar, caso curse mais de um;
- Sistema oferece menu audível de semestres cursados pelo aluno;
- Aluno informa semestre;
- Sistema informa notas em disciplinas do semestre selecionado pelo aluno.

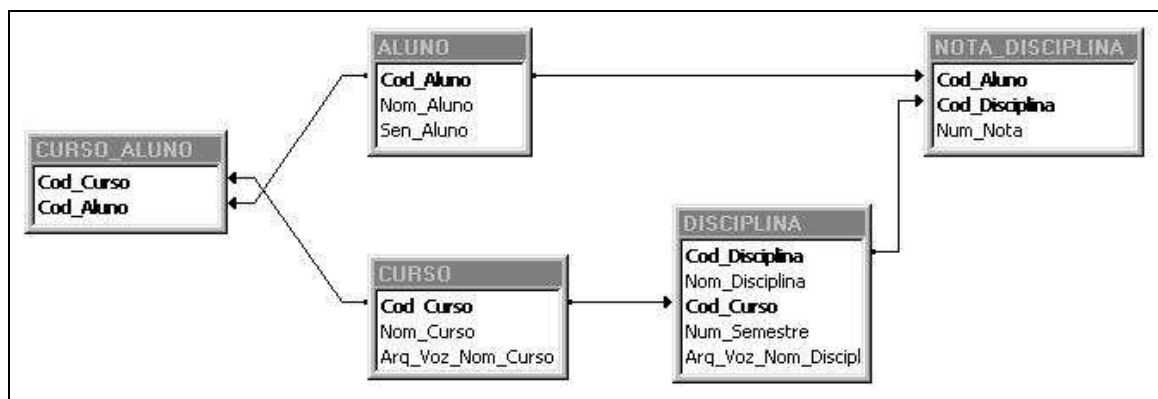
De acordo com as tarefas executadas pelo sistema ao atender uma chamada telefônica, foi definido o fluxo de funcionamento da rotina de atendimento do sistema, conforme mostra a FIGURA 31.

FIGURA 31 – FLUXO DE FUNCIONAMENTO DO SISTEMA



A parte do armazenamento dos dados do sistema foi definida através do modelo de entidade e relacionamento, que é representado na FIGURA 32. Neste, pode-se ver que “um” ALUNO pode cursar “uma ou várias” DISCIPLINAS, enquanto uma DISCIPLINA pode ter “um ou vários” ALUNOS. Por isso, foi criada a tabela associativa NOTA_ALUNO. Da mesma maneira, “um” ALUNO pode cursar “um ou vários” CURSOS, enquanto “um” CURSO tem “um ou vários” ALUNOS. Por isso, foi criada a tabela associativa CURSO_ALUNO. Além disso, “um” CURSO têm “uma ou várias” DISCIPLINAS.

FIGURA 32 – MODELO DE ENTIDADES E RELACIONAMENTOS



O QUADRO 5 mostra o dicionário de dados das tabelas.

QUADRO 5 – DICIONÁRIO DE DADOS DAS TABELAS

Colunas da tabela: ALUNO		
Cod_Aluno	Número(05)	- Código do Aluno;
Nom_Aluno	Texto(50)	- Nome do Aluno;
Sen_Aluno	Texto(50)	- Senha do Aluno;
Colunas da tabela: CURSO		
Cod_Curso	Número(08)	- Código do Curso;
Nom_Curso	Texto(50)	- Nome do Curso;
Arq_Voz_Nom_Curso	Texto(50)	- Arquivo de voz do nome do curso;
Colunas da tabela: CURSO_ALUNO		
Cod_Curso	Número(08)	- Código do Curso;
Cod_Aluno	Número(05)	- Código do Aluno;
Colunas da tabela: DISCIPLINA		
Cod_Disciplina	Número(08)	- Código da Disciplina;
Nom_Disciplina	Texto(50)	- Nome da Disciplina;
Cod_Curso	Número(08)	- Código do Curso;
Num_Semestre	Número(02)	- Número do Semestre da disciplina;
Arq_Voz_Nom_Discipl	Texto(50)	- Arquivo de voz do nome da disciplina;
Colunas da tabela: NOTA_DISCIPLINA		
Cod_Aluno	Número(05)	- Código do Aluno;
Cod_Disciplina	Número(08)	- Código da Disciplina;
Num_Nota	Número(4,2)	- Nota do aluno na disciplina;

A base de dados do sistema será atualizada através de uma rotina de importação de arquivos texto formatados no *layout* definido no QUADRO 6, que será executada semestralmente pelo operador do sistema. O *layout* do arquivo de importação de dados de cada tabela, deve seguir a mesma seqüência dos campos da tabela, conforme o dicionário de dados, sendo que a informação correspondente a cada coluna da tabela deve vir separada por vírgula (“,”).

sistema que foi escolhido um banco de dados compatível com SQL, visto que esta já se tornou uma linguagem padrão de acesso a dados.

Os principais componentes da biblioteca ASYNC PRO 32 utilizados no protótipo foram o *TapdTapiDevice* e *TapdComPort*. Estes componentes abrangem a praticamente todo o objetivo do trabalho. Além destes dois componentes, foram utilizados os componentes *TapdTapiLog* e *TapdTapiStatus* que, quando usados em conjunto, provêm uma prática ferramenta de auditoria para o operador do sistema.

Funcionamento básico do protótipo, com seus métodos, propriedades e eventos:

- a) O operador executa a aplicação, e seleciona o dispositivo TAPI, através do método *ApdTapiDevice.SelectDevice*;
- b) O sistema seta a propriedade *ApdTapiDevice.EnableVoice := True*. O QUADRO 7 mostra o trecho de código onde isto ocorre.

QUADRO 7 – SELECIONANDO O DISPOSITIVO TAPI

```

procedure TfmTapi.btDispositivoTapiClick(Sender: TObject);
begin
  try
    ApdTapiDevice.SelectDevice;
    ApdTapiDevice.EnableVoice := True;
    AtualizaStatus('Dispositivo TAPI selecionado. ');
  except
    begin
      ShowMessage('Dispositivo TAPI selecionado não suporta extensões de voz! Seleccione outro dispositivo. ');
      lbStatus.Caption := 'Selecione dispositivo TAPI...';
    end;
  end;
end;

```

- c) A partir deste momento o operador pode clicar no botão “Aguardar chamadas”, deixando o sistema em modo de espera por ligações. O método *ApdTapiDevice.AutoAnswer* é que faz o *modem* ficar em modo de espera. O QUADRO 8 mostra o procedimento em que isto ocorre.

QUADRO 8 – DEIXANDO O SISTEMA ESPERANDO POR LIGAÇÃO

```

procedure TfmTapi.btAguardarClick(Sender: TObject);
begin
  if ApdTapiDevice.EnableVoice then
  begin
    ApdTapiDevice.AutoAnswer;
    AtualizaStatus('Aguardando chamada...');
    Etapa := EtpAguardando;
  end
  else
  begin
    ShowMessage('Dispositivo TAPI atual não suporta extensões de voz!
    Selecione outro dispositivo. ');
    lbStatus.Caption := 'Selecione dispositivo Tapi...';
    Etapa := EtpNaoPronto;
  end;
end;
end;

```

d) Quando o sistema recebe uma ligação, é gerado o evento *OnTapiConnect*, e deste evento é chamada o procedimento de boas vindas ao usuário, chamando-se o método *ApdTapiDevice.PlayWaveFile("entrada.wav")* para tocar o arquivo de boas vindas;

e) Logo após, o sistema toca arquivos de voz pedindo que o usuário digite o seu código e senha para validar o acesso às informações. O sistema fica esperando que o usuário termine de digitar todos os dígitos de sua identificação. E cada dígito é reconhecido pelo tom de discagem, o DTMF, que gera o evento *OnTapiDTMF*. Esse evento chama o procedimento onde é tratado, e o dígito é automaticamente reconhecido pelo componente *TApdTapiDevice* e passado por parâmetro para o procedimento. O QUADRO 9 mostra a parte deste procedimento onde o sistema toca o arquivo de voz correspondente a tecla digitada pelo usuário, no teclado do telefone.

QUADRO 9 – RECONHECENDO TECLAS DIGITADAS NO TELEFONE

```

procedure TfmTapi.ApdTapiDeviceTapiDTMF(CP: TObject; Digit: Char;
  ErrorCode: Integer);
begin
  { Eventos do teclado do telefone do usuário }
  if Digit = '*' then
    TocaWave('Asterisco.wav')
  else if Digit = '#' then
    TocaWave('Sustenido.wav')
  else
    TocaWave('Num'+Digit+'.wav');

  (...)
end;

```

f) Depois de validada a identificação, o sistema verifica se o aluno cursa somente um curso, caso curse mais de um, apresenta um menu audível com os cursos cursados pelo aluno, para que o aluno selecione o curso. Se o aluno cursa apenas um curso, o sistema pula esta etapa e passa diretamente para a próxima. O QUADRO 10 mostra um trecho do procedimento em que é validada a identificação do aluno.

QUADRO 10 – VALIDANDO A IDENTIFICAÇÃO DO ALUNO

```

procedure TfmTapi.EtapaMenuCurso();
begin
  { Valida identificação }
  Query.SQL.Clear;
  Query.SQL.Text := ' SELECT NOM_ALUNO'+
                    ' FROM ALUNO '+
                    ' WHERE COD_ALUNO = :PARM_0'+
                    ' AND SEN_ALUNO = :PARM_1';
  Query.Params[0].AsInteger := StrToInt(CodAlu);
  Query.Params[1].AsString := SenAlu;
  Query.Open;
  if not Query.Eof then {Se identificação for valida}
  begin
    AtualizaStatus('O aluno '+Query.FieldName('NOM_ALUNO').AsString+'
está conectado!');

    (...)

  end;
  Query.Close;
end;

```

g) O sistema monta um menu audível com os semestres que o aluno cursou, e pede ao aluno que selecione o semestre da disciplina que deseja consultar (com dois dígitos). Para montar este menu, o sistema dispõe do código do aluno, daí parte da tabela NOTA_DISCIPLINA para saber quais disciplinas o ALUNO cursou, e da tabela DISCIPLINA, agrupa o campo “num_semestre” para obter todos os semestres cursados pelo aluno.

h) Ao selecionar o semestre, o sistema informa, através de voz digitalizada, as notas em disciplinas referentes ao semestre que o aluno selecionou. Para fazer esta consulta, o sistema dispõe do código do aluno e do semestre escolhido, daí parte da tabela NOTA_DISCIPLINA para saber quais disciplinas o ALUNO cursou, e da tabela DISCIPLINA, utiliza o campo “arq_voz_nom_discpl” para chamar o arquivo de voz correspondente ao nome da disciplina. Depois de informar as notas em disciplinas do semestre, o sistema volta a apresentar o menu de semestres (procedimentos da etapa “g”), onde o aluno tem a opção de continuar consultando outros semestres ou encerrar a ligação.

i) Quando o aluno se desconecta do sistema, é gerado o evento *OnTapiFail*, e no procedimento tratador do evento é chamado o método *ApdTapiDevice.CancelCall*. Ao se executar tal método o sistema fica indisponível para receber mais ligações, porém, foi implementado um *Timer* que verifica o *status* do sistema, e quando este se encontra no *status* de espera, este *Timer* chama o mesmo procedimento que é chamado quando o operador clica no botão “Aguardar chamadas”. Isso evita que o operador tenha que ficar clicando neste botão depois de cada ligação.

j) O QUADRO 11 mostra o procedimento *TocaWave*, que teve que ser criado para se ter um controle e padronização na maneira de tocar o arquivo de voz pelo modem.

QUADRO 11 – PROCEDIMENTO DE TOCAR ARQUIVO DE VOZ

```
procedure TfmTapi.TocaWave(ArqWave: String);
begin
  if (btSaidaSom.Tag = 1) then
    ApdTapiDevice.UseSoundCard := True;

  AtualizaStatus('Tocando arquivo Wave:'+ArqWave);
  ApdTapiDevice.InterruptWave := False;
  ApdTapiDevice.PlayWaveFile('c:\AraraAzul\som\' +ArqWave);
  while ApdTapiDevice.WaveState = wsPlaying do
    Application.ProcessMessages;

  ApdTapiDevice.UseSoundCard := False;
  ApdTapiDevice.EnableVoice := True;
end;
```

4.2.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

O protótipo foi batizado com o nome de AraraAzul 1.0. Abaixo, a FIGURA 33 mostra a tela principal do sistema.

FIGURA 33 – TELA PRINCIPAL DO SISTEMA



Esta tela foi desenvolvida para oferecer controle do sistema ao seu operador, ou seja, acesso e manutenção aos dados, e acompanhamento e controle sobre as chamadas atendidas pelo sistema.

Para acessar e manter o banco de dados do sistema, o operador dispõe das telas “Cadastros” e “Rotinas”, que são acessadas pelos botões de nome correspondente.

Para controlar uma chamada atendida pelo sistema, o operador dispõe dos botões “Aguardar Chamada” e “Desligar”. Para acompanhar uma chamada, possui a tela de “Status”, e a linha “Números digitados pelo usuário” para saber o que o usuário digitou no aparelho telefônico.

4.2.2.1 MANUTENÇÃO DOS DADOS DO SISTEMA

Basicamente duas telas dão suporte aos dados do sistema:

- a) Tela de Cadastros;
- b) Tela de Rotinas.

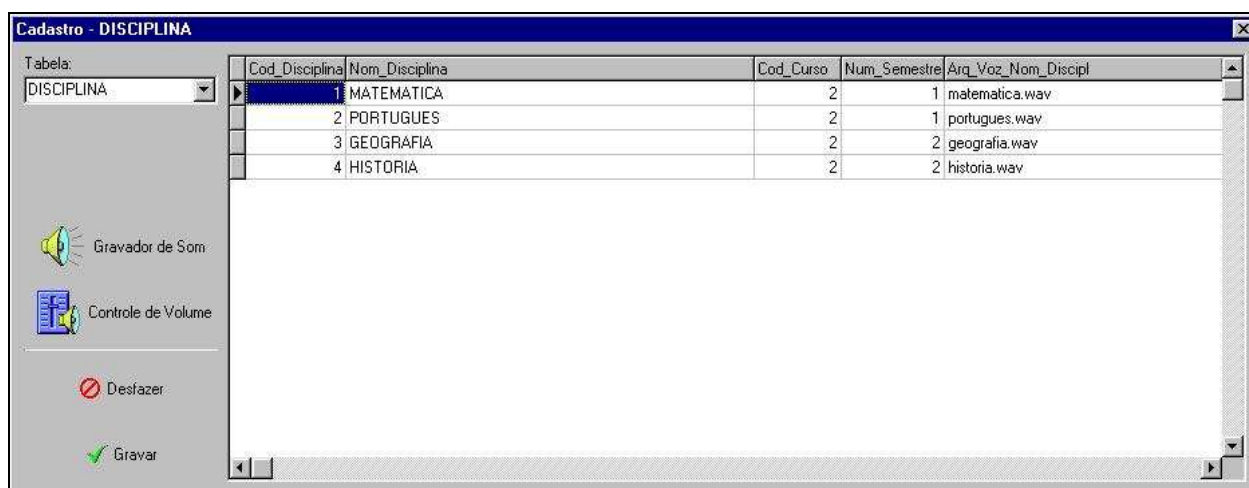
Abaixo a FIGURA 34 mostra os botões da tela principal que dão acesso a estas telas.

FIGURA 34 – BOTÕES DE CADASTROS E ROTINAS



A **tela de Cadastros** serve para o operador consultar a base de dados, bem como associar arquivos de voz com nome de disciplinas às disciplinas cadastradas no sistema. O operador pode fazer modificações apenas na tabela de disciplinas, as demais tabelas servem somente para consulta. A FIGURA 35 mostra a tela de cadastros com a tabela de disciplinas selecionada.

FIGURA 35 – TELA DE CADASTROS



Quando a tabela CURSO ou DISCIPLINA está selecionada, a tela oferece mais dois botões para o operador: “Gravador de Som” e “Controle de Volume”. Estes botões servem para acessar os programas Gravador de Som e Controle de Volume do *Microsoft Windows*.

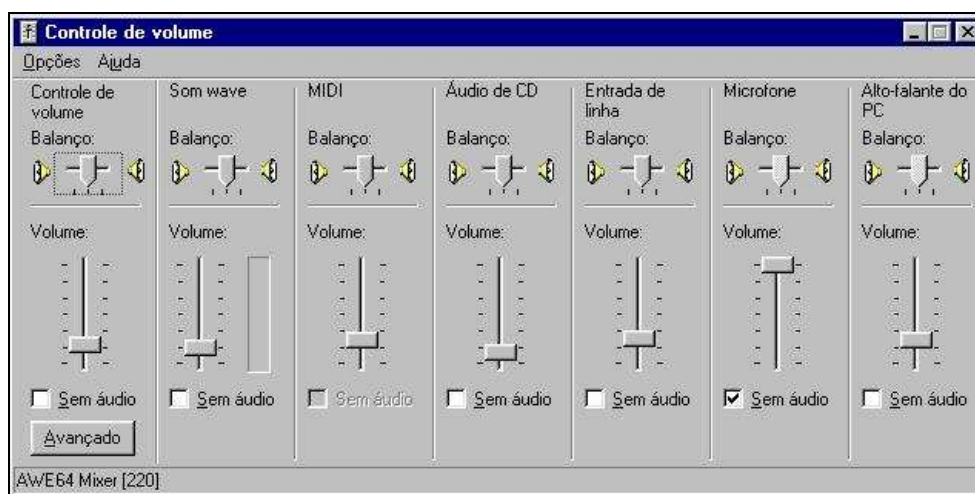
Isto porque estas duas tabelas possuem um campo para associar o arquivo de voz correspondente ao nome do curso e ao nome da disciplina, respectivamente. Com estes dois programas, o operador pode criar seus arquivos de voz e associá-los ao curso ou à disciplina correspondente.

Abaixo, as FIGURAS 36 e 37 mostram os programas Gravador de Som e Controle de Volume, respectivamente.

FIGURA 36 – GRAVADOR DE SOM



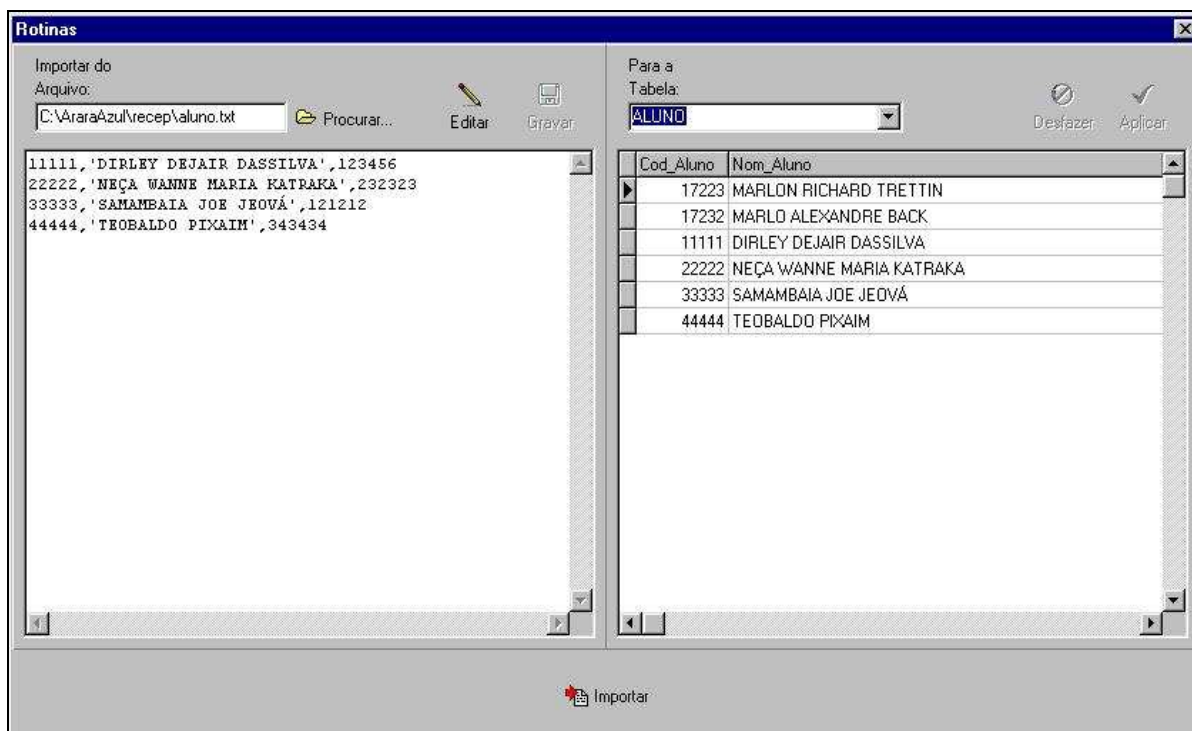
FIGURA 37 – CONTROLE DE VOLUME



A **tela de Rotinas** serve para o operador importar a base de dados para o sistema. Esta pode ser originada de qualquer banco de dados utilizado pela instituição que deseja adotar o sistema. Basta os arquivos de importação virem em formato de texto e no *layout* definido na especificação, que a importação da base da instituição será possível de ser feita. Dessa forma o sistema torna-se mais portátil, e visto que esta importação de dados deva acontecer normalmente duas vezes por semestre, optou-se por esta opção ao invés de fazer acesso direto ao banco de dados da instituição.

Na FIGURA 38, pode-se ver o funcionamento da tela de Rotinas. À esquerda o operador escolhe o arquivo texto de onde se originam os dados, e à direita escolhe a tabela destino para os dados, logo após clica no botão “Importar”.

FIGURA 38 – TELA DE ROTINAS

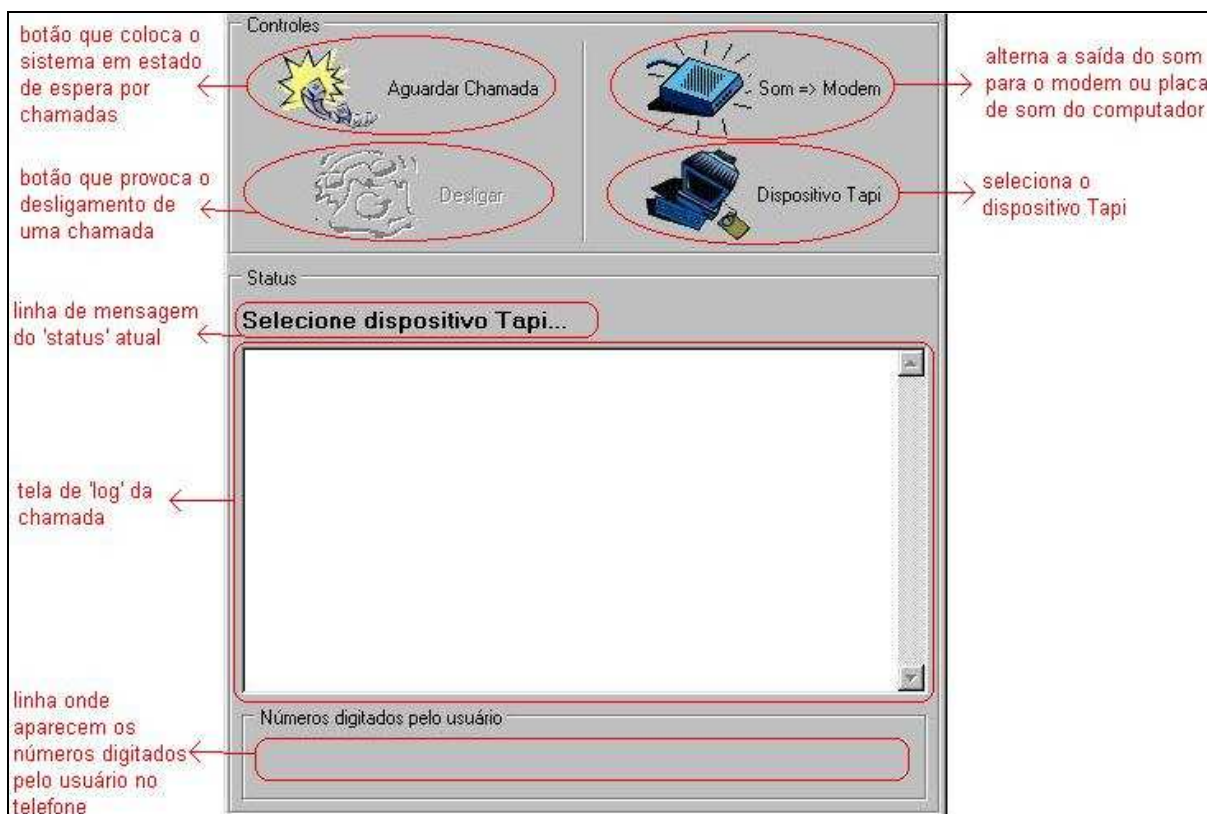


4.2.2.2 CONTROLE E ACOMPANHAMENTO DA CHAMADA

Nesta parte da tela principal, o operador pode acompanhar e controlar as chamadas atendidas pelo sistema. Esta parte da tela é composta pelos botões de “Aguardar Chamada”, “Desligar”, “Som=> Modem / Placa de Som”, “Dispositivo Tapi” e “Tela de Status”.

A FIGURA 39 mostra de forma resumida, a função de cada um destes componentes.

FIGURA 39 – CONTROLE E ACOMPANHAMENTO DA CHAMADA



A parte de *status* da chamada é composta por uma linha de *status*, uma lista de operações que foram executadas (*log*) e uma linha que mostra o que o usuário remoto está digitando no telefone, conectado ao sistema.

4.2.2.3 TELAS AUXILIARES

Ainda na tela principal, existem dois botões que dão acesso as duas telas auxiliares do sistema: “Sobre o AraraAzul”, que contém informações sobre a versão e autor do sistema, e “Ajuda”, que dá acesso a tela de ajuda. Abaixo, a FIGURA 40 mostra a parte da tela principal com estes dois botões.

FIGURA 40 – BOTÕES DAS TELAS AUXILIARES



5 CONCLUSÕES

Qualquer tipo de consumidor, seja de serviços, produtos e até de educação, tem a necessidade de ser bem atendido. Para isso, empresas e instituições disponibilizam diversos meios para o consumidor ter acesso à informação que lhe satisfaça.

Foi objetivado por este trabalho, implementar um protótipo para acesso remoto a banco de dados através da linha telefônica, coisa que há muito, é disponibilizada por instituições bancárias para consulta de saldo, extratos e outros serviços. Este protótipo em particular, teve o objetivo de desenvolver um meio para acesso à notas de alunos em disciplinas. Este meio foi o telefone.

Conforme o problema é apresentado na introdução deste trabalho, pode-se concluir que os seus objetivos foram alcançados, tendo em vista que o protótipo desenvolvido está dentro das especificações realizadas, e dos limites previstos.

5.1 CONSIDERAÇÕES FINAIS

Este trabalho provê uma implementação descomplicada e de fácil assimilação do protótipo desenvolvido, sendo assim, didaticamente vantajoso em relação aos sistemas complexos existentes para propósitos semelhantes. Por ser de fácil assimilação, ele facilita a sua extensão e aprimoramento para versões futuras.

A desvantagem deste protótipo é que, para servir para outro tipo de aplicação, como uma central de atendimento de horóscopo, por exemplo, é preciso que se faça manutenção no seu código fonte. Não sendo ele, portanto, configurável pelo operador.

As limitações do protótipo estão mais relacionadas com o *hardware* utilizado, visto que a biblioteca de componentes ASYNC PRO 32 é uma ferramenta poderosa para a implementação de sistemas com propósitos semelhantes. A única limitação imposta pelo ASYNC PRO 32 é que, na versão *trial*, utilizada no protótipo, não é possível executar a aplicação sem o *Delphi*. A limitação do *hardware* se refere ao *modem* utilizado, que apesar de oferecer recursos de voz, não pode ter seu desempenho e recursos comparados com uma placa de telecomunicações profissional. Esta sim, é recomendada para quem quiser utilizar o protótipo profissionalmente.

5.2 EXTENSÕES

Sugestões para extensão deste trabalho em trabalhos futuros:

- a) Implementar um módulo no qual o operador do sistema possa configurar *scripts* de atendimento, o que tornaria o sistema portátil para praticamente qualquer aplicação;
- b) Implementar recursos de reconhecimento de voz no sistema para que o usuário não precise sempre interagir com o sistema via tons de discagem (DTMF);
- c) Implementar recursos de textificação de voz para melhorar a capacidade de armazenamento do sistema; armazenando as conversas que o usuário tem com atendente em formatos de texto. Isso contribuirá para uma economia significativa de espaço de armazenamento nas empresas que possuem centrais de atendimento, e utilizam este recurso;
- d) Implementar recursos de sintetização de voz, o que extinguirá a necessidade de arquivos de voz no sistema. Isso traz o benefício de os diálogos não precisarem ser pré-gravados em arquivos de voz, podendo ser gerados na hora da interação com o usuário, e além disso o custo de armazenamento destes diálogos se torna muito menor;
- e) Trocar os tipos de arquivos de voz de (.WAV) para (.MP3), economizando mais espaço de armazenamento.

REFERÊNCIAS BIBLIOGRÁFICAS

ALENCAR, Marcelo Sampaio de. **Telefonia digital**. São Paulo: Érica, 1998.

BILLY & GEORGE Software. **Using windows API in VB tutorial**, [S.L.], mar. 2000. Disponível em: <www.geocities.com/SiliconValley/Lab/1632/atc02.htm> . Acesso em: 02/12/2000.

CANTÚ, Marco. **Dominando o Delphi 5 – a bíblia**. São Paulo: Makron Books, 2000.

GIORGETTI, Guido. **Free web site for electronics & communications**, [S.L.], jan. 1998. Disponível em: <<http://users.sienanet.it/users/guido>>. Acesso em: 29/11/2000.

INTER-TEL Incorporated. **Tech knowledge – transforming technology into solutions**, Chandler, abr. 1998. Disponível em: <http://www.inter-tel.com/products/tech_knowledge/tapi.html>. Acesso em: 11/10/2000.

MONTORO, Fábio de Azevedo. **Modem: você precisa dele para navegar nas redes de computadores**. São Paulo: Érica, 1995.

NUNES, José Renato Soares. **Comunicação de dados: conceitos básicos**. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1989.

SHIOZAWA, Ruy Sérgio Cacesse. **Qualidade no atendimento e tecnologia de informação**. São Paulo: Atlas, 1993.

SILVEIRA, Jorge Luís. **Comunicação de dados e sistemas de teleprocessamento**. São Paulo: Makron , McGraw-Hill, 1991.

SOARES Neto, Vicente. **Comunicação de dados: conceitos fundamentais**. São Paulo: Érica, 1993.

SOFTPAC, Inc. **Telephony**, Newtown, jun. 1999. Disponível em: <<http://www.softpac.com/cti.htm>>. Acesso em: 11/10/2000.

TAFNER, Malcon Anderson; LOESCH, Cláudio; STRINGARI, Sérgio. **Comunicação de dados usando a linguagem C**. Blumenau: FURB, 1996.

TCI-MIPS. **Voice modem**, [S.L.], jan. 1996. Disponível em: <http://www.tci-mips.no/teknisk/info/Aopen/Tech9.htm>>. Acesso em: 12/10/2000.

TURBOPOWER Software Company. **About Async Professional**, Colorado Springs, jun. 1999. Disponível em: <http://www.turbopower.com>>. Acesso em: 14/11/2000.

U.S. ROBOTICS. **Sporster voice 33.6 faxmodem manual**. Skokie: U.S. Robotics, 1996.