

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

TIAGO DA ROSA VALÉRIO

**PROTÓTIPO DE INTEGRAÇÃO ENTRE TELEFONE E COMPUTADOR VIA
UNIDADE DE RESPOSTA AUDÍVEL (URA)**

CRICIÚMA, JUNHO DE 2006.

TIAGO DA ROSA VALÉRIO

**PROTÓTIPO DE INTEGRAÇÃO ENTRE TELEFONE E COMPUTADOR VIA
UNIDADE DE RESPOSTA AUDÍVEL (URA)**

Trabalho de Conclusão de Curso para
Obtenção do Grau de Bacharel em Ciências da
Computação da Universidade do Extremo Sul
Catarinense UNESC.

Orientador: Prof. MSc. Paracelso de O. Caldas

CRICIÚMA, JUNHO DE 2006.

Este trabalho é dedicado à Alvaci da Rosa Valério e Divaldo Valério por nunca deixarem de acreditar no meu potencial e incentivo incansável para conclusão de meus estudos e realização profissional.

AGRADECIMENTOS

Agradeço ao professor Paracelso de Oliveira Caldas, pela orientação neste trabalho e o desempenho preocupação em cada etapa da elaboração, sempre buscando o melhor caminho e ao mesmo tempo esclarecendo dúvidas, nunca deixando fugir do tema do trabalho proposto.

A professora Merissandra Côrtes pela dedicação e paciência na correção da metodologia científica deste trabalho.

Ao professor Paulo Martins, pela confiança e profundo conhecimento na área de redes contribuindo para realização deste trabalho.

A Thiago Hoffman por esclarecimentos referentes à utilização da tecnologia Java com sintetização de voz e utilização de funções de *APIs* externas.

Ao professor Wilson Gruber, mostrando os padrões diferentes de comunicação do telefone e computador, esclarecendo assim o grande objeto de estudo das interfaces de programação para realização da integração entre telefone e computador.

A todos professores com quem tive aula e forneceram conhecimento não só na área do curso, mas também para vida pessoal.

A todos os meus colegas de classe, onde compartilhamos momentos bons e ruins para chegarmos até a conclusão do curso.

A todos os meus familiares e amigos, que com uma simples palavra de incentivo, faz com que um acadêmico tire forças de onde não existem mais.

Aos meus pais, pelo apoio e credibilidade, a mim confiados.

A Priscilla Fernandes pelo apoio e paciência, em saber que é preciso realizar certos sacrifícios para podermos conquistar nossos objetivos na vida.

“O pessimista se queixa do vento, o otimista espera que ele mude e o realista ajusta as velas.”

RESUMO

Este trabalho se refere a utilização de novas tecnologias nos serviços via telefone. As linguagens de programação realizam novas implementações, facilitando a comunicação entre telefone e computador, no Java foi implementado a biblioteca *Java Telephony Application Programming Interface (JTAPI)*, com métodos para comunicação entre telefone e computador, trabalhando diretamente nos pinos do *modem*. Para interação entre telefone e computador a *JTAPI* permite a captura da ligação telefônica pelo *modem* fornecendo eventos que permitem identificar as teclas pressionadas no telefone do cliente. Esta tecnologia é restrita aos *modems* que possuem o método *voice* embutido, pois este permite carregar a função *fclass* do java que é responsável pela conexão ao sistema telefônico pelo *modem*. Quanto ao retorno das informações ao cliente deve ser configurado pelo próprio sistema operacional redirecionar o som da caixa do computador para o *modem*. Após o cliente informar o seu código o software monta um *Structured Query Language (SQL)* dinâmico, buscando informação do saldo devedor e próximo vencimento e utiliza tecnologia de voz sintetizada para enviar esta informação para o cliente pela caixa de som ou pelo *modem* conforme configuração do sistema operacional. Este processo abrange todas as empresas com um número de atendentes, que quando solicitados pelos seus clientes, buscam as informações no sistema da empresa e retornam pelo telefone. O protótipo tem o objetivo de auxiliar às pequenas e médias empresas fazendo uma conexão direta entre o cliente com seu telefone e o sistema da empresa.

Palavras – Chave: *JTAPI*, Telefone, Computador, Comunicação.

ABSTRACT

This work relates the use of new technologies in the services saw telephone. The programming languages carry through new implementations, facilitating the communications between telephone and computer, in the Java were implemented the library Java Telephony Application Programming Interface (JTAPI), with methods for communication between telephone and computer, working direct in the bolts of the modem. For interaction between telephone and computer the Jtapi allows to the capture of the telephone linking for the modem supplying events that allow to identify the keyboard keys pressured in the telephone of the customer. This technology is restricted to modems that the possess the inlaid method voice, this allows to load the function fclass of Java it that it is responsible for the connections to the telephonic systems for the modem. How much to the return of information to the customer it must be configured by the proper operational system redirecting the sound of the box of the computer for the modem. After the customer to inform its code software mounts a dynamic Structured Query Language (SQL), searching information of the debt balance and next expiration and uses technology of synthetized voice to send this information for the customer for the box of sound or the in agreement modem configuration of the operational system. This process encloses all the companies with a number of attendants, who when requested for its customers, they search the information in the system of company and return for the telephone. The archetype has the objective of assisting small companies making a direct connection enters the customers with its telephone and the system of the company.

Words – Keys: *JTAPI*, Telephone, Computer, Communication.

LISTA DE FIGURAS

Figura 1. Rede totalmente interconectada.	19
Figura 2. Computador centralizado.	19
Figura 3. Hierarquia de dois níveis.	20
Figura 4. Rota de uma chamada a pequena distância.	20
Figura 5. Rota de um circuito típico de uma chamada a média distância.	21
Figura 6. O uso das transmissões analógica e digital para uma chamada entre dois computadores. A conversão é feita entre os <i>modems</i> .	23
Figura 7. Os canais tem sua frequência aumentada, cada qual com um valor diferente, logo após são combinados.	26
Figura 8. Multiplexação por divisão de comprimento de onda.	27
Figura 9. Multiplexação por divisão de tempo.	30
Figura 10. Modulação por 3 <i>bits</i> por amostra.	33
Figura 11. Modulação por 4 <i>bits</i> por amostra.	33
Figura 12. Supressores de eco.	35
Figura 13. Padrão RS-232-C.	37
Figura 14. Reconhecimento de fala.	39
Figura 15. Sistema Gerencial de Banco de Dados.	46
Figura 16. Os objetos de telefonia em espera.	56
Figura 17. Primeiro e terceiro grupo do controle de chamada da rede telefônica.	62
Figura 18. Modelo de chamada para dois grupos.	64
Figura 19. Modelo de chamada para duas conexões simultâneas.	64
Figura 20. Modelo de chamada para dois terminais ativos.	65
Figura 21. Modelo de chamada de três grupos.	65

	22
Figura 22. Estado finito de máquina do objeto chamada.	66
Figura 23. Estado finito de máquina do objeto conexão.	67
Figura 24. Estado finito de máquina do objeto terminal conexão.	68
Figura 25. A aplicação cria um observador para o terminal.	70
Figura 26. A aplicação cria o objeto chamada.	70
Figura 27. A aplicação estabelece a chamada.	71
Figura 28. O protocolo <i>stack</i> envia uma mensagem de <i>setup</i> .	71
Figura 29. O protocolo <i>stack</i> envia uma mensagem alertando.	72
Figura 30. O usuário B responde a chamada.	72
Figura 31. O protocolo <i>stack</i> envia uma mensagem de conexão.	72
Figura 32. Primeiro e terceiro grupo do controle de chamada de IP para o telefone.	75
Figura 33. Sistema Intelicon com <i>pabx</i> .	80

LISTA DE SIGLAS

ACD	<i>Automatic Call Distribution</i>
ADSI	<i>Analog Display Service Interface</i>
API	<i>Application Programming Interface</i>
ARCNET	<i>Attached Resource Computer Network</i>
ASAI	<i>Definity CallVisor Adjunct Swith</i>
CCCIT	<i>Corporate Core Compensation Implementation Teams</i>
CDMA	<i>Code Division Multiple Access</i>
CLDC	<i>Connected Limited Device Configuration</i>
CLID	<i>Calling Line Identification</i>
CSTA	<i>Computer Supported Telecommunication Aplications</i>
CTI	<i>Computer Integrated Telephony</i>
DCE	<i>Data Circuit-Terminating Equipment</i>
DDL	<i>Data Definition Language</i>
DML	<i>Data Manipulation Language</i>
DNS	<i>Domain Name System</i>
DTE	<i>Data Terminal Equipment</i>
DTMF	<i>Dual Tone Multi-Frequency</i>
ECTF	<i>Enterprise Computer Telephony Forum</i>
EIA	<i>Eletronic Industries Association</i>
FFT	<i>Windowed Fast Fourier Transform</i>
FDM	<i>Frequency Division Multiplexing</i>
FSM	<i>Finite State Machine</i>
G3PD	<i>Definity G3 PABX</i>

HMM	<i>Hidden Markov Models</i>
IEEE	<i>Institute of Eletrical and Eletronic Engineers</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization of Standardization</i>
IVR	<i>Interactive Voice Response</i>
JTAG	<i>Join Test Action Group</i>
JTAPI	<i>Java Telephony Application Programming Interface</i>
KTS	<i>Key Telephony System</i>
LBS	<i>Location-Based Services</i>
LLC	<i>Logical Link Control</i>
MIDP	<i>Mobile Information Device Profile</i>
MVIP	<i>Multivendor Integration Protocol</i>
NML	<i>Netware-Loadable Module</i>
ODBC	<i>Open Database Connectivity</i>
OSI	<i>Open System Interconnect</i>
PCM	<i>Pulse Code Modulation</i>
PIN	<i>Personal Identification Number</i>
QAM	<i>Quadratura por Amplitude</i>
RFC	<i>Request For Comments</i>
RMS	<i>Record Management System</i>
SAC	<i>Serviços de Atendimento ao Consumidor</i>
SCSA	<i>Signaling Computing System Architeture</i>
SDH	<i>Synchronous Digital Hierarchy</i>
SDL	<i>Storage Definition Language</i>
SMTP	<i>Simple Mail Transfer Protocol</i>

SNMP	<i>Simple Network Management Protocol</i>
SONET	<i>Synchronous Optical Network</i>
SQL	<i>Structured Query Language</i>
STF	Sintetizador Texto-Fala
TAO	<i>Telephony Application Objects</i>
TAPI	<i>Telephony Application Programming Interface</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TDM	<i>Time Division Multiplexing</i>
TDMA	<i>Time Division Multiple Access</i>
TSAPI	<i>Telephony Service Application Programming Interface</i>
UMS	<i>Unified Messaging System</i>
URA	Unidade de Resposta Audível
VDL	<i>Vision Definition Language</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVO GERAL	15
1.2	OBJETIVOS ESPECÍFICOS	15
1.3	JUSTIFICATIVA	16
1.4	ESTRUTURA DO TRABALHO	16
2	SISTEMA TELEFÔNICO	18
2.1	ESTRUTURA DO SISTEMA TELEFÔNICO	18
2.2	TRANSMISSÃO DE INFORMAÇÕES	21
2.3	PROBLEMAS NA TRANSMISSÃO	23
3	MÉTODOS DE TRANSMISSÃO DE INFORMAÇÕES	25
3.1	MULTIPLEXAÇÃO POR DIVISÃO DE FREQUÊNCIA	25
3.2	MULTIPLEXAÇÃO POR DIVISÃO DE COMPRIMENTO DE ONDA	27
3.3	MULTIPLEXAÇÃO POR DIVISÃO DE TEMPO	28
3.4	<i>SONET / SDH</i>	30
3.5	COMUTAÇÃO	31
4	MODEMS	32
4.1	FUNCIONAMENTO DOS <i>MODEMS</i>	32
4.2	INTERFACE ENTRE O <i>MODEM</i> E O COMPUTADOR	35
5	UTILIZAÇÃO DE VOZ NO COMPUTADOR	39
5.1	RECONHECIMENTO DE VOZ	40
5.2	APLICAÇÕES COM RECONHECIMENTO DE VOZ	41
5.3	SINTETIZAÇÃO DE VOZ	43
6	BANCO DE DADOS	45
6.1	SISTEMA DE BANCO DE DADOS	45
6.2	LINGUAGEM PARA MANIPULAÇÃO DE DADOS	48
6.3	CLASSIFICAÇÃO DO SISTEMA GERENCIAL DE BANCO DE DADOS	49
7	INTEGRAÇÃO TELEFONE E COMPUTADOR	50
7.1	<i>CALL CENTER</i> (CENTRO DE CHAMADAS)	50
7.2	FUNÇÕES DA INTEGRAÇÃO ENTRE TELEFONE E COMPUTADOR	52
7.3	INTERFACE ENTRE <i>SWITCH</i> E <i>HOST</i>	54
7.4	INTERFACE DE PROGRAMAÇÃO DA APLICAÇÃO	57
7.5	RECURSO DE ARQUITETURA DO CTI	58
8	JAVA TELEPHONY APPLICATION PROGRAMMING INTERFACE	61
8.1	LIGAÇÃO ENTRE TELEFONE E COMPUTADOR	61
8.2	MODELOS DE CHAMADA	63
8.3	ESTADO FINITO DE MÁQUINA	66

9	TRANSIÇÕES DO MODELO DE CONFIGURAÇÃO	70
9.1	FUNCIONAMENTO DO MODELO DE CHAMADA	70
9.2	VÁRIOS NÍVEIS DE DETALHE	73
9.3	ENDEREÇAMENTO	75
10	UNIDADE DE RESPOSTA AUDÍVEL	77
10.1	CONCEITO DE URA	77
10.2	TIPOS DE APLICAÇÕES	78
11	TRABALHOS CORRELATOS	80
11.1	SISTEMA INTELICON	80
11.2	URA CTI <i>SERVER</i>	81
11.3	UNIDADE DE RESPOSTA INTERATIVA INFORMATEC	81
11.4	PLATAFORMA MULTISERVIÇOS URA	82
12	PROTÓTIPO DE INTEGRAÇÃO TELEFONE E COMPUTADOR	84
	CONCLUSÃO	90
	REFERÊNCIAS	93
	BIBLIOGRAFIA RECOMENDADA	96

1 INTRODUÇÃO

Vive-se em uma sociedade capitalista, na qual empresários procuram aperfeiçoar novas tecnologias reduzindo custos e proporcionando qualidade em seus serviços e produtos, para seus clientes. Devido a grande concorrência no mercado empresarial, esses clientes tornam-se cada vez mais exigentes e ao mesmo tempo procuram conhecer melhor os seus direitos como consumidores. Assim para atender esta expectativa de mercado seria um grande diferencial às empresas que prestassem esclarecimentos dos direitos do consumidor e fornecessem uma diversidade de serviços para estes clientes.

Existem empresas que adotaram, o serviço de tele-atendimento, cujo funcionamento consiste no cliente ligar para a empresa e ser prontamente atendido, com suas dúvidas esclarecidas. Esse atendimento ao cliente também possui opções de fornecimento de vários serviços como orçamentos, informação sobre o vencimento das parcelas do cliente ou ainda a sua situação do crédito em relação à empresa. Para as empresas poderem ampliar ainda mais os seus domínios, elas necessitam de um plano de atendimento 24 horas ou ainda a ampliação do seu quadro de funcionários. Outro fator é a necessidade de reduzir custos e ao mesmo tempo se manterem competitivas no mercado, no que se refere ao preço de seus serviços e produtos e a qualidade que os mesmos são oferecidos.

O protótipo de integração entre telefone e computador via URA baseia-se na eliminação ou pelo menos redução ao número de atendentes neste tipo de serviço, tendo como objetivo a conexão direta entre os clientes e as informações desejadas no sistema da empresa, eliminando o quadro elevado de atendentes. Os clientes podem chegar a informações pretendidas, respondendo as solicitações do protótipo feitas pelo telefone, o

protótipo fará novas solicitações ou retornará a informação solicitada. Os clientes podem, por exemplo, saber qual o próximo vencimento em aberto, escolhendo as opções correspondentes e informando o seu código como cliente da empresa, e o sistema irá fazer uma busca no banco de dados, retornando a informação sobre qual a data do próximo vencimento do cliente.

1.1 OBJETIVO GERAL

Desenvolver um protótipo para uma aplicação de comunicação entre telefone e computador, utilizando tecnologia Java e Unidade de Resposta Audível (URA).

1.2 OBJETIVOS ESPECÍFICOS

Podemos citar como objetivos específicos do trabalho em questão:

- a) realizar a comunicação entre telefone e computador;
- b) realizar a captura de valores digitados via teclado telefônico utilizando a biblioteca *Java Telephony API (JTAPI)* do Java, via conversão do sinal do *modem*;
- c) utilizar Java executando comandos *Structured Query Language (SQL)* com base nos valores digitados pelo cliente;
- d) desenvolver o protótipo de integração entre o telefone e o computador;
- e) demonstrar a integração entre telefone e computador, ligando de um telefone celular e recebendo Unidade de Resposta Audível (URA) do computador;
- f) utilizar sintetização de voz como retorno da aplicação para o usuário.

1.3 JUSTIFICATIVA

Desenvolver um protótipo realizando a integração entre telefone e computador via Unidade de Resposta Audível, agilizando o atendimento telefônico nas grandes e médias corporações.

Este protótipo vai realizar uma conexão direta entre o cliente com seu telefone e o sistema corporativo da empresa, onde estão às informações desejadas por este cliente, que atualmente são fornecidas por uma comunicação direta entre o cliente e os atendentes desta empresa, via telefone.

Nas empresas existe a necessidade de um enorme quadro de funcionários, devido à abrangência desta área de tele-atendimento e ainda a concorrência faz com que as empresas mantenham no ar um serviço destes 24 horas. Estas informações solicitadas pelos clientes são consultadas no sistema e informadas pelos atendentes, o protótipo visa realizar esta conexão direta entre os clientes e o sistema da empresa eliminando os atendentes.

Utilizando o protótipo de integração entre telefone e computador via Unidade de Resposta Audível, os empresários possuem a solução para manter este serviço de atendimento 24 horas, podendo reduzir os seus custos com funcionários e manter a qualidade na prestação de seus serviços.

1.4 ESTRUTURA DO TRABALHO

Este trabalho é composto por uma série de pesquisas com o intuito de contemplar a fundamentação teórica e auxiliar na prática da integração entre telefone e

computador. Com base neste objeto de estudo o segundo capítulo foi destinado ao histórico e estruturação do sistema telefônico.

Os métodos de transmissão de informações pelo sistema telefônico foram descritos no terceiro capítulo.

O *modem*, equipamento responsável pela conversão do sinal analógico para digital foi descrito no quarto capítulo.

A utilização de voz no computador foi descrita no quinto capítulo, pois o retorno de valores do banco de dados para o cliente, é convertido para voz.

O sexto capítulo foi destinado ao banco de dados, pois este é essencial para o funcionamento deste trabalho.

A integração entre telefone e computador, suas funções e componentes necessários foram descritos no sétimo capítulo.

Quanto a API responsável pelo controle da chamada no computador, os possíveis modelos de chamada e o estado finito de máquina de cada objeto que realiza o controle de chamada, foram descritas no oitavo capítulo.

O nono capítulo é respectivo ao funcionamento da chamada controlado pelo computador, sendo uma descrição detalhada do seu funcionamento e cada passo que deve ser realizado neste processo.

O sistema de Unidade de Resposta Audível, que é a união de todos os itens anteriores esta descrito no décimo capítulo.

O décimo primeiro e décimo segundo capítulos, visam mostrar aplicações da área de telefonia e a aplicação prática desenvolvida deste trabalho.

2 SISTEMA TELEFÔNICO

Ao realizar a comunicação entre dois computadores próximos, utiliza-se o cabo, porém se esta distância ultrapassar o limite de alcance do sinal, a solução proposta se torna inviável, devido a quantidade de cabo utilizado e da infinidade de equipamentos, gerando alto custo. Em muitos países a passagem destes cabos por propriedades públicas é ilegal, obrigando então a utilização da melhor maneira possível dos recursos de telecomunicações existentes, para realizar esta comunicação (TANENBAUM, 1996).

2.1 ESTRUTURA DO SISTEMA TELEFÔNICO

Quando Alexander Graham Bell, patenteou o telefone existiu uma grande procura pelo seu invento. Inicialmente ele era vendido aos pares e o usuário devia conectá-los com um fio, os elétrons eram retornados no solo. O usuário para utilizar o aparelho para se comunicar com outras residências, deveria conectar fios a todas elas (FERNANDES, 2004).

Alexander Graham Bell notou que desta maneira em um ano, as cidades ficariam tomadas por fios, gerando uma total desorganização. Foi criada a *Bell Telephone Company* que criou a primeira estação de comutação em *New Haven, Connecticut* no ano de 1878. Era ligado um fio da estação de comutação até a residência ou escritório do usuário. Quando o usuário fosse efetuar uma ligação, este girava uma manivela no aparelho telefônico que gerava um som que chamava a atenção do operador da companhia telefônica, esse manualmente conectava o emissor da chamada ao receptor com um *jumper*, este modelo pode ser verificado na Figura 2.

As estações de comutação multiplicaram-se rapidamente, logo os usuários sentiram a necessidade de fazer ligações interurbanas, assim a *Bell System* teve de conectar as estações de comutação umas as outras, como na Figura 1, o que se tornou inviável devido a grande distância entre elas. A solução adotada foi à criação das estações de comutação de 2º nível que se conectavam as demais, como na Figura 3.

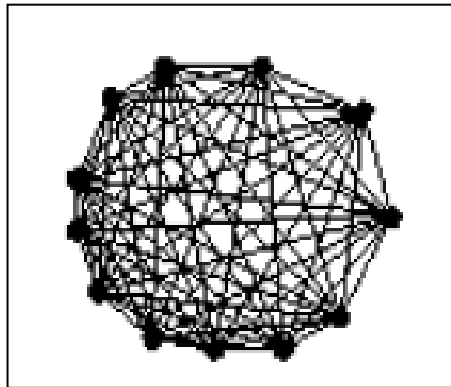


Figura 1 . Rede totalmente interconectada.
Fonte : FERNANDES, A. (2004).

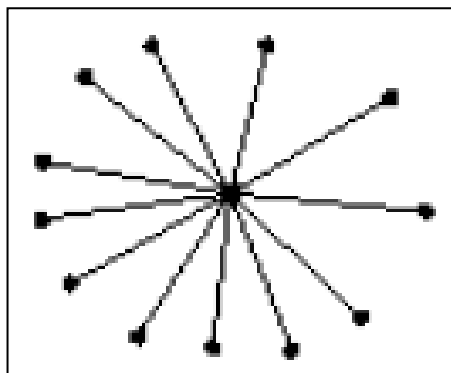


Figura 2 . Computador Centralizado.
Fonte : FERNANDES, A. (2004).

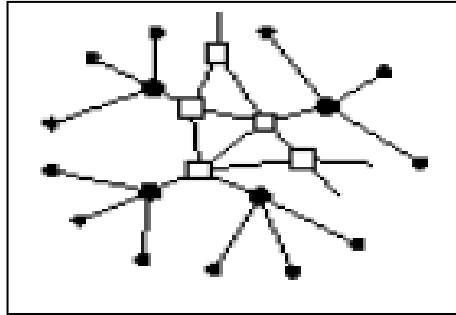


Figura 3 . Hierarquia de dois níveis.
Fonte : FERNANDES, A. (2004).

Atualmente o sistema telefônico é organizado em uma hierarquia de vários níveis e extremamente redundante. Este se encontra na atualidade em um estágio avançado tecnologicamente, porém mantendo a idéia inicial do sistema telefônico. Cada telefone mantém dois fios de cobre que saem do aparelho e se conectam diretamente a estação final mais próxima da companhia telefônica. Normalmente a distância é de 1 a 10 km, sendo menor nas cidades do que no campo.

As conexões realizadas entre o telefone e a estação final são conhecidas como *loop* local. Quando é utilizado o telefone para ligar a outro pertencente à mesma estação final Figura 4, o sistema de comutação cria uma conexão elétrica entre esses dois *loops* locais permanecendo intacta durante toda a chamada, porém se outro telefone pertencer à outra estação local será utilizado outro procedimento (FERNANDES, 2004).

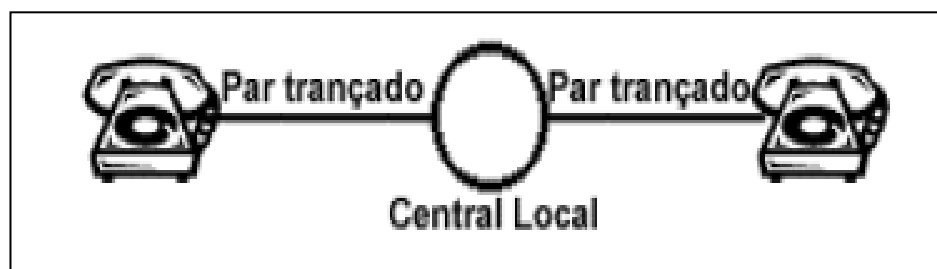


Figura 4 . Rota de uma chamada a pequena distância.
Fonte : TOTTEL, E. (2003).

Cada estação de comutação possui uma ou mais linhas de saída para se conectar às demais estações telefônicas, conhecidas, como estações interurbanas. Essas linhas utilizadas para conectar estações são conhecidas como troncos de conexões interurbanos.

Quando as estações finais do transmissor e do receptor possuem o seu tronco de comunicação interurbana ligado à mesma estação interurbana a conexão pode ser estabelecida dentro dela, Figura 5. Se o transmissor e o receptor não compartilham a mesma estação interurbana o caminho é estabelecido em um ponto mais alto da hierarquia. As estações interurbanas são ligadas por uma rede formada pelas estações principais, locais e regionais (FERNANDES, 2004).



Figura 5 . Rota de um circuito típico de uma chamada a média distância.
Fonte : TOTTEL, E. (2003).

2.2 TRANSMISSÃO DE INFORMAÇÕES

Os recursos de telecomunicações (rede telefônica), utilizados para realizar a comunicação entre computadores, foram projetados com o intuito de transmitir a voz humana de uma forma mais ou menos reconhecível.

Existe uma taxa de perda na transmissão de informações pelo sistema telefônico, esta diferença pode ser explicada fazendo um comparativo entre a comunicação e o computador por um cabo local e por uma linha telefônica de discagem. Ao se realizar esta transferência de dados na memória do equipamento utilizando-se um

cabo local a velocidade varia entre 10^7 e 10^8 *bps*, tendo uma taxa de erros muito baixa, mal podendo ser medida. Um erro por dia a essas velocidades corresponderia a um erro a cada 10^{12} ou 10^{13} *bits* enviados.

Utilizando a linha telefônica resulta uma taxa de transferência de dados na ordem de 10^4 *bps*, com uma taxa de erro de 1 para 10^5 *bits* enviados. A taxa de *bits* vezes o desempenho da taxa de erros seria 11 vezes melhor para a comunicação local do que uma linha telefônica de qualidade de voz.

No sistema telefônico existem vários meios de transmissão, para os *loops* locais é utilizada comunicação via par trançado, anteriormente se utilizava cabos de isolamento separados por 25 cm um do outro nos pólos telefônicos. Em estações de comutação são utilizados cabos coaxiais, microondas e com maior frequência fibras óticas.

O modo de transmissão utilizado era totalmente analógico, sendo o sinal de voz transmitido como uma voltagem elétrica da origem para o destino. Com a utilização dos equipamentos eletrônicos e computadores, se tornou possível à realização dessa transmissão digital, sendo esse sistema composto por apenas duas voltagens. O sistema permite que seja incluso um ou vários amplificadores de sinal para evitar a sua perda, assim, se percorre longas distâncias com uma pequena taxa de erros. Por outro lado no sistema analógico sempre ocorre perda de sinal quando amplificado, sendo a taxa de erro cumulativa conforme o número de ampliações.

As empresas telefônicas têm o custo menor na transmissão digital do que na analógica em razão da facilidade de identificação dos valores. Ainda a manutenção do sistema digital é extremamente simples, pois se um *bit* for transferido ou não, é suficiente para identificar se houve problema na transmissão.

Nos últimos tempos a transmissão analógica dominou as comunicações, na maioria dos troncos de comunicação que utilizam transmissão digital. Os *loops* locais permaneceram com sinal analógico devido ao alto custo para convertê-lo. Assim quando se envia dados por um computador pela linha telefônica é necessária a utilização de *modem* para converter o sinal digital para analógico do *loop* local. Em seguida o sinal é convertido para digital para ser transmitido pelos troncos de longa distância. O próximo passo é a conversão novamente para analógico no *loop* local na outra extremidade, a receptora, e, finalmente ele é convertido para digital novamente por um *modem* de outro computador, como verificado na Figura 6 (TOTTEL, 2003).

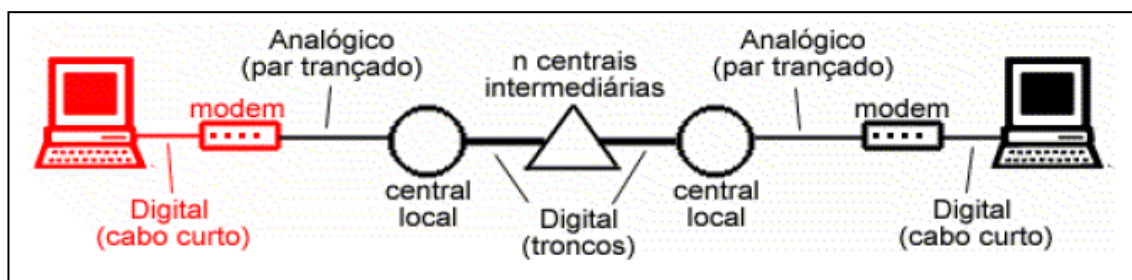


Figura 6 . O uso das transmissões analógica e digital para uma chamada entre dois computadores. A conversão é feita entre *modems*.

Fonte : TOTTEL, E. (2003).

2.3 PROBLEMAS NA TRANSMISSÃO

Na sinalização analógica é realizada uma variação de tensão representando o fluxo de informações, porém os dados recebidos não são os mesmos que foram transmitidos, em razão de o meio de transmissão não ser perfeito. Esse fato pode provocar erro na transmissão digital (DANTAS, 2002).

Na transmissão encontramos três problemas que são atenuação, retardo no tempo e ruído. Atenuação é a perda de força do sinal à medida que ele se propaga, podendo aumentar a quantidade dessa perda conforme maior à distância a ser percorrida

pelo sinal. Essa perda é medida em decibéis por quilometro, se a atenuação for grande demais talvez o receptor não consiga detectar o sinal. Em muitos casos, se sabe qual a atenuação gerada pelo sinal para ele chegar a determinado local. Para compensar esta perda poderiam ser utilizados amplificadores, porém com a implementação deste método não seria possível recuperar integralmente a forma original deste sinal.

O sinal de transmissão pode ser visto como uma seqüência de coeficientes de uma série de Fourier, cada coeficiente é atenuado por um diferente volume de freqüência, o que resulta em um outro espectro de Fourier e em outro sinal. Outro problema é a distorção causada pelo retardo que é ocasionado pela diferença de velocidade existente entre os diferentes coeficientes de Fourier. Na transmissão digital os coeficientes rápidos de um *bit*, podem alcançar os coeficientes lentos do *bit* seguinte, misturando os *bits* e ocasionando erro na recepção do sinal (DANTAS, 2002).

O ruído é gerado por alterações de energia no meio de transmissão. O ruído térmico é ocasionado pelo deslocamento aleatório dos elétrons em um fio. A linha cruzada é ocasionada pelo acoplamento indutivo entre dois fios. O ruído por impulso, é provocado dentro de outros fatores, por grampos na linha de força. Na transmissão digital o ruído por impulso pode provocar a perda de um ou mais *bits*. Com o intuito de diminuir o número de erros nestas transmissões, são estudadas várias maneiras de realizar essa transferência de informação pelo sistema telefônico.

3 MÉTODOS DE TRANSMISSÃO DE INFORMAÇÕES

As companhias telefônicas perceberam que o custo de instalação de um tronco de banda larga e um tronco de banda estreita entre duas estações era quase o mesmo, então foi elaborado esquemas para multiplexar conversões em um único tronco físico. Podemos dividir estes esquemas em duas categorias, *Frequency Division Multiplexing (FDM)* e *Time Division Multiplexing (TDM)*. Utilizando a *FDM* o tronco é dividido em frequências e cada usuário utiliza uma faixa de frequência diferente e na *TDM* é realizado um revezamento entre os usuários em que cada um deles em determinado momento utiliza a faixa toda. (TANENBAUM, 1996).

Compreendendo melhor a *FDM* e *TDM*, pode se analisar a transmissão de rádio, era uma linha de transmissão e cada estação operadora. Cada operadora possui uma frequência sendo que a faixa é implementada de maneira a não criar interferência nas demais, então essa transmissão é do tipo *FDM*. Agora se esta estação quer alternar esta frequência enviando, por exemplo, um pouco de música e depois um pouco de propaganda utilizando a mesma frequência então se tem *TDM*.

3.1 MULTIPLEXAÇÃO POR DIVISÃO DE FREQUÊNCIA

Utilizando três canais telefônicos de nível de voz multiplexados, os filtros limitam a largura de banda de 3000 *Hz* para cada canal. Utilizando vários canais para multiplexar são alocadas faixas de banda de 4000 *Hz* para cada canal, com o intuito de assegurar que eles estejam bem separados (GALLO, 2003).

Estes canais terão sua frequência aumentada, cada um com valor diferente, os quais depois podem ser combinados, assim eles ocuparão o mesmo espectro, como

pode ser verificado na Figura 7. Mesmo possuindo intervalos entre os canais, existe uma sobreposição entre canais adjacentes, porque os filtros não possuem extremidades uniformes. Essa sobreposição indica que um pico forte na extremidade de um canal será sentido no adjacente como ruído não térmico.

Devido à utilização de *FDM* se tornar difundida em todo mundo, houve a necessidade de criar uma padronização para ela. O mais difundido é o padrão com 12 canais de voz de 4000 *Hz* multiplexados na banda de 60 a 108 *kHz*, o que pode ser chamado de grupo, esta banda pode ainda ser utilizada por outro grupo. Algumas companhias criaram uma linha privada juntando estes grupos, podendo unir cinco grupos formando um super-grupo, depois podendo unir cinco ou dez super-grupos e criar o grupo-mestre (FALBRIARD, 2002).

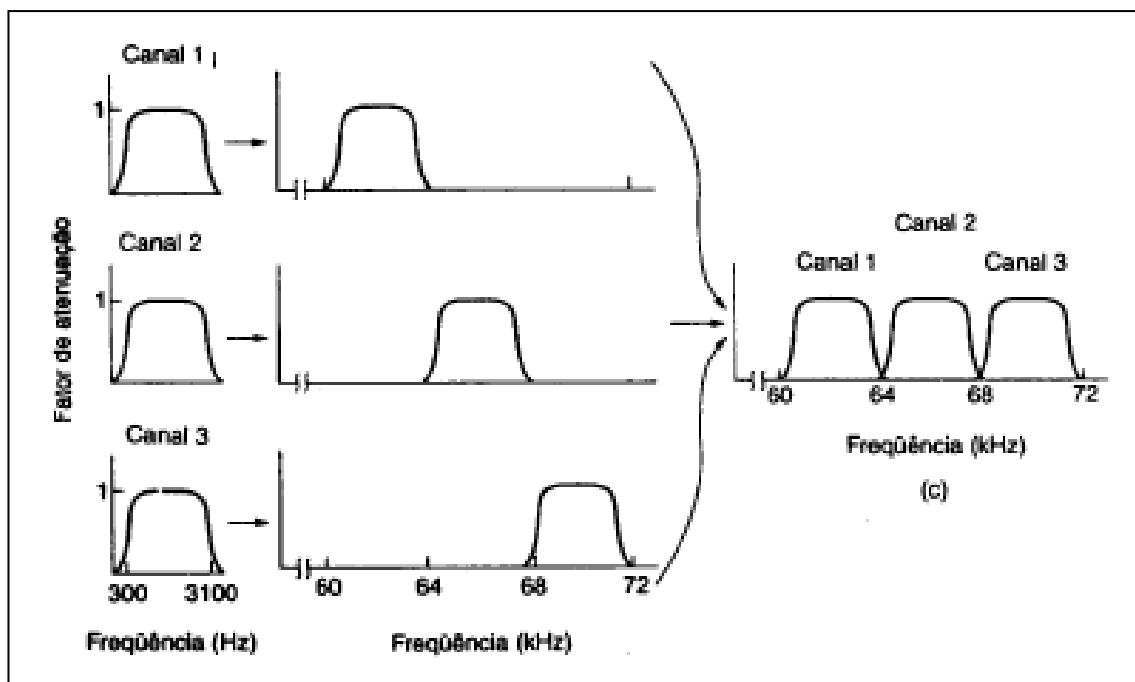


Figura 7 . Os canais tem sua frequência aumentada, cada qual com um valor diferente, logo após são combinados.
Fonte : GALLO, M. (2003).

3.2 MULTIPLEXAÇÃO POR DIVISÃO DE COMPRIMENTO DE ONDA

Utilizando canais de fibra ótica ocorre uma variação na multiplexação por divisão de frequência, se tornando multiplexação por divisão de comprimento de onda, os dois fios chegam até um prisma cada um com sua energia e faixa diferente, então são refletidos pelo prisma e transmitidos por uma fibra compartilhada até outro prisma que divide este sinal novamente, Figura 8 (GALLO, 2003).

Esse tipo de multiplexação é bastante difundido, pois a banda de uma única fibra é de aproximadamente 25000 *Ghz*, existindo a possibilidade de utilizar a multiplexação de vários canais nesta única fibra, sempre respeitando o princípio de que cada canal deve possuir uma frequência diferente.

No sistema de comprimento de onda fixo, existem muitas fibras de entrada e de saída sendo em geral, o acoplador uma estrela passiva onde a luz de cada fibra, ilumina a esta estrela. Mesmo espalhando essa energia por *n* saídas e diluí-las em fator *n*, este sistema ainda é prático para centenas de canais (DANTAS, 2002).

A luz de uma fibra ótica que está a 1,50206 *micra* se deslocar para uma fibra de saída, e todas as fibras de saída utilizam filtros sintonizáveis para que esta fibra possa ser definida como 1,50206 *micra*, estes filtros são criados a partir de interferômetros *Fabry-Perot* ou *Mach-Zehnder*.

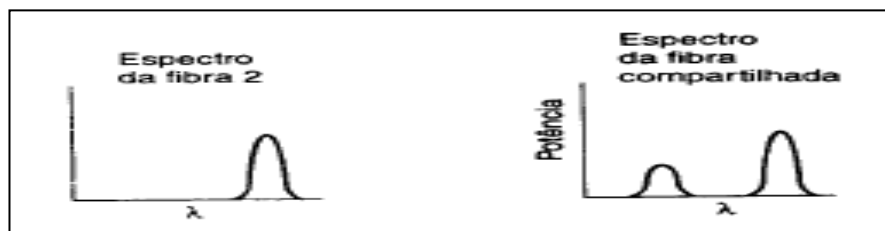


Figura 8 . Multiplexação por divisão de comprimento de onda.
Fonte : DANTAS, M. (2002).

3.3 MULTIPLEXAÇÃO POR DIVISÃO DE TEMPO

A multiplexação por divisão de tempo vem sendo muito utilizada, pois pode se trabalhar diretamente com circuitos digitais, enquanto na multiplexação por divisão de frequência mesmo com o advento das fibras ópticas e canais de microondas, ainda necessita-se do meio analógico. No entanto, um pouco de desvantagem na multiplexação por divisão de tempo é que só pode ser realizada, à transmissão de dados digitais por meio dela, como na Figura 9 (GALLO, 2003).

Os *loops* locais utilizam sinal analógico na implementação deste tipo de multiplexação. O sinal deve ser convertido para digital na estação final, pois nela todos *loops* locais chegam juntos para serem combinados em troncos de saídas.

Segundo Dantas (2002), os sinais analógicos são convertidos para digitais por um dispositivo conhecido como *Codec*, gerando um número de 7 à 8 *bits*, este dispositivo com 8000 amostragens por segundo, pois de acordo com o teorema de *Nyquist* este é o número de amostragens ideal para captura das informações da largura de banda do canal telefônico, técnica esta conhecida como *Pulse Code Modulation (PCM)*.

O surgimento da transmissão digital e a falta de padronização dela, acabou gerando uma certa incompatibilidade entre as linhas de transmissões de muitos países. Problema sério, pois para corrigir esta diferença na padronização de transmissão, sendo necessário a utilização de caixas pretas, as quais representam enormes custos para as companhias. Alguns países utilizam o método de concessionária, formada por 24 canais de voz multiplexados, realizando uma amostragem alternada dos sinais analógicos enviados para um *codec*. Cada um dos 24 canais por sua vez, consegue inserir, 8 *bits* no fluxo de saída. Nessa transmissão 7 *bits* são dados e 1 é para controle, gerando 7 X

$8.000 = 56.000 \text{ bps}$ e $1 \times 8.000 = 8.000 \text{ bps}$ de informações de sinalização por canal (SOARES, 1995).

O sinal de voz quando digitalizado utiliza técnicas estatísticas para reduzir o número de *bits* necessários por canal, além da voz pode se implementar estas técnicas para reduzir o número de *bits* de qualquer sinal analógico. Os métodos de compactação utilizados partem do princípio que a mudança de sinal é mais lenta que a frequência de amostragem, assim as informações digitais de nível 7 e 8 *bits*, se tornam redundantes.

A modulação de código de pulso diferencial produz a diferença entre o valor atual e o anterior e não na amplitude digitalizada, com variações de ± 16 em uma escala de 128, as quais podem ser consideradas inviáveis, pois 5 *bits* podem ser suficientes ao invés de 7. se o sinal gerar muitas variações na sua captura, a própria codificação gera um processo de amostragem mais minucioso para recuperar as informações perdidas, utilizando voz nessa transmissão, pode-se desconsiderar o erro (DANTAS, 2002).

Neste método de compactação, se consegue uma variação utilizando uma amostragem com valores diferenciados entre +1 e -1 de seu predecessor. O processo pelo qual, à nova amostragem, fica acima ou abaixo do predecessor é conhecido como modulação delta. Esta modulação poderá apresentar problemas se o sinal alternar muito rapidamente, pois assim as informações serão perdidas. Um processo de aperfeiçoamento da *PCM* diferencial é a codificação por previsão, nela são extrapolados os valores anteriores com o intuito de prever os posteriores, codificando a diferença entre o valor atual e o previsto, porém o transmissor e o receptor devem utilizar o mesmo algoritmo de previsão.

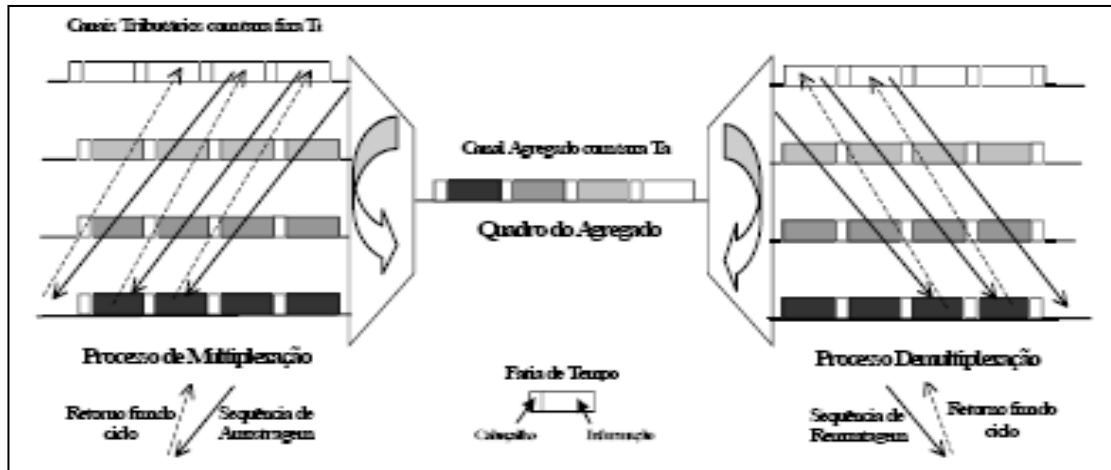


Figura 9 . Multiplexação por divisão de tempo.
 Fonte : GALLO, M. (2003).

3.4 SONET / SDH

Com a utilização de fibras óticas, inicialmente cada companhia possuía seu próprio sistema óptico. Posteriormente quando o AT&T foi dividida, as companhias telefônicas tiveram que se unirem à diversas concessionárias de comunicação a longa distância, tornando obrigatório à criação de uma padronização para este meio. O padrão criado foi o *Synchronous Optical Network (Sonet)*, que posteriormente teve a adição de um conjunto de recomendações paralelas do *Corporate Core Compensation Implementation Team (CCCIT)*. Estas recomendações são conhecidas como *Synchronous Digital Hierarchy (SDH)*, podem ser diferenciadas do *Sonet* apenas por pequenos detalhes (FALBRIARD, 2004).

Este padrão foi criado com 4 objetivos principais, primeiro ele deveria realizar a ligação entre diferentes concessionárias em rede, para isso foi definido um padrão para sinalização que respeitasse o comprimento das ondas, a sincronização, a estrutura de enquadramento e outros aspectos. Em segundo lugar foi necessário unificar os sistemas digitais baseados em canais *PCM* de 64 *Kbps* que realizavam esta combinação, mas de diferentes formas. Em terceiro existia a necessidade de multiplexar

vários canais, com o intuito de ampliar a velocidades ainda maiores, algo na casa de *gigabits* e por último era necessário fornecer recursos de operação, administração e manutenção.

3.5 COMUTAÇÃO

O sistema telefônico pode ser dividido em duas partes, a planta interna (*loops* e troncos locais) e planta externa (comutadores), podendo os circuitos de comutação ser dividido em comutação de circuitos e comutação de pacotes. Na comutação de circuitos quando se realiza uma ligação pelo computador ou com a utilização do aparelho telefônico, o aparelho de comutação procura um meio físico para ligar o aparelho de origem, da ligação até o receptor da mesma (FABIANO, 2004).

Neste tipo de comutação é determinado primeiro todo o trajeto que a ligação irá percorrer para depois a mesma ser iniciada. Dessa forma não se encontra congestionamento, pois após definir o trajeto da ligação o único atraso na entrega de suas informações é o próprio de tempo de propagação do sinal.

Estes vários métodos de transmissão de informações e toda estrutura do sistema telefônico, motivou os fabricantes de equipamentos da área de telecomunicações a aperfeiçoarem seus equipamentos, até certa hora que sentiu-se a necessidade de todos estes equipamentos conversarem da mesma maneira, foi onde surgiram os protocolos de comunicação.

4 MODEMS

Os sinais utilizados no sistema telefônico são digitais limitados por uma faixa de frequência de 300 *Hz* a 3 *kHz*. Utilizando sinal digital dessa forma, o sinal recebido do outro lado não terá uma forma de onda quadrada, devido a efeitos de capacitância e de indutância no sinal. Deste modo a sinalização digital só pode ser realizada a baixas velocidades e em curtas distâncias, porém utilizando a sinalização analógica, se pode contornar esses problemas (SOARES, 1995).

4.1 FUNCIONAMENTO DOS *MODEMS*

Na sinalização analógica é utilizado um tom contínuo na faixa entre 100 e 2000 *Hz*, chamado de portadora senoidal. A sua amplitude, frequência ou fase podem ser moduladas para transmitir informação. Na modulação de amplitude são utilizados dois níveis diferentes de voltagem para representar 0 e 1. Na modulação por frequência são utilizadas dois ou mais tons. Na modulação de fase mais comum a onda portadora é sistematicamente deslocada de 45, 135, 225 ou 315 graus em intervalos uniformes (COMER, 1997).

Cada mudança de fase transmite dois *bits* de informação, sendo que para tal é utilizado um dispositivo que aceita um fluxo serial de *bits* de entrada e produz uma portadora modulada na saída (ou vice-versa), dispositivo este que é chamado *modem* (modulador - demodulador). O aparelho é colocado entre o computador e o sistema telefônico. Nas Figuras 10 e 11, pode ser verificado respectivamente a modulação de 3 e 4 *bits* por amostra.

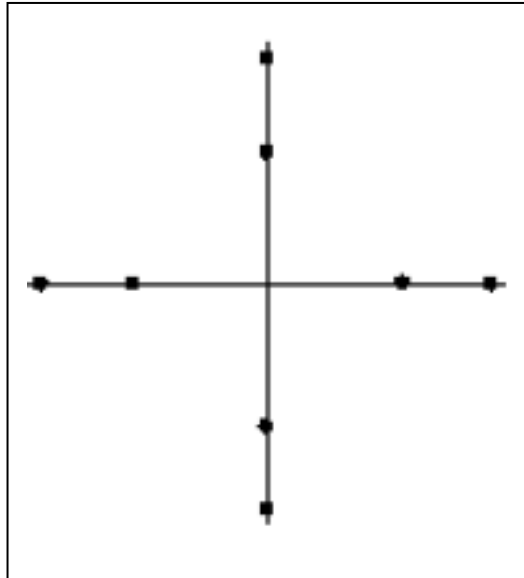


Figura 10 . Modulação por 3 bits por amostra.
Fonte : SOARES, L. (1995).

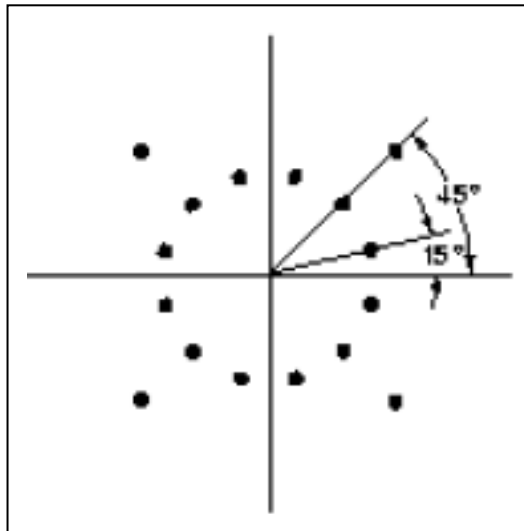


Figura 11 . Modulação de 4 bits por amostra.
Fonte : SOARES, L. (1995).

Segundo Dodd (2000), existem *modems* avançados onde são utilizados dois níveis de amplitude por mudança de fase, e a amplitude é indicada pela distância de origem. O esquema de modulação é diferente utilizando intervalos de 30 graus,

possuindo oito mudanças de fase em uma amplitude legal, porém as outras quatro permitem dois valores resultando no total de dezesseis combinações. Quando se utiliza esse esquema para transmitir 9600 *bps* em uma linha de 2400 *baud*, se chama essa transmissão de Quadratura da Amplitude (QAM).

Pode acontecer de se encontrar ecos na junção da malha do assinante, pois ela possui um circuito com dois fios enquanto o tronco possui um circuito com quatro fios. Para exemplificar este efeito, se pode dizer que ele acontece quando se fala algo no telefone e depois de algum retardo escutamos novamente a voz. Para eliminar este problema são instalados supressores de ecos nas linhas com mais de 2000 km, pois em linhas curtas esse efeito é tão rápido que se torna imperceptível à audição humana.

O supressor de eco é um aparelho que ao detectar a voz humana trafegando por uma direção, trava todos os outros sinais que percorrem a direção oposta. Quando a primeira pessoa começa a falar ele permite o tráfego apenas nesta direção, quando esta para de falar ele alterna o sentido permitindo apenas a segunda pessoa falar, como pode ser verificado na Figura 12 (DODD, 2000).

A utilização dos supressores de eco foi vital para o sistema telefônico, pois sem a sua utilização o sistema telefônico poderia ser incomodo para seus usuários, estudos psiquiátricos indicam que o eco pode ocasionar gagueira e confundir os usuários. Na comunicação de dados este aparelho pode acarretar uma série de problemas, primeiro impedindo a transmissão *full duplex* onde os dados transitam nos dois sentidos ao mesmo tempo e até a transmissão *half duplex* onde é utilizado um sentido por vez na transmissão.

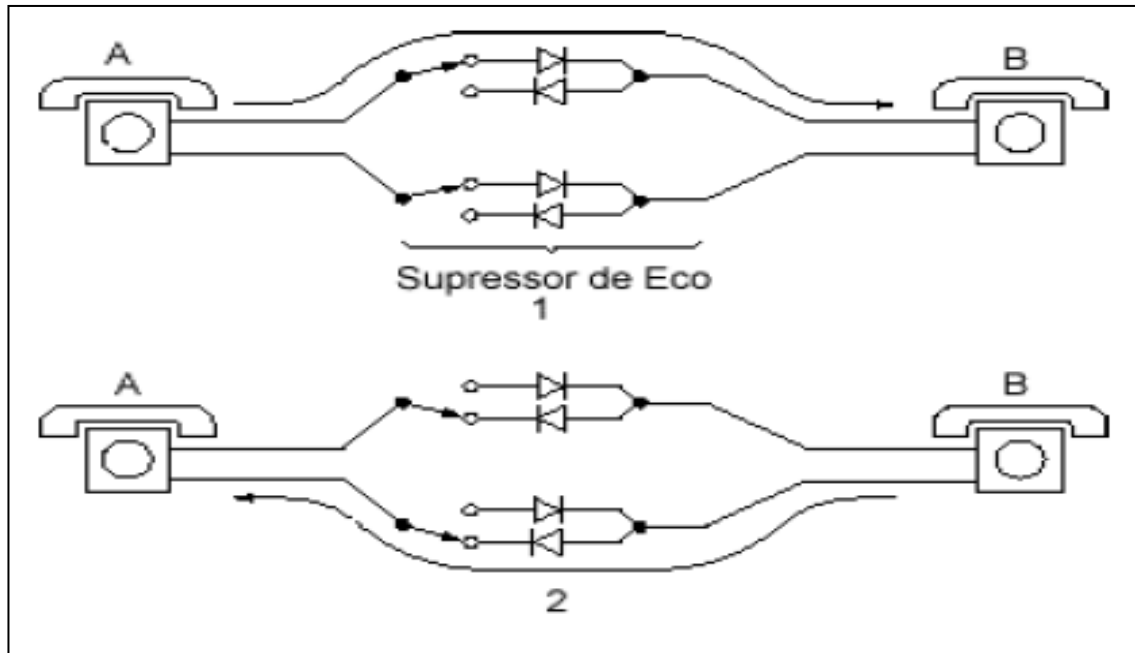


Figura 12 . Supressores de eco.
Fonte : DODD, A. (2000).

O tempo necessário para troca de sentido pode ser substancial. Este equipamento foi projetado para fazer o chaveamento com a voz humana e não dados digitais. Para resolver este problema foi implementado nos supressores de eco o seu desligamento, quando os mesmos receberem um tom puro de 2100 Hz , assim permanecendo enquanto estiver presente uma portadora, com isso, certos sinais de controle que ativam e desativam as funções internas de controle ficam na banda do usuário (COMER, 1997).

4.2 INTERFACE ENTRE O MODEM E O COMPUTADOR

A interface entre o *modem* e o computador é um exemplo de protocolo da camada física, nela devem ser especificadas as interfaces mecânica, elétrica, funcional e procedural. Os padrões mais conhecidos na camada física são o RS-232-C e o RS-449 (DODD, 2000).

O RS-232-C é a terceira revisão do RS-232, este padrão foi criado pela *Electronic Industries Association (EIA)*, uma organização comercial de fabricantes eletrônicos, sendo chamado de *EIA RS-232-C*. Nos padrões o terminal de computador é chamado *Data Terminal Equipment (DTE)*, e o *modem* é chamado de *Data Circuit-Terminating Equipment (DCE)*.

A especificação mecânica é para um conector de 25 pinos com largura de $47,04 \pm 0,13$ mm, e com outras dimensões igualmente bem especificadas. A fila superior tem pinos enumerados de 1 a 13, a fila inferior tem pinos numerados de 14 a 25. A especificação elétrica do RS-232-C é de que uma voltagem mais negativa de -3 volts é 1 binário e uma voltagem mais positiva de +4 volts é 0 binário. Sendo utilizadas taxas de dados de até 20 *Kbps* e cabos de até 15 metros (COMER, 1997).

Na especificação funcional são indicados quais circuitos são conectados a cada um dos 25 pinos e qual o seu significado, geralmente são implementados nove pinos, o restante é omitido. O computador ligado, ativando o *Data Terminal Ready* (pino 20). O *modem* ligado ativa o *Data Set Read* (pino 6). Ao detectar uma porta na linha telefônica, o *modem* liga o *Carrier Detect* (pino 8). O *Request to Send* (pino 4) indica que o terminal deseja transmitir dados. O *Clear to Send* (pino 5) indica que o *modem* está pronto para receber dados.

Os dados são transmitidos por circuito *Transmit* (pino 2) e recebidos no circuito *Receive* (pino 3). Também possui outros circuitos para selecionar a taxa de dados, testar o *modem*, fazer o *clock* dos dados, detectar sinais de chamada e enviar dados no sentido inverso, como pode ser verificado na Figura 13.

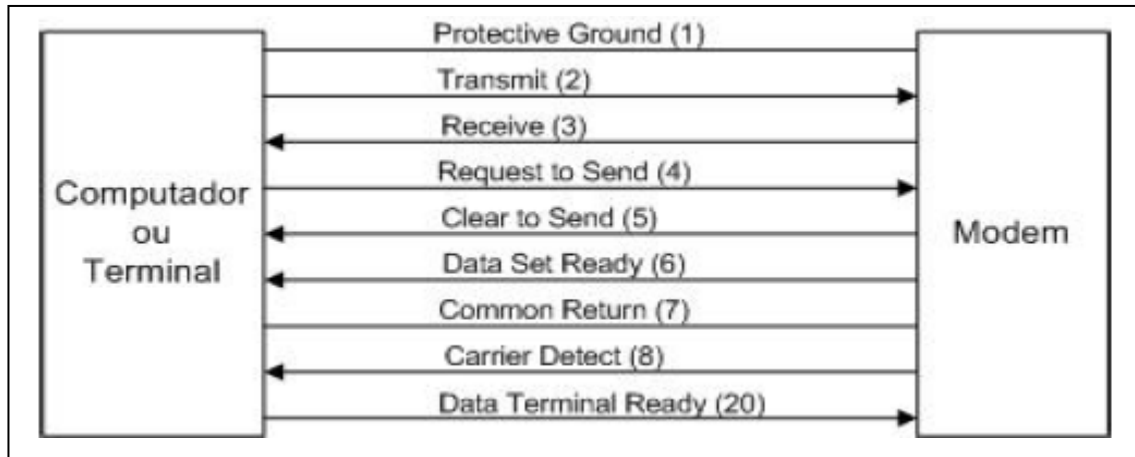


Figura 13 . Padrão RS-232-C.
Fonte : DODD, A. (2000).

O protocolo 232 é a especificação procedimental, ou seja, como devem ocorrer os eventos, sendo baseados em pares de reação-ação. Quando dois computadores querem ser conectados utilizando o RS-232-C e nenhum deles é *modem*, é utilizado um equipamento chamado *null modem* que liga a linha de transmissão de uma máquina na linha de recepção da outra, eles também podem cruzar algumas das outras linhas de modo similar (SOARES, 1995).

O RS-232-C possui limitações de taxa de dados até 20 *kbps* e comprimento máximo de cabo de 15 metros, os quais se tornaram cada vez mais incômodos. A *EIA* foi obrigada a criar um novo padrão que fosse compatível com o antigo ou um novo padrão que não fosse compatível com o antigo, mas, ao mesmo tempo satisfizesse todas novas necessidades do mercado. Resumindo a *EIA* escolheu os dois padrões.

O padrão criado foi o RS-449, que na verdade são três padrões em um, as interfaces mecânicas, funcionais e procedimentais. A interface elétrica é utilizada em outros dois padrões: o RS-423-A similar ao RS-232-C que utiliza transmissão balanceada, ou seja, todos os circuitos compartilham o mesmo pino terra e o RS-442-A que usa transmissão balanceada em que cada um dos circuitos principais utilizada dois fios, assim este padrão pode transmitir com uma velocidade de até 2 *Mbs* em cabos de

60 metros, e chegar a velocidades maiores conforme for diminuindo a distância destes cabos (FABIANO, 2004).

No padrão RS-449 foram incluídos muitos circuitos que não existiam no padrão RS-232-C, como por exemplo, para testar o *modem* localmente ou remotamente. No entanto com estes novos circuitos não era mais possível manter o padrão com 25 pinos, agora sendo necessário um conector de 37 pinos e outros com 9 pinos, o conector de 9 pinos utilizando apenas se o segundo canal (reverso) estiver sendo utilizado.

O *modem* permitiu a ligação entre o computador e o sistema telefônico, então as tecnologias na área de integração entre estes dois meios vem crescendo vertiginosamente. Os desenvolvedores visam aperfeiçoar esta ligação, criando *softwares* de acesso remoto a computadores via telefone, porém para ter um retorno deste computador para o telefone foi necessário o estudo e a implementação da utilização voz no computador.

5 UTILIZAÇÃO DE VOZ NO COMPUTADOR

A expressão reconhecimento de voz é utilizada com vários sentidos, que indicam tecnologias diferentes. O processamento de voz pode ser aplicado em quatro áreas principais: comando de voz, fala natural, síntese de voz e autenticação de voz (DANTAS, 2002).

O reconhecimento de palavras caracteriza-se por utilizar apenas um pequeno trecho de fala, identificando o tipo de ação que o sistema deve tomar. Este processamento torna-se simplificado, sendo que o sistema já sabe as palavras que podem ser utilizadas. Este é o caso das centrais de atendimento telefônico, onde o usuário pode usar voz em vez de botões.

O reconhecimento de fala natural envolve uma ou mais frases, isto é, várias palavras que tenham um sentido semântico, Figura 14. A fala reconhecida é então convertida em texto. O tipo de aplicação mais comum para essa tecnologia é o ditado de documentos, para o uso em processadores de texto, escrita de *e-mails*, entre outros.

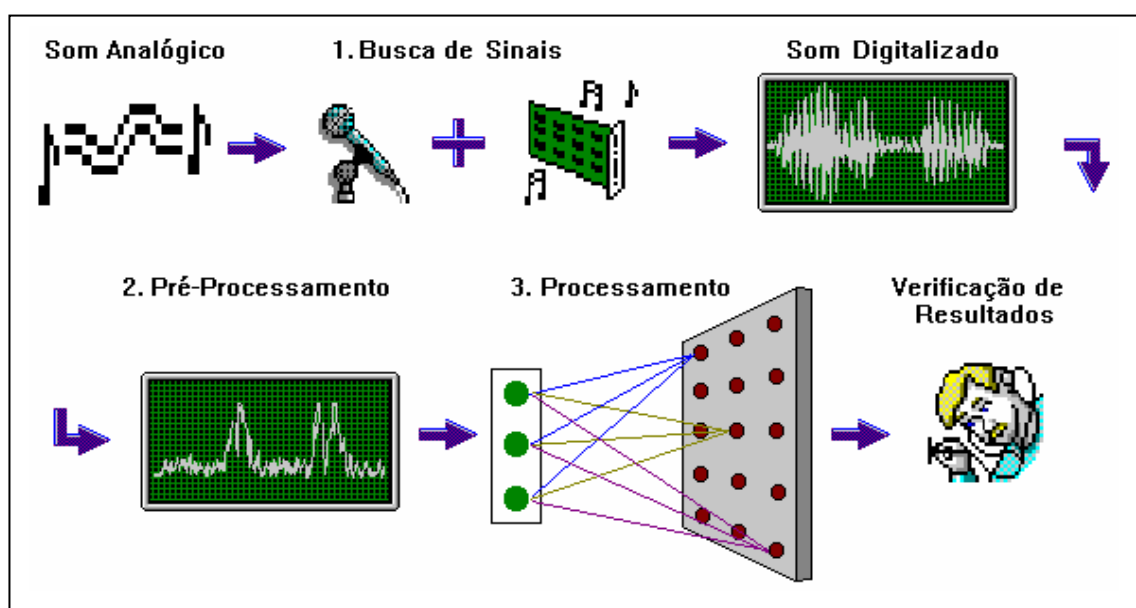


Figura 14 . Reconhecimento de fala.
Fonte : DANTAS, A. (2002).

Segundo Barros (2002), a síntese de voz é o processo contrário ao do reconhecimento da fala. O sintetizador recebe um texto na forma digital e transforma-lo em ondas sonoras, ou em outras palavras, fazendo uma leitura em voz alta. Um programa de síntese de voz é útil nas situações em que o usuário não pode desviar a atenção para ler algo ou não tem acesso ao texto escrito, seja porque a informação está distante ou porque o usuário tem alguma deficiência visual.

A autenticação baseia-se, por sua vez no fato de que a voz é única para cada pessoa e pode ser utilizada para identificar alguém. Os sistemas de autenticação podem ser aplicados para permitir o acesso de uma pessoa para uma determinada função.

5.1 RECONHECIMENTO DE VOZ

O reconhecimento de voz evoluiu bastante nos últimos anos. Inicialmente o reconhecimento, processava-se apenas em modo discreto, isto é, o usuário fazia pausas entre uma palavra e outra ditada. O reconhecimento tornou-se mais inteligente, uma vez que possui um conjunto de regras gramaticais incorporadas, permitindo assim perceber melhor o que há para ser ditado (BARROS, 2002).

O reconhecimento de voz usa diferentes técnicas para reconhecer a voz humana. Funciona como um *pipeline* que transforma os sinais áudios digitais provenientes da placa de som em fala reconhecida. Estes sinais passam por diversas etapas, ao longo das quais são aplicados métodos matemáticos e estatísticos de forma a tentar compreender o que está sendo ditado.

Quando o usuário fornece um comando de voz pelo microfone, este é transmitido para a placa de som, sendo o sinal analógico amostrado 16000 vezes por segundo. Depois este é transformado para digital por uma técnica chamada *Pulse Code*

Modulation (PCM). Esta forma de sinal é um *stream* de amplitudes representando o sinal analógico (DANTAS, 2002).

O *software* de reconhecimento de voz não pode trabalhar diretamente com base neste *stream*, sendo muito complicado procurar padrões que possam ser relacionados com o que está atualmente a ser ditado. Este é transformado no conjunto de bandas de frequência discreta, usando uma técnica chamada *Windowed Fast Fourier Transform (FFT)*. É uma amostragem do sinal de áudio a cada centésimo de segundo, mas desta vez no domínio de frequência. Assim, a *stream* de entrada é agora representada por um conjunto de bandas de frequências discretas, podendo ser facilmente identificados os componentes de frequência de um som. A partir destes componentes, é possível aproximar-nos do que o ouvido humano ouve.

O *software* de reconhecimento de voz possui uma base de dados de milhares de frequências ou fonemas. Um fonema é a menor unidade de fala de um idioma ou dialeto. A dicção de fonemas é diferente entre si, tal que ao substituir um fonema na palavra, esta passa a ter um significado totalmente diferente. O *software* compara o fonema na base de dados, ao encontrá-lo é atribuído ao sinal de entrada o número identificador do fonema na base de dados, também chamado *feature number*.

5.2 APLICAÇÕES COM RECONHECIMENTO DE VOZ

O reconhecimento de voz representa ganhos significativos no processo de transcrever documentos para processadores de texto. Tirando vantagem do fato de que o ser humano é capaz de ditar sete vezes mais rápido que ao escrever, consegue-se em alguns casos um melhor desempenho em 60% (DANTAS, 2002).

Na indústria da saúde que se depara atualmente com fatores críticos de sucesso como a redução das despesas e aumento da eficiência. O reconhecimento de voz pode ajudar as equipes médicas eliminando a necessidade de transcrever manualmente os relatórios médicos, bastando para isso o uso de um pequeno aparelho portátil, parecido com um gravador, enquanto se procede ao diagnóstico dos pacientes.

Nas seguradoras, os ganhos podem ser imensos. Imaginando que um inspetor de seguros está a avaliar um sinistro no terreno. Ao usar o reconhecimento de voz, é possível que ao mesmo tempo em que descreve o caso, os dados do processo comecem a chegar a seguradora, permitindo que o cliente veja o seu caso resolvido muito mais rapidamente.

Nas centrais telefônicas automáticas, o usuário pode dizer naturalmente que deseja falar com uma determinada pessoa e o sistema repassa a chamada para o ponto correspondente. Em caso de dúvida, por exemplo, no caso em que se pede para falar com uma determinada pessoa e existem mais pessoas com o mesmo nome na empresa/organização, o sistema interage com o cliente dando alternativas para que a escolha seja feita.

O reconhecedor de voz também necessita saber quando é que um fonema acaba e outro começa. Para isso, utiliza-se uma técnica matemática conhecida como *Hidden Markov Models (HMM)*. Admitindo que o reconhecedor de voz, que registrou os *feature numbers* correspondentes a dois fonemas consecutivos de uma palavra, torna-se complicado, a olho humano, distinguir onde começam e acabam os fonemas, sobretudo se estes dois fonemas tiveram algum *feature number* em comum. É aqui que entra a técnica das *HMM*, que existe na explosão combinatória de dar possibilidade para qualquer fonema a ser seguido de qualquer outro fonema, ligados por meio de transições

com pesos associados, até que se torna possível distinguir com clareza onde começa e acaba o fonema (BARROS, 2002).

5.3 SINTETIZAÇÃO DE VOZ

Segundo Dantas (2002), a sintetização de voz consiste na fala por computador de textos escritos. Os sintetizadores texto-fala são programas de computador que admitem como entrada um texto irrestrito escrito em língua natural e produzem como resultado o som correspondente à leitura deste texto. Quando submetido a um Sintetizador Texto-Fala (STF), o texto passa por uma cadeia de transformações dando origem a uma série de códigos intermediários antes da obtenção do código final (sinal de áudio). A cada passo são inferidas novas propriedades sobre o texto nos seguintes níveis :

- a) nível léxico são observados os aspectos referentes a formação das palavras que constituem a linguagem. O processamento léxico (ou morfológico), por exemplo, determina a seqüência de caracteres que representa uma determinada palavra válida, um número ou uma abreviatura;
- b) nível sintático são analisados a estruturação do texto em parágrafos, períodos, orações e o papel de cada termo dentro destas alturas;
- c) nível prosódico são verificados os aspectos referente à pronuncia do texto, tais como : duração das palavras/sílabas, acentuação, ritmo e entonação. Algumas características prosódicas do texto encerram um grau de conteúdo semântico, o que pode vir a dificultar seu tratamento;

- d) nível fonético é tratado o problema da divisão de um termo transcrito ortograficamente em unidades fonéticas (grupo de um ou mais fonemas) e sua correspondente tradução em código fonético.

Um STF é, portanto, constituído de duas unidades básicas: um módulo de processamento lingüístico que cuida da análise do texto-fonte de um modo geral e de um sintetizador que é capaz de gerar o código do som do texto a partir de um conjunto adequado de parâmetros.

A utilização de voz no computador gerou uma série de avanços tecnológicos, socializando pessoas com deficiências visual, oral entre outras e também permitiu a utilização deste recurso com aplicações de diversas áreas, tendo como grande aliado desta tecnologia a consulta de valores em banco de dados e sintetização destas posteriormente.

6 BANCO DE DADOS

Segundo Pereira (2002), os sistemas de processamento de informações são desenvolvidos e aperfeiçoados, devido à importância fundamental que o controle das informações geradas sobre nossas vidas e das corporações. Podemos citar ferramentas desta área como : processadores de texto (editoração eletrônica), planilhas (cálculos com tabelas de valores) e Sistemas de Gerenciamento de Banco de Dados (SGBD).

6.1 SISTEMA DE BANCO DE DADOS

O banco de dados pode ser definido como um conjunto de informações inter-relacionadas representando determinado domínio, como uma lista telefônica ou um sistema de controle de estoque. No entanto deve existir uma maneira de se controlar este conjunto de informações a fim de aumentá-lo, diminuí-lo ou alterá-lo. Para esses casos foram desenvolvidos os sistemas de gerenciamentos de banco de dados que tratam de softwares com recursos específicos que permitem a manipulação destas informações (MONTEIRO, 2004).

O sistema de banco de dados é responsável pela manutenção de registros. Este sistema é formado por dados, hardware, software e usuários. Podemos considerar o sistema de banco de dados como uma sala de arquivos eletrônicos que possuem uma série de métodos, técnicas e ferramentas que visam sistematizá-las. Um exemplo de sistema de banco de dados pode ser visto na Figura 15.

O banco de dados tem como principais objetivos a abstração de dados. O usuário não é colocado a par das transições internas do banco de dados, pois o usuário só deseja realizar suas alterações no banco de dados. Prover independências de dados às

aplicações, armazenamento das informações do sistema em um conjunto diferente possuindo um controle de acesso com usuário e senha (PEREIRA, 2002).

Segundo Monteiro (2004), a organização que utiliza um bom sistema de banco de dados possui uma série de vantagens, dentre elas podemos destacar:

- a) rapidez na manipulação e no acesso das informações;
- b) redução no esforço humano (desenvolvimento e utilização);
- c) controle integrado de informações distribuídas fisicamente;
- d) redução de redundância e de inconsistência de informações;
- e) compartilhamento de dados;
- f) aplicação automática de restrições de segurança;
- g) redução de problemas de integridade.

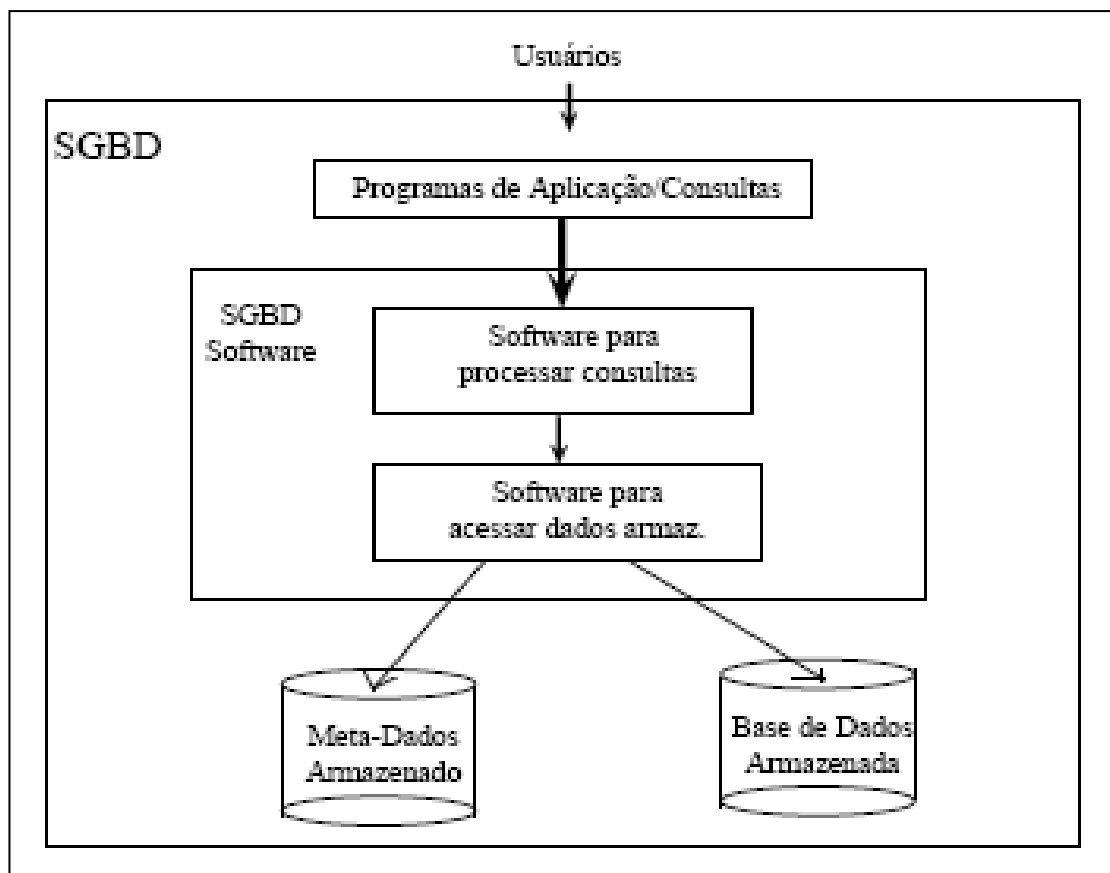


Figura 15 . Sistema Gerencial de Banco de Dados.
Fonte : MONTEIRO, E. (2004).

A abstração de dados realizada pelo sistema de banco de dados pode ser dividida em três níveis : nível de visão dos usuários, nível conceitual e nível lógico. O nível lógico é o mais baixo da abstração, descreve como realmente os dados estão armazenados englobando estruturas complexas de baixo nível. O nível conceitual mostra as informações armazenadas e seus respectivos relacionamentos, o banco de dados é descrito de estruturas praticamente simples que podem envolver estruturas complexas no nível físico. O nível de visões do usuário descreve partes dos bancos de dados, de acordo com as necessidades de cada usuário (PEREIRA, 2002).

Os modelos lógicos de dados são apresentados por um conjunto de ferramentas conceituais para descrição dos dados, dos relacionamentos entre os mesmos e das restrições de consistência e integridade. Estes modelos podem ser divididos em: baseados em objetos e baseados em registros. Modelos lógicos baseados em objetos descrevem os níveis conceituais e de visões do usuário, enquanto o modelo lógico baseado em registros descreve também os dados nos níveis conceituais e de visões de usuário (MONTEIRO, 2004).

O banco de dados é estruturado em registros de formatos fixos de diversos tipos, cada tipo de registro tem sua coleção de atributos e existe linguagem para expressar consultas e atualizações.

Segundo Pereira (2002), a manipulação das informações dentro de um banco de dados pode ser realizada de quatro maneiras: recuperação de informações armazenadas, inserção de novas informações, exclusão de informações e modificação de dados armazenados. A linguagem de manipulação de dados permite ao usuário acessar ou manipular dados, vendo-os de forma como são definidos no nível de abstração mais alto do modelo de dados utilizado.

O sistema de gerenciamento de banco de dados é o módulo que fornece a interface entre os dados de baixo nível armazenados no banco de dados e os programas aplicativos ou as solicitações submetidas ao sistema, este *software* manipula todos os acessos ao banco de dados pela interface do usuário.

6.2 LINGUAGEM PARA MANIPULAÇÃO DE DADOS

Na definição dos esquemas conceitual e interno utiliza-se uma linguagem chamada *Data Definition Language* – Linguagem de Definição de Dados (*DDL*). O sistema gerenciador de banco de dados possui um compilador *DDL* que permite executar declarações com o intuito de identificar as descrições dos esquemas armazenando no catálogo do SGBD. A *DDL* é utilizada em SGBDs onde a separação entre níveis internos e conceitual não é muito clara (VALDURIEZ, 2001).

No SGBD onde a separação entre os níveis conceitual e interno é bem clara, é utilizada uma outra linguagem, a *Storage Definition Language* – Linguagem de Definição de Armazenamento (*SDL*) para especificação do esquema interno. A especificação do esquema conceitual fica por conta da *DDL*. Quando o SGBD utiliza na arquitetura três esquemas, deve ser utilizada uma linguagem de definição de visões, a *Vision Definition Language* – Linguagem de Definição de Visões (*VDL*).

Segundo Monteiro (2004), quando o esquema estiver compilado e o banco de dados com informações, se usa uma linguagem para fazer a manipulação de dados, a *Data Manipulation Language* (*DML*).

6.3 CLASSIFICAÇÃO DO SISTEMA GERENCIAL DE BANCO DE DADOS

Segundo Valdúriez (2004), na classificação de um Sistema Gerencial de Banco de Dados (SGBD), se deve verificar o modelo de dados em que ele está baseado. Na grande maioria os SGBDs são baseados no modelo relacional, alguns utilizam modelo conceitual e alguns em modelos orientados a objetos. Outras classificações são:

- a) usuários : SGBD mono-usuário, normalmente utilizado em computadores pessoais ou multi-usuário utilizado em estações de trabalho, mini-computadores ou computadores de grande porte;
- b) localização : SGBD localizado ou distribuído, no localizado todos os dados estarão em uma máquina (ou em único disco) e distribuído os dados estarão distribuídos por diversas máquinas (ou diversos discos);
- c) ambiente : ambiente homogêneo é o ambiente composto por um único SGBD e um ambiente heterôgeneo é o ambiente composto por diversos SGBDs.

A utilização de voz no computador buscando valores do banco de dados gerou uma série de avanços tecnológicos, socializando pessoas com deficiência visual e oral, permitindo à utilização deste recurso em aplicações de diversas áreas, dando ênfase as aplicações de integração de telefone e computador que permitem ao computador controlar uma chamada telefônica com a possibilidade de interagir com o usuário pelo teclado do telefone.

7 INTEGRAÇÃO TELEFONE E COMPUTADOR

Segundo Barros (2002), pode-se considerar o telefone e o computador como as duas maiores invenções da humanidade, porém a cada dia, elas estão se aproximando tornando inevitável à necessidade de troca de informações. Esta tecnologia é conhecida como *Data Circuit-Terminating Equipment (DCE)*, que proporciona uma série de serviços, com destaque para o *call center*.

7.1 CALL CENTER (CENTRO DE CHAMADAS)

A utilização do *call center* vem crescendo, se tornando uma grande aliada estratégica das grandes companhias. Pode-se verificar na quantidade de atendimentos ao público realizado por companhias telefônicas, serviços prestados pelos bancos, Serviços de Atendimento ao Consumidor (SAC) utilizado principalmente por grandes indústrias ou ainda serviços de número 0800 e 0900 gratuitos oferecidos por instituições públicas e empresas privadas (GALLO, 2002).

Segundo Barros (2002), o *call center* funciona com um número de atendentes conforme a proporção de atendimento almejada pela organização. Estes atendentes ficam na frente de terminais interligados por uma rede, sendo controlado por um *software* desenvolvido para realizar o controle deste *call center*, integrado a uma rede telefônica. Este serviço possui um único número e quando o *pabx* recebe a ligação, o *software* específico do *call center*, dispara uma mensagem para o usuário, paralelamente localizando o primeiro atendente livre e re-encaminhando esta ligação.

O operador após atender a ligação solicita informações ao usuário como cpf, código do cliente ou conta bancária, estas informações são adicionadas ao *software* que

irá retornar ao atendente as informações necessárias do usuário para que ele possa continuar com o processo de atendimento. Estas informações podem ser também referentes a ligações atendidas anteriormente. O *software*, monitora toda a conversa gravando em fitas, sendo que este serve para suporte legal ou mesmo para auditoria do atendimento.

A integração computador – telefone pode ser verificada em vários pontos de um *call center*, muitas centrais telefônicas são programadas para economizar tempo do operador, fazendo uma espécie de re-direcionamento da ligação, por exemplo, destinar as ligações de uma organização para o setor desejado economizando tempo da corporação (BARROS, 2002).

O *call center* permite a recuperação de mensagens mais longas, gravadas no computador (textos de ajuda ou instrução), ou sintetizar mensagens compostas como (número de telefone ou saldo bancário), passando-as pelo telefone. O operador pode discar números pelo teclado do computador fazendo com que toda operação do *call center* seja otimizada por um *software* de controle, o que ajuda a diminuir o tempo parado dos operadores, redistribuindo as chamadas e controlando a produtividade de cada atendente.

Este serviço de realização de chamadas, do próprio *call center* pode ser utilizado por diversas aplicações como : levantamento de opiniões e o grau de satisfação dos usuários, *telemarketing* (venda de bens e serviços pelo telefone). O *software* pode montar, uma lista de número telefônicos a serem chamados, dividir as ligações proporcionalmente aos atendentes conforme o grau de disposição de cada um e realizar as ligações automaticamente redirecionando-a ao atendente correspondente (GALLO, 2002).

Muitas pessoas possuem esta tecnologia e não sabem, por exemplo, que os *modems* mais modernos, integram voz, fax e dados como o *RapidComm Voice* da *US Robotics* (maior fabricante de *modems* do mundo), além de permitirem a utilização de microfone e alto-falantes ao microcomputador para discar, falar ao telefone, enviar e receber *fax*, acessar um provedor *Internet*, estabelecer comunicação de dados ponto-a-ponto, entre outros.

Com as transformações das redes telefônicas de analógicas para digitais, como nos celulares *Code Division Multiple Access (CDMA)* e *Time Division Multiple Access (TDMA)*, proporcionam ainda mais a integração entre telefones e computadores. Pode-se prever que integrando os banco de dados das grandes organizações telefônicas com os telefones e computadores dos usuários com a *Internet*, pode significar um impulso ao comércio eletrônico entre outras coisas (BARROS, 2002).

7.2 FUNÇÕES DA INTEGRAÇÃO ENTRE TELEFONE E COMPUTADOR

Segundo Toledo (2000), a integração entre telefone e computador foi desenvolvida para realizar o controle de chamadas telefônicas pelos computadores. A integração entre telefone e computador, facilita a utilização da interface comparada com as interfaces naturais do telefone que oferecem funções de acesso, limitada pelo teclado telefônico. Esta oferece ainda plataformas e funções que permitem executar rapidamente serviços do telefone com maior flexibilidade. Podem-se classificar as funções do CTI em três categorias: controle de chamada, processamento de informações e direcionamento de dados do cliente. As funções do controle de chamada incluem:

- a) *setup* da chamada e serviços relacionados como os serviços de cronômetro e serviços de tela;

- b) serviços relacionados com rota de serviço de atendimento automático e serviço de rota alternativa;
- c) serviços de interface de rede como geração e detecção de tom, *setup* da chamada/detecção de release e detecção do sinal da ligação;

O processamento de informações inclui:

- a) processamento de voz e *fax* como gravação e aviso de voz, envio de voz e *fax*, *broadcast*, *storing* e *forwarding*;
- b) *Dual Tone Multi-Frequency (DTMF)* processamento de dígitos, sintetização de texto para palavra e reconhecimento de palavras (como um comando de voz de reconhecimento oral), verificação da fala, corte sobre voz e palavras marcadas;
- c) *login* de chamadas como monitoramento e gravação *on-line* e conta da chamada.

Na transferência de dados do cliente fornece informações pessoais sobre a transferência para todos os grupos da chamada. Esta transferência utiliza identificadores da convocação/chamada para buscar as informações da convocação/chamada do banco de dados e associar a informação com a chamada durante o processo, assim a chamada pode ser processada de maneira eficiente (TOLEDO, 2000).

A arquitetura da integração entre telefone e computador consiste em três componentes: a interface entre *switch* e *host*, interface de programação da aplicação e o recurso da arquitetura CTI. A interface entre o *switch* e o *host* fornece uma conexão entre o *switch* e o *host* (servidor CTI). A *API* permite ao desenvolvedor de *software* criar novos serviços e funções para o sistema de CTI. O recurso de arquitetura do CTI conduz o telefone à arquitetura do computador como o reconhecimento de palavra e aparelhos de *fax*.

7.3 INTERFACE ENTRE SWITCH E HOST

A interface entre *switch* e *host* fornece uma conexão entre *switch* e *host* do computador (um servidor CTI). Tradicionalmente *switches* são sistema fechados que não fornecem interface para controle externo. Nas aplicações de telefone-computador, é necessário a utilização de *switch* para ser controlado pelo servidor CTI rodando as aplicações. Este método de interface é chamado *CTI link*, que é diferente para uma interface de linha de telefone. Visando suportar controle externo, muitos fabricantes de *switch* oferecem interfaces como *Lucent's Definity G3* e *Nortel's Meridian Link*, estes são conectados no *switch* via *CTI link*, o servidor CTI fornece o meio telefônico para programadores com *API* padrões como a *TAPI* e *TSAPI* (NETO, 2001).

A interface do *CTI link*, possui duas tecnologias: *Computer Supported Telecommunication Applications (CSTA)* e *Switch to Computer Application Interface (SCAI)*. O desenvolvimento da *CSTA* foi iniciado pela *European Computer Manufacturates Association (ECMA)*. O *SCAI* é uma versão *ANSI* da interface *CSTA*, porém é muito mais complicado. As utilidades das aplicações *SCAI*, incluem transmissão simultânea de voz e dados, re-discagem de chamadas e suporte de transferência e cronômetro de chamadas pelo computador (SPENCER, 1999).

Atualmente, os fabricantes de *pabx* têm adaptados *CSTA* como padrão para indústria. O *CSTA* possui como base a definição da *TAPI*, mas os serviços *CSTA* também correspondem para função de chamada da *TSAPI*. O protocolo *CSTA* especifica o comando *sets* e estruturas de dados para comunicação entre o *switch* e o *host*. Por este protocolo, o *host* acessa os serviços de telefone do *CSTA* do *switch* ou fornece serviços computacionais do *CSTA* para o *switch*.

Na especificação dos serviços, o *CSTA* define um modelo de processamento do telefone para o *host* do computador e um modelo de processamento do computador para o *switch*. Cada um destes modelos consiste em setar estados para os objetos, estes estados podem existir para cada objeto, e a realização dos estados de transição de objetos do telefone incluindo dispositivo, conexão e chamada, esta relação é demonstrada na figura 16 (NETO, 2001).

O objeto dispositivo pode ser um dispositivo físico (um botão, uma linha, um tronco ou uma estação) ou um dispositivo lógico (um grupo de dispositivos ou um grupo de distribuição automática de chamadas *Automatic Call Distribution (ACD)*). O objeto dispositivo tem atributos que incluem o tipo, perfil, identificador e estado do dispositivo que podem ser monitorados e manipulados pela função computacional.

O objeto de chamada descreve uma sessão lógica entre as pessoas que estão realizando as chamadas e as que estão recebendo as chamadas. O comportamento da chamada, incluindo estabelecimento e desligamento da chamada, pode ser observado e manipulado pela função computacional. O objeto de chamada que representa uma sessão de chamadas tem atributos como identificador, e, estado tem operações como iniciar ou terminar. Um ou mais dispositivos podem estar envolvidos na chamada em estágios diferentes. O dispositivo na chamada pode ser substituído por outro quando a chamada é transferida e duas chamadas podem se fundir em uma única chamada com a operação de conferência.

O objeto conexão representa a relação entre uma chamada e um dispositivo, que tem atributos como identificador e estados e operações como espera ou termina. Muitos serviços *CSTA*, como chamada em espera e terminação de chamada, são feitos por meio de operações nas conexões abrangendo uma chamada.

Segundo Neto (2001), o serviço *SCAI* fornece uma boa administração, as mensagens específicas do evento da chamada do cliente que são associadas logicamente com a distribuição de chamada automática na central dos clientes (serviços de *ACD*)/*Centrex*. Estas mensagens específicas do evento originadas no *switch* da chamada são processadas pelos computadores dos clientes e pelo *software* da gerência da chamada para executar várias transmissões de dados, telecomunicações e/ou funções administrativas.

O serviço *SCAI* também suporta mensagens *request* que são enviadas do computador dos clientes ao *switch* servidor. Estes prestam serviços de manutenção às funções de *switching* iniciadas das mensagens do pedido dentro do *switch* digital que são associadas com o serviço dos clientes *ACD*/*Centrex*. Os serviços de mensagens *request* e chamadas eventuais ajustam, quando utilizadas com um cliente apropriado de gerenciamento de chamadas.

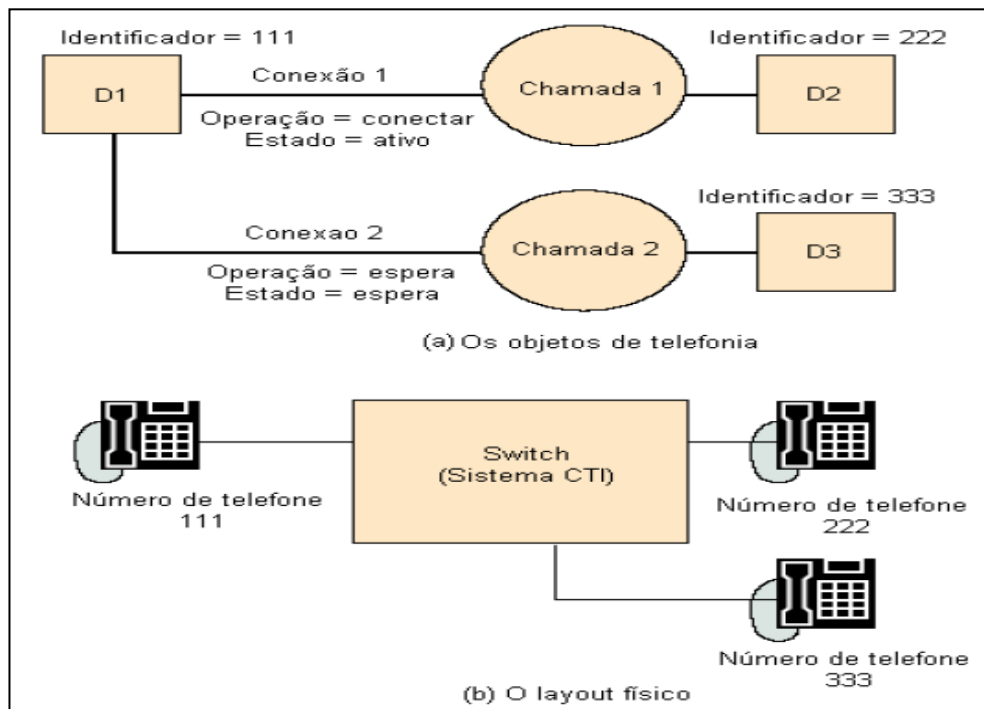


Figura 16 . Os objetos de telefonia em espera.
Fonte : NETO, S. (2001).

7.4 INTERFACE DE PROGRAMAÇÃO DA APLICAÇÃO

Segundo Dodd (2000), a interface de programação da aplicação habilita os desenvolvedores de *software* criar novas aplicações de telecomunicações. *TAPI* e *Microsoft's Telephony Applications Programming Interface (TSAPI)* desenvolvidas para computadores que utilizam a plataforma *Windows* permitem os desenvolvedores implementar aplicações para o telefone. Uma versão inicial da *TAPI* (*TAPI* 1.3) foi lançada em Novembro de 1993, utilizada para gerar uma nova versão (*TAPI* 1.4) lançada para o *Windows* 95. Estas versões não suportam o modelo de chamada com o *ITU-T Q.931*.

A *TAPI* permite a criação de aplicações de reconhecimento de dados para iniciar e terminar chamadas, progresso do monitor, detecção do *CLID* (identificação da linha de chamada), performance na identificação, ativar características como manter, conferência, parar e escolher. Pode-se redirecionar e iniciar chamadas, responder e rotear chamadas recebidas, gerar e detectar sinais *DTMF*. Esta *API* permite ainda múltiplas aplicações compartilhem uma simples linha telefônica (SPENCER, 1999).

Pode-se aceitar diferentes tipos de chamadas recebidas na mesma linha telefônica, habilitando estes dispositivos para compartilhar uma linha telefônica permitindo aos usuários realizar o uso mais eficiente destas linhas. A *TAPI* fornece acesso para vários serviços da rede telefônica como:

- a) serviços comuns do telefone, que suportam um tipo de informação (voz ou dados) por chamada e um canal por linha;
- b) *ISDN*, que suportam simultaneamente voz e dados por chamada e múltiplos canais por linha;
- c) Serviço de rede digital, que suportam comunicação de dados;

d) Outros serviços como *Centrex*, *PBX* e *Key Telephone System (KTS)*.

Posteriormente foram criadas as versões da *Microsoft TAPI (TAPI 2.0 e TAPI 2.1)* que moveu a configuração do controle de chamada do primeiro grupo para configuração do controle de chamada do terceiro grupo (DODD, 2000).

7.5 RECURSO DA ARQUITETURA DO CTI

Segundo Spencer (1999), para implementar e habilitar as funções de um CTI, é necessário recursos de *hardware*, estes recursos referem-se a placas de interface para união de dados e voz, placa de processamento de voz, placa de *fax*, placa de reconhecimento e sintetização de palavras. O recurso da arquitetura do CTI é tipicamente, um sistema aberto com suporte para utilização de fabricantes diferentes. Os padrões para recurso da arquitetura do CTI são *Multivendor Integration Protocol (MVIP)*, *Signaling Computing System Architecture (SCSA)* e *ECTF H.100/H.110*.

O *MVIP* é composto por vários padrões abertos, com suporte para integração de processamento de voz, protocolos de comunicação de *fax*/dados, e outras tecnologias de computador que necessitam conectar a rede telefônica. A *MVIP* fornece um padrão aproximado para integrar tecnologias de computador com *pabx* proprietários e outros sistemas telefônicos comutadores.

MVIP-90 é o padrão original *MVIP* que especifica a multiplexação por divisão de tempo, mas com capacidade de 512X64 *kb/s* e a capacidade de distribuir o circuito comutado dentro do computador com o *software* de controle. O *H-MVIP* é uma extensão do *MVIP-90* que permite ao sistema compartilhar 37072X64 *kb/s*, com maior capacidade de comutação, a especificação *MVIP* inclui uma interface de *driver* de

aparelho comum, sendo transparente aos programadores de aplicações e utilizando placas de recursos de *hardware* re-manufaturados (LIMA, 2001).

A arquitetura aberta para recurso de integração entre telefone e computador (SCSA), além de incluir o modelo de *hardware* que especifica o CTI, inclui o tempo real de comutação de dados e o modelo de *software* que identifica o serviço para processamento de informações e controle da chamada.

Segundo Spencer (1999), o modelo de software *SCSA* é conhecido como *Telephony Application Objects (TAO)*, que identifica o objeto com *set* da interface e serviços para manipulação de informações de uma chamada e acesso dos elementos do controle da chamada. Adicionando na interface e serviços, defini-se uma interface padrão de mensagens flexíveis, independentes de fabricantes e nível de *hardware*.

A *Enterprise Computer Telephony Fórum (ECTF)*, especifica uma alta capacidade chamada H.1000 CTI, fornecendo um modelo de compatibilidade com produtos existentes. O novo CTI *Bus* foi julgado como solução final para as indústrias, tendo como principais características:

- a) aumento da capacidade : 4096 *slots* rodando ao mesmo tempo a 8 *MHz*;
- b) redundância : dois *clocks* e linhas de sincronização, para que se perder um *clock*, todos aparelhos possam ser sincronizados com outro *clock*;
- c) operação : 16 das 32 linhas de dados podem ser selecionadas para funcionar a 2 *MHz* para *MVIP-90* ou 4 *MHz* para o modo *SCBus*. O H.100 também suporta o aparelho específico, com um canal opcional de mensagens.

Estes métodos e componentes participantes da integração entre o telefone e o computador são necessários em forma real, permitindo aos desenvolvedores abstrair estas funções do sistema telefônico e utilizarem remotamente pelo computador, com

este conceito um consórcio de fabricantes e empresas relacionadas à telefonia criaram a *JTAPI*.

8 JAVA TELEPHONY APPLICATION PROGRAMMING INTERFACE

A *Java Telephony Application Programming Interface (JTAPI)*, foi projetado por um consórcio de indústrias de computadores e da área de telefonia, visando a criação de uma *API* portátil, capaz de controlar uma chamada integrando telefone e computador. Os requisitos na modelagem da *API* eram bem simples, facilidade de execução em todos os sistemas operacionais e integração entre o primeiro e o terceiro grupo do controle de chamadas (HUGHES, 2003).

A integração entre telefone e computador vai permitir à pequenas empresas acessar características das centrais de chamada de alta tecnologia. A *JTAPI* fornece uma abstração de serviços que normalmente seriam prestados apenas por uma central telefônica (DEITEL, 2002).

8.1 LIGAÇÃO ENTRE TELEFONE E COMPUTADOR

A *JTAPI* é um *API* orientado a objeto, que permite a criação de aplicativos em java que utilizam recursos do telefone. Existem outras *APIs* similares específicas para outras plataformas como *Telephony Application Programming Interface (TAPI)* para *Microsoft Windows* e *TSAPI* para *Novell Netware* (HOOGE, 1996).

O objetivo de criação da *JTAPI* é realizar a ligação entre o aplicativo java e o sistema telefônico, o ponto de localização da *JTAPI* definirá o grau de controle que a ligação terá sobre a chamada.

No primeiro grupo do controle de chamada a interface estará localizada no terminal, onde a aplicação terá o mesmo grau de controle que o usuário normal do telefone. No terceiro grupo do controle de chamada a interface estará localizada dentro

do próprio sistema telefônico, conforme o acesso interno deste sistema, permitindo a aplicação aumentar a capacidade de controle em relação ao primeiro grupo do controle de chamada. Na Figura 17, o exemplo da *JTAPI* no primeiro e terceiro grupo do controle de chamada (LIMA, 2001).

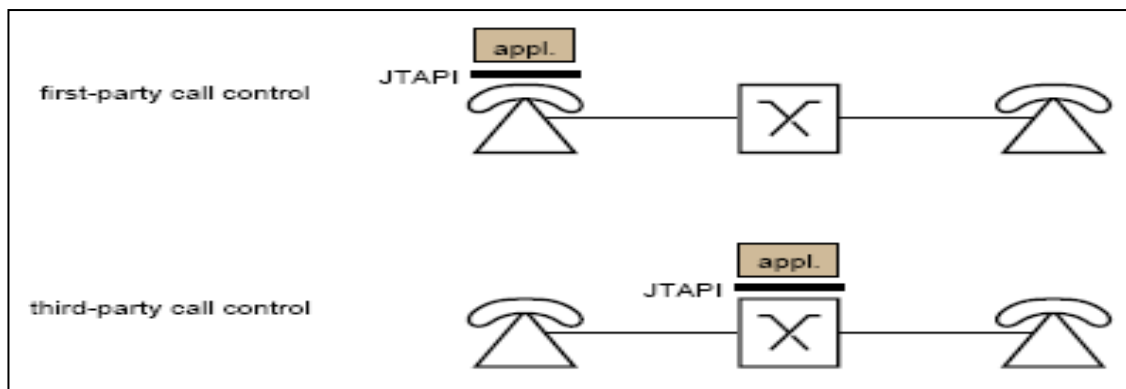


Figura 17 . Primeiro e terceiro grupo do controle de chamada da rede telefônica.
Fonte : HOOGE, A. (1996).

No processo desenvolvimento da *JTAPI*, um dos principais objetivos era suportar vários cenários que ocorressem durante o controle da chamada, que permitiu ao sistema telefônico utilizar o terceiro grupo do controle da chamada, da mesma maneira usada no primeiro grupo de controle da chamada. Na visão do terceiro grupo da chamada, o final local e o final remoto são simétricos não possibilitando distingui-los (TOLEDO, 2000).

Segundo Lima (2001), o projeto da *JTAPI* é um modelo de chamada com alto nível de abstração independente da tecnologia, descrevendo a chamada como um estado finito de máquina o *Finite State Machine (FSM)* que permite a transição entre os estados de chamada envolvidos. O modelo de chamada é genérico cobrindo vários cenários de chamada, sendo capaz de descrever, por exemplo:

- a) uma chamada entre dois grupos;
- b) múltiplas chamadas para o mesmo terminal;

- c) uma conferência entre múltiplos grupos;
- d) uma configuração de chamada que alerta múltiplos terminais.

O modelo descreve a chamada como um todo ou em grupos, definindo cinco classes base (no sentido de tipos de objeto), duas classes descrevem partes do controle, sendo que estes objetos são persistentes e independentes da chamada. O usuário é representado pelo objeto endereço, onde o atributo principal deste objeto é a identificação do usuário. O terminal telefônico é representado por um objeto terminal, e o seu atributo principal é o endereço do terminal (TOLEDO, 2000).

As outras três classes descrevem a chamada, estes objetos são instanciados e criados dinamicamente durante a chamada, portanto não são persistentes durante todo o controle. A cada inclusão destes objetos é gerado um estado finito de máquina, o objeto chamada é criado para cada chamada realizada. O objeto conexão é criado para cada usuário participante da chamada, sendo responsável por conectar o objeto do endereço do usuário com o objeto chamada. O objeto conexão do terminal é criado para cada objeto participante da chamada, ele será responsável por conectar o objeto terminal com o objeto conexão (HOOGE, 1996).

8.2 MODELOS DE CHAMADA

Segundo Spencer (1999), o modelo de chamada de dois grupos é composto pelo objeto conexão, anexado no objeto de chamada, um para cada participante. Este modelo é simetricamente competente (ele não pode distinguir entre a entidade local e a remota) porque ele fornece uma visão do terceiro grupo do controle da chamada. A Figura 18 é um exemplo deste modelo, esta representação será importante para a extensão no caso de uma conferência de chamadas com três ou mais participantes.

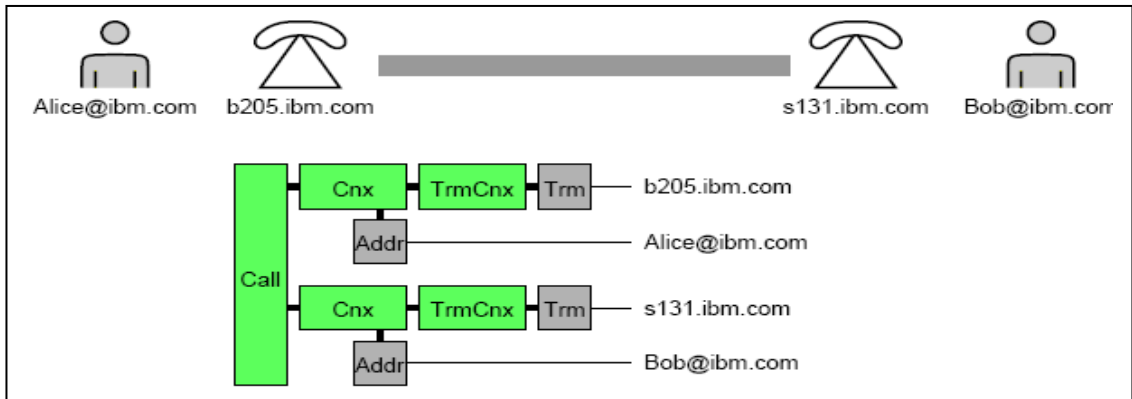


Figura 18 . Modelo de chamada para dois grupos.
 Fonte : SPENCER, R. (1999).

No modelo de duas chamadas simultâneas, todos os objetos da chamada relacionados tem seus números duplicados. O objeto endereço e o objeto do terminal do usuário que têm duas chamadas são anexados para duas conexões e dois objetos do terminal conexão. A Figura 19 mostra o exemplo, de um usuário que tem duas chamadas simultâneas para o mesmo terminal.

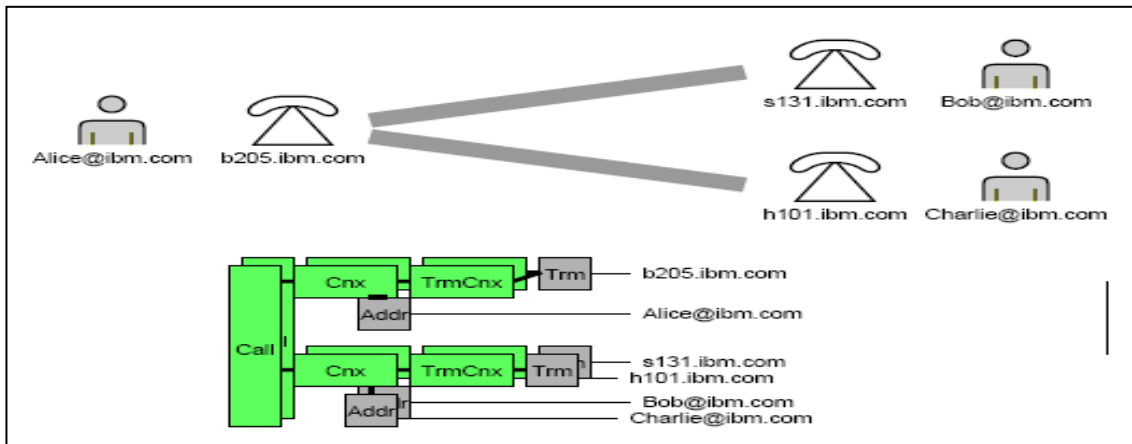


Figura 19 . Modelo de chamada para duas conexões simultâneas.
 Fonte : SPENCER, R. (1999).

Segundo Spencer (1999), o modelo de chamada com dois terminais ativos realiza a separação do objeto terminal conexão do objeto chamada. Na Figura 20 são representados como terminais de conexão: Bob, Alice e Charlie, sendo que Bob tem múltiplas aparências, que significam que quando Bob é chamado, vários terminais são

tocados. Estas aparências múltiplas são representadas pelos dois objetos terminal conexão, ligados ao objeto conexão de Bob, sendo um para cada terminal. Quando um dos terminais responde a chamada de outro terminal ele é desconectado (nos termos do modelo de chamada o objeto terminal conexão vai para o estado desabilitado).

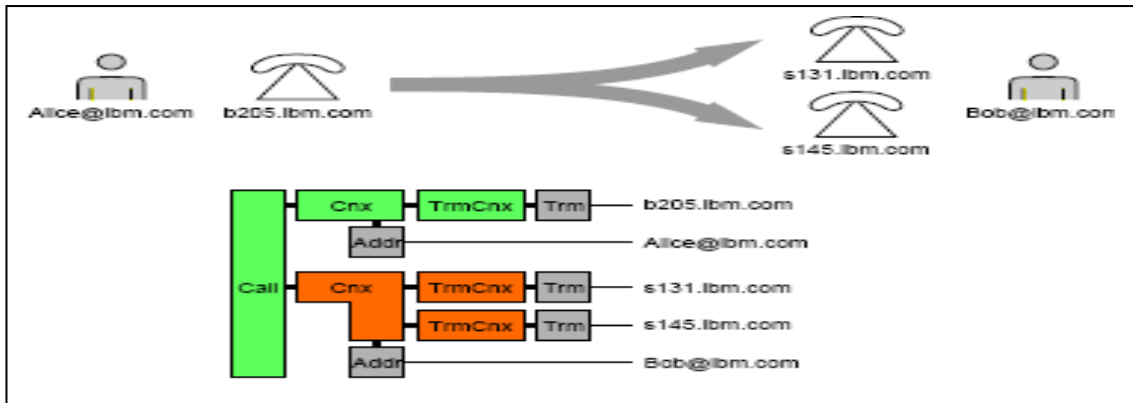


Figura 20 . Modelo de chamada para dois terminais ativos.
 Fonte : SPENCER, R. (1999).

No modelo de chamada para três grupos ocorre a separação dos objetos conexão dos objetos chamada, o modelo de chamada será a próxima extensão de uma chamada básica com dois participantes. O modelo simples adiciona a terceira perna com a conexão, endereço, terminal conexão e objeto terminal para o terceiro participante. A Figura 21, mostra o exemplo de uma conferência de chamadas com três participantes.

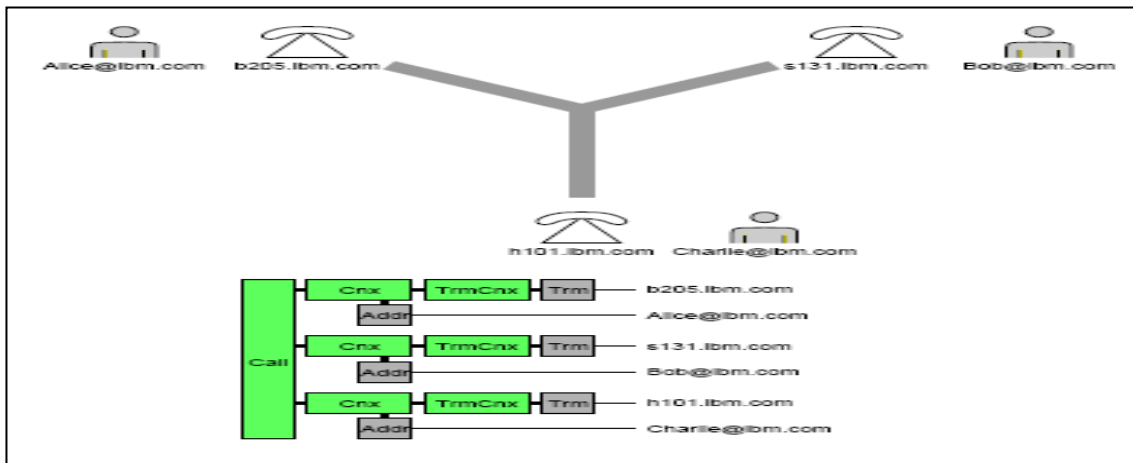


Figura 21 . Modelo de chamada de três grupos.
 Fonte : SPENCER, R. (1999).

8.3 ESTADO FINITO DE MÁQUINA

Neste descreve-se a relação dos objetos que compõem a chamada e seus estados finitos de máquina em mais detalhes. No caso serão mostrados os estados de transição de cada objeto que faz parte do controle da chamada.

Segundo Toledo (2000), o objeto chamada é criado para cada chamada, o estado deste depende do número de objetos conexão. A Figura 22 corresponde aos estados finitos de máquina deste objeto.

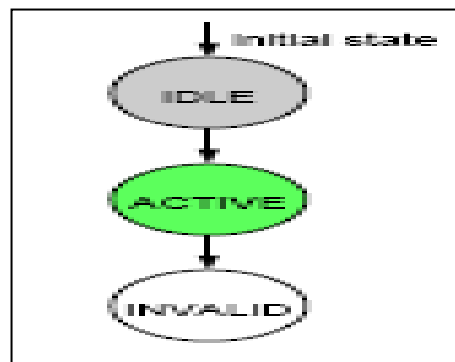


Figura 22 . Estado finito de máquina do objeto chamada.
Fonte : TOLEDO, A. (2000).

Estados finitos de máquina do objeto chamada:

- a) inativo : este é o estado inicial para todas as chamadas, neste estado a chamada tem zero conexões;
- b) ativo : uma chamada com alguma ligação corrente ativa e o no estado da chamada com uma ou mais conexões associadas devem estar neste estado;
- c) inválido : este é o estado inicial para todas as chamadas, estes objetos perdem todos os seus objetos conexão, via uma transação do objeto conexão no estado conexão desconectada (*Connectiom Disconnected*).

Este estado de chamada possui zero conexões e estes objetos de chamadas não podem ser usadas para uma ação futura.

O objeto conexão é criado para cada usuário participante da chamada, estas conexões são do objeto endereço do usuário com o objeto chamada. A Figura 23 corresponde aos estados finitos de máquina deste objeto (TOLEDO, 2000).

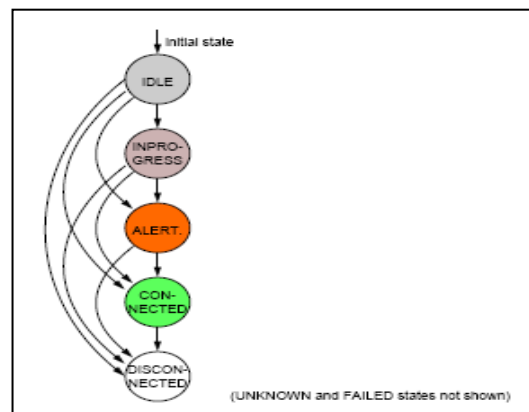


Figura 23 . Estado finito de máquina do objeto conexão.
Fonte : TOLEDO, A. (2000).

Estados finitos de máquina do objeto conexão:

- inativo : este é o estado inicial para todas nova conexões. Conexões que estão no estado inativo não são parte ativa da chamada telefônica, porém suas referências para os objetos chamada e endereço são válidas;
- sem progresso : este estado implica na conexão, que representa o destino final da chamada telefônica, e no processo de conexão do destino. Sob certas circunstâncias a conexão não pode iniciar progredindo este estado, pacotes de extensão forma elaborados neste caso e em várias outras situações;
- alerta : este estado implica que o endereço será notificado de uma chamada na outra chegada;

- d) conectado : este estado implica que uma conexão e o endereço são partes ativas de uma chamada telefônica. Em termos comuns, duas pessoas conversando com uma outra são representadas por conexões no estado conectado (*Connection Connected*);
- e) desconectado : este estado implica uma conexão a não ser muito tempo parte de uma chamada telefônica, contudo estas referências para a chamada e endereço ainda permanecem válidas. Uma conexão neste estado é interpretado enquanto anteceder o início da chamada telefônica;
- f) falhado : este estado implica que uma conexão para aquele final de chamada teve falha por algum motivo. Um motivo porque uma conexão está no estado falhado, é porque a parte estava ocupada.

Segundo Toledo (2000), o objeto terminal conexão é criado para cada terminal telefônico participando da conexão. Eles conectam o objeto terminal com o objeto conexão. A Figura 24 corresponde aos estados finitos de máquina deste objeto.

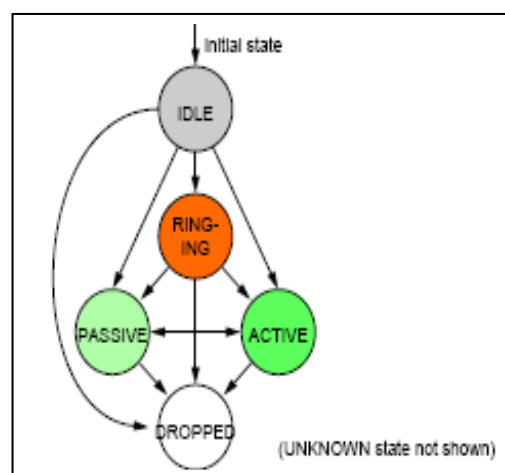


Figura 24 . Estado finito de máquina do objeto terminal conexão.
Fonte : TOLEDO, A. (2000).

Estados finitos de máquina do objeto terminal conexão:

- a) inativo : este é o estado inicial para todos terminal conexão. O objeto terminal conexão, não fica neste estado por muito tempo, normalmente transitam para outro estado rapidamente;
- b) tocando : este estado indica que o terminal esta tocando, indicando que o terminal tem uma chamada em comum;
- c) ativo : este estado indica que o terminal é a parte ativa de uma chamada telefônica. Este habitualmente implica no grupo da conversa que o terminal é parte de uma chamada telefônica;
- d) passivo : este é o estado que o terminal é parte de uma chamada telefônica, mas não esta ativo. Este pode implicar que um recurso do terminal será usado e pode limitar ações do terminal;
- e) derrubado : este estado indica que um telefone particular tem permanecido fora da chamada telefônica.

Tendo a definição da *JTAPI*, os possíveis modelos de chamadas e os estados de transição de cada objeto que vai compor o controle de chamada, é necessário definir minuciosamente como a *JTAPI* vai controlar esta chamada.

9 TRANSIÇÕES DO MODELO DE CONFIGURAÇÃO

Esta seção mostra o estado de transição do modelo de chamada para um modelo de configuração, o cenário é uma chamada de dois grupos onde o usuário A chama o usuário B. Note que são modelos de chamada de duas interfaces *JTAPI*, o terminal do usuário A e o terminal do usuário B.

9.1 FUNCIONAMENTO DO MODELO DE CHAMADA

Depois de inicializado o provedor de serviços relata dois objetos no espaço do endereço, o terminal representando o terminal local e o endereço representando o usuário local. A aplicação adiciona um observador para o terminal local, Figura 25.

A aplicação cria um objeto de chamada para estabelecer uma chamada, como se pode ver na Figura 26, depois a aplicação estabelece a chamada, Figura 27.

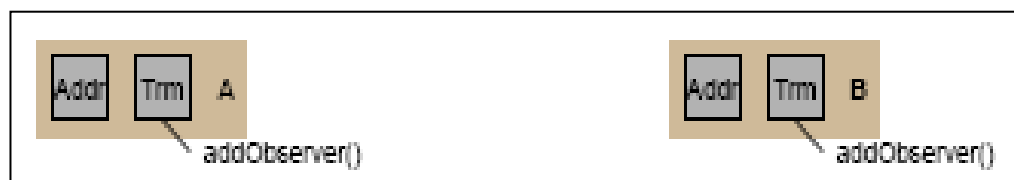


Figura 25 . A aplicação adiciona um observador para o terminal.
Fonte : SPENCER, R. (1999).



Figura 26 . A aplicação cria o objeto chamada.
Fonte : SPENCER, R. (1999).



Figura 27 . A aplicação estabelece a chamada.
 Fonte : SPENCER, R. (1999).

O provedor de serviços cria uma chamada com duas situações, cada uma possui uma conexão e um terminal conexão, depois de criado o estado finito de máquina é inativo. A chamada local transita conectado/ativo. O protocolo *stack* envia uma mensagem de *setup*, Figura 28.

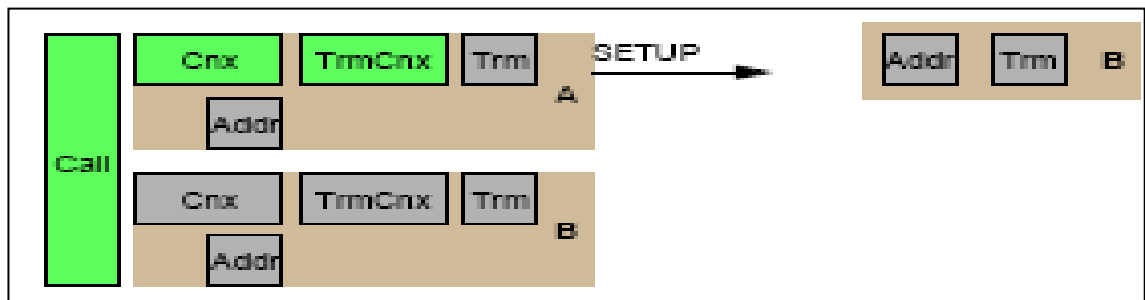


Figura 28 . O protocolo *stack* envia uma mensagem de *setup*.
 Fonte : SPENCER, R. (1999).

A situação da chamada remota transita em conectado/ativo e a situação da chamada local transita em alertando/tocando. Na Figura 29 o protocolo *stack* envia de volta uma mensagem alertando, a situação da chamada remota transita em alertando/tocando e a aplicação notifica que a chamada esta vindo de A. Na figura 30 o usuário B está alertado e o sistema espera enquanto ele responde a chamada. O usuário B chama a aplicação para responder e a situação da chamada local transitará em conectado/ativo. O protocolo *stack* envia a mensagem conexão na Figura 31 e a situação da chamada remota transitará em conectado/ativo, então a chamada está estabilizada.

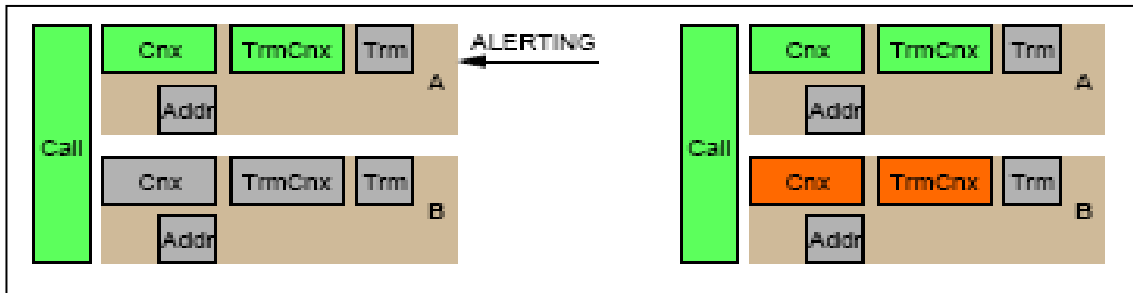


Figura 29 . O protocolo *stack* envia uma mensagem alertando.
 Fonte : SPENCER, R. (1999).

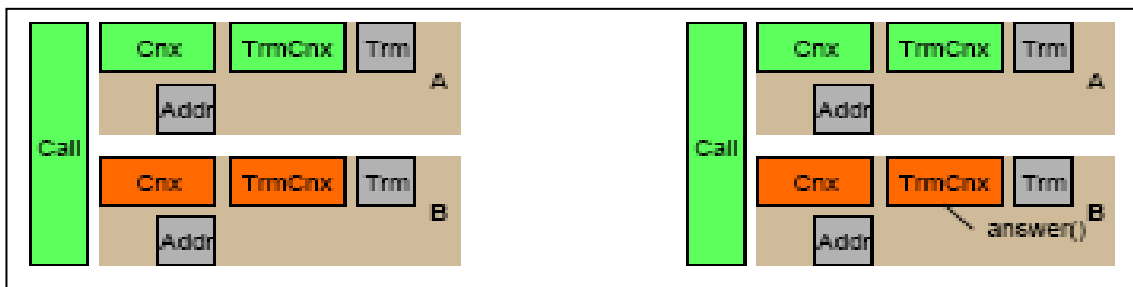


Figura 30 . O usuário B responde a chamada.
 Fonte : SPENCER, R. (1999).

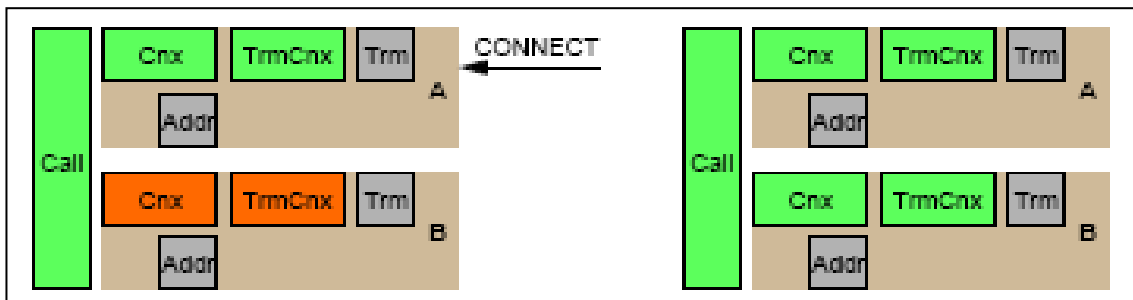


Figura 31 . O protocolo *stack* envia uma mensagem de conexão.
 Fonte : SPENCER, R. (1999).

Segundo Hooge (1996), um dos princípios da *JTAPI* é que o modelo de chamada do provedor de serviço para a aplicação é independente da localização da interface. A *JTAPI* pode estar localizada dentro do sistema telefônico ou no terminal participante da chamada. O exemplo demonstrado neste capítulo está desprezando a propagação da sinalização de mensagens, sendo que o modelo de chamada nos terminais A e B são idênticos.

A representação de uniformidade não indica o controle direto desta, o fato que uma aplicação possui o objeto uniforme não significa que ele terá o mesmo controle

sobre eles. Por exemplo, uma aplicação no terminal A é apenas aberta para manipular os objetos conexão e terminal conexão, quando anexadas ao terminal A, no objeto terminal. Quando uma aplicação tenta invocar um método de um objeto, ele não terá o controle sobre o provedor de serviços que dispara uma exceção.

9.2 VÁRIOS NÍVEIS DE DETALHE

A *JTAPI* utiliza a herança para obter um *designer* final, onde objetos são simples. A aplicação pode escolher entre utilizar uma visão básica de entidades e chamadas ou uma visão mais detalhada revelando mais informações sobre diferentes aspectos. Os objetos fornecem uma visão melhor detalhada, utilizando herança no objeto de visão básica adicionando estes estados e métodos. A visão básica é fornecida pelo pacote *core*, enquanto a visão detalhada é fornecida pelos pacotes *callcontrol* e *callcenter*.

Muitas interfaces de programação são construídas utilizando a relação mestre/escravo entre a aplicação e o provedor de serviços. A aplicação possui o controle, iniciando a ação e o provedor de serviços responde a estas ações. No caso da *API* do telefone é diferente, sendo esta natural de um sistema telefônico onde a chamada é manipulada por vários componentes (NETO, 2000).

Primeiro é realizado o controle de usuários da chamada, via seus terminais. Deste modo uma aplicação pode atuar no primeiro grupo ou terceiro grupo do controle da chamada, elas não são excluídas para que várias chamadas não relacionadas manipulem a mesma chamada. O provedor de serviços relata as ações externas (e da continuação também às ações da aplicação local) onde está escolhido o modelo de chamada. Observando o controle de uma chamada pela *JTAPI*, podemos verificar que

ela é uma *API* simétrica, onde o controle pode ser exercido tanto acima quanto abaixo da *API*.

Segundo Spencer (1999), a *JTAPI* estabelece o uso do evento *observer()*, em objetos padrões para informar a aplicação sobre a mudança de estado do modelo de chamada. A aplicação registra o objeto utilizando o evento *observer()* com o provedor de serviços, com o propósito de observar o modelo da chamada, o qual é o sujeito. Embora o estado finito de máquina no modelo de chamada altere o provedor de serviços gerando um evento do objeto que descreve a alteração do estado. O provedor de serviços então chama a aplicação do objeto que utiliza o *observer()* e passa esta referência para o evento do objeto.

A aplicação não pode agir apenas como componente na chamada, pois uma aplicação bem escrita deve estar preparada para não assumir os caminhos que deverão seguir os estados de transação. Esta aplicação deve sempre estar bem preparada para receber alterações inesperadas de estados resultantes de ações externas da chamada. Quando uma aplicação é designada para uma interface de usuário o modelo, controle e visão padrão podem ser aplicados com sucesso.

O modelo de chamada corresponde ao modelo padrão e a aplicação tem fornecido a visão e o controle. A visão mostra o estado da chamada em uma representação gráfica. O controle aceita a inserção do usuário e inicia as ações da chamada, porque o modelo de chamada relata todas as ações, que serão disparadas externamente ou pela aplicação, a visão e o controle serão guardados separados na aplicação se comunicando apenas por meio do modelo de chamada (LIMA, 2001).

9.3 ENDEREÇAMENTO

Ao se utilizar o sistema telefônico tradicional, o controle de chamadas pode ser executado no primeiro grupo via uma interface no terminal ou no terceiro grupo via uma interface dentro do sistema telefônico. A utilização do controle de chamadas pode ser verificada em ambos os grupos na Figura 32. Na arquitetura ponto-a-ponto as funções e controle de chamada são totalmente implementadas pelos terminais, tanto que estes estão próximos uns dos outros. No caso do servidor *proxy* deve ser introduzido para intervir no protocolo do controle de chamada (SPENCER, 1999).

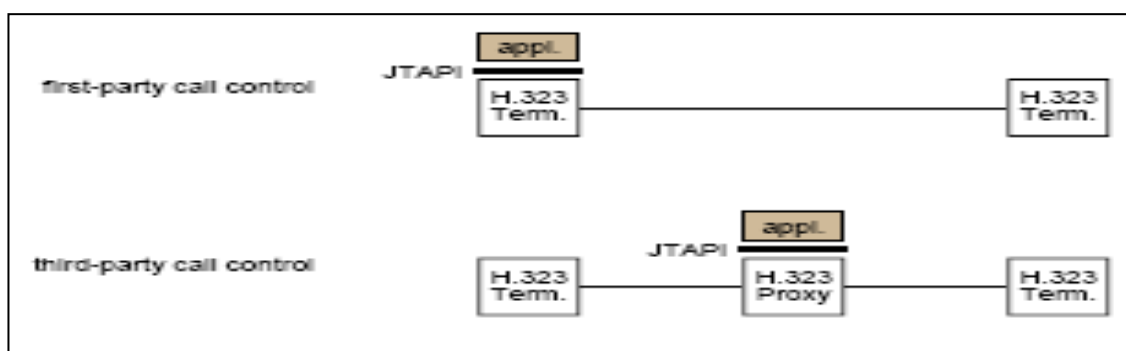


Figura 32 . Primeiro e terceiro grupo do controle de chamada no *Ip* do telefone.
Fonte : SPENCER, R. (1999).

O modelo de chamada da *JTAPI* distingue entre o identificador do usuário e o terminal separando em objetos endereço e objeto terminal. Portanto ele está diretamente para o mapa de identificadores de usuários de um *IP* do sistema telefônico, que poderá ser encontrado na forma de endereço de *e-mail* referenciando para o objeto endereço (TOLEDO, 2000).

Segundo Spencer (1999), os terminais telefônicos são *internets hosts* cada um com seu endereço *IP* (também pode ser representado por um nome de domínio) este é o mapa de endereços para o objeto terminal. No modelo de chamada *JTAPI* os objetos

endereços e objetos terminais são estáticos, mais precisamente eles são duas espécies de objetos endereço e terminal.

Os objetos terminais e objetos endereços sendo estáticos fazendo que o provedor de serviços, tenha conhecimento antecipado sobre qualquer chamada existente. Na primeira parte da interface do controle de chamada estes são objetos para o terminal local e o usuário deste terminal. Estes objetos são informados como sendo espaço do endereço do provedor.

Estes objetos não são necessariamente no espaço de endereço do provedor, mas aparecem porque eles participam da chamada. A aplicação não pode acessá-los, exceto utilizando os objetos de chamada, portanto o provedor de serviços é livre para criá-los dinamicamente quando eles forem necessários. A *JTAPI* permite que o relacionamento entre a identificação do usuário e endereço do terminal é mais dinâmico do que este (LIMA, 2001).

Tendo conhecimento do funcionamento do modelo de chamada, da *API* que permite a manipulação desta chamada e outros objetos que compõem a ligação com o sistema telefônico, vamos descrever o que seria um sistema de unidade de resposta audível.

10 UNIDADE DE RESPOSTA AUDÍVEL

Segundo Lima (2003), esta tecnologia é também conhecida como *Interactive Voice Response (IVR)*, funciona como interface telefônica para um sistema computadorizado, ou seja, um sistema localizado no *front-end* do computador que permite a entrada de dados via teclado telefônico e pela voz humana. O usuário pode receber mensagens de voz gravadas (digitalizadas ou sintetizadas).

A utilização desta tecnologia para fornecer dados de entrada é um processo simples devido à facilidade de fornecer estas informações via teclado telefônico. Porém torna-se um pouco mais complexo o fornecimento de palavras pela complexidade de informá-las via teclado telefônico.

10.1 CONCEITO DE URA

AS tecnologias de reconhecimento de voz que permitem aos usuários fornecer os dados de entrada apenas conversando ao telefone, possuem grandes avanços, porém ainda restritos em seus vocabulários. Na saída de informações o processo é mais simples, pois basta converter o texto em voz para poder transmitir. A aplicação mais utilizada de URA é a disponibilização por telefone, *fax* ou *e-mail* de informações disponíveis em um banco de dados (SAMPAIO, 2003).

Os Sistemas de Unidade de Resposta Audível (URA) possibilitam acesso, armazenam registros e efetuam vendas 24 horas, o que atrai a atenção de muitos usuários, pois podem utilizar este serviço, eliminando de certa forma custos com atendentes. A URA funciona principalmente com uma série de perguntas acopladas, com o intuito de rotear a chamada e realizar o seu processamento a fim de economizar

tempo, encaminhando a ligação para o setor desejado sem um segundo intermediário. A tecnologia de URA independentemente de estar a cargo de agentes ou recursos de atendimento automático, supera as demais tecnologias (identificador de chamadas) neste tipo de aplicação.

Este sistema é utilizado para substituir o modo de espera, podendo agregar valor ao tempo, que na maneira antiga era perdido na fila de espera. As aplicações mais atuais enquanto aguardam o atendimento de um agente utilizam chamadores com URA, que mantêm suas posições na fila de espera possibilitando a opção de atendimento automático, utilizando atendimento humano apenas em último caso (LIMA, 2003).

10.2 TIPOS DE APLICAÇÕES

Segundo Sampaio (2003), a URA possui várias aplicações entre elas: Bancos (Tele-Saldos, extratos, financiamento), Concessionárias de Telefonia (disque horóscopo, disque piada, serviço de despertador, hora certa), outras possibilidades (agendamento, cobrança telefônica, consulta lista de preços, estoques, taxas, programações).

Hoje existem diversos ramos de nossa economia, utilizando este tipo de tecnologia, o que demonstra como ela proporciona resultados aos seus investidores. Dentre os usuários desta tecnologia podemos citar empresas de diversos ramos como: indústria, comércio, condomínios, hospitais, *shopping centers*, agências bancárias, *call centers*, escolas, universidades, clubes, prestadores de serviços, hotéis, órgãos e empresas governamentais.

Segundo Lima (2002), podemos citar como características de uma aplicação URA:

- a) captura de dados;
- b) disponibilização de informações via mensagens de voz pré-gravadas, *fax* ou visuais;
- c) comunicação com aplicações externas;
- d) possibilitar a gravação de mensagens em estúdio profissional;
- e) possibilitar a interrupção do menu de opções, caso o usuário conheça o passo seguinte;
- f) habilitar ECO em qualquer ponto da aplicação;
- g) caixas postais, em quantidade ilimitada para armazenamento de mensagens de voz ou *fax*;
- h) modularidade que permita o crescimento do sistema de acordo com a demanda;
- i) flexibilidade na implementação de novos serviços;
- j) monitoração das ligações *On-line*;
- k) transferência sincronizada de voz e dados;
- l) envio de *fax* ao cliente na mesma ligação;
- m) permitir a criação de inúmeros ramais virtuais com caixas postais;
- n) gerenciamento de relatórios com recursos de construção de acordo com as necessidade específicas;
- o) possibilita transformar cada canal de atendimento em multi-aplicações, tratando-os na estratégia determinada.

11 TRABALHOS CORRELATOS

Neste capítulo serão abordados, alguns trabalhos da mesma linha deste, ou seja, integração entre telefone e computador utilizando Unidade de Resposta Audível.

11.1 SISTEMA INTELICON

Este sistema é composto de uma unidade de resposta audível, que transforma as informações retirados de banco de dados em voz digitalizada. Integrado com o banco de dados, a URA Intelicon possibilita criar sistema de auto-atendimento com racionalização de linhas telefônicas e diminuição de custos com atendentes, pois funciona 24 horas por dia. A URA Intelicon pode ser utilizada de três maneiras:

- integrada a um *pabx* (Figura 33);
- integra a um *pabx* virtual (disponibilizada pela companhia telefônica);
- sem *pabx* (atendimento e transferência para ramais pelo próprio Intelicon).

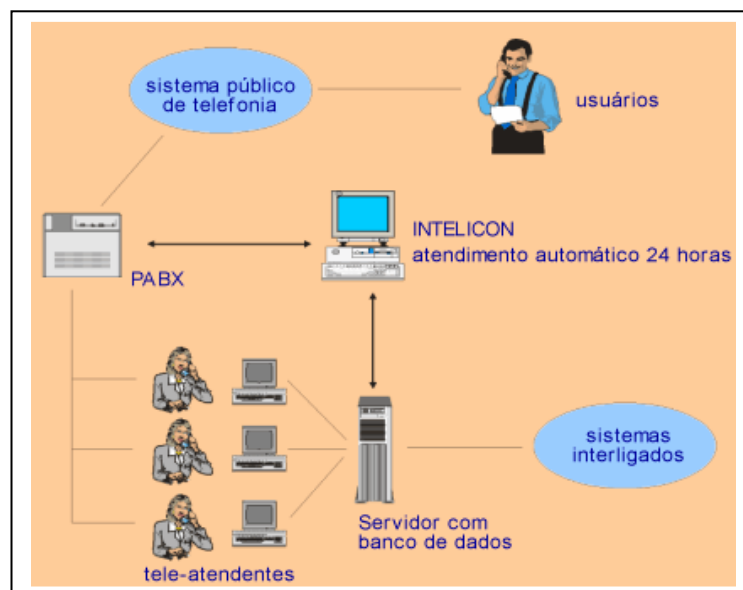


Figura 33 . Sistema Intelicon com *pabx*.
Fonte : GRANDE, D. (2005).

11.2 URA CTI *SERVER*

O URA CTI *Server* é um equipamento que pode ser conectado em troncos ou canais analógicos, permitindo o atendimento de chamadas telefônicas, respostas de voz e interação com o usuário e banco de dados. Este equipamento é conhecido no mercado como unidade de resposta audível.

O *software* registra cada chamada no banco de dados permitindo emitir relatórios sobre o perfil de utilização dos usuários e bilhetagem das chamadas. Com a URA também é possível à gravação de recados em *Hard Disks*, para posterior recuperação das chamadas. Este *software* possui como principais características:

- a) capacidade de 4 à 24 canais por módulos de URA;
- b) possibilidade de interconexão de módulos em rede, deixando quase sem limite a quantidade de portas de atendimento;
- c) base de dados centralizada armazenando dados de cada ligação;
- d) alarmes para espaços em disco, banco de dados, entre outros;
- e) relatório estatístico por período mostrando quantidade de ligações, tempo total e tempo médio de ligações;
- f) acesso ao gerenciador restrito por senha.

11.3 UNIDADE DE RESPOSTA INTERATIVA INFORMATEC

A Unidade de Resposta Interativa Informattec atende, responde e encaminha chamadas telefônicas automaticamente, a partir de quatro troncos simultaneamente.

Este *software* roda em *Windows NT*, interagindo com o usuário via teclado telefônico e acessa banco de dados (remoto ou local) em diversas plataformas, para fornecer informações solicitadas via voz ou fax.

Como banco de dados podem ser utilizados: *Oracle, SQL, Informix, Sysbase, Dataflex, Acess* entre outros. A unidade de resposta audível já foi implementada em várias áreas com sucesso como:

- a) empresas : fornecimento de cotações e preços, informações sobre a situação de ordens de serviços e acompanhar a situação de ordens de pedido;
- b) prefeituras : fornece via voz ou *fax*, todos os valores referentes a impostos, informa sobre eventos, licitações de obras e matrículas escolares;
- c) água, esgoto e gás : informa via voz ou *fax*, valores de contas vencidas e a vencer, locais e datas para pagamentos e datas de corte.

11.4 PLATAFORMA MULTISERVIÇOS URA

A Plataforma Multiserviços URA é resultado de intensa pesquisa e árduo trabalho de engenharia de desenvolvimento, se constituindo em uma solução para diversas necessidades do mercado. Apresentando uma grande variedade de facilidades, produtos e serviços integrados.

A utilização da tecnologia *Windows NT*, aliada a integração de diversas soluções torna a plataforma multiserviços URA parte significativa da *Computer Telephony Integration*. Este *software* possui como principais características:

- a) número ilimitado de portas;

- b) plataforma *Windows NT*;
- c) manutenção remota;
- d) inclui sistema de *hardware* e *software* exclusivo do sistema;
- e) relatório de gerenciamento;
- f) banco de dados residente.

12 PROTÓTIPO DE INTEGRAÇÃO TELEFONE E COMPUTADOR

O trabalho foi implementado na área de integração entre telefone e computador, utilizando a tecnologia de Unidade de Resposta Audível.

Esta tecnologia é baseada na inserção de uma interface de unidade de resposta audível no computador que possui acesso ao sistema telefônico. O computador terá propriedades de captura desta ligação do meio telefônico.

A interface possui funções para capturar os *DTMFs* (digito do teclado telefônico) e processar as informações capturadas de acordo com a funcionalidade definida para o aplicativo e retornar o processamento de informações para o meio telefônico com o auxílio da sintetização de voz.

Na definição da aplicação desenvolvida, optou-se pela área pública, para contribuir à sociedade proporcionando à pessoas comuns acesso a esta tecnologia. Outra razão prende-se ao fato de que em razão do custo baixo maior amplitude de uso terá o *software*.

Para prestação de serviços de tele-atendimento, conforme o porte e área de abrangência da organização, ela necessita de um grande número de funcionários, ainda mais se este atendimento for disponível 24 horas por dia. Com a inserção do Protótipo de Unidade de Resposta Audível a figura do atendente será eliminada ou pelo menos reduzida. O fato do serviço estar no ar 24 horas por dia, não agrega custos, pois as máquinas não receberão horas extras, trazendo assim uma enorme redução de custos a organização.

Atualmente quando se fala em automação, ou seja, qualquer tipo de serviço que substitua uma pessoa, logo é criado uma barreira, pois todos têm medo, de que as máquinas façam o serviço e as pessoas percam seus empregos. As pessoas devem

procurar sempre se aperfeiçoar conhecendo novas tecnologias, tendo em vista que o mercado do trabalho está em constante mudança, devido ao avanço tecnológico.

O Protótipo de Integração entre Telefone e Computador foi desenvolvido para fornecer um serviço de informações aos contribuintes da prefeitura quanto ao Imposto Predial e Territorial Urbano. Será fornecido o valor devido e qual o próximo vencimento das parcelas do contribuinte.

O funcionamento do sistema consiste em: o contribuinte liga para prefeitura, sua ligação será direcionada a um computador que estará conectado ao meio telefônico pelo recurso da *JTAPI* em modo de espera. A ligação do cliente será capturada pelo *modem* do computador. Este aplicativo vai disparar uma mensagem solicitando que o contribuinte informe seu código de cliente pelo teclado telefônico e confirmado com o dígito #. O aplicativo receberá informações do teclado do telefone do cliente, que captura os *DTMFs*, informados por este usuário e após confirmar com a tecla #. O aplicativo montará um comando *Structured Query Language* buscando informações do cliente contidas no banco de dados do sistema corporativo da empresa. Estes valores serão convertidos para voz utilizando a tecnologia de sintetização de voz e depois retornará estes valores convertidos para o telefone do usuário.

Inicialmente com a utilização da *JTAPI*, foi definida a implementação a ser utilizada de integração entre telefone e computador, buscando do computador o terminal de conexão utilizado para capturar a ligação, este terminal conexão é o *modem* ligado ao sistema telefônico que adiciona este terminal em modo de observação, ou seja, aguardando o recebimento de uma chamada telefônica, este método é especificado pela seguinte implementação:

```
Provider myprovider = null;
try
{
    JtapiPeer peer = JtapiPeerFactory.getJtapiPeer("net.xtapi.XJtapiPeer");
    myprovider = peer.getProvider(null);
}
catch (Exception e)
{
    e.printStackTrace();
}
try
{
    /// No Windows, por padrão os terminais são numerados
    Terminal terminal = null;
    terminal = myprovider.getTerminal("0");
    System.out.println("Buscou terminal: " + terminal);
    terminal.addCallObserver(new Secretaria() );
}
catch (Exception e)
{
    e.printStackTrace();
    System.exit(0);
}
```

No caso acima deve-se utilizar uma variável de objeto terminal, buscando o terminal que é a linha telefônica ligada ao *modem* do computador, para utilização desta implementação o programador deve possuir o código referente ao terminal desejado, utilizando a função *getTerminal()* passando como parâmetro o número do terminal de conexão.

Deste modo, a aplicação esta em modo de observação, ou seja, aguardando uma chamada. A *JTAPI* fornece o controle da chamada telefônico, dividindo os eventos e atributos que a compõem em três objetos que são: conexão, chamada e terminal conexão. O programador deve utilizar os eventos e atributos de cada objeto para realizar o controle da chamada telefônica.

Quando o usuário disca para o número do telefone, onde o computador esta em modo de observação, é disparado o evento *TermConnRinginEv*, por este evento o usuário sabe que tem uma ligação para ser capturada, então o programador utiliza a função *getTerminalConnection()*, que vai capturar a ligação telefônica pelo terminal definido no script anterior.

Com a chamada capturada o programador precisa capturar os *DTMFs* (digito do teclado telefônico), para fazer a interação com o cliente. Quando o usuário pressiona o teclado telefônico a *JTAPI* dispara o evento *MediaTermConnDtmfEv*, então o programador utiliza a função *getDtmfDigit()* para pegar o valor digitado pelo teclado telefônico.

Neste protótipo foi definido que após o usuário apertar a tecla #, indicará que o usuário terminou de informar o código do cliente, então o valor anterior a # será armazenado em uma variável e será montado um comando *Structured Query Language*, para buscar as informações do cliente no banco de dados da prefeitura, o processo foi definido pela seguinte implementação:

```
char digit = ((MediaTermConnDtmfEv)ev).getDtmfDigit();
System.out.println("Usuário discou: " + digit);
if (digit == '#')
{
    if (codigo.length() == 0)
    {
        return;
    }
    Integer valor = new Integer(codigo.toString());
    System.out.println("Processando: " + valor);
    new Processador(valor.intValue()).processa();
    codigo = new StringBuffer();
}
```

Quando na implementação é criada uma instância do objeto processador, o aplicativo está passando o código digitado pelo cliente no teclado telefônico, para ser montar os comandos de banco de dados. Com as informações recuperados do banco de dados, que são valor devido pelo contribuinte e próximo vencimento, necessita-se de uma maneira de enviar estas informações ao contribuinte.

Nesta implementação foi definida a utilização da sintetização de voz, no caso o programador deve passar as variáveis para uma classe responsável pela sintetização e retorna as informações ao telefone do contribuinte.

A sintetização de voz é implementada como um simples compilador que reconhece as palavras e dispara um som pré-definido para cada palavra reconhecida. Porém neste caso o programador deve passar sempre uma frase por extenso, pois na grande maioria destas APIs de sintetização de voz, não reconhecem valores que não são do tipo texto.

Para poder transformar as variáveis com valor e data em texto, foram criadas duas classes, com a função de receber um valor numérico ou um valor data e desmembrar esta variável transformando em uma string para poder ser passada para API de sintetização de voz.

Quanto ao retorno de voz para o telefone do cliente, não trata-se da aplicação, pois a JTAPI não fornece nenhum serviço de retorno de informações a chamada telefônica. Este serviço é restrito a parte de *hardware*, para retornar as informações ao telefone deve ser utilizado um *modem* com tecnologia *Via Voice*, que permite que o *modem* retorne também informações ao meio telefônico por meio de sons.

Na parte de testes da aplicação desenvolvida a maior dificuldade, foi à captura da chamada telefônica, pois além de utilizar o evento que indica que o telefone esta chamando e capturar a ligação pelo terminal definido, é necessário a configuração do *modem*.

Cada *modem* possui um padrão de comandos, que indicam quando ele vai transmitir ou receber informações, entre uma série de outras funções. A JTAPI força o programador a informar os comandos do *modem* que indicam que ele está recebendo uma ligação, o comando para iniciar a parte de retorno, o codificador utilizado, a porta onde o *modem* esta conectado, além de uma série de outros parâmetros.

Na implementação do protótipo ainda não foi realizada o retorno das informações ao telefone do cliente, as informações estão sendo retornadas apenas nas

caixas de som do computador. Neste caso para redirecionar o som da caixa de computador para o telefone, existe opções de configuração no próprio modem, porém deve informar os comandos corretos para utilização de sons pelo modem .

CONCLUSÃO

O protótipo de integração entre telefone e computador via unidade de resposta audível foi desenvolvido para atuar na área de tele-atendimento, fazendo uma ligação direta entre os clientes e o sistema corporativo da empresa. Este processo elimina ou pelo menos reduz o número de atendentes destas empresas, pois com ele o cliente obtém as informações desejadas, obtidas por uma interação com o sistema corporativo da empresa.

O material teórico deste trabalho, deu-se por meios de dados bibliográficos existentes sobre os itens que o compõem, além de consultas na Internet e conversas com profissionais da área de telefonia.

A funcionalidade do protótipo foi projetada com base, nos sistemas desta área disponíveis no mercado e consultas a usuários deste tipo de serviço, sempre buscando agregar melhorias ao trabalho proposto.

Existem no mercado interfaces que permitem aos programadores capturar ligações telefônicas para o computador, utilizando funcionalidades do *modem*. Estes recursos são possíveis devido as *API* de programação serem desenvolvidas para visar esta ligação entre *modem* e o sistema telefônico, pois o meio telefônico utiliza um padrão diferente de comunicação do computador.

Os responsáveis pela criação destas *APIs* de programação, tem como principal objetivo o encapsulamento das funções de controle da chamada, ou seja, os programadores que utilizarem estas *APIs* não precisam conhecer como é realizada a comunicação entre aparelhos com os padrões diferentes. Assim pode-se criar objetos que controlam esta chamada utilizando os eventos e atributos de uma chamada telefônica, que são: chamada, conexão e terminal conexão.

Estes objetos possuem atributos e eventos que permitem o controle do objeto e a maneira que este atuará no controle da chamada. O objeto conexão representa a ligação entre a chamada e o dispositivo representado pelo objeto terminal conexão. O objeto chamada representa a sessão lógica entre a pessoa que está realizando a ligação e a pessoa que está recebendo a ligação. O objeto terminal conexão pode representar o dispositivo físico ou lógico que faz ou recebe a ligação.

O grande objeto de estudo deste trabalho é utilizar a abstração dos serviços telefônicos, com a utilização da *JTAPI* que controla esta chamada dividindo-a em objetos, diminuindo o custo e facilitando o acesso de pequenos empresários a esta tecnologia.

Existe uma limitação deste trabalho em relação aos operadores de telefonia, pois este trabalho de captura pelo *modem* é restrito à sinalização analógica, se o cliente utilizar uma operadora com sinalização digital, não será possível a captura da ligação pelo *modem*.

Ao término do trabalho, verificou-se que a grande dificuldade da integração entre telefone e computador, está à nível de hardware, principalmente nos padrões de comunicação utilizados diferentes, pelo computador e pelo sistema telefônico.

Este fato pode ser verificado na definição dos comandos do *modem* para indicar à *JTAPI*, que o computador está recebendo uma chamada, para cada *modem* utilizado existe um comando diferente, tanto que a *JTAPI* fornece um arquivo específico para os programadores informarem os comandos utilizados pelo *modem* no controle da chamada telefônica.

Obteve-se grande dificuldade na captura da ligação do *modem*, pois além de implementar a rotina para captura deste, é necessária a configuração do *modem* com

seus comandos indicando que este recebendo uma ligação, além de informar em qual porta ele está conectado.

A JTAPI permite utilizar como terminal de conexão o modem ligado a linha telefônico, ou utilizar os terminais de um rede, definindo cada um como um telefone. A dificuldade neste caso ocorre porque o comando *getterminals* busca os terminais da rede e o comando *getterminal* busca o *modem* ligado a rede telefônica.

No início do trabalho, Unidade de Resposta Audível era tratado como um *hardware* que disparava mensagens, no entanto durante o trabalho adquiriu-se conhecimento de que trata-se de um conjunto, englobando *hardware* e *software* utilizados para fazer este tratamento de recebimento e envio de informações entre telefone e computador.

Quanto ao retorno das informações do cliente, foi realizado o estudo, porém não foi terminada a implementação até a conclusão deste trabalho. No estudo realizado vários fatores indicam que esta funcionalidade está diretamente ligada ao arquivo de configuração da *JTAPI* para controle do *modem*, no caso deve ser especificado os comandos corretos do *modem* para retorno ao sistema telefônico.

Embora realizada a implementação na área de integração e computador, propõe-se uma série de trabalhos futuros:

- a) o protótipo foi criado com base em um banco de dados específico, propõe-se uma maneira de configurar este protótipo para qualquer banco de dados;
- b) permitir ao usuário criar opções de consulta, com o sistema alterando dinamicamente as consultas de *SQL*;
- c) continuidade deste estudo para retornar estas informações ao telefone do cliente.

REFERÊNCIAS

ANSCHUTZ, A. **Digital cellular telecommunications**. New Jersey : Prentice Hall, 1997.

BARROS, Lindeberg. **Redes de computadores : dados, voz e imagem**. São Paulo : Érica, 2002.

COMER, Douglas. **Internet working with TCP/IP**. São Paulo : Prentice Hall, 2001.

CTI Server. Disponível em: < <http://www.pacttelecon.com.br/ura.html>>. Acesso em: 10 fev. 2005.

DANTAS, Mário. **Tecnologia de redes de comunicação e computadores**. Rio de Janeiro : Axcel, 2002.

DEITEL, Harvey. **Advanced Java : complete training course**. New Jersey : Prentice Hall, 2001.

DODD, Annabel. **O guia essencial para telecomunicações**. Rio de Janeiro : Campus, 2000.

FABIANO, Anteu. **Infra-estrutura, protocolos e sistemas operacionais de lans**. São Paulo : Érica, 2004.

FALBRIARD, Claude. **Protocolos e aplicações para redes de computadores**. São Paulo : Érica, 2002.

FERNANDES, Alexandre. **Redes de computadores : fundamentos**. São Paulo : Érica, 2004.

GALLO, Michael. **Comunicação entre computadores e tecnologias de rede**. Rio de Janeiro : Thomsom Learning, 2003.

HUGHES, Merlim. **Java : networking programming**. New Jersey : Prentice Hall, 2004.

HOOGE, H. **The communicating PC**. São Paulo : Makron Books, 1996.

INTELICON. Disponível em: <<http://www.delgrande.com.br/uraintelicon.html>>. Acesso em: 15 mai. 2005.

LIMA, Valter. **Telefonia e cabeamento de dados**. São Paulo : Érica, 2001.

MILANO, Paulo. **Comunicações movies : tecnologias e aplicações**. São Paulo : Érica, 2002.

MONTEIRO, Emiliano. **Projeto de sistemas e banco de dados**. Rio de Janeiro : Brasport, 2004.

MULT. URA. Disponível em: <http://www.comodity.com.br/portugues/plat_ura.html>. Acesso em: 01 jun. 2005.

NETO, Soares. **Telecomunicações tecnologia de centrais telefônicas**. São Paulo : Érica, 2001.

PEREIRA, Maurício. **Projeto de banco de dados : uma visão prática**. São Paulo : Érica, 1996.

PEREIRA, Willian. **Fundamentos de banco de dados**. São Paulo : Érica, 2002.

SAMPAIO, Marcelo. **Telefonia celular digital**. São Paulo : Érica, 2004.

SAMPAIO, Marcelo. **Telefonia digital**. São Paulo : Érica, 2003.

SOARES, Luís. **Redes de computadores**. Rio de Janeiro : Campus, 1995.

SPENCER, Roberts. **Essential JTAPI**. New Jersey : Prentice Hall, 1999.

TANENBAUM, A. **Computer networks**. São Paulo : Prentice Hall, 1996.

TOLEDO, Adalton. **Redes de acesso em telecomunicações**. São Paulo : Makron Books, 2000.

TOTTEL, Ed. **Redes de computadores**. Porto Alegre : Bookman, 2003.

URA Informatec. Disponível em: <http://www.informatec-sp.com/predUra_e.html>. Acesso em: 10 mai. 2005.

VALDURIEZ, Patrick. **Princípios de sistemas de banco de dados distribuídos**. Rio de Janeiro : Campus, 2001.

BIBLIOGRAFIA RECOMENDADA

BARNES, David. **Programação orientada a objetos com java**. São Paulo : Makron Books, 2004.

BRODGEN, Bill; MINNICK, Chris. **Guia do desenvolvedor java**. São Paulo : Makron Books, 2002.

HORSTMANN, Cay. **Conceitos de computação com o essencial java**. Porto Alegre : Bookman, 2005.

LAFORE, Robert. **Estruturas de dados e algoritmos em java**. Rio de Janeiro : Ciência Moderna, 2005.

MOREIRA, Luís. **Java para iniciantes**. Rio de Janeiro : Ciência Moderna, 2002.

REESE, George. **Jdbc e java : programação para banco de dados**. São Paulo : Berkeley, 2002.

SANTOS, Rafael. **Introdução à programação orientada a objetos usados em java**. Rio de Janeiro : Campus, 2002.

SCHILDT, Herbert. **A arte de java**. Rio de Janeiro : Campus, 2003.

SIKORA, Makael. **Java : guia prático para programadores**. Rio de Janeiro : Campus, 2003.