

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

PROTÓTIPO DE UM SISTEMA DE VOIP (VOZ SOBRE IP)

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

JOSÉ MÁRCIO DE SOUSA

BLUMENAU, NOVEMBRO/2001.

2001/2-29

PROTÓTIPO DE UM SISTEMA DE VOZ SOBRE IP

JOSÉ MÁRCIO DE SOUSA

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

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Francisco Adell Péricas — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Francisco Adell Péricas

Prof. Sérgio Stringari

Prof. Miguel A. Wisintainer

AGRADECIMENTOS

Em primeiro lugar, a Deus, por tudo.

Aos meus pais, José Orlai e Celeuma, pela compreensão, apoio e carinho durante estes anos de estudo.

Ao meu orientador, prof. Francisco Adell Péricas, pela orientação sempre oportuna, esclarecedora, inteligente e pelo seu incentivo, fazendo com que este trabalho pudesse ser realizado.

A minha esposa Kerly e minha filha Maryane pela compreensão e dedicação durante estes anos de estudo.

A todos aqueles que, direta ou indiretamente, contribuíram para a realização deste trabalho.

SUMÁRIO

AGRADECIMENTOS	iii
SUMÁRIO.....	iv
LISTA DE FIGURAS	vi
LISTA DE ABREVIACÕES	vii
RESUMO	ix
ABSTRACT	x
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS.....	3
1.2 ESTRUTURA DO TRABALHO.....	3
2 TELEFONIA IP.....	4
2.1.1 AMBIENTE PARA TELEFONIA IP.....	6
2.1.2 VOZ.....	7
2.2 VOIP.....	7
2.2.1 TCP/IP.....	9
2.2.2 H.323.....	13
2.2.2.1 PILHA DE PROTOCOLOS	16
2.2.3 RTP/RCTP – REAL TIME PROTOCOL/REAL TIME CONTROL PROTOCOL.....	19
2.2.3.1 RTP.....	20
2.2.3.2 RTCP CONTROLE E GERENCIAMENTO	23
2.2.3.3 FORMATOS DE <i>PAYLOAD</i>	24
2.3 <i>MULTICAST</i> NA INTERNET	25
3 TECNOLOGIAS UTILIZADAS.....	27
3.1 ANÁLISE ESTRUTURADA.....	27
3.1.1 MODELO ENTIDADE-RELACIONAMENTO (MER)	28
3.1.2 DIAGRAMA DE FLUXO DE DADOS (DFD)	29
3.2 BORLAND C++.....	31
4 DESENVOLVIMENTO DO PROTÓTIPO	32
4.1 ESPECIFICAÇÃO DO PROTÓTIPO	32
4.2 IMPLEMENTAÇÃO	40
4.2.1 UTILIZANDO DA BIBLIOTECA.....	40

4.2.2 CRIANDO UMA SESSÃO	42
4.2.3 DEFININDO UMA SESSÃO ATIVA	42
4.2.4 USO DA BIBLIOTECA JTHREAD e JMUTEX.....	45
4.3 UTILIZAÇÃO DO PROTÓTIPO DO SISTEMA	46
5 CONCLUSÃO	53
5.1 SUGESTÕES PARA TRABALHOS FUTUROS	54
5.2 DIFICULDADES ENCONTRADAS	54
REFERÊNCIAS BIBLIOGRÁFICAS	55

LISTA DE FIGURAS

Figura 1 - Telefonia IP entre dois computadores.....	6
Figura 2 – Arquitetura do protocolo TCP/IP.....	10
Figura 3 – Estrutura do <i>header</i> UDP.....	13
Figura 4 – Pilha de protocolo H.323.....	17
Figura 5 – Cabeçalho RTP.....	22
Figura 6 – Tipos de endereços <i>Multicast</i>	25
Figura 8 – Exemplo de um fluxo de dados.....	30
Figura 9 – Diagrama macro do protótipo.....	33
Figura 10- Cadastro de usuário.....	34
Figura 11- Fluxograma do processo para criar uma sessão VoIP.....	35
Figura 12- Fonte do botão discar.....	36
Figura 13- Fluxograma do botão discar.....	37
Figura 14- Fonte do botão desligar.....	38
Figura 15- Diagrama de contexto do protótipo.....	38
Figura 16- Diagrama de fluxo de dados do protótipo.....	39
Figura 17- Dicionário de dados.....	40
Figura 18- Representa a criação de uma sessão.....	41
Figura 19- Componente utilizados na Biblioteca JVOIP.....	44
Figura 20- Inicialização da JMutex.....	45
Figura 21- Inicialização da JThread.....	46
Figura 22- Tela inicial do protótipo.....	46
Figura 23- Cadastro.....	47
Figura 24- Cadastro de usuários.....	47
Figura 25- Selecionando operações.....	48
Figura 26- Operações.....	49
Figura 27- Seleção por Nome.....	50
Figura 28 – Seleção por Ramal.....	51
Figura 31- Selecionando o sobre.....	52
Figura 32- Sobre.....	52

LISTA DE ABREVIações

CNAME - Canonical Name

CSRC – Contributing Source

DARPA - Defense Advanced Research Project Agency

DPCM -Diferencial Pulse Code Modulation

FTP - File Transfer Protocol

IGMP - Internet Group Management Protocol

IP - Internet Protocol

ITU-T – International Telecom Union

LAN - Local Area Network

MCU – Multipoint Control Unit

NGN - Next Generation Network

PCM - Pulse Code Modulation

PSTN - Public Switched Telefone Network

QoS – Qualidade de Serviço

RAS - Regitration Admission and Status

RTCP - Real Time Control Protocol

RTP - Real Time Protoco

RTPC - Rede telefônica Comutada

SMTP - Simple Mail Transfer Protocol

SSRC – Synchronization Source

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

VAD - Vocie Activety Detection

VoIP - Voz sobre IP

RESUMO

A popularidade da Internet como meio de baixo custo tem despertado o interesse por tecnologias para a comunicação de voz utilizando o protocolo IP (*Internet Protocol*). Este trabalho especifica e implementa um protótipo de um sistema VoIP (voz sobre IP), desenvolvido para a comunicação *full-duplex* de voz, de computador-a-computador (PC to PC), em ambiente Windows. O controle do estabelecimento da comunicação entre dois computadores foi feito utilizando biblioteca *freeware* H.323.

ABSTRACT

The Internet popularity as a low cost communication media raised the interest about technologies for voice communication using the IP protocol (Internet Protocol). This work intends to specify and implement a prototype of a VoIP system (voice over IP). The prototype was developed for full-duplex voice communication, from computer to computer (PC to PC), on a Windows environment. The control of the communication establishment between two computers was done using a H.323 freeware library.

1 INTRODUÇÃO

Soluções baseadas em *Internet Protocol* (IP) têm sido propostas em substituição aos modelos de telefonia convencional, com inúmeras vantagens. Esse trabalho propõe-se a apresentar uma possibilidade de uso desta tecnologia em substituição ao sistema telefônico convencional, aproveitando todas as facilidades da tecnologia IP.

O método básico para comunicação telefônica é estabelecer um circuito entre dois assinantes: isto se faz ainda hoje na maioria das ligações tradicionais. Apesar da telefonia ter evoluído para circuitos digitais e multiplexados, a presença do circuito é indispensável na comunicação. A telefonia pública hoje é conhecida como *Public Switched Telephone Network* (PSTN) ou Rede Telefônica Pública Comutada (RTPC). Na telefonia o usuário é conhecido sempre por assinante.

Com a utilização de redes de pacotes para tráfego de voz elimina-se a necessidade da presença de um circuito. Dentro destes conceitos, a voz é empacotada e transmitida em redes de computadores juntamente com os dados. O protocolo IP é o utilizado para este processo. A rede que estiver habilitada com esse protocolo poderá trafegar também voz, por isso poderá ser um ponto fundamental na sua escolha.

Segundo Oliveira (2001), existem aspectos relacionados com a natureza do sinal de voz que permitem adicionar mecanismos de compressão, sendo eles: distribuição não uniforme de amplitudes, correlação entre amostras sucessivas, correlação entre ciclos sucessivos, fator de inatividade ou percentual de silêncio, densidade espectral média não uniforme confirmando a redundância de informações e densidade espectral instantânea, ou presença de formatos que se mantêm inalterados durante 20 a 30 ms.

Uma possibilidade interessante é a supressão de silêncio. Em uma conversação telefônica apenas 40% do tempo o canal voz está ativo, ou seja, o usuário está falando. Um mecanismo conhecido como *Voice Activity Detection* (VAD) é usado para perceber a presença do silêncio e removê-la.

Para realizar uma chamada são necessários protocolos de controle e sinalização para executar algumas tarefas com localização do usuário, notificação de chamada, início de transmissão de voz, finalização de transmissão de voz e desconexão.

Hoje em dia a arquitetura mais difundida no mercado para transmitir voz sobre *Local Area Network* (LAN) é o padrão H.323, proposto pela *International Telecom Union* (ITU-T). Este padrão é uma pilha de protocolos que está focado na conexão e controle da chamada, que são separados da transmissão de conteúdo (voz) entre os computadores.

Para a transmissão do fluxo de voz, utiliza-se o protocolo do modelo de referência *Transmission Control Protocol/Internet Protocol* (TCP/IP) chamado *Real Time Protocol* (RTP). O RTP utiliza o serviço de transporte *User Datagram Protocol* (UDP) para transmitir os pacotes. O RTP é necessário porque na Telefonia IP uma taxa de transmissão constante é fundamental, enquanto que a perda de pacotes pode ser desprezada.

Uma preocupação desse trabalho é a utilização de tecnologias abertas que estão se tornando padrão de mercado para não torná-lo dependente de tecnologias proprietárias.

Quanto ao protótipo, será implementado um software para a comunicação entre vários computadores utilizando a telefonia sobre IP, através da utilização dos protocolos do padrão H.323 do *International Telecom Union* (ITU-T) que farão o transporte da voz entre os recursos de rede utilizados, inclusive aqueles que permitem a mobilidade, com o propósito de reduzir sensivelmente os custos inerentes à telefonia convencional utilizado atualmente na empresa.

Alguns softwares comerciais e *freewares* que fornecem transmissão de voz sobre IP são implementados utilizando o padrão H.323, entre eles: Microsoft NetMeeting e Netscape Conference.

Segundo Craig (2001), está disponível também na forma de biblioteca de desenvolvimento a pilha de protocolos que implementa o padrão H.323 com código fonte em C e C ++, compatível para as plataformas Linux e Windows. Esta pilha foi desenvolvida por um grupo de pesquisadores e pode ser encontrada no projeto chamado de OpenH323. Como alternativa, também há a biblioteca distribuída livremente pela *Vovida Networks*.

O RTP é o protocolo padrão para transporte de dados que necessitam de transmissão em tempo real, tais como áudio e vídeo, e pode ser usado por diversos tipos de serviços de mídia interativos, tal como a telefonia IP. O protocolo é composto por transmissão de dados e por controle a chamada de Rede Telefônica Pública Comutada (RTCP). Os dados vêm de um

protocolo (RTP) leve que provê o suporte a aplicações com características de tempo real, incluindo reconstrução de mensagens, detecção de perdas, segurança, selo de tempo e identificador de conteúdo.

1.1 OBJETIVOS

O presente trabalho tem como objetivo desenvolver um protótipo de um software de Sistema de Voz sobre IP (VoIP) para a Internet através da especificação de um sistema telefônico baseado na arquitetura de uma Intranet.

O trabalho visa especificar uma solução alternativa para a utilização da infra-estrutura de rede na comunicação entre funcionários de uma empresa que não dependa da infra-estrutura de telefonia tradicional, possibilitando que empresas obtenham uma sensível redução de custos operacionais.

1.2 ESTRUTURA DO TRABALHO

A seguir é apresentada uma síntese dos capítulos constantes desse trabalho.

O capítulo de introdução apresenta uma visão geral do presente trabalho, o contexto em que está inserido, sua importância e objetivo.

O segundo capítulo apresenta uma fundamentação do tema Telefonia IP e VoIP, demonstrando uma visão geral do assunto.

O terceiro capítulo apresenta as tecnologias e ferramentas utilizadas para o desenvolvimento do software deste trabalho.

O quarto capítulo apresenta o software desenvolvido nesse trabalho. Descreve o protótipo de software, suas características, sua especificação, principais telas.

O quinto capítulo apresenta as conclusões desse trabalho e as sugestões para que o mesmo possa ter continuidade e seja melhorado.

2 TELEFONIA IP

Segundo Oliveira (2001), há décadas, desde a invenção do telefone, que a exigência básica para uma comunicação telefônica é o estabelecimento de um circuito entre os dois assinantes. Isto se dá ainda hoje na maioria das ligações telefônicas realizadas. Apesar do circuito ser digitalizado e multiplexado, sua presença é indispensável na comunicação telefônica, tratada por convencional, e que representa a telefonia pública hoje utilizada, conhecida como *Public Switched Telephone Network* (PSTN) ou Rede Telefônica Pública Comutada (RTCP).

Uma revolução está para acontecer no ambiente de telecomunicações. A necessidade da presença de um circuito está acabando em função da utilização de redes de comutação de pacotes para tráfego de voz. Dentro deste conceito, a voz é empacotada e transmitida em redes compartilhadas, juntamente com dados. E o protocolo preferido para este transporte é o *Internet Protocol* (IP). A esta nova concepção de rede telefônica dá-se o nome de *Next Generation Network* (NGN).

Segundo Monteiro (2000), o termo telefonia IP, Telefonia Internet ou ainda Voz sobre IP aplica-se à utilização de redes baseadas no protocolo IP na camada de rede para transporte de voz, através da Internet. As vantagens da utilização dessa tecnologia são:

- a) compartilhamento da rede para o tráfego de voz com o tráfego de dados (*e-mail*, etc);
- b) unificação de redes de transporte, sinalização e gerência sobre a mesma rede, com economia de infra-estrutura e manutenção;
- c) meio de transmissão de baixo custo, comparado ao sistema telefônico convencional;
- d) possibilidade de compactação e supressão de silêncio, reduzindo a largura de banda utilizada;
- e) utilização de rede já instalada;
- f) possibilidade de oferecer outros serviços como correio de voz, *call-center* via Internet, etc;
- g) possibilidade de uso de diversos serviços com *e-mail*, fax, voz, *web*, com o auxílio de reconhecimento e síntese de voz.

Um engano sobre a telefonia IP é pensar que seu maior benefício é para chamadas de longa distância de baixo custo. Apesar das ligações de longa distância realmente estarem promovendo o uso da telefonia IP, as razões pelas quais as companhias são atraídas é a facilidade de criação de serviços e a consolidação de suas redes. A principal vantagem de Voz sobre IP (VoIP) sobre a rede telefônica pública comutada é a facilidade de adição de novos serviços e funcionalidades, assim como a significativa diminuição dos custos de implantação e manutenção por parte das companhias telefônicas. Por exemplo, pode-se criar uma videoconferência pela definição de como o fluxo de vídeo é codificado e decodificado. Fazer videoconferência sobre a RTPC pode ser um processo difícil e caro, porque a RTPC tem conceito rigoroso de circuitos, todos eles com largura de banda normalmente de 64 kbps. Para aplicações que exijam largura de banda maior, existe a necessidade de mais de um circuito para transmissão de um mesmo fluxo de vídeo. Da mesma forma, como a telefonia IP é baseada em software é mais fácil adicionar serviços como correio de voz e outros.

A segunda vantagem para a telefonia IP é a consolidação das redes de dados. Atualmente, companhias telefônicas mantêm duas redes, uma para voz e outra para dados. Como redes de voz são 4 ou 5 vezes mais caras que as redes de dados, o uso de VoIP elimina a rede de voz, gerando grande redução de custos.

Pode-se ainda prever a integração de diversos serviços sobre o sistema telefônico, utilizando tecnologias agregadas como reconhecimento e síntese de voz. Por exemplo, é possível ouvir todos os *e-mails* recebidos por um telefone convencional, utilizando a síntese de voz e transmissão sobre IP para que estes *e-mails* cheguem por telefone. Desta forma, serviços como fax, WWW e outros podem ser acessados através de telefones comuns.

Será possível também a unificação das três redes hoje existentes para manutenção do sistema telefônico: a rede de transporte de voz, a rede de sinalização e a rede de gerência. Estas redes possuem hoje infra-estruturas independentes e muitas vezes sobre redes físicas diferentes. Com o uso da telefonia IP, unificam-se todas estas redes sobre a mesma infra-estrutura utilizando sempre o mesmo protocolo de rede: o protocolo IP.

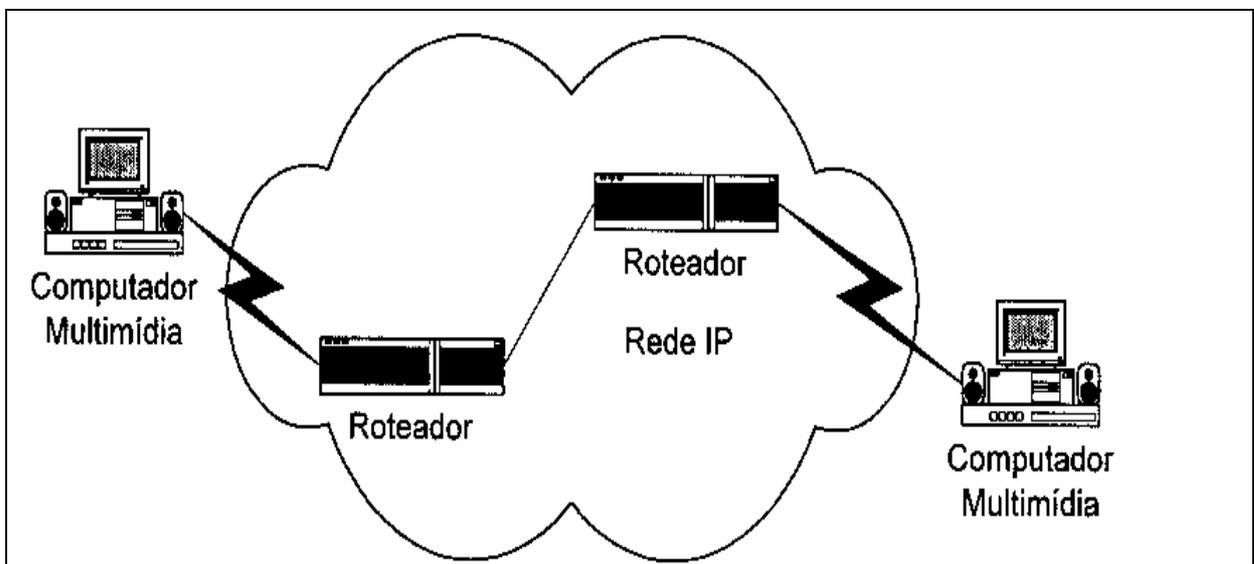
Um engano comum referente ao VoIP é a qualidade final de transmissão de voz. Se a rede estiver sobrecarregada, podem ser inseridos atrasos que podem atrapalhar a qualidade

final da voz. Todavia, se a telefonia IP é usada em uma rede privada dedicada, a qualidade final de voz está bem próxima do nível escolhido desejado pelos clientes.

2.1.1 AMBIENTE PARA TELEFONIA IP

De acordo com Oliveira (2001), o termo telefonia IP não é limitado somente a chamadas entre dois computadores. Na realidade, a telefonia IP refere-se ao fluxo de voz sobre pacotes transportados sobre redes que utilizam o protocolo IP para terminais onde o fluxo é decodificado em voz novamente. Esses terminais podem tanto ser computadores quanto telefones convencionais modulados.

Figura 1 - Telefonia IP entre dois computadores



O desenho mostrado na figura 1 é o mais simples da telefonia IP e é composto por dois usuários se comunicando sem que haja utilização de rede telefônica convencional. A codificação da voz é feita pelos computadores envolvidos e a transmissão é feita através da rede IP. Vários softwares estão disponíveis para esta aplicação, podendo utilizar um protocolo proprietário ou padrão, permitindo a interação de softwares de diferentes fabricantes.

2.1.2 VOZ

A conversão humana é uma forma de onda mecânica com frequências principais na faixa que vai de 300 Hz a 3,4 kHz, com alguns padrões de repetição definidos em função do timbre de voz e dos fonemas emitidos durante a conversão. O primeiro problema da telefonia em geral é a reprodução com qualidade da voz humana em um terminal à distância.

Em um ambiente de telefonia totalmente analógico isto é possível pela transmissão da forma de onda entre os interlocutores através de meio metálico, com possíveis ampliações analógicas. Isto, porém, representava um custo alto pela impossibilidade de se utilizar o meio físico para a transmissão de mais de um canal de conversação. Com o advento da telefonia digital, a voz é codificada em formato digital, que pode ser multiplexado no tempo de forma a compartilhar meios de transmissão. A representação digital de áudio oferece algumas vantagens: alta imunidade a ruído, estabilidade e reprodutividade.

Duas formas podem ser usadas para a codificação da voz humana:

- a) baseado na forma de onda;
- b) baseado nos padrões de voz.

A codificação baseada na forma de onda já é utilizada hoje na telefonia convencional para digitalizar a voz, de forma a permitir a multiplexação dos circuitos. A técnica hoje utilizada é conhecida por *Pulse Code Modulation* (PCM).

Os dispositivos responsáveis pela codificação de voz são conhecidos como *voice codecs*, ou simplesmente vocoders, como será tratado neste documento. Estes elementos são responsáveis pela codificação da voz em um fluxo de bits, possivelmente utilizando técnicas de compressão de voz e supressão de silêncio.

2.2 VOIP

De acordo com Monteiro (2001), Voz sobre IP ou VoIP é uma tecnologia que permite a digitalização e codificação de voz e o empacotamento de dados IP para a transmissão em uma rede que utilize TCP/IP. Devido ao volume de dados gerado por uma aplicação VoIP, esta tecnologia se encontra em funcionamento em redes corporativas privadas, mas se a rede

base para transporte desta aplicação for a Internet, não é aconselhável que seja utilizada para fins profissionais, pois o TCP/IP não oferece padrões de Qualidade de Serviço (QoS) comprometendo desta forma a qualidade da voz. A qualidade da voz fica dependente do tráfego de dados existentes no momento da conversa.

Uma grande diferença entre uma aplicação de dados e uma aplicação de voz é que uma aplicação de voz é sensível ao atraso. Em uma rede IP não é possível garantir um atraso constante o que pode tornar uma aplicação de voz em tempo real, como por exemplo, uma ligação telefônica, um serviço de baixa qualidade com a voz entrecortada e muitas vezes inteligível.

De acordo com Shulzrinne (2001), a tecnologia de Voz sobre IP, conhecida pela abreviação VoIP, vem se popularizando rapidamente. Institutos de pesquisa apontam crescimento exponencial da utilização desta tecnologia, principalmente por parte de médias e grandes empresas.

Voz sobre IP é um conceito relativamente simples: transformar voz em mais uma aplicação IP dentro de uma rede de dados que utilize IP como protocolo de nível de rede. Aliás, esta simplicidade é que permite transmitir dados e voz dentro de uma mesma rede, completamente anárquica e dispersa, a custos relativamente baixos. A grande diferença entre as aplicações de dados, excluindo-se multimídia, e as de voz é a incapacidade de uma rede de oferecer atraso constante a uma aplicação de voz on-line, como é uma ligação telefônica, que causa retardos indesejáveis para os usuários. Na prática, esta limitação introduz voz entrecortada e muitas vezes ininteligível. A capacidade de uma rede em oferecer atrasos constantes é chamado de QoS.

De acordo com Monteiro (2000), transportar voz sobre uma rede de dados não é uma solução recente. Há anos a Embratel comercializa soluções baseadas em equipamentos TDM mas a solução é de alto custo, visto que há grande desperdício de largura de banda. Soluções baseadas em *Frame-Relay* também já são fornecidas há muito tempo, mas como *Frame-Relay* é um protocolo de enlace, a aplicação de voz estará disponível apenas nos pontos remotos onde for possível implantar esta tecnologia. Recentemente a tecnologia ATM também vem sendo adotada para este fim, mas seu alto custo e seu baixo nível de capilaridade tem sido um empecilho para o seu crescimento.

De acordo com Shulzrinne (2000), basicamente três fatores contaram para o crescimento da tecnologia de Voz Sobre IP: o desenvolvimento e padronização de protocolo que permite QoS em redes IP; o desenvolvimento acelerado de métodos de compressão de voz e a explosão da Internet.

Independente da tecnologia adotada, o movimento de integração entre voz e dados na mesma infra-estrutura de rede, há alguns anos já é esperado. As vantagens são claras, pois os custos envolvidos na manutenção de equipes técnicas, infra-estruturas diferenciadas e ligações internacionais são reduzidos com a integração. O aumento do leque de novas aplicações, da disseminação de microcomputadores pessoais (para funcionamento como terminal multimídia), das capilaridades redes IP e da banda de transmissão disponível para o usuário, contribuíram em muito para Voz sobre IP (VoIP) tornar-se uma realidade.

Contudo, a diferença de preço entre o terminal telefônico convencional e um equipamento para uso de VoIP, ainda é um forte fator limitante para uso desta última solução em larga escala. Além disso, a alta disponibilidade das redes telefônicas convencionais aliadas à falta de qualidade de serviço e de confiabilidade da rede, originalmente herdada do IP, são aspectos de peso na comparação entre os ambientes legado e VoIP.

Apesar e tais desvantagens, e devido aos enormes benefícios introduzidos pela integração entre telefonia e IP, a mudança de cenário de comunicação de voz e dados atual para uma realidade integrada em larga escala, onde os meios de transmissão deverão servir aos dois “mundos” de forma transparente ao usuário, é uma realidade almejada.

2.2.1 TCP/IP

Segundo Viana (2001), a arquitetura de rede TCP/IP é o nome que se dá a toda a família de protocolos utilizados pela Internet. Esta família de protocolos foi desenvolvida pela *Defense Advanced Research Project Agency* (DARPA) no Departamento de Defesa dos Estados Unidos (DoD).

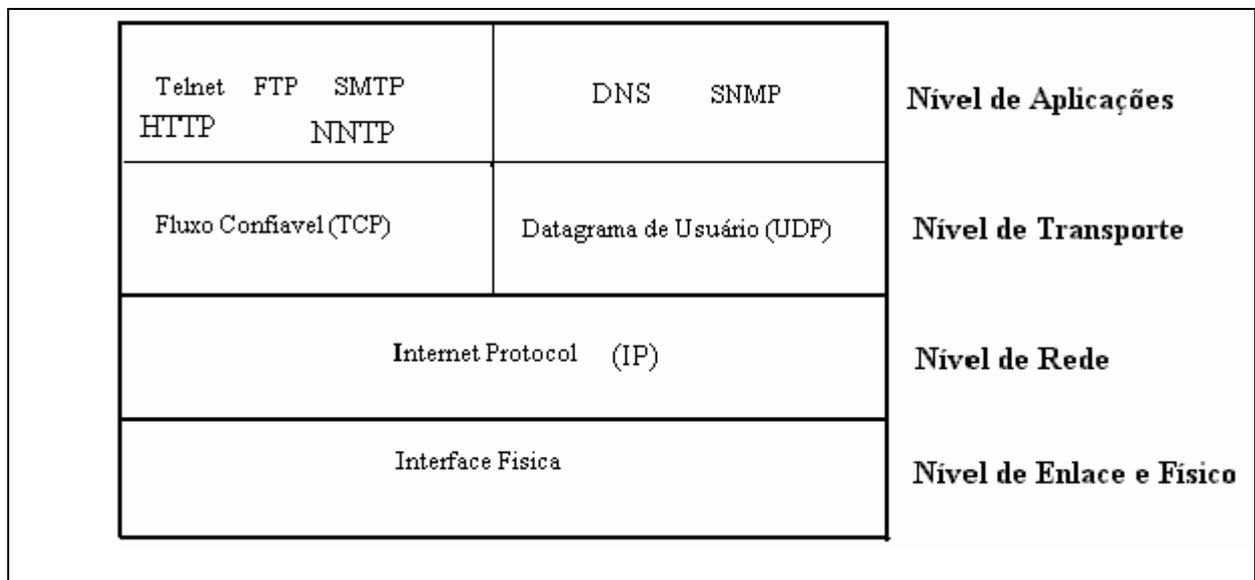
Este conjunto de protocolos foi desenvolvido para permitir aos computadores compartilharem recursos numa rede. Toda a família de protocolos inclui um conjunto de padrões que especificam os detalhes de como comunicar computadores, assim como também convenções para interconectar redes e rotear o tráfego.

Oficialmente esta família de protocolos é chamada modelo de referência *Transmission Control Protocol / Internet Protocol* (TCP/IP), comumente referenciada só como TCP/IP, devido a seus dois protocolos mais importantes TCP e IP.

O TCP é o protocolo da camada de transporte, que oferece um serviço confiável e orientado à conexão.

Assim como o TCP, outro protocolo da camada de transporte é o *User Datagram Protocol* (UDP), sem conexão e não confiável.

Figura 2 - Arquitetura do protocolo TCP/IP



Desde sua origem, o protocolo IP foi desenvolvido e implementado como um protocolo de comunicação para controle de tráfego baseado na regra do melhor esforço (*Best-effort Service*), que não provê nenhum mecanismo de qualidade de serviços e conseqüentemente, nenhuma garantia de alocação de recursos da rede. Na época, ninguém imaginava que a Internet se tornaria a grande rede mundial que é atualmente. E, desse rápido crescimento da Internet, a tendência atual é a integração de voz (telefonia) e dados numa única infra-estrutura de redes de pacotes IP. Essa emergente e crescente demanda pelos serviços IP *Telephony*, como chamado pelo mercado, provocou uma corrida frenética dos fabricantes de equipamentos de redes para desenvolver protocolos que garantissem qualidade de serviços fim-a-fim.

TCP/IP é portanto uma arquitetura de rede simples popularizada entre computadores com o sistema operacional UNIX, Windows e outros. Atualmente o TCP/IP está presente na maioria dos sistemas operacionais e é utilizado por muitas empresas em uma variedade de aplicações. TCP/IP é, especificamente, um protocolo de comunicação de dados projetado para aplicações não sensíveis ao atraso tais como: e-mail, web,ftp, etc.

A arquitetura do TCP/IP incorpora alguns protocolos que fornecem diversos serviços:

- a) o IP aceita dados segmentados (chamado *protocol data units*, ou PDUs) através de um host e os envia pela Internet através de gateways até o destino. A entrega não é confiável porque alguns PDUs nunca chegam ao destino;
- b) o *Transmission Control Protocol*(TCP) fornece mecanismos de transporte que garantem entrega livre de erros, sem pedras ou duplicação, e reconstituição de PDUs para corresponder à ordem enviada. O TCP gerencia a transferência entre dois processos chamados *transport users*. Permite multiplexação, gerência de conexões, transporte de dados, registro de erros, e diversas outras capacidades;
- c) o *User Datagram Protocol* (UDP) opera no mesmo nível que o TCP, oferecendo serviço de conexão para troca de mensagens. Mais rápido que o TCP, porque exclui recursos de confiabilidade do TCP, é adequado para aplicações orientadas para transações;
- d) o *File Transfer Protocol* (FTP) é projetado para passar arquivos ou parte de arquivos de um sistema para outro. Passa dados pelo TCP através de interface do sistema operacional. Os arquivos transferidos podem usar conjunto de caracteres ASCII, bem como incluir algumas opções de compressão de dados e mecanismos de identificação de senhas para controle de acesso;
- e) o *Simple Mail Transfer Protocol* (SMTP) fornece o mecanismo mas não a interface gráfica para correio eletrônico de rede. O protocolo permite que usuários enviem correspondência de uma LAN bem como da Internet;
- f) a função *TELNET* fornece um padrão de emulação de terminal, permitindo que terminais conectem e controlem aplicações operando em *host* remoto como se fosse usuário local. Tem um módulo de usuário para traduzir códigos de terminais em código específicos de rede e um módulo de servidor para interagir com processos e aplicações e conduzir terminais;
- g) o *Simple Network Management Protocol* (SNMP) é um protocolo que suporta a

troca de mensagens de gerência de rede entre *hosts*. Os *hosts* podem incluir um centro de gerência de rede, às vezes chamado de *host central*. O SNMP é projetado para operar sobre o UDP;

- h) o *Hypertext Transfer Protocol* (HTTP) é o protocolo que define como programas de navegação Internet e servidores Internet devem interagir de maneira a transferirem arquivos WWW.

Com TCP/IP, um processo leva dados para um *host*, que por sua vez os encaminha para outro processo. Um *host* transmissor fornece à rede o endereço do *host* receptor para garantir encaminhamento correto pela rede.

O protocolo UDP é restringido a portas e *sockets*, e transmite os dados de forma não orientada à conexão. Ele nada mais é do que uma interface para o protocolo IP. Esse protocolo substitui o protocolo TCP quando a transferência de dados não precisa estar submetida a serviços como controle de fluxo.

A função básica do UDP é servir de multiplexador ou demultiplexador para o tráfego de informações do IP. Assim como o TCP trabalha com portas que orientam adequadamente o tráfego de informação a cada aplicação de nível superior. Essas portas são:

- a) **porta de destino:** É uma parte do datagrama (uma extremidade) que indica o aplicativo ao qual deve-se enviar a informação que chega;
- b) **porta de origem:** Localiza-se no outro extremo do datagrama e indica o aplicativo que enviou a mensagem. Pode ser usado para um reenvio, ou quando não é utilizado, é preenchido com zeros.

A figura 3 mostra a estrutura do cabeçalho do UDP.

Figura 3 - Estrutura do *header* UDP



2.2.2 H.323

De acordo com Craig (2001), com os avanços nos sistemas de comunicações e com a popularização de serviços sobre as redes de computadores, surgem diariamente novas oportunidades e também novos desafios. Antigamente, os aplicativos de intercâmbio de dados não necessitavam de suporte para executar, por exemplo, operações em tempo real, contudo essa realidade mudou.

Dentro desse contexto temos o padrão H.323. Esse padrão descreve como áudio, vídeo, dados e informações de controle podem ser gerenciados em uma rede baseada em pacotes para disponibilizar serviços de conversação em equipamentos H.323.

O padrão H.323 provê uma arquitetura ação de dados multimídia, para redes baseadas no protocolo IP. O H.323 permite também que produtos de multimídia e aplicações de fabricantes diferentes possam interoperar de forma eficiente e que os usuários possam se comunicar sem preocupação com a velocidade da rede.

H.323 é uma recomendação da *Internation Telecom Union* (ITU-T), organismo que define padrões para redes de computadores e telecomunicações. Estas redes incluem TCP/IP em cima de *Ethernet*, *Fast Ethernet* e *Token Ring*.

A especificação H.323 foi aprovada em 1996 pelo Grupo de estudos 16 do ITU e sua versão 2 foi aprovada em janeiro de 1998. O H.323 é parte de uma série padrões de comunicações que permitem vídeo conferência e VoIP através de redes.

Segundo Domingues (2000), a recomendação H.323 tem como uma de suas características a flexibilidade, pois pode ser aplicada tanto à voz, quanto a vídeo conferência e multimídia. Aplicações H.323 estão se tornando populares no mercado corporativo por várias razões, dentre elas podemos citar:

- a) o H.323 define padrões de Voz para uma infra-estrutura existente, além de ser projetada para compensar o efeito de latência em LANs, permitindo que os clientes possam usar aplicações de voz sem mudar a infra-estrutura de rede;
- b) as redes baseadas em IP estão ficando mais velozes, além da largura de banda para redes com arquitetura Ethernet estarem migrando de 10 Mbps para 100 Mbps, e a Gigabit Ethernet está fazendo progressos no mercado;
- c) o H.323 provê padrões de interoperabilidade entre LANs e outra redes;
- d) o fluxo e dados em redes pode ser administrado. Com o H.323, o gerente de rede pode restringir a quantidade de largura de banda disponível para conferências e voz. O suporte à comunicação *Multicast* também reduz exigências de largura de banda;
- e) a especificação H.323 tem o apoio de muitas empresas de comunicação e organizações, incluindo a Intel, Microsoft, Cisco e IBM. Os esforços destas companhias estão gerando um nível mais alto de consciência no mercado.

De acordo com Nóbrega (2001), a ITU-T propôs o padrão H.323, sendo mais difundido atualmente, especialmente por ser o precursor da Telefonia IP e ser o primeiro padrão a tratar deste tema. As principais características deste padrão são:

- a) especifica algoritmos padrões de compressão que devem ser implementados de forma a garantir compatibilidade, conhecidos como áudio *codecs* ou *vocoders*;
- b) cria protocolos utilizados para o controle da chamada, estabelecimento dos canais de comunicação e negociação de qualidade de serviço;
- c) permite a interoperabilidade com outros terminais de voz, como telefonia convencional, RDSI, voz sobre ATM e outros, permitindo assim a construção de gateways;
- d) descreve elementos ativos do sistema e suas funções.

O protocolo H.323 utiliza em suas diversas funcionalidades uma família de recomendações ITU-T: H.225 para conexão, H.245 para controle, H.332 para conferências, H.335 para segurança, H.246 para interoperabilidade com RTPC e a série H.450.x para serviços suplementares. Todos estes padrões fazem parte da série H de recomendações.

Verifica-se também na recomendação H.323, os elementos que compõem uma rede de telefonia IP. Estes elementos aparecem na figura 4 e podem ser definidos como segue:

Terminal H.323 – Computador onde está implementado o serviço de telefonia IP, atuando como terminal de serviço de telefonia IP, como terminal de voz, vídeo e dados, através de recursos multimídia. Esses são os clientes da LAN que fornecem comunicação em tempo real e nas duas direções. Todos os terminais H.323 têm que suportar o H.245, Q.931, *Registration, Admission and Status* (RAS) e RTP.

Os terminais H.323 podem também incluir o protocolo de conferencia de dados T.120, codificadores de vídeo e suporte para MCU. Um terminal H.323 pode comunicar com outro terminal, um *gateway* ou um MCU;

Gateway H.323 – Elemento situado entre uma rede IP e outra de telecomunicações, como o sistema telefônico convencional (RTPC), rede integrada de serviços digitais (RDSI), rede de telefonia celular; de forma a permitir a interoperabilidade entre as duas redes.

Um *gateway* H.323 é um ponto final da rede que fornece comunicação em tempo real nas duas direções entre terminais H.323 em uma rede IP e outros terminais ITU em uma rede comutada ou para outro *gateway* H.323. Eles executam a função de translação entre diferentes formatos de dados.

Os *gateways* são opcionais em uma LAN onde os terminais se comunicam diretamente, mas quando os terminais precisam se comunicar com um ponto final em outra rede, a comunicação se faz via *gateway* através dos protocolos H.245 e Q.931.

Gatekeeper – Ele é o componente mais importante de um sistema H.323 e executa a função de gerente. Ele atua como ponto central para todas as chamadas dentro de sua zona (é a agregação do *gatekeeper* e dos terminais registrados nela), e fornece serviços aos pontos finais registrados. Algumas das funcionalidades que os *gatekeepers* fornecem são:

- a) tradução de endereços: tradução de um endereço alias (o endereço alias fornece um método alternativo de endereçamento de um ponto. Ele pode ser um endereço de e-mail, um número telefônico ou algo similar.) para um endereço de transporte. Isto é feito usando-se uma tabela de tradução que pode ser atualizada através de mensagens de registro;
- b) controle de admissão: o *gatekeeper* pode permitir ou negar acesso baseado em autorização de chamada, endereço de fonte e destino, etc;
- c) sinalização de chamada: o *gatekeeper* controla o processo de sinalização entre dois pontos finais que querem se conectar;
- d) autorização de chamada: o *gatekeeper* pode rejeitar chamadas de um terminal devido a falhas de autorização através do uso de sinalização H.225. As razões para rejeição poderiam ser acessos restrito durante alguns períodos de tempos ou acesso de certos terminais ou *gateways*;
- e) gerenciamento de largura de faixa: Controle do número de terminais que podem acessar simultaneamente a rede. Através do uso da sinalização H.225, o *gatekeeper* pode rejeitar chamadas de um terminal devido à limitação de largura de faixa;
- f) gerenciamento da chamada: O *gatekeeper* pode manter uma lista de chamadas H.323 em andamento. Essa informação pode ser necessária para indicar que um terminal chamado está ocupado, e fornecer informações para a função de gerenciamento de largura de faixa.

2.2.2.1 PILHA DE PROTOCOLOS

Segundo Monteiro (2000), a figura 4 mostra a pilha de protocolos do H.323. OS pacotes de áudio, e vídeo e registro usam o UDP enquanto que os pacotes de dados e controle usam o TCP.

Figura 4 - Pilha de protocolo H.323

Aplicações de áudio	Aplicações de Vídeo	Controle e Gerenciamento			
G.711 G.723 G.728 G.729	H.261 H.263	RTCP	H.225.0	H.225.0 Canal de Sinalização de Chamada	H.245 Canal de Controle
RTP			Canal RAS	X.224 Classe 0	
UDP			TCP		
Nível de Rede (IP)					
Nível de Enlace					
Nível Físico					

A H.323 fornece três protocolos de controle: sinalização de chamada H.225.0, RAS H.225.0 e o controle de mídia H.245. O H.225.0 é usado em conjunto como o H.323 e fornece a sinalização para controle de chamada. Para fazer uma chamada de uma fonte para um receptor, um canal RAS H.225 é usado. Após o estabelecimento da chamada, o H.245 é usado para negociar o fluxo de mídia.

Aplicação e Áudio Codecs e Vídeo Codecs - utiliza o protocolo *Real Time Control Protocol* (RTP) para transmissão de pacotes.

H.225.0: RAS Registration, Admission and Status – o canal RAS é usado para comunicação entre pontos finais e o *gatekeeper*. Uma vez que as mensagens RAS são enviadas utilizando-se o UDP (um protocolo não confiável), o uso de *timeouts* e retransmissão de mensagens é recomendado. Os procedimentos definidos por um canal RAS são:

- descoberta do *gatekeeper*: este é o processo utilizado pelos pontos finais para determinar o *gatekeeper* no qual eles devem se registrar;
- registro de ponto final: este é o processo pelo qual um ponto final junta-se uma zona e informa ao *gatekeeper* sobre seus endereços e de transporte;
- localização do ponto final: um ponto final ou *gatekeeper* que possui um endereço alias de um ponto final e deseja determinar suas informações de contato pode

utilizar uma mensagem específica;

- d) mensagens: o canal RAS também é usado para transmissão de mensagens de admissão, mudança de largura de faixa, status e desligamento.

RTP/RTCP Real Time Protocol / Real Time Control Protocol -usado para transporte do fluxo de pacotes multimídia, com características de tempo real, executando também funções de estatísticas de qualidade de serviços.

H.225.0: Sinalização de chamada - o canal de sinalização de chamada é usado para carregar mensagens de controle H.225. Em redes que não possuem uma gatekeeper, as mensagens de sinalização de chamadas são passadas diretamente entre o ponto que chamou e o que foi chamado, utilizando-se um endereçamento de sinalização de chamada. Em redes que contem um gatekeeper, a troca de mensagens de admissão iniciais são feitas entre o ponto que chama e o gatekeeper, através de mensagens de endereçamento RAS.

H.245: Controle de conferência e mídia - o H.245 é o protocolo de controle de mídia que os sistemas H.323 utilizam depois que a fase de estabelecimento de chamada foi completada. O H.245 é usado para negociar e estabelecer todos os canais de mídia conduzidos pelo RTP/RTCP.

O H.245 é usado para possibilitar o uso de canais, o Q.931 é usado para a sinalização e o estabelecimento da chamada, o RTP é o protocolo de transporte em tempo real que carrega os pacotes de voz enquanto que o RAS é usado para interação com o gatekeeper. As funcionalidades oferecidas são as seguintes:

- a) determinação de mestre e do escravo: o H.245 designa um Controlador de Multiponto (MC) que é responsável pelo controle central em caos onde uma chamada é estendida a uma conferência.;
- b) troca de capacidades: o H.245 é usado para negociar as capacidades quando uma chamada é estabelecida. A troca de capacidades pode ocorrer em qualquer momento durante a chamada, portanto possibilitando renegociar a qualquer momento;
- c) controle do canal de mídia: após os pontos finais de uma conferência terem trocado capacidades, eles podem abrir e fechar um canal lógico de mídia;
- d) controle de conferência: em conferências, o H.245 fornece aos pontos finais anúncios mútuos e estabelece o modelo de fluxo de mídia entre todos os pontos

finais.

Q.931 – Mensagem trocadas entre terminais para sinalização de chamada

Multipoint Control Unit (MCU) – o MCU é um ponto final da rede que fornece a capacidade de três ou mais terminais e *gateways* de participarem de uma conferência multiponto. O MCU consiste de um Controlador Multiponto MC e Processadores Multiponto MP. O MC determina as capacidades comuns dos terminais usando o H.245 mas ele não executa a multiplexação de áudio, vídeo e dados. A multiplexação dos fluxos de mídia é feita pelo MP sobre o controle do MC.

2.2.3 RTP/RCTP – REAL TIME PROTOCOL/REAL TIME CONTROL PROTOCOL

De acordo com Nóbrega (2001), o RTP é um protocolo padrão para transporte de dados com características de tempo real, como vídeo e áudio, que pode ser usado em diferentes serviços como mídia sob demanda e interativos. O protocolo é composto por uma parte de transmissão de dados e outra de controle, chamada RTCP (*Real Time Control Protocol*). A parte de dados consiste de um protocolo leve, que provê suporte para aplicações com características de tempo real, incluindo reconstrução temporal de mensagens, detecção de perdas, segurança, selo de tempo e identificação de conteúdo.

O protocolo de transporte envolve acompanhar o fluxo de bits gerados pelo codificador de mídia, normalmente o telefonia IP, quebrando em pacotes, enviando-os pela rede e reproduzindo o fluxo de bits no receptor. O processo é complexo porque pacotes podem ser perdidos, ter atrasos variados e serem entregues fora de ordem. O protocolo de transporte deve permitir ao receptor detectar estas perdas. Ele deve também transportar informações de temporização de forma que o receptor possa fazer também compensação para o atraso.

Algumas funcionalidades do RTP são: seqüenciamento, sincronismo intramídia, identificação de conteúdo, identificação de quadro, identificação de origem.

Já o RCTP, que acompanha o RTP, provê informações adicionais sobre seus participantes, tais como retorno de informações de qualidade de serviço, sincronismo intermídia e identificação do usuário.

Segundo Oliveira (2001), RTCP necessita que todos os participantes enviem estas informações periodicamente. O protocolo usa o mesmo endereço do RTP, porém em porta diferente. Nem todas as aplicações RTP utilizam o RTCP, ou seja, pode ser dispensável para algumas aplicações.

Aplicações em tempo real, tais como VoIP e fluxo de vídeo, têm um número de requisitos que as distinguem dos serviços de dados tradicionais da Internet:

- a) **seqüência:** Os pacotes devem ser reordenados em tempo real no receptor, caso eles cheguem fora de ordem. Se um pacote é perdido, ele deve ser detectado e compensado sem retransmissão;
- b) **sincronização intra-mídia:** O intervalo de tempo que existe entre pacotes sucessivos deve ser transmitida ao receptor como informação de controle. Por exemplo, nenhum dado é geralmente enviado durante períodos de silêncio na fala. A duração desse silêncio deve ser reconstruída adequadamente. Se um número de diferentes mídias estão sendo usadas em uma única sessão, devem haver meios de sincronizá-las. Com isso, é possível sincronizar o sinal de voz com o de vídeo. Isso também é conhecido como *lip-sync*;
- c) **identificação do payload:** Na Internet, freqüentemente é necessário modificar a deflação de mídia dinamicamente para ajustar-se à disponibilidade de largura de faixa ou a novos usuários que se juntam ao grupo. Algum tipo de mecanismo é necessário para identificar a codificação utilizada em cada pacote;
- d) **identificação de frame:** Vídeo e voz são enviados em unidades lógicas chamadas *frames*. É necessário indicar para o receptor onde é o início ou fim do *frame*, de forma a auxiliar no sincronismo da entrega dos dados.

Os serviços descritos acima são providos por um protocolo de transporte. Na Internet, o RTP é usado para esse objetivo. O RTP tem dois componentes: o primeiro é o próprio RTP, e o segundo é o RTCP.

2.2.3.1 RTP

De acordo com Oliveira (2001), o RTP é geralmente usado em conjunto com o UDP, mas pode fazer uso de qualquer protocolo de nível inferior baseado em pacotes. Quando um *host* deseja enviar um pacote de mídia, formata-o para empacotamento, adiciona qualquer

cabeçalho específico, adiciona o cabeçalho RTP, e o passa para a carga de um nível mais baixo.

O RTP fornece algumas funcionalidades que vão além de resequenciamento de detecção de perda:

- a) **multicast-friendly:** O RTP e o RTCP foram projetados para *multicast*. De fato, eles foram projetados para desde pequenos grupos *muticast*, como aqueles usados em uma ligação telefônica compartilhada por três pessoas, até para muitas pessoas. Como em eventos *broadcast*;
- b) **independência da mídia:** O RTP fornece serviços à mídia de tempo real genéricas, tais como voz e vídeo. Os cabeçalhos e semânticas de qualquer codificação específica são definidos para cada mídia em especificações separadas;
- c) **mixers e Traslators:** Os *Mixers* são dispositivos que pegam mídias de vários usuários, as misturam e enviam o fluxo resultante. Os *Traslatos* pegam um único fluxo de mídia, converte-o para outro formato envia-o;
- d) **retorno da QoS:** O RTCP permite aos receptores fornecer um retorno a todos os membros de um grupo, enviando informações sobre a qualidade da recepção. As fontes RTP podem usar essa informação para ajustar a sua taxa de comunicação, enquanto que outros receptores podem determinar se os problemas na qualidade do serviço são locais ou gerais. Observadores podem usar essa informação para o gerenciamento da QoS;
- e) **controle de sessão livre:** Com o RTCP, os participantes podem trocar informações de identificação tais como nome, e-mail, número telefônico, e mensagens breves;
- f) **criptografia:** Os fluxos de mídia RTP podem ser criptografados usando-se chaves que são trocados por algum método não RTP.

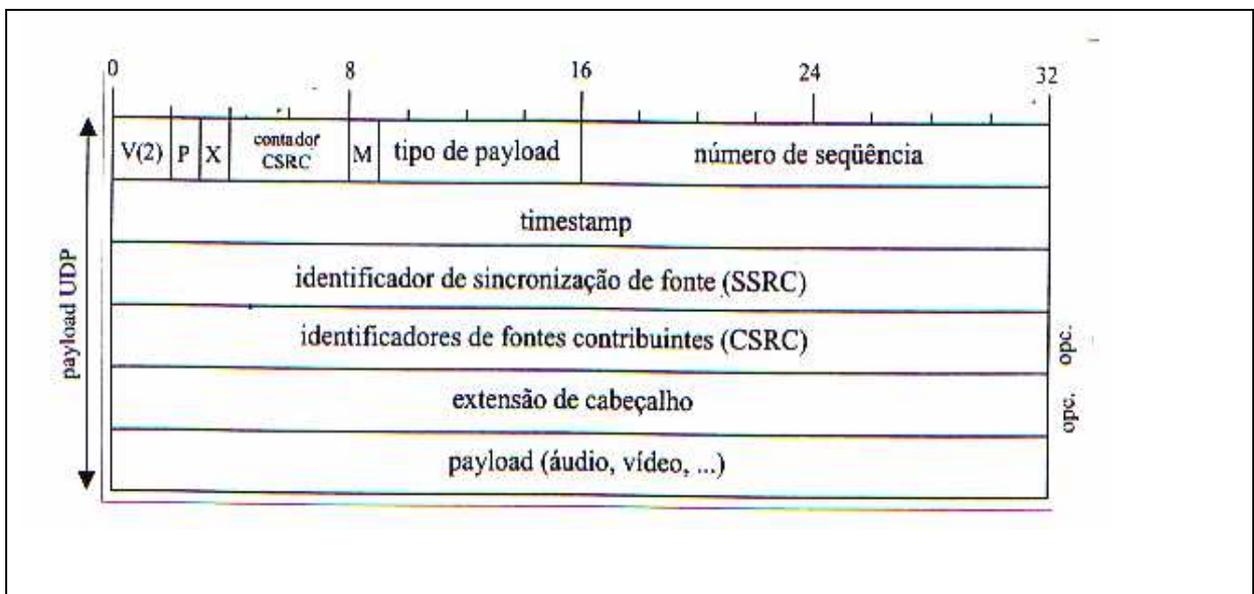
O cabeçalho RTP, como mostra a figura 5, tem um tamanho de 12 bytes. O campo V indica a versão do protocolo. O *flag X* sinaliza a presença de um cabeçalho estendido entre o cabeçalho fixo e o *payload*. Se o bit P está setado, a carga estará preenchida com bits extras (*padding*) para assegurar um alinhamento apropriado para criptografia.

Os usuários participantes de um grupo *multicast* são distinguidos por um Identificador de Sincronização de Fonte (SSRC) que é um número aleatório de 32 bits. Ter um o

identificador de chamada de aplicação permite distinguir facilmente fluxos vindos do mesmo *Mixer* ou *Translator* e associar os relatórios do receptor.

Como descrito anteriormente, um *Mixer* combina fluxo de mídia de varias fontes. Na telefonia atual, os participantes podem ter dificuldades em distinguir quem está falando em um instante de tempo qualquer. A lista SSRC contribuinte Identificador de Fontes Contribuintes (CSRC), cujo comprimento é indicado pelo campo contador CSRC, lista todos os SSRC que contribuem para o conteúdo do pacote. Em uma áudio conferência, ela iria listar todos os locutores ativos.

Figura 5 - Cabeçalho RTP.



O RTP suporta a noção de *framing* dependente de mídia para auxiliar no processo de reconstrução e reprodução. O bit de marcação M fornece a informação para esse objetivo. O tipo de *payload* identifica a codificação utilizada no pacote. O número de seqüência incrementa seqüencialmente de um pacote para o próximo, e é usado para detectar perdas e restaurar a ordem dos pacotes. O *timestamp*, incrementado com a frequência de amostragem da mídia, indica quando o *frame* de mídia foi gerado.

2.2.3.2 RTCP CONTROLE E GERENCIAMENTO

De acordo com Monteiro (2000), o RTCP é o protocolo de controle auxiliar do RTP. Os transmissores e receptores de mídia periodicamente enviam pacotes RTCP para o mesmo grupo *multicast* (mas em diferentes *ports*) usado para distribuir os pacotes RTP. Cada pacote RTCP contém um número de elementos, geralmente um relatório do transmissor (SR) ou um relatório do receptor (RR) seguido de descrição de fonte (SDES). Cada um serve para uma função diferente:

- a) **relatório do transmissor (SR):** São gerados pelo usuário que estão também enviando mídia (fontes RTP). Eles descrevem a quantidade de dados enviados até o momento, bem como correlacionam o *timestamp* amostrado do RTP com o tempo absoluto para permitir a sincronização entre mídias diferentes;
- b) **relatórios do receptor (RR):** são enviados pelos participantes da sessão RTP que estão recebendo mídia. Cada um desses relatórios contém um bloco para cada fonte RTP no grupo. Cada bloco descreve a taxa de perda dessa fonte. O bloco também indica o último *timestamp* e o atraso desde o último relatório do transmissor recebido, permitindo que as fontes estimem suas distâncias aos receptores;
- c) **descrição de fonte (SDES):** são pacotes usados para controle de sessão. Eles contêm *Canonical Name* (CNAME), um identificador único global similar em formato a um endereço de e-mail. O CNAME é usado para associar diferentes fluxos de mídia gerados pelo mesmo usuário. Os pacotes SDES também identificam o participante através de seu nome, e-mail e número telefônico. Isso fornece uma forma simples de controle da sessão. As aplicações clientes podem mostrar as informações de nome e e-mail na interface do usuário. Isso possibilita aos participantes da sessão saber mais sobre os outros participantes. Se um usuário está saindo, ele inclui a mensagem BYE. Finalmente, elementos de aplicação (APP) podem ser usados para adicionar informações específicas da aplicação nos pacotes RTCP.

Desde que o relatório do transmissor, os relatórios do receptor e os pacotes SDES contêm informações que podem mudar continuamente, é necessário enviar esses pacotes periodicamente. Se os participantes de uma sessão RTP simplesmente enviam pacotes RTCP em um período fixo, a largura de faixa usada em um grupo *multicast* poderia crescer

linearmente com o tamanho do grupo, o que é indesejável. Ao invés disso, cada membro da sessão conta o número de outros membros da sessão que ele ouve (através de pacotes RTCP). O período entre pacotes RTCP para cada usuário é então ajustado para ser linearmente escalonado com o número de membros do grupo.

2.2.3.3 FORMATOS DE PAYLOAD

Segundo Domingues (2000), os mecanismos descritos anteriormente permitem ao RTP fornecer os requisitos necessários ao transporte de áudio e vídeo genéricos. Entretanto, codificadores particulares terão requisitos adicionais de informação que precisam ser transmitidos. Para suportar isso, o RTP permite que formatos de *payload* sejam definidos para codificadores particulares. Esses formatos de *payload* descrevem a sintaxe e a semântica do *payload* RTP. A semântica particular de um *payload* é comunicada pelos bits indicadores de tipo de *payload* RTP. Esses bits são mapeados para os atuais codificadores e formatos via uma ligação de nomes, registrados pela *Internet Assigned Numbers Authority* (IANA). Isso possibilita que uma fonte RTP mude o formato de codificação no meio do fluxo sem sinalização explícita. Mais ainda, qualquer um pode registrar um nome, contanto que ele não esteja sendo usado, através de procedimentos específicos. Isso possibilita que o RTP seja utilizado com qualquer tipo de codificador desenvolvido por qualquer pessoa.

Formatos de *payload* para várias mídias foram definidos, tais como os codificadores de vídeo H.323, H.261, JPEG e MPEG, e vários codificadores de áudio e vídeo são suportados com formatos de *payload* mais simples.

Segundo Leopoldina (2001), mais ainda, formatos de *payload* RTP estão sendo definidos para fornecer alguns serviços genéricos. Um formato para codificação de áudio redundante permite ao usuário transmitir o conteúdo de áudio usando múltiplas codificações, cada uma atrasada da anterior, a uma taxa de comunicação menor. Isso possibilita que pacotes sejam recuperados a partir dos pacotes subseqüentes, embora com uma qualidade de codificação mais baixa. Um formato de *payload* está sendo definido com mecanismo de FEC, permitindo a recuperação de pacotes perdidos de uma maneira independente da codificação. Um outro formato está sendo introduzido para multiplexar mídias de múltiplos usuários em um único pacote. Isto é particularmente útil para a substituição de troncos entre *gateways* na

telefonia na Internet, o que pode fornecer uma redução no *overhead* dos cabeçalhos dos pacotes.

2.3 MULTICAST NA INTERNET

De acordo com Tanenbaun (1997), normalmente, as comunicações IP são feitas entre um transmissor e um receptor. Entretanto, para algumas aplicações, é interessante que um processo seja capaz de transmitir dados para um grande número de receptores simultaneamente. Dentro os exemplos dessa estratégia estão à atualidade ações para vários corretores e o controle de chamadas de teleconferência digital.

O IP aceita *multicast*, usando endereços classe D que identificam um grupo de *hosts*. Estão disponíveis 28 bits para identificar grupos; portanto, pode haver mais de 250 milhões de grupos ao mesmo tempo. Quando um processo envia um pacote para um endereço classe D, é feita uma tentativa de entrega-lo a todos os membros do grupo endereçado, mas não há qualquer garantia de que isso realmente acontecerá. É provável que alguns membros não obtenham o pacote.

São aceitos dois tipos de endereços de grupo: endereços permanentes e temporários. Um grupo permanente está sempre presente e não precisa ser estabelecido. Cada grupo permanente tem um endereço de grupo permanente. Alguns exemplos de endereços de grupo permanente são mostrados na figura 6.

Figura 6 – Tipos de endereços Multicast

224.0.0.1	Todos os sistemas de uma LAN
224.0.0.2	Todos os roteadores de uma LAN
224.0.0.5	Todos os roteadores OSPF de uma LAN
224.0.0.6	Todos os roteadores OSPF designados em uma LAN

Para usar um grupo temporário, primeiro é necessário criá-lo. Um processo pode solicitar que seu *host* se conecte a um grupo ou se desconectar dele. Quando o último processo de um *host* deixa um grupo, o grupo passa a não mais existir no *host*. Cada *host* controla os grupos que pertencem seus processos atuais.

O *multicast* é implementado por roteadores *multicast* especiais, que podem ou não ser colocados como roteadores-padrão. Uma vez por minuto aproximadamente, cada roteador *multicast* envia um processo de hardware *multicast* (ou seja, chamada de enlace de dados) para *host* da LAN (endereço 224.0.0.1) solicitado que sejam informado aos quais seus processos pertencem. Cada *host* envia respostas para todos os endereços classe D nos quais está interessado.

Esses pacotes de consulta e respostas utilizam um protocolo chamado *Internet Group Management Protocol (IGMP)*. Ele possui apenas dois tipos de pacotes: consulta e resposta, cada um com um formato fixo simples, contendo algumas informações de controle na primeira palavra do campo de carga útil e um endereço classe D na segunda palavra. O IGMP é descrito na RFC 1112.

O roteamento *multicast* é feito com base no método *spanning tree*. Cada roteador *multicast* troca informações com seus vizinhos usando um protocolo de vetor de distância modificado. Dessa forma, cada um dos vizinhos é capaz de construir uma *spanning tree* que abrange todos os membros de um grupo. Várias otimizações são usadas para podar a árvore, a fim de eliminar roteadores e redes que não interessem a determinados grupos.

3 TECNOLOGIAS UTILIZADAS

Para a realização deste trabalho foram utilizadas algumas ferramentas e tecnologias, a fim de especificar a análise e desenvolver o protótipo. Neste capítulo são abordadas as análises estruturadas, a ferramenta de fluxogramas livres e o ambiente de programação C++.

3.1 ANÁLISE ESTRUTURADA

Segundo Fischer (1990), o objetivo da análise estruturada é produzir uma especificação estruturada, que é um tipo de documento dos requisitos funcionais. As especificações estruturadas diferem dos documentos tradicionais de requisitos funcionais porque estes se baseiam em texto, ao passo que as especificações estruturadas baseiam-se em gráficos. A análise estruturada não impede que os requisitos mudem, mas podem acomodar estas mudanças e minimizar seu impacto, ajudando a organizar os módulos do software de maneira funcionalmente destacada e sustentável.

A análise estruturada fornece uma abordagem sistemática, etapa por etapa, para desenvolver a análise e produzir uma especificação de sistema novo e melhorado. Para alcançar este objetivo ela centraliza-se em uma comunicação clara e concisa. Por isto usa-se o método de decomposição funcional, *top-down*, para definir os requisitos do sistema. A especificação do sistema produzido pela análise estruturada é um modelo decomposto e *top-down* do sistema a ser constituído. Devido à especificação ser um modelo gráfico conciso e fácil compreensão, o usuário pode familiarizar-se bem com o sistema antes de sua implementação. Isto possibilita a identificações de erros e interpretações falsas, durante o processo de desenvolvimento. Como a especificação é dividida em partes menores, as mudanças nos requisitos que ocorrem durante o ciclo de vida são mais facilmente alternadas (Martin, 1991).

Segundo Gane (1984), análise estruturada é a construção de um modelo lógico de um sistema, utilizando de técnicas gráficas capazes de levar usuários, analistas e projetistas a formarem um quadro claro e geral do sistema e de como suas partes se encaixam para atender às necessidades daqueles que dele precisam.

É composta de um conjunto de técnicas e ferramentas em constante evolução, nascido do sucesso da programação e do projeto estruturados (Gane, 1984).

Segundo Demarco (1989), projeto estruturado é uma estratégia para a produção de projeto *top-down*, com alta capacidade de manutenção facilmente testável. O projeto estruturado começa com as características hierárquicas, fazendo com que você trate as questões em ordem de importância, ao invés de tratá-las na ordem em que o computador irá encará-las.

A programação estruturada é um conjunto de convenções que o programador pode seguir para produzir o código estruturado, tendo regras de codificações impondo limites sobre o uso das estruturas básicas de controle ajudando a organização do código fonte do sistema (Martin, 1991) (Yourdon, 1993).

3.1.1 MODELO ENTIDADE-RELACIONAMENTO (MER)

A sua criação deve-se a Peter Chen, baseada numa notação desenvolvida por James Martin denominada de Modelo de Informação (João, 1993).

O modelo conceitual de dados é representado por um gráfico denominado de diagrama de entidade e relacionamento (DER), que detalha as associações existentes entre as entidades de dados e utilizam componentes semânticos próprios (João, 1993).

Esse gráfico facilita a visualização da organização e o inter-relacionamento entre os objetos que são conectados uns aos outros através de relacionamentos representando um conjunto de conexões entre objetos representados por losango.

Segundo Yourdon (1993), o diagrama de entidade e relacionamento é uma ferramenta de modelagem que é utilizada para modelar regras particulares de importância para a empresa e o relacionamento entre elas, sendo utilizado para identificar e organizar a informação utilizada pela empresa, não apenas a informação armazenada.

Enfatiza os principais objetos ou entidades com que o sistema lida, bem como a relação entre os objetos que correspondem aos locais de armazenagem de dados do DFD (João 1993).

As entidades são os objetos definidos, podendo ser objetos específicos como pessoas ou uma nota fiscal, ou conceitos abstratos, com posições e serviços, sendo representadas por um retângulo no DER, tendo um único nome que deverá refletir o tipo do objeto (Demarco 1989), (Yourdon 1993).

Outro componente do DER são os relacionamentos, que segundo João (1993), são associações ou ligações que mostram como uma entidade ou grupo de entidades se relaciona com a outra entidade ou grupo.

Cada ocorrência do relacionamento corresponde a uma associação de exatamente uma ocorrência de cada entidade que participa do relacionamento (Demarco, 1989).

Segundo Yourdon (1993), há notações que se referem à cardinalidade dos relacionamentos entre as entidades, sendo elas:

- a) relacionamento 1:1 - relacionamento que denota a ligação de um registro com somente um da outra entidade;
- b) relacionamento 1:N - relacionamento que denota a ligação de um registro com múltiplas ocorrências da outra tabela;
- c) relacionamento N:N - relacionamento que denota a ligação de vários registros com múltiplas ocorrências da outra entidade.

3.1.2 DIAGRAMA DE FLUXO DE DADOS (DFD)

Segundo Yourdon (1993), o diagrama de fluxo de dados (DFD), identifica as funções do sistema e como elas utilizam as informações e as transferem entre as funções existentes.

É uma ferramenta *top-down* e se estende sucessivamente para os níveis de maior detalhamento, são multidimensionais e utilizam-se de recursos gráficos, permitindo o compartilhamento do modelo entre a comunidade possuindo o foco centrado na questão e no seu refinamento (João, 1993).

Segundo Demarco (1989), o DFD é composto por quatro elementos básicos:

- a) fluxo de dados, representados por flechas;
- b) processos, representados por círculos;
- c) arquivos, representados por linhas retas;
- d) fontes ou destinos de dados, representados por caixa.

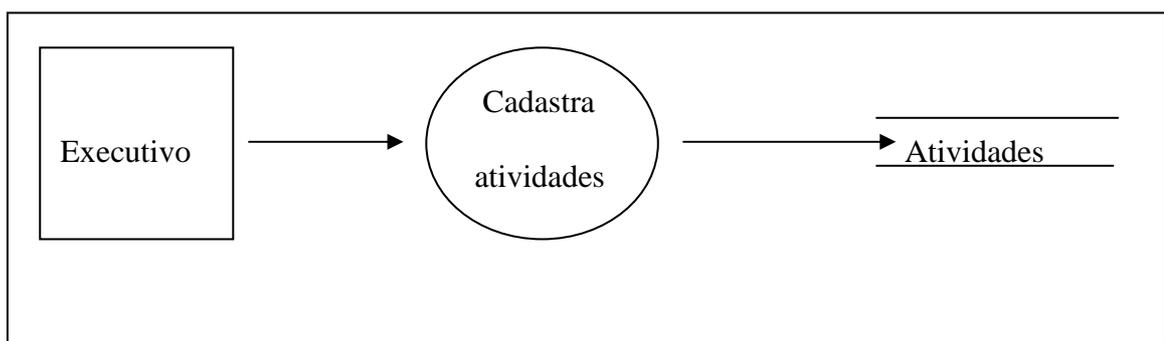
Segundo Gane (1984), o fluxo de dados identifica os processos, entidades, ou depósitos de dados nas suas extremidades anotando uma descrição do conteúdo de cada fluxo ao longo de sua extensão, sendo a mais significativa possível.

O processo é um componente procedural do sistema, operando sobre os dados, podendo executar operações aritméticas ou lógicas com os dados para produzir algum resultado, devendo ser usado um nome significativo para definir a operação executada pelo processo (Martin 1991).

O arquivo, ou depósito de dados, são repositório de dados produzido por um processo, preservados por um determinado período, e então utilizados por um outro processo (João 1993).

A fonte ou destinos de dados é uma pessoa ou empresa, repousando fora do contexto do sistema, que é o originador ou o receptor de dados do sistema (Demarco 1989).

Figura 8 – Exemplo de um fluxo de dados



3.2 BORLAND C++

Segundo Norton (1992), o Borland C++ é uma linguagem visual de programação. Ele é um descendente do C, mais possui uma interface visual prática. O C++ pode usar ou criar novas bibliotecas de funções, chamadas de DLLs, e seus programas podem responder a elas e iniciar qualquer evento do windows.

O C++ trabalha basicamente com o conceito de projeto que seria um conjunto de programas. O C++ segue o paradigma da programação orientada a objetos. Neste tipo de programação, variáveis e funções são agrupadas em conjuntos que são chamados de classes. Para se usar uma função ou variável de uma determinada classe é preciso colocar antes de seu nome um prefixo conhecido como objeto (seria o nome do objeto) daquela classe.

Segundo Dorfman (1996), uma possibilidade de se trabalhar com interface gráfica, é colocar na janela uma série de elementos como botões simples, botões de rádio, onde você pode entrar com dados etc. São chamados de *controls*, e são tirados da caixa de ferramentas. Cada *control* possui uma série de características, chamadas de propriedades, que tem valores *defaults* podem ser alterados. Uma propriedade é uma variável presa a um *control*.

4 DESENVOLVIMENTO DO PROTÓTIPO

Para o desenvolvimento do protótipo utilizou-se a linguagem de programação C++ 5.0 da Borland para Windows, baseado nas bibliotecas descritas por LIESENBORG(2001), escritos em C++.

Os softwares modernos procuram ter uma interface mais simples e amigável com os usuários proporcionando um melhor entendimento e facilidade de seu uso. Essas interfaces permitem o desenvolvedor criar visualmente a interface como o usuário utilizando-se apenas do mouse.

4.1 ESPECIFICAÇÃO DO PROTÓTIPO

Para a especificação do protótipo deste Trabalho de Conclusão de Curso foram utilizadas linguagens gráficas livres (fluxogramas), que representam o funcionamento dos processos e funções deste protótipo de software Voz sobre IP (VoIP).

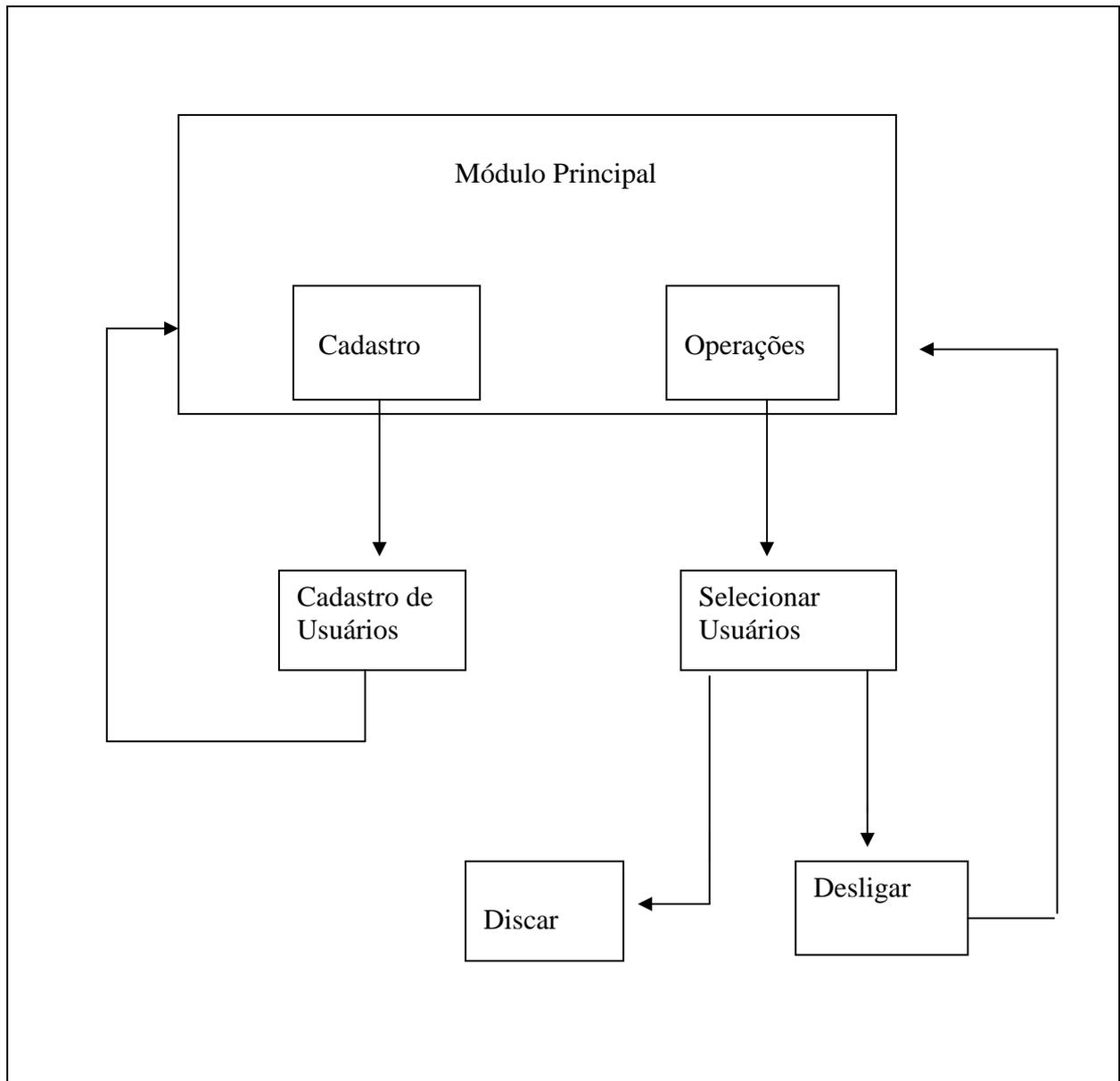
As linguagens gráficas expressam seus processos e fluxos através de um conjunto de cadeias de símbolos alfanuméricos ou especiais. As informações e ações a serem tomadas podem ser vistas com maior rapidez e clareza através da sua representação precisa e detalhada, bem como, o entendimento dos processos podem ser visualizados passo a passo.

As linguagens gráficas livres estão baseadas em princípios, não tem uma forma especificamente definida e sua simbologia gráfica é arbitrária.

O protótipo VoIP foi desenvolvido com o objetivo de reduzir custos operacionais da corporações com ligações entre seus escritórios.

Para um melhor entendimento dos processos que ocorrerão no protótipo seguem as suas descrições.

Na figura 9 é apresentado um diagrama macro das principais funcionalidades do protótipo Voz sobre IP (VoIP).

Figura 9 – Diagrama macro do protótipo

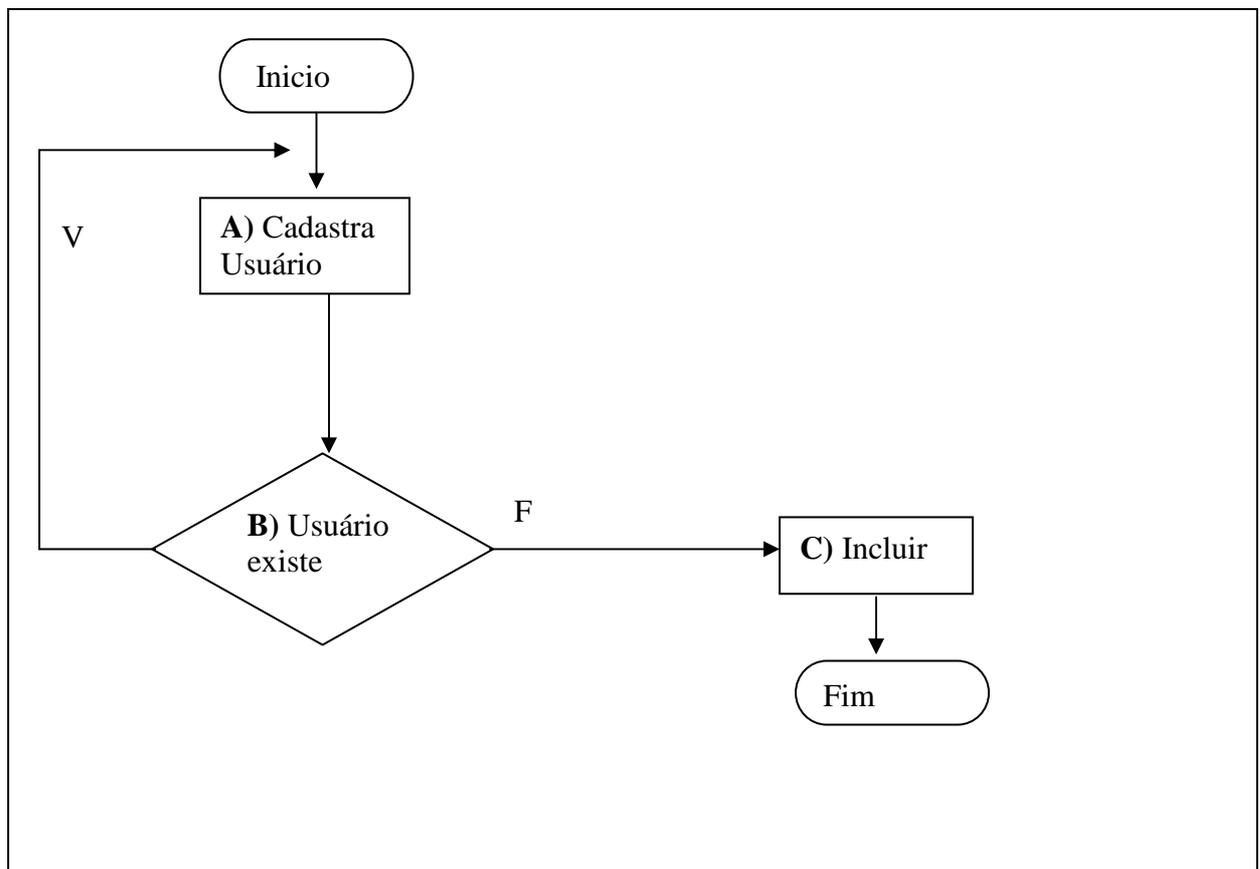
No módulo principal há duas opções que são: **Cadastro e Operações**

Na opção de **Cadastro** o gerente tem a opção de efetuar o cadastramento de usuário sendo que este cadastro será utilizado para efetuar a comunicação entre dois usuários. Neste cadastro de usuários têm-se como informações principais os campos **Ramal** e o campo chave da tabela que identifica o ramal a ser discado; **Nome** que identificará o nome do usuário; **IP**, onde será cadastrado o endereço IP da máquina onde o usuário receberá as ligações VoIP.

Na opção de **Operações** o gerente selecionará os usuários cadastrados anteriormente para que possa ser criada uma sessão VoIP. A seguir o gerente deverá pressionar o botão **Discar** para estabelecer a comunicação e quando deseja encerra a ligação deverá ser pressionado o botão **Desligar** destruindo a sessão criada anteriormente.

A figura 10 ilustra o fluxograma do cadastro de usuários na tabela.

Figura 10 – Cadastro de usuário



A seguir será explicado cada processo acima mostrado na figura 10:

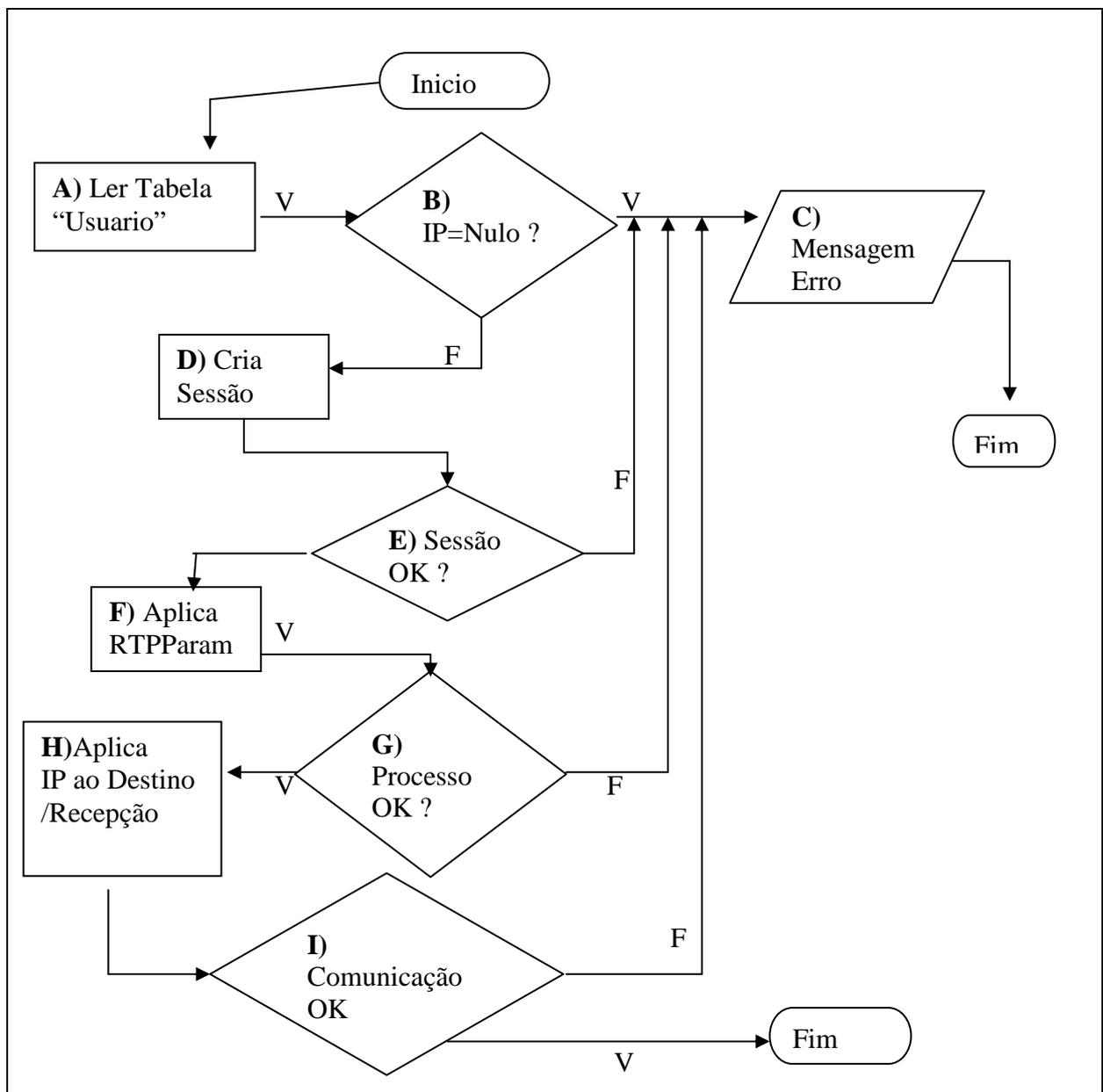
Processo A: neste processo será efetuado o cadastramento dos dados pertinentes aos usuários;

Processo B: neste processo será feita a verificação se o usuário já foi cadastrado. Caso o usuário não tenha sido cadastrado passa-se para o **Processo C** senão e retornado ao **Processo A** entrando com os novos dados.

Processo C: neste processo são armazenados os dados na tabela de **Usuários**.

A figura 11 contém um fluxograma do protótipo para efetuar uma chamada através de uma sessão *VoIP*

Figura 11 - Fluxograma do Processo para Cria uma Sessão VoIP



A seguir serão descritos os processo do fluxo de uma Sessão VoIP conforme mostrado na figura 11:

Processo A: neste processo e resgatado o IP do usuário.

Porcesso B: é verificado se o IP é nulo caso seja é apresentado uma mensagem de erro conforme o **Processo C** e o programa é abortado. Caso o IP seja válido passa-se ao **Processo D**.

Processo D: ocorre o processo de criação de uma sessão VoIP.

Processo E: verifica-se a sessão criada no **Processo D** é válida. Caso seja válida passa-se ao **Processo F** caso contrário é apresentado uma mensagem de erro conforme o **Processo C** e o programa e abortado.

Processo F: e executada a função RTPParams para que seja efetuado a comunicação em tempo real.

Processo G: caso a função RTPParams retorne erro, é apresentado uma mensagem de erro conforme o **Processo C** e o programa e abortado, caso contrário passa-se para o **Processo H**.

Processo H: neste processo é aplicado o IP de destino para que a comunicação possa ser estabelecida passando também o endereço IP de quem efetuou a chamada.

Processo I: caso aconteça um erro de comunicação entre o IP destino e o IP de recepção é apresentado uma mensagem de erro conforme o **Processo C** e o programa e abortado. Caso contrário a comunicação é estabelecida.

A figura 12 apresenta o código fonte do processo de criação de uma sessão VoIP.

Figura 12 - Fonte do Botão Discar

```
void __fastcall TFrmOperacoes::btnDiscarClick(TObject *Sender)
{
    unsigned long ip;
    AnsiString tempStr;

    if (session)
        BtnDesligarClick(Sender);
    session = new JVOIPSession;
    checkError(session->Create(params));
```

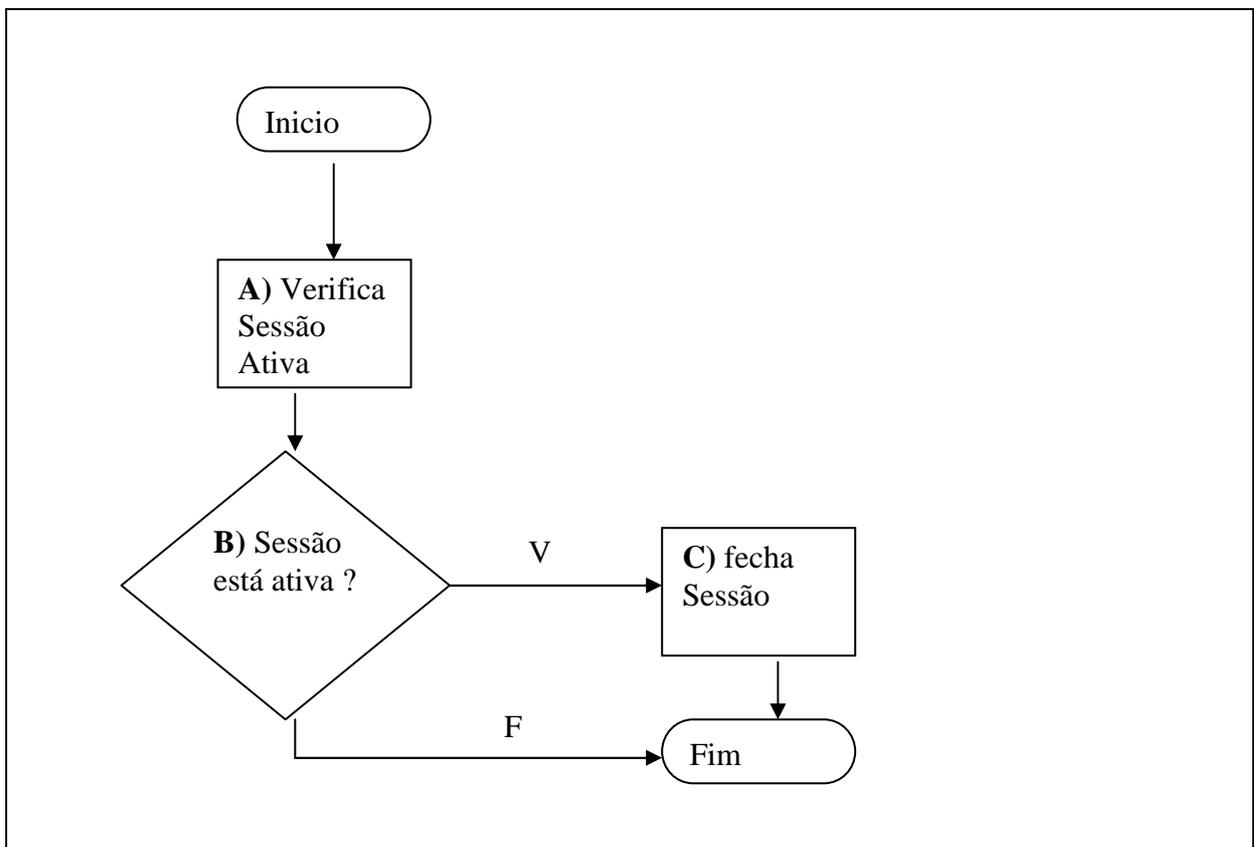
```

tempStr = (AnsiString) DBLkpListNome->KeyValue;
ip = inet_addr(tempStr.c_str());
ip = ntohl(ip);
checkError(session->AddDestination(ip,5000));
checkError(session->AddToAcceptList(ip,5000));
}

```

A seguir na figura 13 será mostrado o fluxo do botão desligar para que uma sessão VoIP possa ser finalizada.

Figura 13 - Fluxograma do Botão Desligar



A seguir será descrito os processo do fluxo do término de uma Sessão VoIP conforme mostrado na figura 14

Processo A: verifica se há comunicação entre o IP origem e o IP destino

Processo B: verifica se há sessão VoIP está ativa e em funcionamento: caso esteja é passado para o **Processo C**.

Processo C: fecha a sessão e libera a comunicação entre o IP origem e o IP destino.

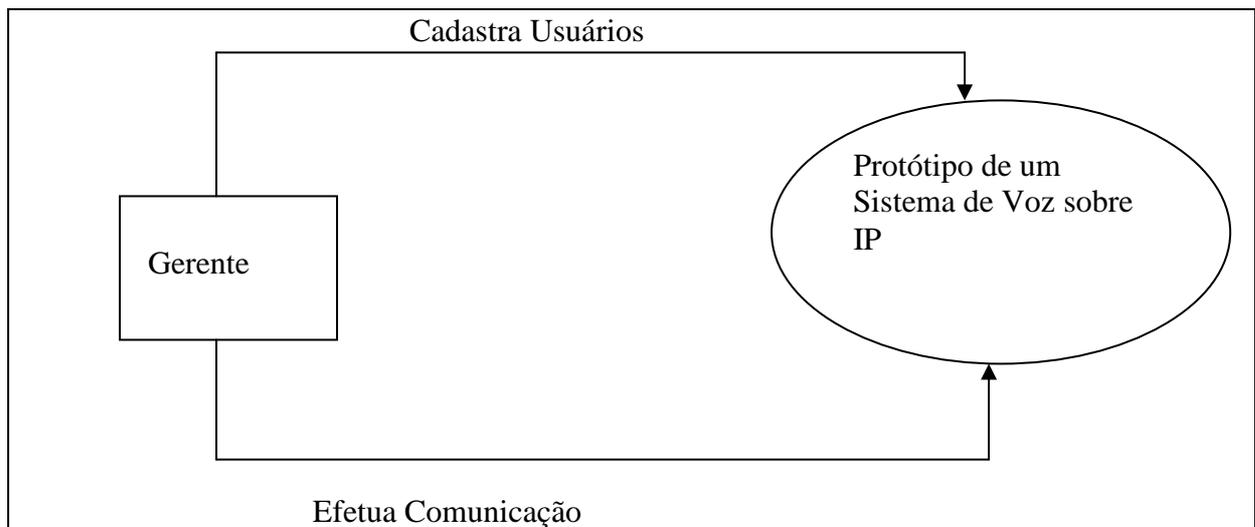
A código fonte do botão desligar é mostrado conforme a figura 14.

Figura 14 - Fonte do Botão Desligar

```
void __fastcall TFrmOperacoes::BtnDesligarClick(TObject *Sender)
{
    session->ClearDestinations();
    session->ClearAcceptList();
    checkError(session->Destroy());
    delete (session);
    session = NULL;
}
```

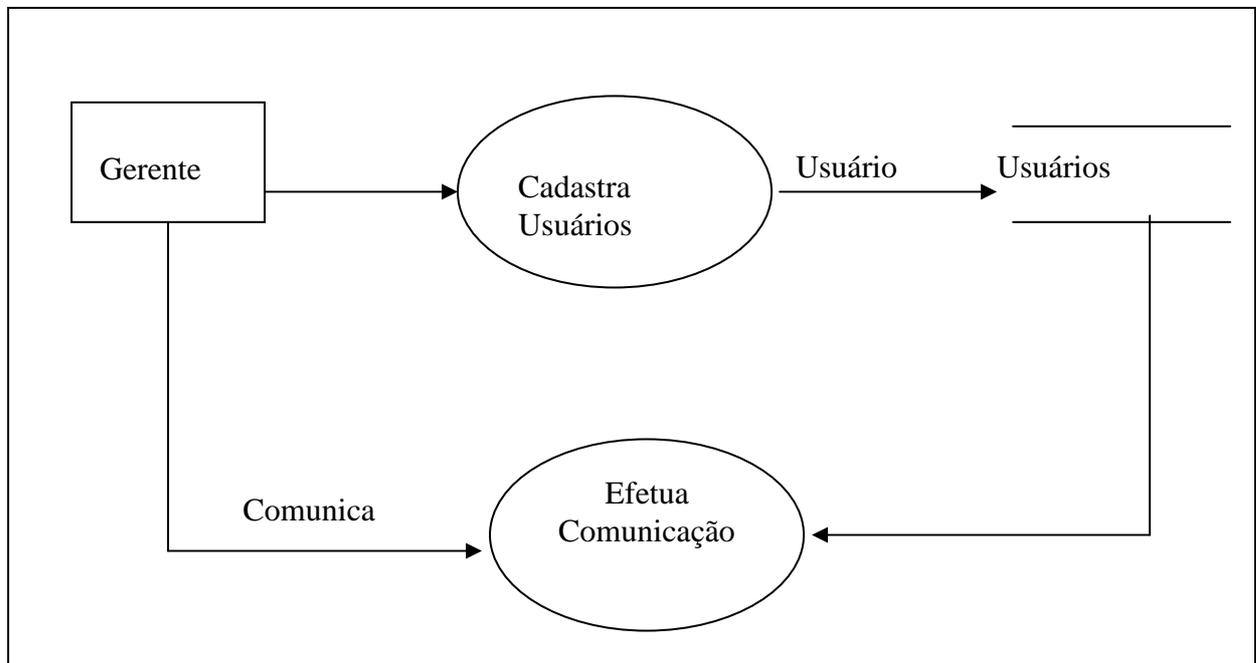
Na figura 15 é mostrado um digrama de contexto do protótipo de um sistema VoIP.

Figura 15 - Diagrama de contexto do protótipo



No protótipo de um sistema de Voz sobre IP, o gerente cadastra os usuários através do fluxo de dados “Cadastra Usuários” e o mesmo acontece com o fluxo de dados “Efetua Comunicação”.

Na figura 16 é mostrado um diagrama do fluxo de dados do protótipo VoIP.

Figura 16 - Diagrama de fluxo de dados do protótipo

No diagrama de fluxo de dados, o gerente cadastra usuários através do processo “Cadastrar Usuários” e esses dados são armazenados em uma tabela denominada de “Usuários”.

Ao efetuar a comunicação, é necessário saber os dados do usuário. A obtenção desses dados é representada pelo fluxo de dados obtido na da tabela “Usuários” que alimenta o processo de “Efetuar Comunicação”

A seguir, no dicionário de dados, está descrita a tabela envolvida no desenvolvimento do protótipo bem como seus respectivos campos.

Figura 17 - Dicionário de dados

Tabela : Usuario	
Nome	Tipo
Ramal	NUMBER
Nome_Usuario	Alfanumerico (30)
IP	Alfanumerico (15)

Como está apresentado na figura 17, o protótipo de um sistema de Voz sobre IP.

4.2 IMPLEMENTAÇÃO

A seguir será descrito a utilização das bibliotecas JVOIPLIB, JThread, JMutex, bem como a implementação do protótipo.

4.2.1 UTILIZANDO A BIBLIOTECA

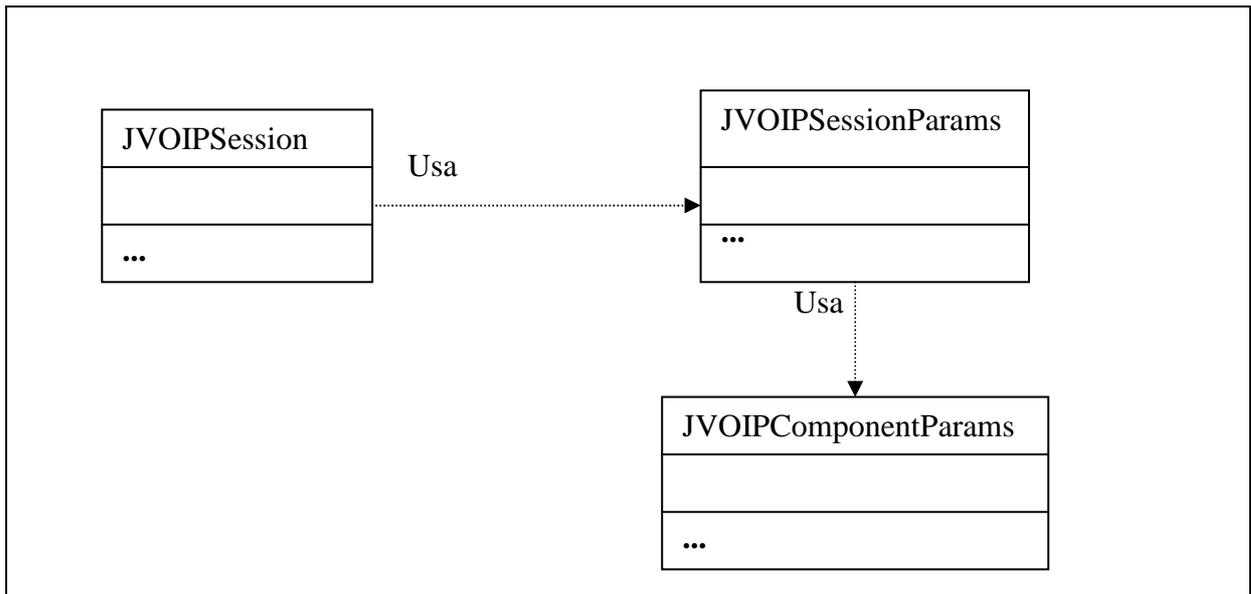
Em primeiro lugar, o nome 'JVOIPLIB' representa *Jori's VoIP Library*. Esta biblioteca faz a criação de voz sobre uma rede IP (VoIP).

Segundo Liesenborgs (2001), os componentes necessários para uma sessão de VoIP. Os componentes têm um ou mais implementações. Em módulos de emulações e estas liberações, estão incluídas nos módulos abaixo:

- a) contribuição da placa de som;
- b) compressão de DPCM;
- c) RTP transmissão de dados;
- d) um misturador de voz elementar;
- e) um *script* de localização.

Basicamente, a biblioteca JVOIPLIB tem os componentes necessários para a utilização de VoIP, sendo assim, acrescenta-se os componentes necessários para uma sessão JVOIP como será mostrado na figura 18.

Figura 18 - Representa a criação de uma sessão



A classe `JVOIPSession` é o conceito central da biblioteca JVOIP, cujo objetivo é acrescentar a VoIP a uma sessão, com os parâmetros desejados, como a taxa de transmissão, ou atribuir a sessão a um grupo *multicast*. Na classe `JVOIPSession` podem-se especificar vários parâmetros, como mostrado nos exemplos a seguir:

- a) **LeaveAllMulticastGrup(ip:unit32) :int** – caso este parâmetro seja setado a classe `JVOIPSession` irá enviar a sua mensagem apenas para um grupo de *Multicast* em específico passando como parâmetro o IP do grupo de destino;
- b) **DeleteFromIgnoreList (ip:unit32, port:unit16) : int** – este parâmetro tem por finalidade de remover algum IP desejado de uma lista de chamadas recebendo com parâmetro o IP e a porta em que o usuário deseja recusar a chamada;

Na `JVOIPComponent` assim como na `JVOIPSession` também têm-se vários parâmetros que podemos setar como será mostrado alguns exemplos abaixo:

- a) **GetComponentName () : String** – este parâmetro requer o nome do grupo de *multicast* em que o usuário deseja se conectar;

- b) **GetComponentDescription () : String** - este parâmetro caso seja setado irá trazer uma descrição para que o usuário que receber a chamada possa saber quem esta chamando como por exemplo.

Na classe JVOIPCompParams que uma classe derivada da classe JVOIPComponent que entre tantos podem ser setados alguns com veremos a seguir:

- a) **GetParameterName (): String** – este parâmetro levava os parâmetros setados anteriormente na classe JVOIPComponent através do parâmetro **GetComponentName**.
- b) **GetParameterValue (): String** - este parâmetro e levado para VoIP o valor setado na classe JVOIPComponent como o parâmetro **GetComponentDescription**

4.2.2 CRIANDO UMA SESSÃO

Uma sessão de VoIP é representada através da classe JVOIPSession, que cria uma sessão chamando a função *create* que leva um parâmetro do tipo JVOIPSessionParams (como sugere o nome esta classe contém os parâmetros de criação de uma sessão).

O JVOIPSessionParams classifica e leva parâmetros para o intervalo de amostra, como porta, taxa de transmissão, tipo de compressão entre outros. Estes parâmetros são armazenados na classe derivada que é a JVOIPComponetParams.

Por exemplo, se especificar na JVOIPSessionParams deverá usar o módulo de transmissão RTP em uma outra porta que não é a porta base que é definida por *default* como (5000), o que pode ser feito através da JVOIPSessionParams associando o paramento JVOIPRTPTransmissionParams para a porta que se deseja .

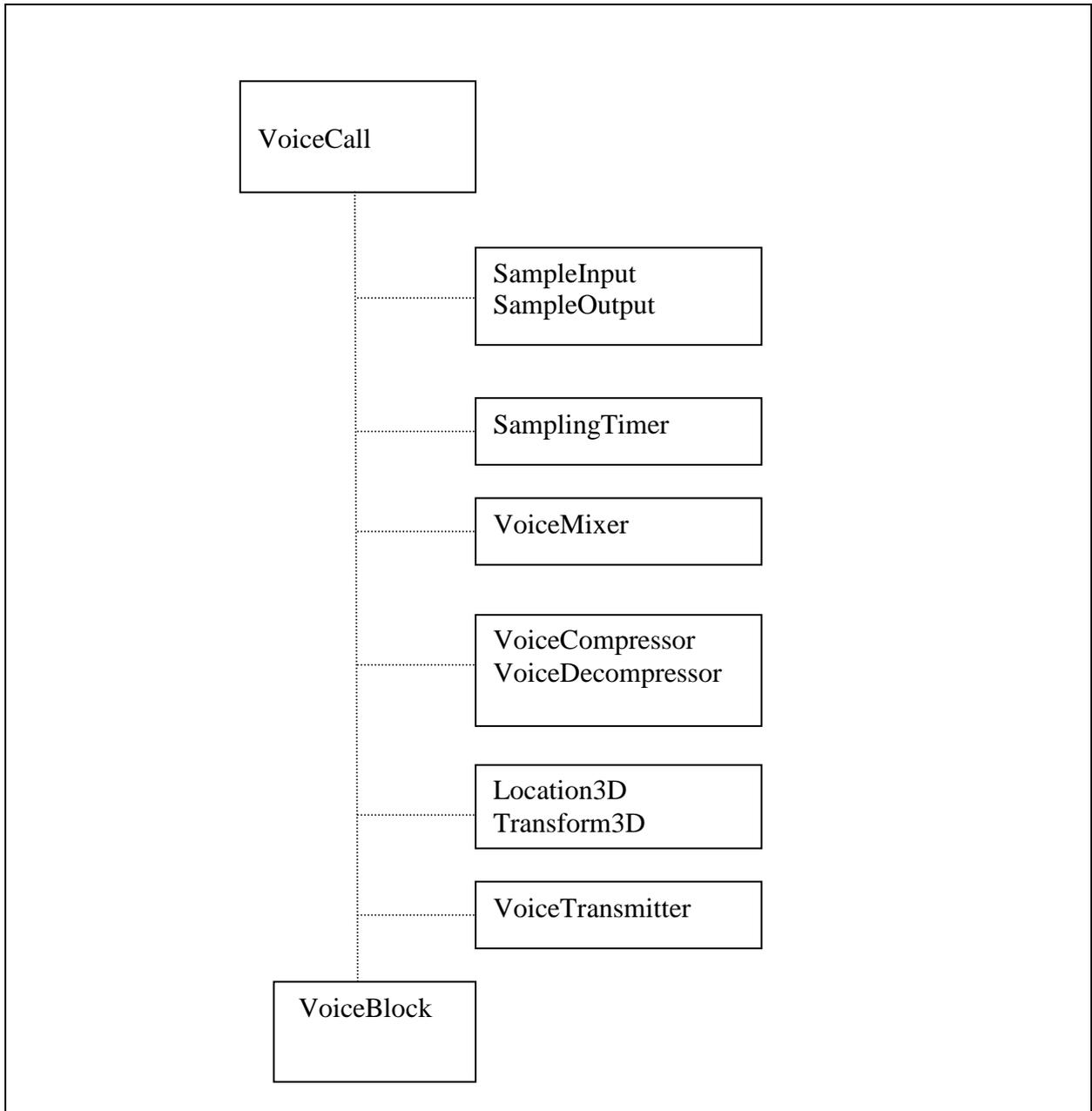
4.2.3 DEFININDO UMA SESSÃO ATIVA

Uma vez que a sessão está ativa, pode-se adicionar vários atributos, tais como taxa de transmissão, tempo de transmissão, etc. As funções para fazer isto são semelhantes às funções usadas pela JVOIPSessionParams.

Se há uma sessão ativa e em funcionamento, ainda têm-se de especificar para onde os dados de VoIP devem ser enviados. Na classe `JVOIPSession` deverão ser especificados os destinatários de *multicast*, capazes de receber os dados que são enviados a um *multicast* em específico, para unir-se e juntar-se a grupos de multicast deve-se fazê-lo usando as funções `JoinMulticastGroup` e `LeaveMulticastGroup` podendo especificar também quais dados estão disponíveis.

Pode-se aceitar todos os dados de entrada ou podem ser especificada a combinação de IP - portas que devem ser ignoradas ou as que deveriam ser aceitas. No caso de IP-porta aceita, todos os pacotes que vêm de outras portas são descartadas.

A figura 19 ilustra os componentes necessários para tornar uma sessão ativa e em funcionamento utilizando bibliotecas `JVOIPLib` e `JThread` e `JRTPLIB` conforme definido por (Liesenborgs, 2001).

Figura 19 - Componentes utilizados na Biblioteca JVOIP

A classe **VoiceCall** é a classe que prepara juntamente com as outras classes ilustradas na figura 19:

SampleInput / SampleOutput: esta classe tem por responsabilidade pegar e reconstituir os sinais da voz;

SamplingTimer: esta classe tem por responsabilidade indicar o intervalo de tempo entre o IP origem e o IP destino;

VoiceMixer: esta classe é responsável por juntar os sinais enviados pela conexão VoIP criada;

VoiceCompressor / VoiceDecompressor: esta classe é a responsável para comprimir e descomprimir a voz tornando assim a voz em um sinal;

Location3D / Transform 3D: esta classe tem a função de adicionar o IP de destino e o IP de origem ao bloco gerado pela VoiceCall;

VoiceTransmitter: esta classe prepara o bloco gerado pelas classes anteriores para ser transmitido dentro de uma sessão VoIP;

VoiceBlock: esta classe contém a voz empacotada e juntamente com a classe VoiceTransmitter são as responsáveis pela transmissão de VoIP entre o IP origem e o IP destino.

4.2.4 USO DA BIBLIOTECA JTHREAD E JMUTEX

Segundo Liesenborgs (2001), o pacote de JThread atualmente, só contém duas classes, que são a JThread e JMutex.

JMutex

A definição da classe de JMutex é mostrada na figura 20. Para usar um exemplo deste tipo, deve-se chamar a função de *Init* primeiro. Para conferir se o *mutex* já foi inicializado, deve-se conferir o valor de retorno de *IsInitialized*, conforme a figura 20

Figura 20 - Inicialização da JMutex

```

Class Jmutex
{
public:
    Jmutex ();
    ~Jmutex();
    int Init();
    int Lock();
    int Unlock();
    bool IsInitialized();
};

```

JThread

A Classe JThread é utilizada para criar um *Thread* adiciona-se os membros em sua classe, conforme e descrito na figura 21 .

Figura 21 - Inicialização da JThread

```

Class Jthread
{
    public
    JThread ();
    Virtual ~Jthread ();
    int Start ();
    int Kill ();
    virtual void *Thread () =0;
    bool IsRunnig ();
    void *GetReturnValue();
}

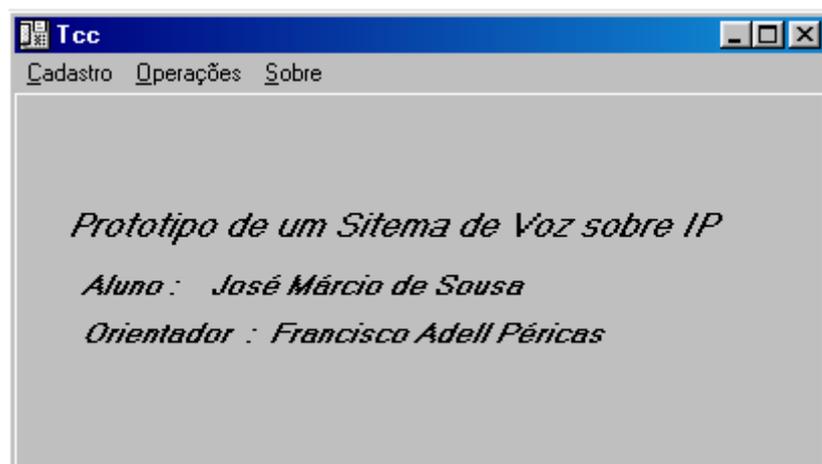
```

A principal característica da biblioteca é oferecer suporte ao *Real Time Transport Protocol* (RTP).

4.3 UTILIZAÇÃO DO PROTÓTIPO DO SISTEMA

Neste item serão mostradas as telas do protótipo acompanhadas de uma breve explicação sobre cada tela.

Figura 22 - Tela inicial do protótipo

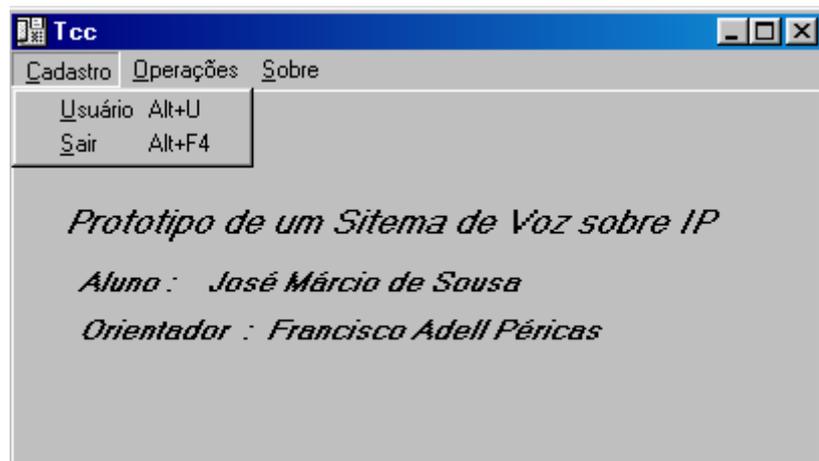


A partir desta tela (figura 22) o usuário poderá selecionar uma das opções abaixo:

- **Cadastro**
- **Operações**
- **Sobre**

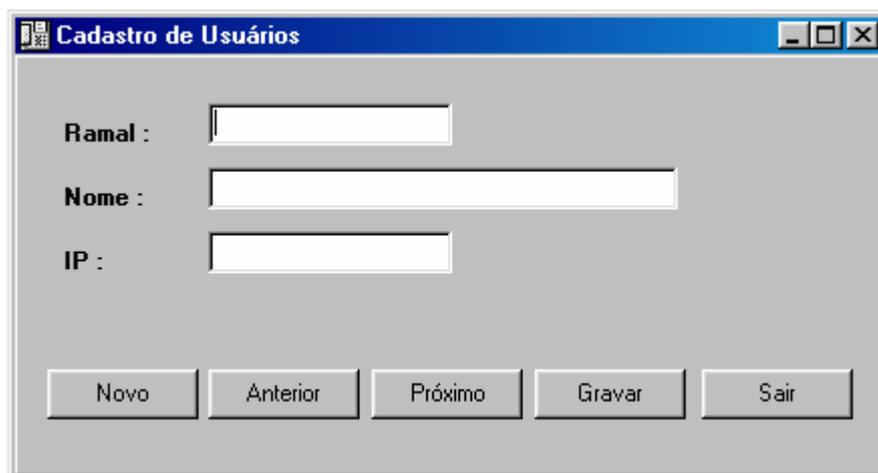
Selecionado a opção de **Cadastro** aparecerá uma tela solicitando que seja selecionada uma das opções abaixo conforme e mostrado na figura 23.

Figura 23 - Cadastro



Após ter selecionado a opção **Cadastro -> Usuário** aparecerá uma tela de cadastro de usuário conforme mostrado na figura 24.

Figura 24 - Cadastro de Usuários

A imagem mostra uma janela de formulário com o título "Cadastro de Usuários". Ela contém três campos de entrada: "Ramal", "Nome" e "IP". Abaixo dos campos, há cinco botões: "Novo", "Anterior", "Próximo", "Gravar" e "Sair".

Todos os campos da figura 24 devem ser preenchidos pois o protótipo se utiliza desta tabela para efetuar a chamada a usuários cadastrado neste módulo.

A seguir será descrito cada campo da tela juntamente com suas características:

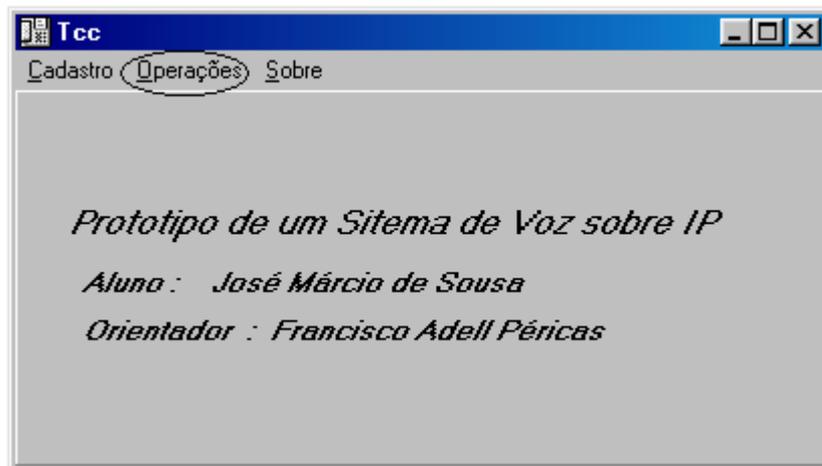
Ramal -> Campo numérico onde é informado o ramal onde os usuários estão localizados. Este é o campo chave da tabela.

Nome -> Campo onde é cadastrado o nome do usuário.

IP -> Campo onde é cadastrado o endereço IP onde o usuário passará a utilizar o sistema.

Na Figura 25 será selecionado o módulo de operações conforme é mostrado na tela abaixo.

Figura 25 – Selecionando operações



Quando for selecionado o módulo operações, aparecerá uma tela com as funções do módulo, conforme a na figura 26.

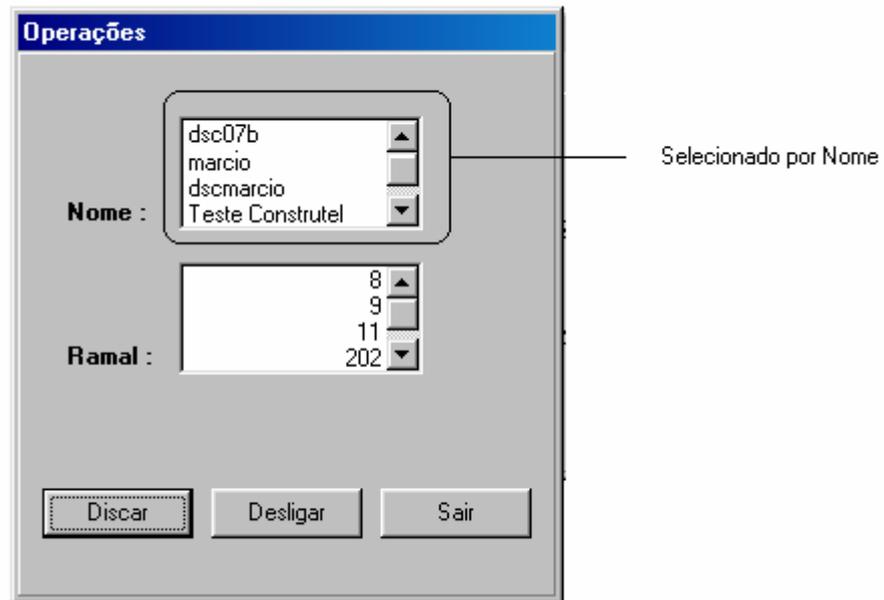
Figura 26 – Operações

The image shows a software window titled "Operações". It contains two dropdown menus. The first dropdown, labeled "Nome", has a list of options: "dsc07b", "marcio", "dscmarcio", and "Teste Construtel". The second dropdown, labeled "Ramal", has a list of options: "8", "9", "11", and "202". Below the dropdowns are three buttons: "Discar", "Desligar", and "Sair". The "Discar" button is highlighted with a dashed border.

O módulo de operações é basicamente onde acontece toda a transação do protótipo onde o usuário escolhe com quem ele quer falar no momento em que deseja efetuar a conexão. Esta tela possui os seguintes campos **Nome**, **Ramal** e os seguintes botões **Discar**, **Desligar**, **Receber** e **Sair**.

A tela de operações possui 2 opções para efetuar uma conexão VoIP que são: através do campo **Nome** ou através do **Ramal**. Nas figuras 27 e 28 estão demonstrados, passo a passo, como efetua-se uma conexão selecionando uma das 2 opções

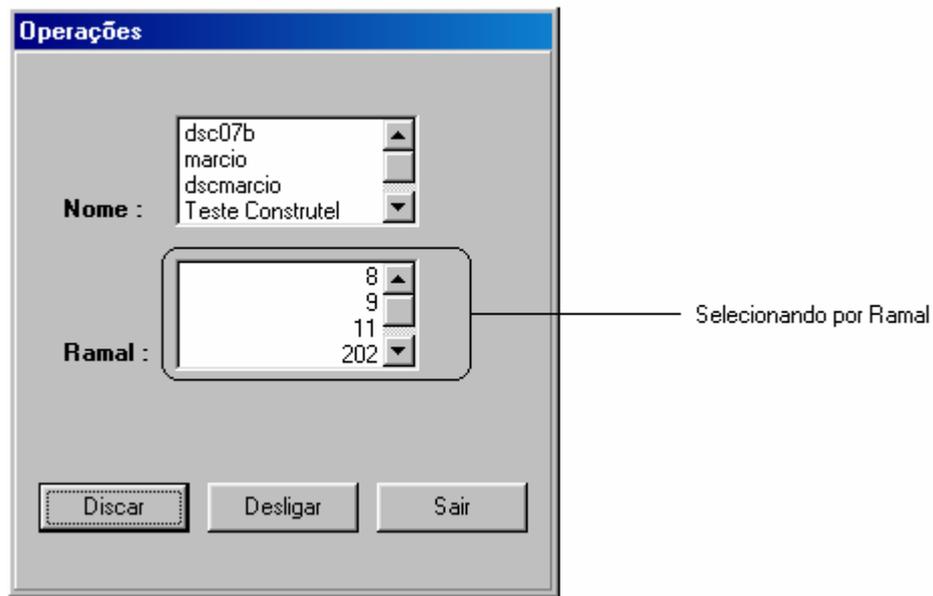
Selecionando por Nome

Figura 27- Seleção por Nome

De acordo com a figura 27 seleciona-se o **Nome do usuário** desejado e em seguida aperta-se no botão **Discar** tornando sessão ativa.

Quando deseja-se encerrar a ligação o botão Desligar deverá ser pressionado.

Selecionando por Ramal

Figura 28- Seleção por Ramal

De acordo com a figura 28 o usuário deseja efetuar uma conexão VoIP através do Ramal. Após ter selecionado o ramal desejado pressiona-se o botão **Discar** e tornando a sessão ativa.

Quando desejar-se encerrar a ligação deve-se pressionar o botão **Desligar** e assim a sessão VoIP ativa estará finalizada.

Descrevendo a funcionalidade dos botões Discar, Desligar, Receber, Sair.

O usuário deve selecionar uma das opções **Nome** ou **Ramal** para que possa ser feita uma busca na tabela de Cadastro de Usuários, enviando o número do IP correspondente ao Nome ou Ramal selecionado.

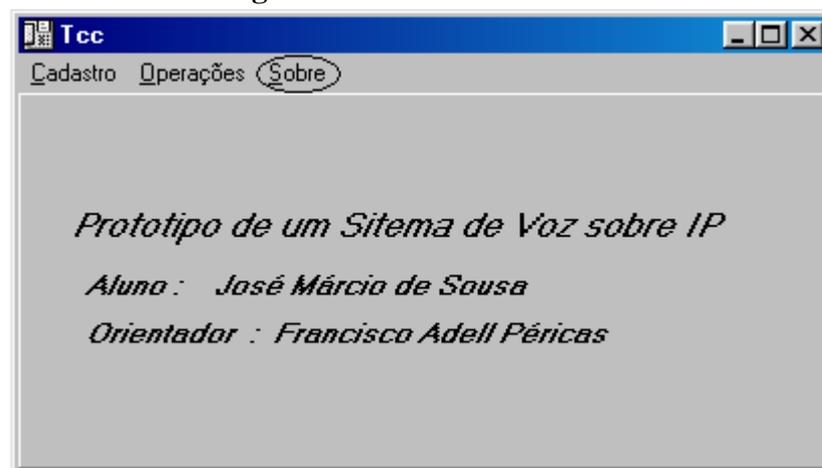
Discar -> Este botão tem por objetivo discar da mesma forma que em um ambiente de telefonia comum, só neste caso o endereço IP previamente cadastrado no banco de dados. Assim que o outro usuário atender sua ligação à conversação entre os computadores esta feita e permanecerá ativa até que seja pressionado o botão Desligar.

Desligar -> Este botão tem como finalidade fechar uma sessão ativa criada pela biblioteca JOVOIPSession.

Sair -> Esta opção permite que o usuário encerre o programa e volte a tela principal.

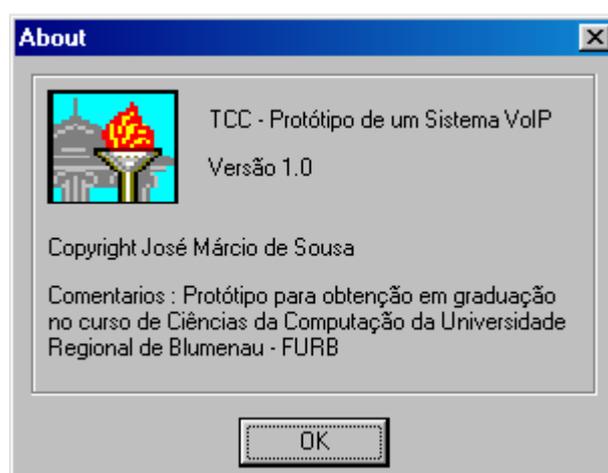
Selecionando a opção Sobre como e mostrado na figura 29.

Figura 29 - Selecionando o sobre



Nesta janela irá aparecer uma breve descrição do protótipo com sua versão e autor do protótipo como é mostrado na figura 30.

Figura 30 - Sobre



5 CONCLUSÃO

O tema VoIP é uma área recente com evolução contínua, e portanto é um tema pouco conhecido e difundido no meio acadêmico e profissional. Devido a pouca difusão do tema encontrou-se dificuldade no levantamento de material referente ao assunto proposto neste trabalho.

A especificação e implementação do protótipo foram às etapas mais específicas já que exigiram um conhecimento aprofundado das características do padrão H.323 e de suas funcionalidades e também da biblioteca JVOIP.

Para a implementação do protótipo proposto neste trabalho, foi utilizada a biblioteca sugerida por Jori Liesenborgs, a qual oferece o recurso de comunicação entre os terminais VoIP. Sem a utilização destes componentes, ficaria inviável estabelecer a comunicação entre os terminais, teria-se então que desenvolver toda a parte de comunicação e liberação da voz e supressão dos sinais de ruído entre outros componentes.

O propósito deste trabalho, foi de extrema importância para conhecimento sobre o funcionamento das Redes de Computadores e de Rede de Telefonia no que diz respeito a protocolos de comunicação, uma vez que todo o processo de VoIP é baseado sobre protocolos IP, RTP e UDP. Além do estudo da biblioteca JVOIP e do padrão H.323 os quais juntamente com os protocolos já mencionados, fazem a comunicação de VoIP acontecer.

Para a implementação deste protótipo utilizando-se da linguagem C++ na sua versão 5.0 que oferece recursos e suporte à aplicação deste gênero, fornecendo uma interface de fácil interpretação para o usuário.

Dadas às características atuais da Internet, em especial a sua implementação no Brasil, conclui-se que a comunicação de voz em tempo real sobre redes IP pode ocorrer de maneira satisfatória dentro de áreas metropolitanas ou de redes corporativas. Entretanto quando distâncias maiores são consideradas, os problemas de degradação da qualidade de comunicação podem comprometer significativamente ou mesmo inviabilizar a realização da conversão.

A aplicação de VoIP desenvolvida foi testada em uma rede local e permitiu uma comunicação de qualidade. Desta forma notou-se que a utilização desta técnica deve ser avaliada antes de sua utilização em grande escala já que teremos outros fatores a serem analisados como a Internet, Links e qualidade dos serviços de telefonias prestados hoje pelas operadoras.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Como complemento a este trabalho, sugere-se uma implementação equivalente através do protocolo SIP ao invés do H.323.

Uma outra alternativa seria, a integração entre VoIP com imagem (vídeo conferência) e também, a confecção de placas para interligar Centrais Comutadas a uma comunicação VoIP, criando assim um telefone IP.

5.2 DIFICULDADES ENCONTRADAS

Durante a elaboração deste trabalho encontrou-se dificuldade na análise da biblioteca VoIP, para a qual foi preciso ter um grande entendimento de redes de computadores e também de redes telefônicas.

E uma outra dificuldade foi à utilização da ferramenta de desenvolvimento da Borland C++ 5.0 já que não era familiarizado com este ambiente.

REFERÊNCIAS BIBLIOGRÁFICAS

CRAIG, Southeren. **Open H.323 project**, [S.l.], ago. 2001. Disponível em: <<http://www.openh323.org>>. Acesso em: 12 set. 2001.

DEMARCO, Tom **Análise estruturada e especificação de sistemas**. Rio de Janeiro: Campus 1989.

DOMINGUES, Miriam Lúcia Campos Serra. **Protocolos de Dados para Conferências Multimídia**. 2000. Dissertação de Mestrado (Ciências da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

DORFMAN, Len. **C++ BY EXAMPLE OBJECT-ORIENTED ANALYSIS DESIGN & PROGRAMMING0**. New York: McGraw-Hill,Inc. 1996.

GANE, Chris; Sarson, Trish. **Análise estruturada de sistemas**. Rio de Janeiro: LTC- Livros Técnicos e Científicos, 1984.

JOÃO, Belmiro do Nascimento. **Metodologia de desenvolvimento de sistemas**. São Paulo: Érica, 1993.

LEOPOLDINA, Graciela Machado. **Modelos de Comunicações para Videoconferência**, São Paulo,[2000?]. Disponível em: <<http://www.cirp.usp.br/secoes/screde/videoconf/videoconf.html>>. Acesso em: 05 out. 2001.

LIESENBORGS, Jori. **JVOIPLIB**, Beringen, ago.2001. Disponível em: <<http://lumumba.luc.ac.be/~jori/jVoIPlib/jVoIPlib.html>> .Acesso em: 10 set. 2001.

LIESENBORGS, Jori. **JRTPLIB**, Beringen, ago.2001. Disponível em: <<http://lumumba.luc.ac.be/~jori/jrtplib/jrtplib.html>> .Acesso em: 10 set. 2001.

LIESENBORGS, Jori. **JThread**, Beringen, ago.2001. Disponível em: <<http://lumumba.luc.ac.be/~jori/jthread/jthread.html>> .Acesso em: 10 set. 2001.

MARTIN, James. **Técnicas estruturadas e case**. Tradução Lúcia Flavia Silva. São Paulo: Makron, McGraw-Hill, 1991.

MONTEIRO, Rafael Flister. **Implementação de transporte robusto de voz em rede baseadas em protocolos IP**. 2000. 85 f. Dissertação de Mestrado (Engenharia Elétrica) - Universidade Federal de Minas Gerais, Belo Horizonte.

NÓBREGA, Obionor. Um Algoritmo adaptativo de transporte para serviços de Voz sobre IP, Simpósio Brasileiro de Redes de Computadores, 2001 Curitiba **Anais...** Curitiba UFPR, 2001.

NORTON, Peter. **Programando em Borland C++ para Windows**.Rio de Janeiro: Berkeley Brasil Editora 1992.

OLIVEIRA, Sérgio. Telefonía IP para ambientes móveis usáveis: Simpósio Brasileiro de Redes de Computadores, 19. 2001, Florianópolis. **Anais...** Florianópolis: UFSC, 2001. p. 542-558.

SHULZRINNE, Hennig. **Object oriented VoIP library**. Página pessoal, Columbia USA, nov. 2000. Disponível em: <<http://www.cs.columbia.edu/~hgs/rtp/jVoIPLib.html>>. Acesso em: 07 jun. 2001.

TANENBAUN, Andrew S. **Redes de Computadores**. Rio de Janeiro:Editora Campus, 1997.

VIANA, A.C., Guedes, L. G. R., Rezende, J. F., and Duarte, O. C. M. B. - **Avaliação de um Ambiente de Diferenciação de Serviço Proporcional em Tráfego TCP**, 19.,2001 Florianópolis.**Anais...** Florianópolis UFSC, 2001, pp. 226 – 241.

YOURDON, Inc. **Yourdon systems method**. New Jersey: Prentice Hall,1993.