

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

FERRAMENTA DE APOIO AO SUPORTE A CLIENTES
UTILIZANDO VOZ SOBRE IP (VOIP)

JEAN CARLOS ELOY LUCHTENBERG

BLUMENAU
2006

2006/1-22

JEAN CARLOS ELOY LUCHTENBERG

FERRAMENTA DE APOIO AO SUPORTE A CLIENTES

UTILIZANDO VOZ SOBRE IP (VOIP)

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Francisco Adell Péricas - Orientador

BLUMENAU
2006

2006/1-22

FERRAMENTA DE APOIO AO SUPORTE A CLIENTES
UTILIZANDO VOZ SOBRE IP (VOIP)

Por

JEAN CARLOS ELOY LUCHTENBERG

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Francisco Adell Péricas – Orientador, FURB

Membro: _____
Prof. Sérgio Stringari, Mestre – FURB

Membro: _____
Prof. Paulo Fernando da Silva, Mestre – FURB

Blumenau, 05 de julho de 2006

Dedico este trabalho à minha família que me apoiou desde o começo de minha graduação, à minha namorada, ao meu orientador e a todos os amigos, especialmente aqueles que me ajudaram diretamente na realização deste.

AGRADECIMENTOS

À Deus, em primeiro lugar, pelo seu imenso amor e graça.

À minha família que sempre esteve presente.

À minha namorada, que suportou por dois semestres momentos muito difíceis longe de mim.

Aos meus amigos, pelos empurrões e cobranças.

Ao meu orientador, Francisco Adell Péricas, por ter acreditado na conclusão deste trabalho e pelo apoio que me concedeu.

Os bons livros fazem “sacar” para fora o que a
pessoa tem de melhor dentro dela.

Lina Sotis Francesco Moratti

RESUMO

Este trabalho especifica e implementa uma ferramenta de apoio ao suporte a clientes de uma empresa de software utilizando voz sobre IP para conversação entre cliente e suporte. Além da comunicação por voz este trabalho permite a troca de mensagens de texto entre os usuários de uma mesma empresa ou no atendimento aos clientes.

Palavras-chave: VoIP. Suporte. Multimídia. *Bufferring*. *Jitter*.

ABSTRACT

This work specifies and implements a client supporting tool for a software-house with voice over IP for client/support conversation. Besides the voice communication this work allows text messages exchange between users from the same enterprise or client attending.

Key-words: VoIP. Support. Multimedia. *Bufferring*. *Jitter*.

LISTA DE ILUSTRAÇÕES

Figura 1 – Cabeçalho do protocolo RTP	16
Figura 2 – Componentes do H.323	18
Figura 3 – Exemplo de <i>invite</i>	19
Figura 4 – Exemplo de <i>busy here</i>	20
Figura 5 – Comunicação por voz com atrasos	21
Figura 6 – Arquitetura geral em três camadas	24
Figura 7 – Controle de acesso	26
Figura 8 – Registro e Atendimento de solicitações	27
Figura 9 – Início de comunicação	28
Figura 10 – Diagrama de atividades anterior à implantação	31
Figura 11 – Diagrama de atividades posterior à implantação	35
Quadro 1 – Envio de mensagens	41
Quadro 2 – Envio de mensagens (servidor)	43
Quadro 3 – Envio de mensagens (servidor)	44
Figura 13 – Tela de login na ferramenta	46
Figura 14 – Tela principal	47
Figura 15 – Comunicação entre usuários	48
Figura 16 – Registro de solicitações	49
Figura 17 – Controle de solicitações	50
Figura 18 – Atendimento a solicitações	51
Figura 19 – Servidor de aplicação	52

LISTA DE SIGLAS

DLL – *Dynamically Linked Library*

ERP – *Enterprise Resource Planning*

HTTP – *HyperText Transfer Protocol*

IETF – *Internet Engineering Task Force*

IP – *Internet Protocol*

MSN – *Microsoft Network*

PCM – *Pulse Code Modulation*

RF – *Requisito funcional*

RFC – *Request for Comments*

RNF – *Requisito não-funcional*

RR – *Receiver Reports*

RTCP – *Real-time Control Protocol*

RTP – *Real-time Transport Protocol*

SDES – *Source Descriptions*

SMTP – *Simple Mail Transfer Protocol*

SR – *Sender Reports*

SIP – *Session Initiation Protocol*

TCP – *Transmission Control Protocol*

UDP – *User Datagram Protocol*

UML – *Unified Modeling Language*

VOIP – *Voice over IP*

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVOS DO TRABALHO	12
1.2 ESTRUTURA DO TRABALHO	12
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 VOZ SOBRE IP	14
2.1.1 Digitalização e compressão.....	14
2.1.2 Protocolos.....	15
2.1.2.1 Real-time Transport Protocol	15
2.1.2.2 Real-time Control Protocol.....	16
2.1.2.3 H.323	17
2.1.2.4 Session Initiation Protocol	18
2.1.3 Problemas na comunicação	21
3 DESENVOLVIMENTO DO TRABALHO.....	23
3.1 ARQUITETURA GERAL	23
3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	25
3.3 ESPECIFICAÇÃO	25
3.3.1 Diagramas de casos de uso.....	26
3.3.2 Diagrama de atividades	29
3.3.3 Diagrama de classes	36
3.4 IMPLEMENTAÇÃO	39
3.4.1 Técnicas e ferramentas utilizadas.....	39
3.4.2 Operacionalidade da implementação	45
3.5 RESULTADOS E DISCUSSÃO	52
4 CONCLUSÕES.....	54
4.1 EXTENSÕES	55
REFERÊNCIAS BIBLIOGRÁFICAS	57

1 INTRODUÇÃO

Muitas empresas não possuem ferramentas específicas para prestação de suporte aos seus clientes. Ao invés disso, utilizam sistemas criados com outros propósitos, mas que acabam atendendo as suas necessidades básicas. A este exemplo tem-se o uso do Microsoft *Network Messenger (MSN Messenger)*, que foi criado para comunicação entre pessoas e é voltado principalmente ao lazer e ao entretenimento, abrindo espaço para distrações por parte dos usuários que deveriam estar concentrados na execução dos seus trabalhos.

Quando se utiliza uma ferramenta não específica como a citada anteriormente, abre-se mão de funcionalidades importantes tais como:

- a) contabilização automática do tempo gasto no atendimento aos clientes;
- b) geração de extratos mais detalhados para os clientes;
- c) integração com o cadastro de clientes da empresa, entre outras.

Além das funcionalidades mencionadas acima, a comunicação por voz vem sendo cada vez mais incorporada a estas ferramentas. Segundo Gurle, Hersent e Petit (2002, p. 15), *Voice over Internet Protocol*, ou VoIP, é o setor de telecomunicações que mais cresce. Seu crescimento está ocorrendo a uma taxa mais veloz do que o crescimento da telefonia móvel. Como esta tecnologia está sendo bem aceita pelo mercado, sistemas de suporte *online* devem estar preparados para trabalharem com ela para serem bem sucedidos e atenderem às atuais necessidades dos usuários.

Em vista disso, as empresas precisam de uma ferramenta específica para dar apoio ao suporte a seus clientes e que atenda às novas exigências do mercado. Sabendo-se desta necessidade, este trabalho de conclusão de curso propõe desenvolver uma ferramenta de apoio ao suporte com VoIP.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver uma ferramenta de apoio ao suporte no atendimento aos clientes que atenda às necessidades da empresa Dessis Informática Ltda, disponibilizando comunicação com o cliente via VoIP.

Os objetivos específicos do trabalho foram:

- a) desenvolver uma ferramenta de apoio ao suporte no atendimento aos clientes para abertura e atendimento a chamados;
- b) utilizar uma biblioteca comercial de voz sobre IP para incluir na ferramenta de apoio ao suporte a clientes comunicação por voz;
- c) utilizar um protocolo padrão de voz sobre IP para realizar a comunicação;
- d) disponibilizar a comunicação *online* entre o suporte e clientes.

1.2 ESTRUTURA DO TRABALHO

O primeiro capítulo apresenta uma introdução com idéias gerais e sem muitos detalhes sobre o assunto que será abordado neste trabalho e sua aplicabilidade.

Já o segundo capítulo apresenta a fundamentação teórica sobre os principais pontos deste trabalho em especial voz sobre IP. Neste capítulo ainda são abordadas outras questões como qual protocolo será utilizado e quais técnicas podem ser utilizadas para que ocorra uma transmissão de boa qualidade.

Em seguida, no terceiro capítulo são apresentados detalhes sobre o desenvolvimento da ferramenta, bem como sua especificação e suas características. Além disso, serão exibidas algumas telas da ferramenta para que se tenha uma idéia melhor de como as coisas funcionam.

E por fim, o quarto capítulo apresenta as conclusões conseguidas até final do desenvolvimento deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados alguns aspectos teóricos e técnicos relacionados com o tema principal deste trabalho que é voz sobre IP.

2.1 VOZ SOBRE IP

Esta seção aborda alguns conceitos envolvidos com a tecnologia de voz sobre IP (VoIP) e está dividida em três subseções, sendo elas: digitalização e compressão, protocolos e problemas na comunicação.

2.1.1 Digitalização e compressão

Segundo Péricas (2003, p. 113), para que o áudio seja transmitido pela rede, é necessário que o mesmo seja digitalizado e comprimido. Isto significa que o áudio precisa passar pela etapa da digitalização, que nada mais é do que a transformação do áudio analógico, representado por ondas, para o áudio digital, representado por bits.

No caso específico do áudio, a técnica básica de digitalização é o PCM (*Pulse Code Modulation*). Codificação de voz normalmente usa o PCM, com taxa de 8000 amostras por segundo, utilizando quantização de 8 bits por amostra, obtendo uma taxa de 64 Kbps. (PERICAS, 2003, p. 113).

Transformações de áudio com frequências tão altas como as descritas acima geram um áudio digital de boa qualidade. No entanto, o arquivo gerado fica muito grande e isto inviabiliza a transmissão “em tempo real” do mesmo pela internet. Então se faz necessário comprimir o áudio digital, para que ele tenha o seu tamanho reduzido e não comprometa o seu recebimento.

Péricas (2003, p. 113) afirma que algumas técnicas de compressão para voz são o

GSM (13 Kbps), o G.729 (8 Kbps) e o G.723.3 (5,3 Kbps a 6,4 Kbps). Pode-se perceber com estes métodos que a taxa diminui consideravelmente, já que com o PCM a taxa necessária é de 64 Kbps e ao se utilizar alguma destas técnicas o arquivo tem seu tamanho diminuído em mais de quatro vezes.

2.1.2 Protocolos

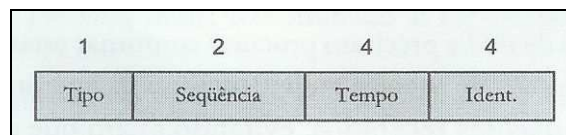
Existem quatro protocolos que merecem destaque neste trabalho, o *Real-time Transport Protocol* (RTP), o *Real-time Control Protocol* (RTCP), o H.323 e o *Session Initiation Protocol* (SI). Ambos são tratados com mais detalhes em seguida.

2.1.2.1 Real-time Transport Protocol

Depois que o áudio foi digitalizado e comprimido ele está pronto para ser transmitido pela rede. Para a transmissão ser realizada, é necessário fazer uso de um protocolo. Péricas (2003, p. 116) afirma: “O protocolo RTP é utilizado para transportar pacotes com formatos de dados para áudio e vídeo”. Isto quer dizer que o áudio é transmitido pelo RTP até o destinatário em tempo real, porém existe o atraso que é abordado mais adiante.

Segundo Gurle, Hersent e Petit (2002, p. 11), o RTP é normalmente usado em cima do *User Datagram Protocol* (UDP), que fornece a noção de porta para multiplexação de transporte e a detecção de erro. A não utilização do *Transmission Control Protocol* (TCP) se deve ao fato de o mesmo realizar a retransmissão nos casos de erros. No entanto falhas são aceitáveis na transmissão multimídia pela rede, já que a mesma deve ser realizada em tempo real com o mínimo de atraso possível. Por este motivo, retransmissões não são admitidas numa comunicação com VoIP.

Conforme Péricas (2003, p. 116), a aplicação servidora de áudio deve encapsular a mensagem em pacotes RTP e enviá-los através de um *socket* UDP à aplicação cliente no destino. Quando a aplicação cliente receber o conteúdo da transmissão, ela precisa abrir os pacotes retirando os seus conteúdos e construindo o *streaming* com as informações que estão no cabeçalho, como pode ser visto na Figura 1.



Fonte: Péricas (2003, p. 116).

Figura 1 – Cabeçalho do protocolo RTP

Conforme pôde ser visualizado na Figura 1, o seu tamanho é de 11 bytes e possui quatro campos, sendo que o primeiro byte corresponde ao tipo de dados que há no pacote, permitindo informar se é um vídeo, áudio ou até mesmo a codificação do áudio. Já o campo *seqüência* informa a ordem dos pacotes na comunicação. O terceiro campo ilustra o momento em que o pacote foi gerado e por fim o quarto campo é utilizado como identificador único do pacote. Além do cabeçalho, existe um campo neste protocolo que leva junto o conjunto de dados, que é o áudio ou vídeo digital comprimido.

2.1.2.2 Real-time Control Protocol

O RTCP é um protocolo que pode ser utilizado em conjunto com o RTP e é responsável por transmitir informações de controle às aplicações clientes.

O RTCP é usado para transmitir aos participantes, de tempos em tempos, pacotes de controle relativos a uma sessão RTP em particular. Esses pacotes de controle podem incluir informações a respeito dos participantes (seus nomes, endereços de email, etc.) e informações sobre o mapeamento dos participantes e suas fontes de fluxo individuais. (GURLE, HERSENT e PETIT, 2002, p. 14).

Como pôde ser visto acima, o protocolo RTCP é capaz de fornecer informações úteis às aplicações clientes sem que seja necessário elas mesmo criarem algum protocolo específico

para tais necessidades.

De acordo com Gurle, Hersent e Petit (2002, p. 14), existem vários tipos de pacotes RTCP:

- a) os *Sender Reports* (SR), que contêm informações de transmissão e recepção para transmissores ativos;
- b) os *Receiver Reports* (RR), que contêm informações de recepção para ouvintes que não sejam também transmissores ativos;
- c) os *Source Descriptions* (SDS), que descrevem vários parâmetros da fonte;
- d) o BYE, que é enviado por um participante quando ele abandona a conferência;
- e) os APP, que são funções específicas de uma aplicação.

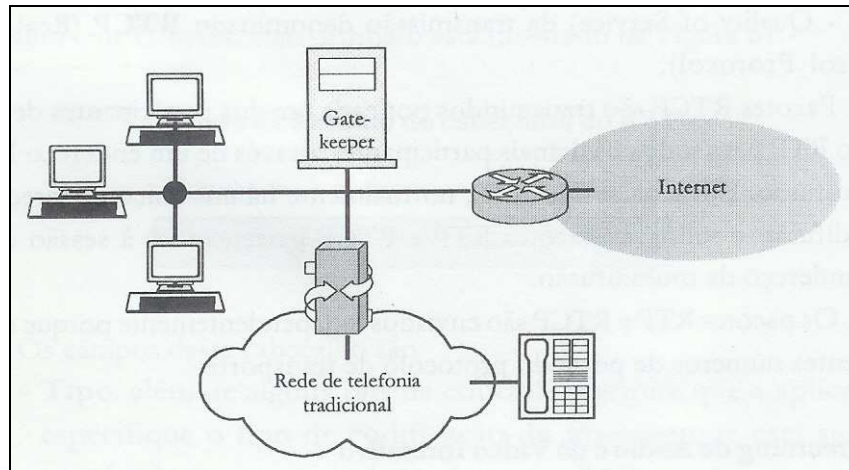
Conforme visto anteriormente, tais pacotes contribuem para a gestão de toda a comunicação multimídia e ficam disponíveis às aplicações clientes. Segundo Péricas (2003, p. 117), os pacotes RTP e RTCP são enviados independentemente porque usam portas de números diferentes. Isto quer dizer que os pacotes dos dois protocolos não são transmitidos em conjunto, já que o RTCP é apenas um protocolo de controle e não de transporte em tempo real, como o RTP.

2.1.2.3 H.323

Gurle, Hersent e Petit (2002, p. 4) afirmam que “O H.323 não atraiu maior interesse do mercado até que a VocalTec e a Cisco fundaram o fórum *Voice over IP* (VoIP) para estabelecer padrões para produtos VoIP”. Com este fórum, o H.323 ganhou maior destaque na área de produtos com voz sobre IP e foi amplamente difundido.

Segundo Péricas (2003, p. 118), os equipamentos H.323 incluem *Gateways* e *Gatekeepers*, onde os *Gateways* permitem a comunicação entre terminais H.323 e aparelhos

telefônicos tradicionais, já os *Gatekeepers* são equipamentos opcionais para tradução de endereços, autorização de acesso, gerência de largura de banda e contabilização e são demonstrados na Figura 2.



Fonte: Péricas (2003, p. 118).

Figura 2 – Componentes do H.323

2.1.2.4 Session Initiation Protocol

O padrão SIP foi desenvolvido pela *Internet Engineering Task Force* (IETF) em 1999 com o objetivo de utilizá-lo para realizar transmissões de voz sobre IP e está definido na *Request for Comments* (RFC) 2543. De acordo com Nazário (2003, p. 23), o SIP é um protocolo de controle da camada de aplicação usado para criar, modificar e terminar sessões com um ou mais usuários (participantes). Isto quer dizer que ele é na verdade um coordenador de toda a comunicação de voz sobre IP.

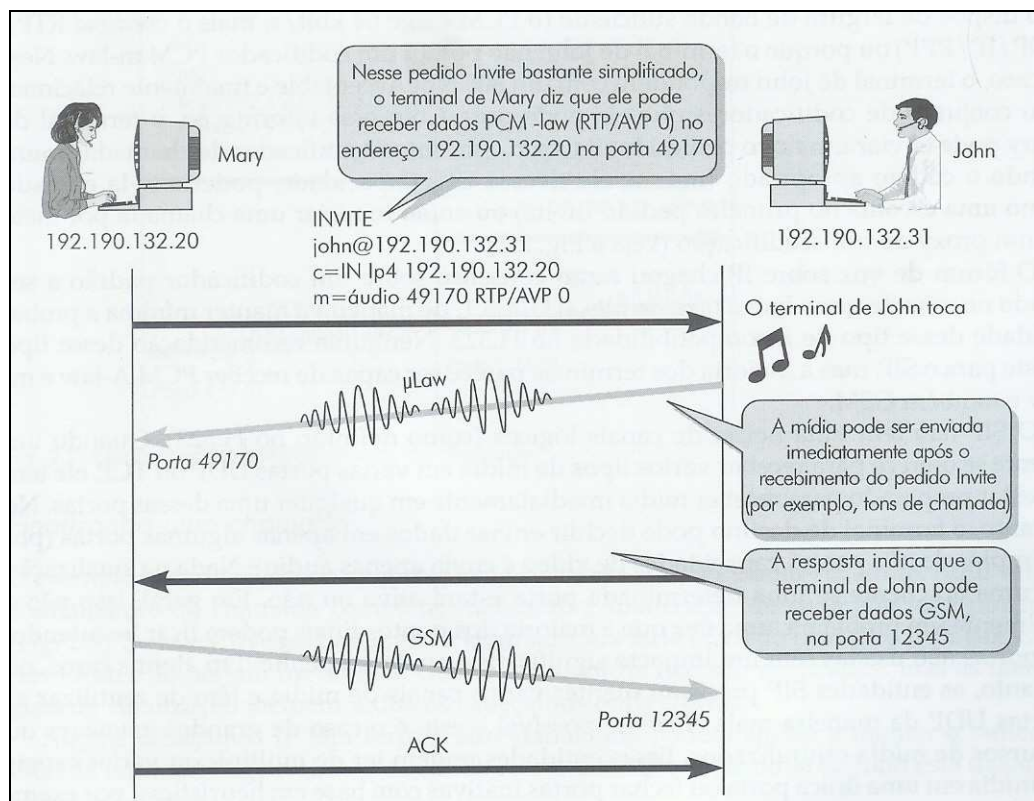
O SIP é um protocolo menos pesado que o H.323 no que diz respeito ao estabelecimento da chamada, pois quando um usuário pretende telefonar para outro, ele inicia a chamada com uma mensagem de convite, que contém informação como a sua identificação, as características da chamada e os serviços que pretende utilizar, enviando-a para um servidor SIP. Quando um servidor SIP recebe esta mensagem de convite, ele localiza o destinatário e reenvia a mensagem de convite para o destino. Durante o re-direcionamento, o servidor SIP pode efetuar a autorização da chamada. (PÉRICAS, 2003, p. 120).

Além disso, o SIP é baseado em texto plano assim como o *HyperText Transfer*

Protocol (HTTP) e o *Simple Mail Transfer Protocol* (SMTP). O SIP possui alguns componentes principais, entre eles, existe o agente do usuário SIP que é a aplicação do usuário final e é o responsável por receber e aceitar chamadas. Além dele, existe o *Servidor Proxy SIP*, que é o encarregado repassar as requisições do agente do usuário para outros servidores SIP. Existe também o *Registrador SIP* que disponibiliza as informações de localidades que recebe dos agentes do usuário e as guarda.

Para realizar chamadas utilizando o SIP, os participantes utilizam pedidos e recebem em troca as respostas, onde um pedido pode retornar várias respostas até que se receba uma resposta final. Antes de tudo, é estabelecida uma conexão para sinalizar os pontos de origem e destino através do envio da mensagem *invite*, que faz o pedido de comunicação. A partir deste momento o destino já está apto a conversar com a origem.

Um exemplo de um *invite* é demonstrado na Figura 3.



Fonte: Gurle, Hersent e Petit (2002, p. 119).

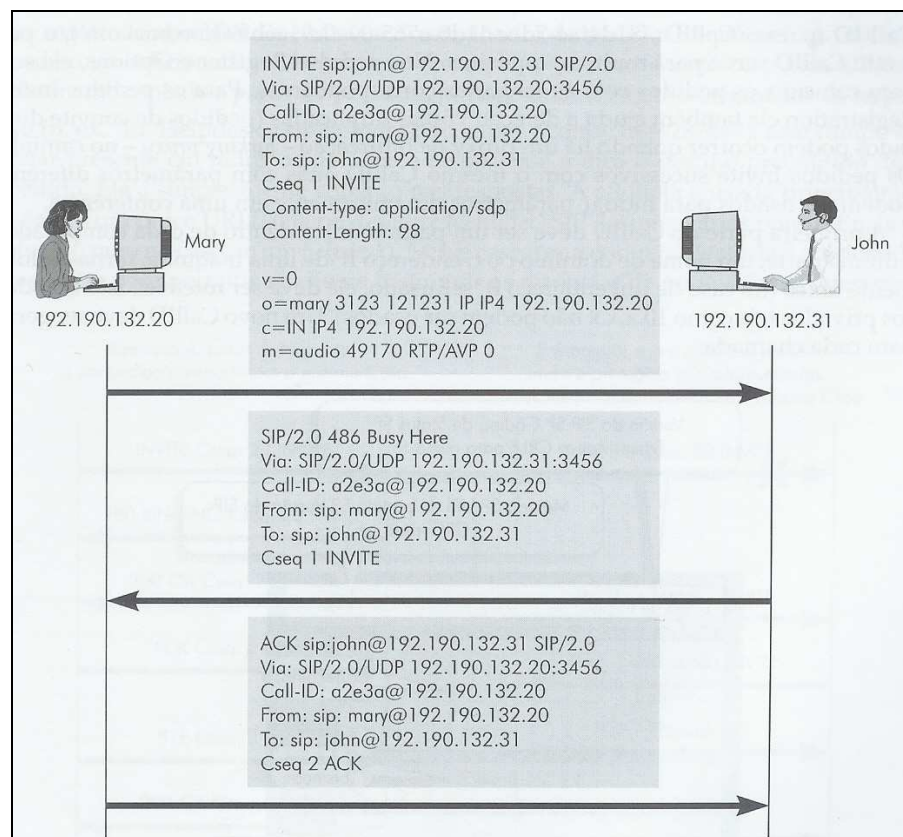
Figura 3 – Exemplo de *invite*

O destino envia uma mensagem de OK à origem para informar que ele aceitou o convite e pode receber os dados da mídia em uma determinada porta. Ao receber a mensagem de OK, a origem envia um ACK com o objetivo de confirmar ao destino o recebimento da mesma. Após estas transmissões, a comunicação entre os participantes continua até que alguém deseje terminar a conversa.

Para finalizar, o participante envia a mensagem BYE. Ao recebê-la, o destinatário precisa enviar um OK com o objetivo de confirmar a finalização do diálogo.

O protocolo SIP também oferece a possibilidade de se rejeitar chamadas, através do uso de alguns códigos, como o 486 que representa a mensagem BUSY HERE, cujo significado é informar que o usuário está ocupado. Após o recebimento da mensagem de rejeição, o participante precisa enviar um ACK.

Um exemplo de um BUSY HERE é demonstrado na Figura 4.



Fonte: Gurle, Hersent e Petit (2002, p. 123).

Figura 4 – Exemplo de *busy here*

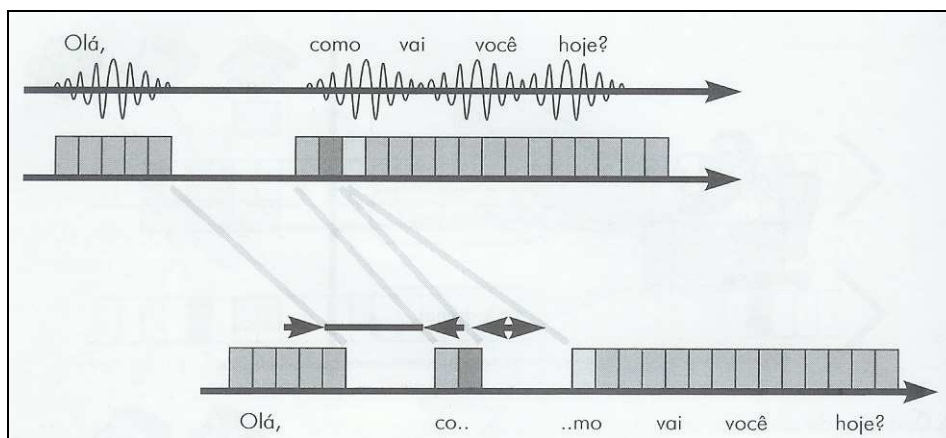
Na Figura 4 a usuária *Mary* inicialmente faz o pedido de *invite* para o usuário *John*, que retorna com a mensagem *busy here*, informando a rejeição à chamada. Para confirmar o seu recebimento, *Mary* envia um ACK.

2.1.3 Problemas na comunicação

Implementações de voz sobre IP devem levar em consideração que em conversações não se pode perceber atrasos muito grandes. Estes atrasos de propagação podem ocorrer devido à perda de pacotes durante a transmissão do emissor ao receptor, que é prevista no projeto do sistema.

O atraso de propagação corresponde ao tempo necessário para a propagação do sinal elétrico ou propagação do sinal óptico no meio sendo utilizado (fibras ópticas, satélite, coaxial, ...) e é um parâmetro imutável onde o gerente de rede não tem nenhuma influência. (MARTINS, 1999, p. 11).

Pode ser visualizado um exemplo da relação entre a transmissão dos pacotes e o seu atraso na Figura 5.



Fonte: Gurle, Hersent e Petit (2002, p. 10).

Figura 5 – Comunicação por voz com atrasos

Na Figura 5 pode ser visualizado o atraso do recebimento dos pacotes de dados. Os

intervalos existentes entre os recebimentos não geram cortes com silêncio no receptor, já que a aplicação deve estar preparada para estes casos e gerar um som mesclado com os últimos pacotes recebidos.

Em um sistema com VoIP, a transmissão deve ocorrer em tempo real, ou seja, o conteúdo deve ser recebido com o menor atraso possível, mesmo que não seja recebido por completo. A espera excessiva por uma resposta na transmissão gera descontentamento por parte dos usuários. E isso é inadmissível em uma transmissão em tempo real.

Além dos atrasos na transmissão, pode ocorrer a alteração da ordem de chegada dos pacotes no receptor. A essa desordem e à variação do atraso no recebimento dos pacotes se dá o nome de *jitter*. Conforme Martins (1999, p. 13), “*Jitter* introduz uma distorção no processamento da informação na recepção e deve ter mecanismos específicos de compensação e controle que dependem da aplicação em questão”. Uma das soluções mais utilizadas para resolver este problema é o uso de *buffer* através da técnica de *buffering*.

O problema causado pelo *jitter* é tratado e resolvido pelo protocolo RTP, que com os campos “Sequência” e “Tempo” do seu cabeçalho, permite à aplicação do destinatário processar os pacotes e ordená-los novamente.

3 DESENVOLVIMENTO DO TRABALHO

Neste capítulo são apresentados os principais aspectos relacionados ao desenvolvimento deste trabalho de conclusão de curso.

3.1 ARQUITETURA GERAL

A ferramenta desenvolvida neste trabalho possui uma arquitetura em três camadas, sendo que a primeira camada é o próprio banco de dados, ou seja, a base de dados que contém as informações necessárias para disponibilização ao usuário quando o mesmo necessitar.

A base de dados desta ferramenta foi modelada seguindo o mesmo padrão de modelagem do outro sistema utilizado pela empresa que faz uso de algumas informações existentes, tais como os seguintes cadastros: clientes, sistemas, programas (telas de cada sistema), funcionários e usuários. Algumas informações dos cadastros utilizados pelo outro sistema são apenas lidas por esta ferramenta, no entanto, existem outras que são também alteradas nos dois sistemas. Desta forma, ambos fazem acesso ao mesmo arquivo do banco de dados.

Este critério de seguir o mesmo padrão de modelagem utilizado pelo outro sistema foi adotado levando-se em consideração que como ambos acessam o mesmo arquivo e as mesmas tabelas, seria necessário modificar a estrutura existente da base de dados e como consequência o outro programa que faz acesso a ela. Este não é o objetivo deste trabalho, muito pelo contrário, a implantação desta ferramenta na empresa deve acontecer da forma mais transparente possível, sem criar dificuldades ou re-trabalhos.

A segunda camada é chamada de servidor de aplicação. Ele é o responsável por acessar diretamente a primeira camada (base de dados) e retornar a informação que for necessária. É

também no servidor de aplicação que estão implementadas algumas regras de negócio da ferramenta, como por exemplo a rotina de conexão dos usuários.

A terceira camada é também conhecida como *front end*. Ela é a aplicação cliente, ou seja, aquela que faz acesso ao servidor de aplicação. Além disso, é nela que estão as telas que o usuário visualiza e interage.

De uma maneira geral, quando o usuário deseja visualizar alguma informação, por meio do *front end* ele faz a solicitação, esta é enviada via *socket* TCP/IP em um formato comprimido para o servidor de aplicação, que pode trabalhar esta solicitação antes de encaminhá-la ao banco de dados.

Para o envio das solicitações de informações pelo *socket*, é utilizado o protocolo TCP/IP padrão, não sendo necessário criar um protocolo específico para tal finalidade.

A arquitetura geral desta ferramenta é demonstrada na Figura 6:

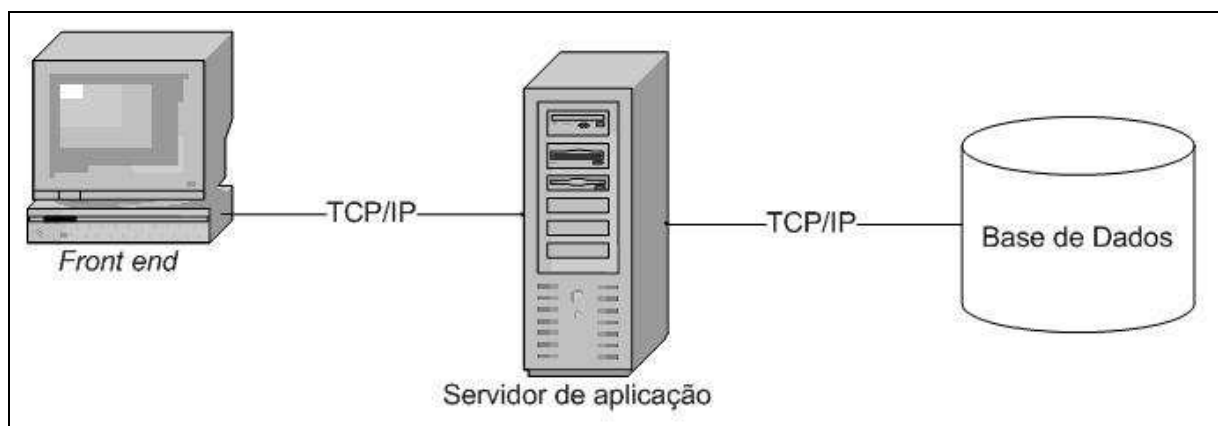


Figura 6 – Arquitetura geral em três camadas

Resumindo, o *front end* tem como função principal exibir as interfaces, ou as telas para interação com o usuário, já o servidor de aplicação é o responsável por receber, processar, enviar as solicitações de informações à base de dados e por fim retornar ao *front end*.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos são classificados em requisito funcional (RF) e requisito não-funcional (RNF). A ferramenta deve:

- a) permitir o usuário ter acesso à ferramenta através do preenchimento de um *login* e uma senha (RF);
- b) permitir a comunicação textual entre os usuários (RF);
- c) possibilitar a comunicação por voz entre os usuários (RF);
- d) contabilizar o tempo de suporte para cada cliente (RF);
- e) integrar-se com o cadastro de clientes da empresa (RF);
- f) ser implementada na linguagem Borland Delphi, utilizando o ambiente Borland Delphi 7 (RNF);
- g) utilizar banco de dados Firebird 1.5 (RNF).

3.3 ESPECIFICAÇÃO

Nesta seção é demonstrado como a ferramenta deste trabalho foi especificada. Para tanto, foram utilizados os diagramas de casos de uso, classes e atividades da *Unified Modeling Language* (UML). Segundo Lima (2005, p. 32), a UML é uma linguagem gráfica para análise, especificação e construção de sistemas para representar projetos orientados a objetos utilizando uma notação comum. Com isto pode-se entender que a utilização desta linguagem de modelagem é importante para uma melhor compreensão e documentação de toda a especificação do projeto em desenvolvimento. Para a elaboração dos diagramas foi utilizada a ferramenta Enterprise Architect da empresa Sparx Systems.

3.3.1 Diagramas de casos de uso

Os primeiros diagramas elaborados na especificação desta ferramenta são os de casos de uso. De acordo com Guedes (2004, p. 26), estes são os diagramas mais gerais e informais da UML, sendo utilizados normalmente nas fases de levantamento e análise de requisitos do sistema, mas podem ser consultados durante todo o processo de modelagem. Além disso, os casos de uso podem ser bem detalhados através dos cenários, que possuem descrições do que exatamente acontece em cada ação de um deles.

Guedes (2004, p. 45) complementa ainda afirmando que estes diagramas procuram, por meio de uma linguagem simples, possibilitar a compreensão do comportamento externo do sistema por qualquer pessoa e é dentre todos os eles o mais abstrato e simples de entender.

O primeiro caso de uso relevante para esta ferramenta é o “Efetua Login” que pertence ao módulo *front end*. Nele, é demonstrado quem são os atores responsáveis por esta ação e identifica o controle de acesso à ferramenta, através do preenchimento de um *login*, uma senha e um status inicial. O status inicial é o modo, ou estado de como o usuário deseja aparecer para os seus contatos.

O caso de uso em questão é demonstrado na Figura 7.

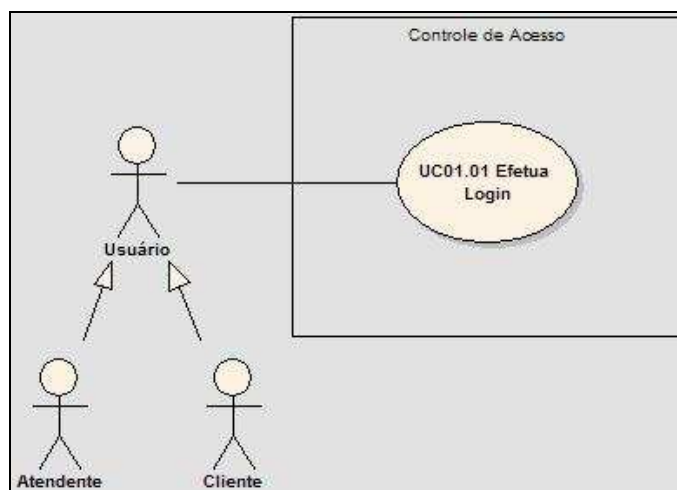


Figura 7 – Controle de acesso

Após o usuário cliente ter acesso à ferramenta através do *login* realizado com sucesso, ele pode registrar uma solicitação. Ela, por sua vez, é gravada diretamente no banco de dados da empresa que fornece o suporte. Ao recebê-la, um técnico pode atendê-la e fornecer o auxílio ao solicitante. Estes dois procedimentos são representados através do diagrama demonstrado na Figura 8.

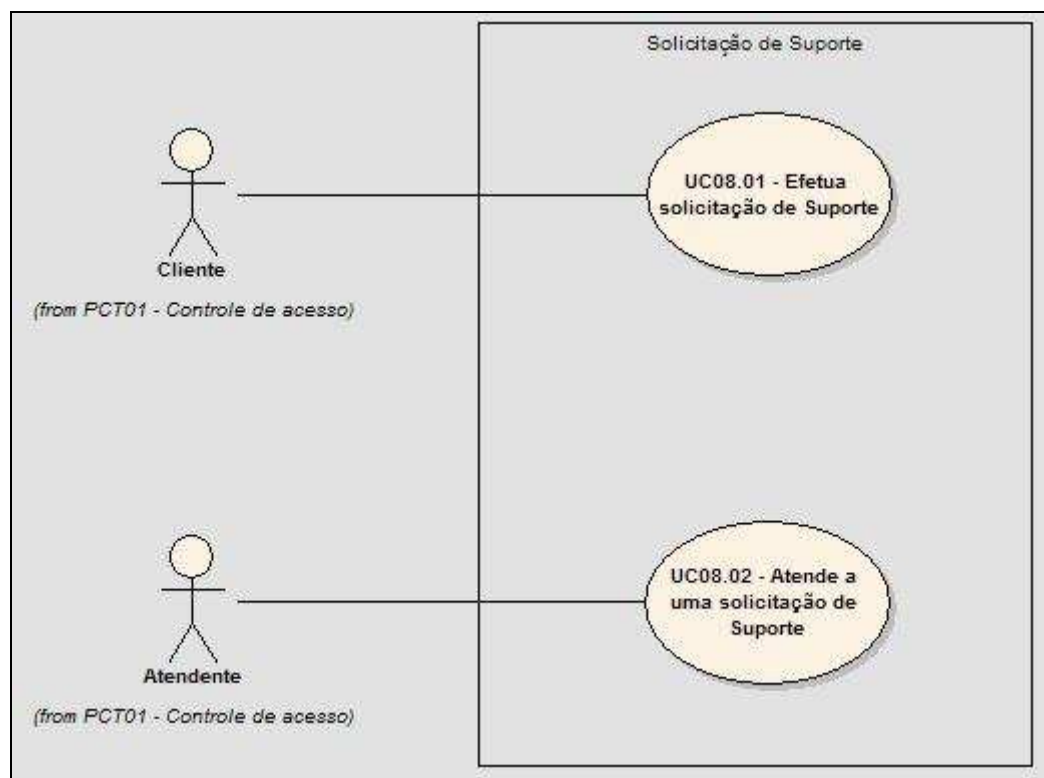


Figura 8 – Registro e Atendimento de solicitações

No caso de uso “Efetua solicitação de suporte” da figura 8, é aberta uma tela para o registro de solicitações. Nela, devem ser preenchidos alguns campos que fornecerão as informações necessárias para o atendente ter conhecimento do que realmente o solicitante deseja. Existem alguns campos que são obrigatórios, tais como o tipo e a prioridade da solicitação, que são carregados do banco de dados. Além deles, é preciso preencher o programa ou tela relacionada com a solicitação, uma descrição resumida e uma detalhada. Existem ainda outros que são obrigatórios de acordo com o tipo de solicitação escolhida, como por exemplo os campos “objetivo” e o “em que situação”, onde o usuário deve

descrever o objetivo da alteração e em que casos acontece o problema descrito.

No caso de uso “Atende a uma solicitação de Suporte”, é aberta uma tela com as informações sobre o registro feito pelo cliente. Nela, os dois usuários podem comunicar-se textualmente em busca de uma solução para o problema em questão. Além disso, eles podem comunicar-se por voz utilizando a tecnologia de voz sobre IP apenas selecionando o botão correspondente.

A partir do momento em que os dois usuários iniciam uma comunicação, seja ela textual ou por voz, é iniciado também o caso de uso início de comunicação que é demonstrado abaixo na Figura 9.

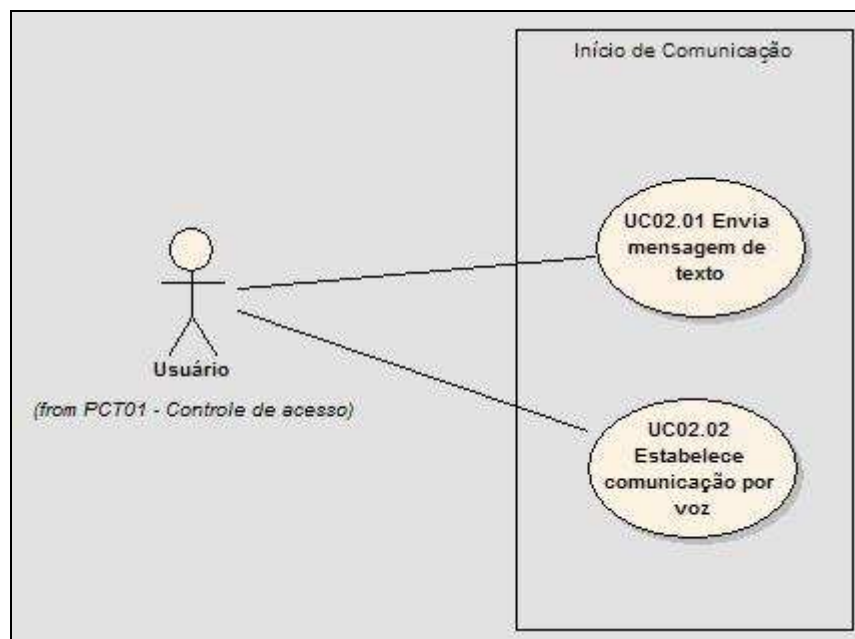


Figura 9 – Início de comunicação

No caso de uso “Envia mensagem de texto”, o usuário (cliente ou atendente) preenche o texto que deseja transmitir ao destinatário, e ao enviá-lo, ele é empacotado e transferido pela rede. Ao recebê-lo, o outro usuário pode continuar comunicando-se normalmente. Já no outro caso de uso da Figura 9, “Estabelece comunicação por voz”, o usuário seleciona o botão correspondente à comunicação por voz. Então é aberta uma tela para a comunicação VoIP entre os dois usuários até que o problema seja resolvido, para então terminar a conversa e o

atendimento.

3.3.2 Diagrama de atividades

O segundo diagrama elaborado neste trabalho é o de atividades. Segundo Guedes (2004, p. 31), ele preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica e concentra-se na representação do seu fluxo de controle. Em outras palavras, este diagrama dá uma noção das atividades que devem ser executadas em um determinado processo.

A empresa que utilizará a ferramenta desenvolvida neste trabalho de conclusão de curso é a *software-house* Dessis Informática Ltda, que é responsável pelo desenvolvimento de *softwares*. A sua linha de produtos é a de gestão empresarial, tanto para o comércio ou para a indústria. Atualmente ela possui três sistemas que são os carros-chefes, ou seja, aqueles que são os maiores responsáveis pelo seu faturamento, um deles para cada segmento de atuação.

No entanto, está sendo desenvolvido um projeto que englobará todas as funcionalidades dos três sistemas em uso em um único. Ao resultado desta junção, dá-se o nome de *Enterprise Resource Planning* (ERP), que é um sistema completo de gestão empresarial.

Com o lançamento deste projeto, o processo de suporte aos clientes mudará um pouco. Atualmente, quando o cliente necessita de alguma alteração em algum *software* aplicativo, ele faz o registro por meio de um sistema específico para tal necessidade, que nada mais é do que um cadastro das solicitações, com informações do que está sendo solicitado com previsão e prioridade. Este chamado é lançado instantaneamente na agenda interna da empresa para que o pessoal do suporte técnico ou do desenvolvimento o analise e eventualmente o execute.

Em caso de dúvidas em relação ao serviço, é feita uma ligação telefônica para o cliente

ou é enviado um e-mail com os questionamentos. Após a análise superficial feita por um programador ou por alguém do suporte sobre a viabilidade e possibilidade de execução da solicitação, ela é agendada diretamente para a pessoa responsável.

Caso não seja possível realizá-lo, o serviço é apontado para algum funcionário do suporte para que o mesmo faça um retorno ao cliente através de um e-mail. Porém, caso a solicitação seja aprovada, a mesma é encaminhada a algum programador que a executará. Ao terminá-la, o serviço é fechado e é enviado um e-mail ao cliente informando sobre uma nova atualização disponível com um texto de resposta sobre o que foi feito.

Além disso, existe outra situação, que é o atendimento feito por telefone, e-mail e pelo MSN Messenger. Neste caso, tenta-se resolver o problema no mesmo momento, porém se isto não for possível, é solicitado ao cliente que registre uma solicitação.

Na Figura 10 é demonstrado através do diagrama de atividades o processo desta forma de trabalho.

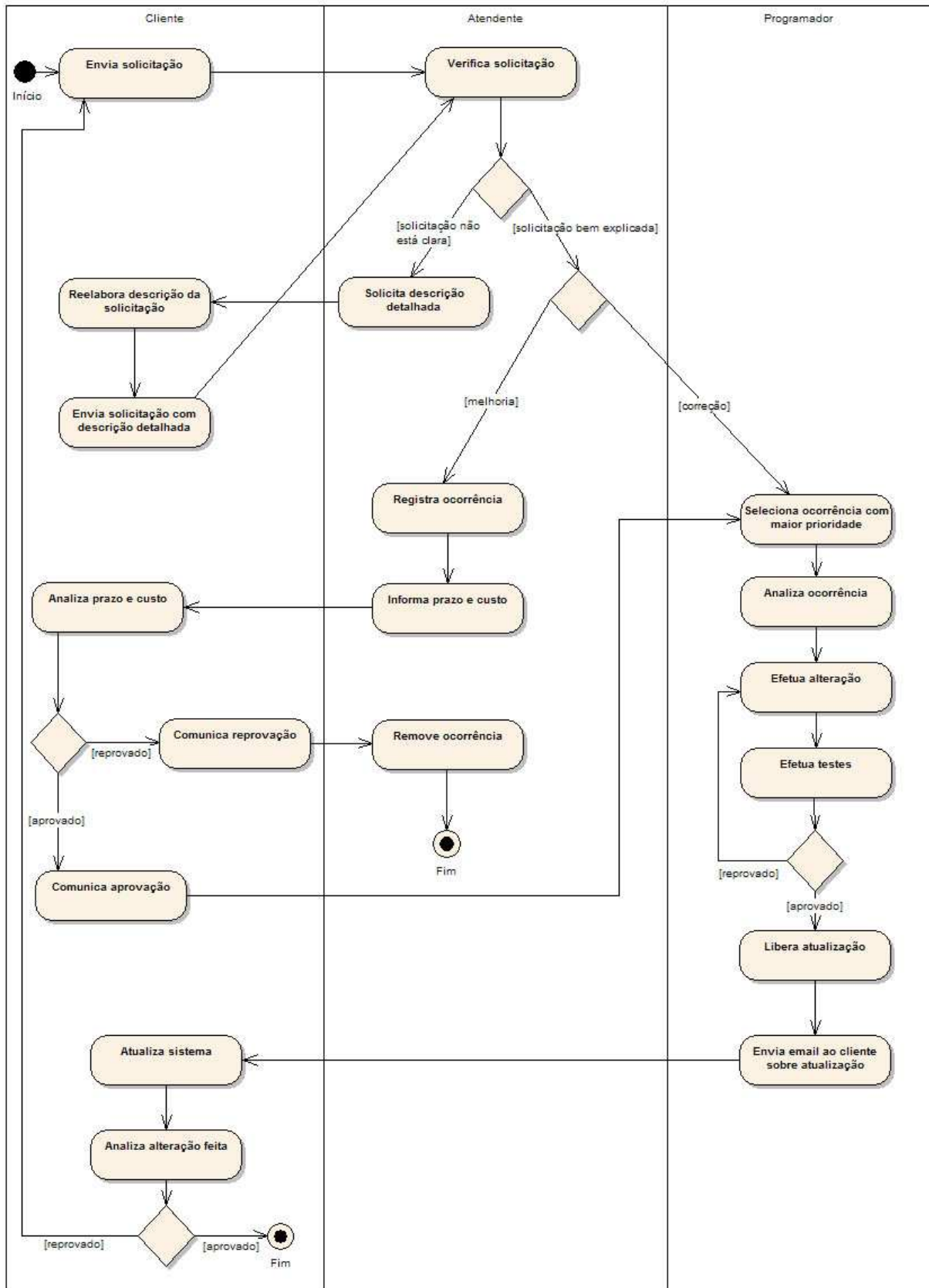


Figura 10 – Diagrama de atividades anterior à implantação

A seguir é descrita a nova forma de trabalho após a implantação da ferramenta aqui desenvolvida.

Quando algum cliente necessita de alguma customização no ERP, ele deve utilizar a ferramenta desenvolvida neste trabalho, que é distribuída juntamente com o sistema comprado. Nela ele descreve o que está necessitando e preenche algumas questões padrões que são geralmente feitas pelo pessoal do suporte. Isso acontece, pois as solicitações que são recebidas sem a utilização desta ferramenta vêm geralmente incompletas e sem muitos detalhes. Desta forma é complicado se fazer uma análise para que a alteração a ser realizada realmente atenda às necessidades do próprio cliente.

Além disso, o cliente deve informar em qual tela, chamada no sistema de aplicativo do ERP, a alteração deve ser realizada, para que o registro seja feito de forma mais detalhada possível e não ocorram erros de interpretação.

Existe também a possibilidade de se registrar uma solicitação de suporte técnico, que não implicará, num primeiro momento, em alguma alteração no sistema. Este tipo de solicitação possui uma prioridade maior e deve entrar na agenda do pessoal responsável pelo suporte para que os mesmos façam o atendimento necessário.

Caso o que foi requisitado implique em uma alteração no ERP, a análise deve ser realizada pelos analistas de sistemas, cargo que hoje não existe na empresa, para que os mesmos façam uma avaliação completa sobre a viabilidade, possibilidade, custos e objetivos da alteração. No entanto, se houver algum problema de interpretação do serviço, a pessoa responsável recorre a esta ferramenta para entrar em contato com o cliente e fazer os devidos questionamentos.

Além disso, não se pode mais fazer tudo da forma que o cliente deseja, mas sim, deve-se fazer de tal forma que fique o ideal, ou seja, é preciso analisar o que será afetado. Para se elevar o nível de qualidade do sistema, é necessário fazer de forma que fique correto e não

implique em não conformidades no ERP. Por este motivo, as alterações levam um tempo maior para serem realizadas, mas com isso eleva-se a qualidade do sistema.

De acordo com o resultado desta análise, a solicitação é encaminhada para o desenvolvimento ou para o suporte dar um retorno ao cliente.

No entanto, caso o registro que foi feito trate de um suporte ao cliente, ele possui uma prioridade maior e é encaminhado diretamente a um responsável do setor correspondente. A pessoa que receber a solicitação pode comunicar-se com o solicitante e resolver o seu problema e/ou dúvida.

Para fazer o atendimento ao usuário, utiliza-se o recurso da comunicação textual, que permite ao atendente trocar informações com o cliente através da troca de mensagens. Além disso, a ferramenta desenvolvida neste trabalho permite a conversação por voz com o uso da tecnologia VoIP que pode facilitar e agilizar a comunicação. Com a conversação existe a informalidade que não chega a ser um problema devido ao registro por escrito prévio da solicitação. Além do mais, é estudada uma forma de se gravar as conversações para se manter um histórico na empresa.

Ao término da etapa de análise e atendimento, o fluxo pode seguir diferentes caminhos. Caso a solicitação tenha sido aprovada por algum analista, ele segue para a agenda do desenvolvimento. Quando chegar o momento da execução do serviço, o programador receberá o requisito bem detalhado, com explicações de como implementar. Além disso, ele é informado sobre as regras de negócio aplicadas na solicitação. Todas estas atitudes são tomadas para que não ocorram erros de interpretação do requisito, que resultaria em falhas no programa e descontentamento por parte do solicitante.

Ao término da implementação, o tempo gasto é contabilizado e a alteração passa pelas fases de teste e documentação para somente então ser liberada ao cliente, que recebe um e-mail de aviso sobre a atualização.

No entanto, caso a solicitação tenha sido encaminhada ao suporte para atendimento ao usuário e já tenha sido encerrada, é contabilizado o tempo gasto e é enviado um e-mail de resposta ao cliente.

Na Figura 11 pode ser visualizado através do diagrama de atividades a nova forma de trabalho.

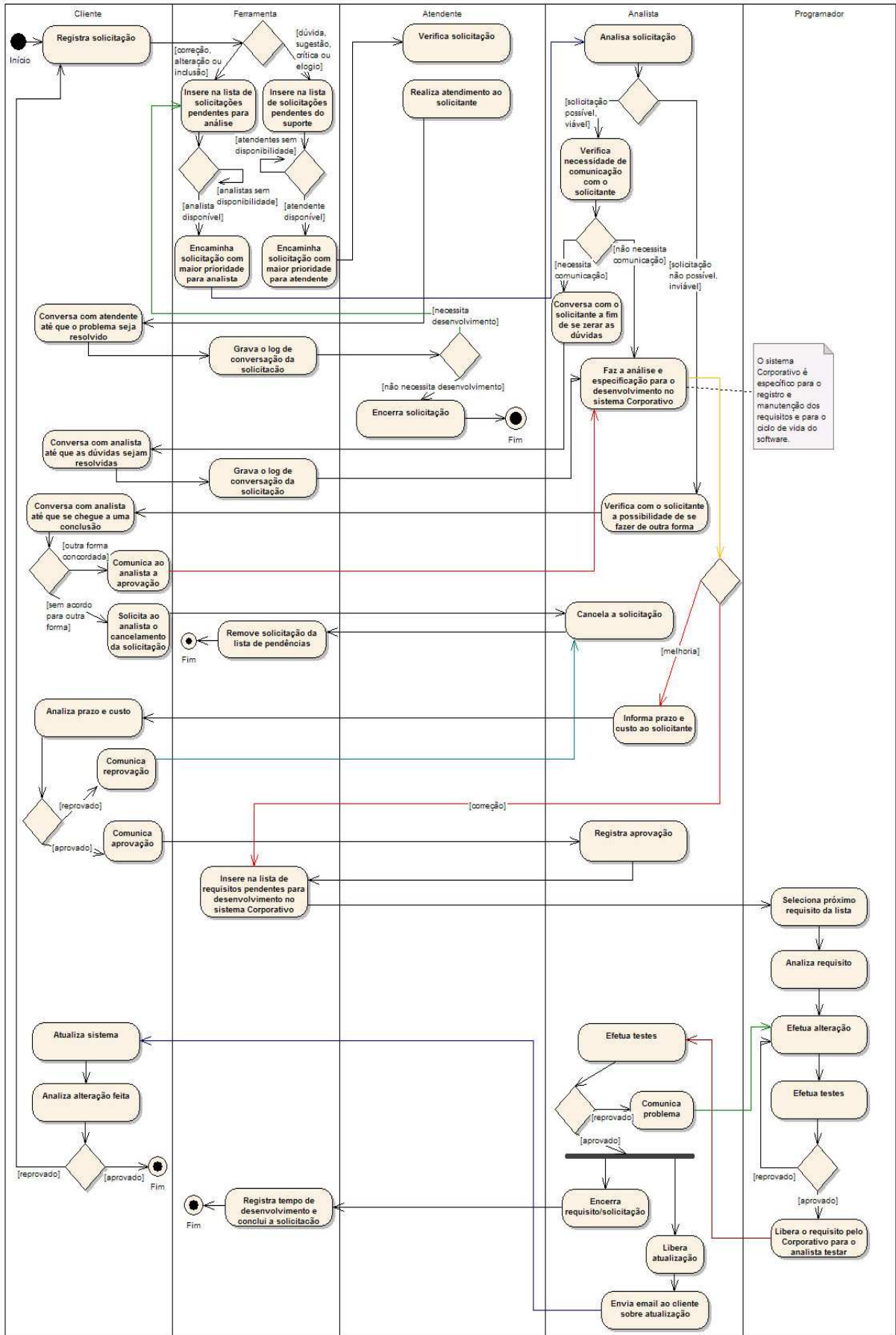


Figura 11 – Diagrama de atividades posterior à implantação

Após a análise das duas figuras, pode-se perceber que além de ser possível fazer uma comunicação textual e por voz com o cliente, também se tem um maior controle das solicitações, uma vez que os textos da comunicação ficam registrados no banco de dados. Fato que não acontecia antes da implantação desta ferramenta, já que se utilizava o *MSN Messenger* e o processo de documentação e registro do tempo em atendimento era manual.

Além disso, nota-se um aumento na estrutura da empresa com o aparecimento de um analista no processo. Seu objetivo é de aumentar ainda mais a qualidade das alterações que são realizadas nos sistemas da empresa. Este é mais um usuário da ferramenta, que pode estar utilizando-a para entrar em contato com o cliente e levantar outros requisitos ou até mesmo para detalhar requisitos mal descritos.

Com estas modificações no processo de trabalho da empresa, impulsionadas pela implantação desta ferramenta, destaca-se o avanço que ela traz em todos os aspectos. Tanto no fator organizacional, com o aumento da estrutura interna da empresa e com uma melhor divisão das tarefas, como por exemplo, a análise das solicitações que passa a ser realizada por uma pessoa específica quanto no fator que diz respeito à otimização.

A utilização da comunicação por voz deixa o processo de troca de informações mais ágil e veloz. Os requisitos da solicitação ficam gravados em registro textual e a conversação se torna eficaz para o seu detalhamento e resolução de dúvidas. Como a comunicação VoIP torna o processo mais informal, estuda-se a possibilidade de se gravar as conversas para se manter o registro. Mas enquanto isso não é feito, é possível registrar textualmente, ao final de um atendimento por voz, um resumo do que foi dialogado.

3.3.3 Diagrama de classes

O terceiro diagrama elaborado neste trabalho é o de classes. De acordo com Guedes

(2004, p. 27), este é o diagrama mais utilizado e o mais importante da UML pois define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos possuídos por cada classe. Isto quer dizer que tudo que acontece na ferramenta utiliza como base as classes, que quando são instanciadas transformam-se em objetos que podem ser manipulados para, por exemplo, enviar uma mensagem de texto para outra pessoa. Este ato ocorre quando se invoca um método de um objeto. Nele é implementado o que exatamente deve ser feito e o que deve acontecer.

Na Figura 12 é demonstrado o diagrama de classes com os seus métodos e atributos relacionados.

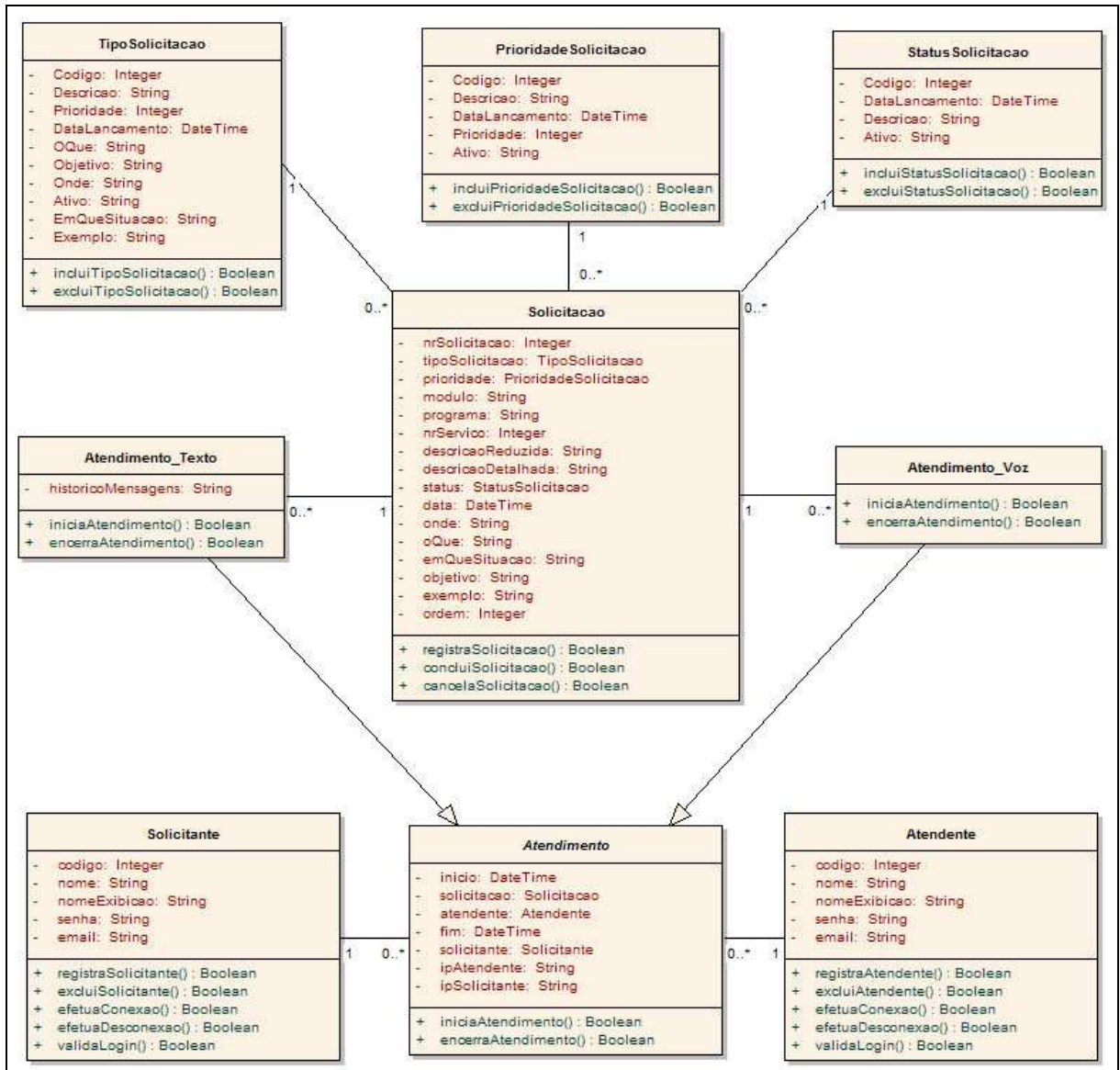


Figura 12 – Diagrama de classes

As classes utilizadas nessa ferramenta são as seguintes:

- “tiposolicitação”: identifica o tipo da solicitação, que pode ser “Correção”, “Melhoria”, “Sugestão”, entre outras. Esta classe também é utilizada para configurar determinados campos da solicitação como obrigatórios ou não, dependendo do tipo da solicitação selecionada;
- “prioridadesolicitação”: identifica a prioridade da solicitação, que pode ser “Urgente”, “Importante”, “Necessário” e “Desejável”. Ela indica a ordem de atendimento da solicitação;
- “statussolicitacao”: identifica o status da solicitação, ou seja, como está o seu

andamento, e pode assumir os seguintes valores “Em aberto”, “Em atendimento”, “Em análise”, entre outros;

- d) “solicitacao”: utilizada para armazenar as informações sobre as solicitações dos usuários;
- e) “atendimento_texto”: utilizada para o registro das mensagens de texto que foram enviadas e recebidas nos atendimentos a uma determinada solicitação;
- f) “atendimento_voz”: utilizada para iniciar ou encerrar o atendimento por voz aos clientes;
- g) “solicitante”: utilizada para armazenar as informações sobre o usuário que registrou a solicitação, bem como código, nome, e-mail, entre outras;
- h) “atendimento”: utilizada para registro dos atendimentos efetuados para uma solicitação, com informação de data e hora de início e término, por exemplo;
- i) “atendente”: utilizada para armazenar as informações sobre o atendente da solicitação, ou seja, o funcionário do suporte ou o analista que está realizando o atendimento ao solicitante.

3.4 IMPLEMENTAÇÃO

A ferramenta proposta neste trabalho foi implementada na linguagem de programação Borland Delphi. Para tanto, foi utilizado o ambiente de desenvolvimento Borland Delphi 7 e o banco de dados Firebird 1.5.

3.4.1 Técnicas e ferramentas utilizadas

Para efetuar a implementação desta ferramenta foi feita a análise e o estudo detalhado

dos requisitos, bem como dos diagramas de casos de uso, de atividades e de classes. Com base nestes diagramas, conseguiu-se levantar as reais necessidades da empresa e que forma seria a mais aconselhável para realizar a implementação.

A divisão em camadas apresenta vantagens significativas em relação ao desenvolvimento de uma ferramenta de apoio ao suporte. Existem alguns pontos que foram determinantes para esta conclusão, como o fato da maior parte da lógica do negócio ser mantida no servidor de aplicação.

Além disso, como se trata de uma ferramenta para suporte, onde deve existir uma lista de usuários e clientes conectados, utiliza-se a mesma arquitetura sem a necessidade de se criar outra em paralelo para apenas permitir a comunicação textual entre os usuários. Desta forma, a arquitetura do servidor de aplicação é utilizada tanto para manter a lógica do negócio como para permitir a comunicação entre atendente e cliente.

Mais precisamente, quando o usuário efetua o *login* através da aplicação cliente (*frontend*), a mesma se conecta ao servidor de aplicação, que faz a autenticação do usuário com sua senha. Caso isto aconteça com sucesso, é enviada à aplicação cliente a lista de usuários registrados com as informações sobre os que estão conectados no momento para que o usuário tenha conhecimento de quais contatos estão disponíveis para uma comunicação instantânea e quais os contatos que estão desconectados. No entanto, a comunicação com usuários desconectados é possível, uma vez que as mensagens enviadas a eles são armazenadas no banco de dados para posterior envio quando se conectarem.

A técnica chamada de *Callback* foi utilizada para que o servidor consiga chamar procedimentos remotos nas aplicações clientes. Isto significa que elas não precisam possuir *timers* para verificação de novas mensagens no servidor ou para atualizar a lista de usuários conectados. Com isto, o servidor é o responsável por informar quando um novo usuário conecta e também por enviar mensagens ao mesmo.

Existem diversas vantagens da utilização de *Callbacks*. Uma delas é a utilização da banda de conexão somente quando necessário, fato que não acontece quando se utilizam *timers*, já que os mesmos devem verificar a intervalos regulares de tempo se existe alguma alteração no servidor.

Quando um usuário envia uma mensagem, a aplicação cliente chama uma função remota no servidor de aplicação, que será responsável por registrar a mensagem enviada no banco de dados e por enviá-la ao destinatário através de uma *Callback*.

Um exemplo de envio de mensagem de texto pode ser visualizado no Quadro 1.

```

try
  vStrEnvio := dmAssistenteCC.SocketConnection.AppServer.
              rfncEnviaMensagem(vStrUsuario, vStrContato, jvreMensagem.Text);
except
end;

if (vStrEnvio <> '') and (Pos('usuário está desconectado', vStrEnvio) = 0) then
begin
  cdsComunicacao.Cancel;
  Mensagem.Exibir(vStrEnvio);
end
else begin

  cdsComunicacao.Post;

  jvdbreHistorico.SelLength := 0;
  jvdbreHistorico.SelStart := jvdbreHistorico.GetTextLen;
  jvdbreHistorico.Perform(EM_SCROLLCARET, 0, 0);

  jvreMensagem.Clear;
  jvreMensagem.SetFocus;

  if (Pos('usuário está desconectado', vStrEnvio) <> 0) then
    Mensagem.Exibir(vStrEnvio);

end;

```

Quadro 1 – Envio de mensagens

No quadro acima tem-se o trecho responsável por enviar uma mensagem através da execução da função remota *rfncEnviaMensagem*, que possui como parâmetros, o remetente, destinatário e a mensagem a ser enviada. Ela foi definida e implementada no Servidor de Aplicação para que o texto passado como parâmetro seja enviado ao destinatário.

A função é executada remotamente através do uso da tecnologia de multicamadas, onde o Servidor de Aplicação roda em conjunto com um Servidor de *Socket* que é responsável por monitorar as conexões em uma determinada porta.

Caso ocorra algum problema ao enviar a mensagem, como por exemplo o destinatário estar desconectado, o usuário é informado, já que a função remota retorna a mensagem do erro ocorrido.

No lado do servidor existe a lista de usuários conectados e para cada conexão existe uma *thread*. Em cada *thread* são armazenadas informações sobre o usuário e sobre o conjunto de dados que o mesmo está manipulando. Quando o método *rfncEnviaMensagem* é chamado, ele se encontra na *thread* do remetente, cabendo ao servidor encontrar a *thread* do destinatário e executar um método nela, que é responsável por enviar via *Callback* a mensagem à aplicação cliente.

Um exemplo da rotina de envio de mensagem de texto na *thread* do servidor pode ser visualizada no Quadro 2.

```

with frmPrincipal.Servers.LockList do
begin
    try

        //Localizar a thread do destinatário e enviar a mensagem pela callback
        vDestinoRemoto := getRemoteDataModule(pUsuarioDestino);

        if Assigned(vDestinoRemoto) then
        begin

            try

                //Gravar no log
                cdsInserereComunicacao.Close;
                cdsInserereComunicacao.Open;
                cdsInserereComunicacao.Append;
                cdsInserereComunicacao.FieldName('CONTADOR').AsInteger := 0;
                cdsInserereComunicacao.FieldName('CONTATOORIGEM').AsString := pUsuarioOrigem;
                cdsInserereComunicacao.FieldName('CONTATODESTINO').AsString := pUsuarioDestino;
                cdsInserereComunicacao.FieldName('DATA').AsDateTime := now;
                cdsInserereComunicacao.FieldName('HISTORICO').AsString := pMensagem;
                cdsInserereComunicacao.Post;
                cdsInserereComunicacao.ApplyUpdates(0);

            except
            end;

            vDestinoRemoto.EnviarMensagem.UsuarioOrigem := pUsuarioOrigem;
            vDestinoRemoto.EnviarMensagem.UsuarioDestino := pUsuarioDestino;
            vDestinoRemoto.EnviarMensagem.Mensagem := pMensagem;

            vDestinoRemoto.TimerEnviaMensagem.Enabled := True;
        end
    end
end

```

Quadro 2 – Envio de mensagens (servidor)

Como as *threads* concorrem entre si ao processador, é necessário implementar um controle para que somente uma delas esteja executando regiões críticas, que são trechos de códigos que não podem ser executados ao mesmo tempo por mais de uma *thread* pois manipulam informações disponíveis a todas elas. Para se resolver tal questão utiliza-se o conceito de *lock* que garante que somente uma *thread* está executando determinado trecho do código até que seja chamado o método *unlock*.

É importante mencionar que somente a *thread* pertencente ao destinatário pode executar a *Callback*. Isto quer dizer que, uma *thread* não pode invocar um método via *Callback* de outra *thread* sendo então necessário que a *thread* ligada ao mesmo usuário faça tal trabalho. A única forma para que um procedimento seja originado em uma *thread* a partir de outra é a utilização de um *timer*.

O *timer* é ativado pela *thread* do remetente na *thread* do destinatário e quando é executado existe um procedimento responsável por enviar a mensagem do remetente via

Callback para o destinatário. Ao receber a mensagem o *frontend* exibe uma janela para comunicação com o remetente.

O objeto de *Callback* é instanciado no momento da conexão do usuário, através da chamada do procedimento remoto *prcSetCallBack* do *front end* no Servidor de Aplicação. O procedimento possui um parâmetro, que é uma referência ao objeto da *Callback* no *front end*. Desta forma, o Servidor de Aplicação pode executar procedimentos remotos pela *Callback* no *front end*.

Um exemplo da rotina do *timer* pode ser visualizado no Quadro 3.

```

TimerEnviaMensagem.Enabled := False;
TTimer(Sender).Enabled := False;

with frmPrincipal.Servers.LockList do
begin

    try

        TimerEnviaMensagem.Enabled := False;
        TTimer(Sender).Enabled := False;

        if (VarType(FCallback) <> varEmpty) and (VarType(FCallback) <> varNull) then
            try
                Self.FCallback.fncRecebeMensagem (EnviarMensagem.UsuarioOrigem,
                    EnviarMensagem.UsuarioDestino, EnviarMensagem.Mensagem);
            except
                end;

        finally
            frmPrincipal.Servers.UnlockList;
        end;

    end;
end;

```

Quadro 3 – Envio de mensagens (servidor)

Na rotina acima, a primeira instrução executada é a desativação do timer, para que o mesmo não seja executado mais de uma vez, ocasionando o re-envio da mensagem ao destinatário. Em seguida, é feito um *lock* na lista de *threads* existentes para que não ocorra conflito de *threads* executando este mesmo trecho de código. Após o *lock* é enviada a mensagem via *Callback* ao *frontend* do destinatário. E por fim, é feito o *unlock* para liberar a lista e permitir que outras *threads* possam executar o mesmo procedimento.

Para incluir a comunicação por voz na ferramenta foi utilizado o controle *ActiveX* da empresa Conaito, que demonstra uma qualidade de áudio excelente com uma baixíssima taxa de transferência, com cerca de 3 ou 4Kb/s. O controle *ActiveX* é pago e a versão utilizada neste trabalho é válida por 30 dias para testes. Além disso, o *ActiveX* não tem seu código fonte liberado para análise e o protocolo utilizado em sua implementação é proprietário.

Foi implementado no trabalho o sistema de detecção de presença de áudio, para utilizar uma taxa de transferência muito menor quando não se está transmitindo áudio efetivamente na comunicação. Neste caso, são transmitidos apenas comandos de controle e a taxa de utilização é de cerca de 0,5Kb/s.

Para que a comunicação por voz funcione, o servidor de voz foi instalado no próprio servidor de aplicação. Desta forma, quando o servidor de aplicação é instanciado, o servidor de voz é inicializado e já está apto a receber conexões de usuários.

Assim que algum usuário se conectar ao servidor, também é feito o seu registro no servidor de voz sobre IP, permitindo dessa forma o usuário efetuar e receber ligações de voz.

Cada usuário pode conversar com apenas um outro usuário, então caso alguém efetue uma chamada a um contato ocupado, é exibido um aviso informando que o mesmo está ocupado e não pode conversar no momento porque já está conversando com outro usuário.

Assim que o usuário desconectar, é desfeito o seu registro no servidor de voz sobre IP, retirando seu nome da lista de usuários conectados e disponíveis para conversações.

3.4.2 Operacionalidade da implementação

Ao executar a ferramenta, é apresentada a tela de *login* para autenticação do usuário com sua senha. O usuário preenche seu *login*, senha e seu estado inicial, que pode ser disponível ou indisponível.

A tela de login pode ser visualizada na Figura 13.



A imagem mostra uma janela de login intitulada "Assistente On-Line Dessim - Login". No topo, há um ícone de um globo terrestre. Abaixo dele, o título "Assistente On-Line Dessim" é exibido em uma fonte azul. O formulário de login contém três campos: "Usuário:" com o texto "Jean Carlos" inserido, "Senha:" com um campo em branco, e "Status:" com o menu suspenso "Disponível" selecionado. Na base da janela, há dois botões: "Entrar" com um ícone de checkmark verde e "Sair" com um ícone de X vermelho.

Figura 13 – Tela de login na ferramenta

Após a autenticação do usuário com sua senha no servidor, é exibida a tela principal da ferramenta. Nela consta a lista de usuários cadastrados, sendo que usuários desconectados aparecem na cor vermelha, usuários indisponíveis aparecem na cor preta e usuários disponíveis aparecem na cor verde. Além disso, existem alguns botões de rápido acesso, como o botão para registro de solicitações e o botão para controle das solicitações já cadastradas. É exibido também no canto direito da tela uma imagem indicativa de conexão com o servidor. Caso o *plug* de tomada esteja com um “x” vermelho, significa que foi perdida a conexão com o servidor.

A tela principal da ferramenta pode ser visualizada na Figura 14.



Figura 14 – Tela principal

A partir da tela principal é possível comunicar-se com outros usuários com apenas um duplo clique em seus nomes. Ao fazer isto é aberta uma tela para comunicação entre os usuários da mesma empresa. Para enviar mensagens ao destinatário, basta digitar o conteúdo e pressionar a tecla *Enter* ou então clicar no botão “Enviar”.

Também é possível conversar via VoIP com o contato. Para tanto, basta pressionar o botão “Falar”. A partir deste momento é estabelecida uma conexão por voz entre os dois usuários. Assim que for desejado encerrar a conversa, basta clicar no botão “Parar” que aparece no lugar do botão “Falar”.

A tela para comunicação entre usuários pode ser visualizada na Figura 15.

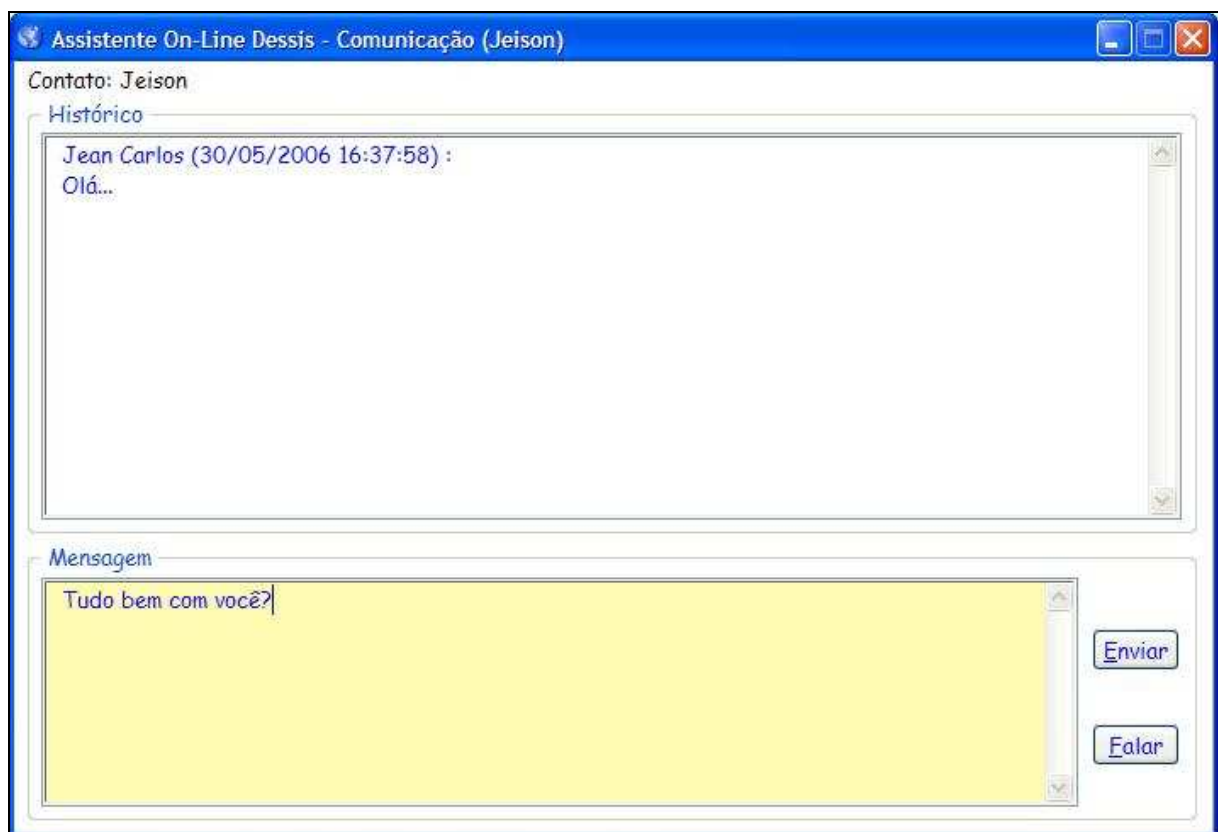


Figura 15 – Comunicação entre usuários

Para efetuar o registro de solicitações basta clicar no primeiro botão da barra de atalhos. O registro de solicitações é utilizado para que sejam cadastrados os pedidos de alterações nos sistemas da empresa, funcionando como ordens de serviço. Nele, o usuário preenche o tipo da solicitação, a prioridade, o programa relacionado, e algumas descrições para um maior entendimento por parte dos atendentes ou analistas da empresa.

A tela para o registro de solicitações pode ser visualizada na Figura 16.

Assistente On-Line Dessis - Registro de Solicitação

Solicitação

Cliente: Teste Contato: []

Tipo: Correção Prioridade: Importante (1)

Programa: FIN-04-I-020 Condições de Pagamento: []

O que: Erro no programa

Onde: No botão SALVAR do cadastro de condições de pagamento

Em que situação: Quando clico no botão SALVAR

Objetivo: Corrigir o erro no programa

Descrição: Erro ao cadastrar condições de pagamento

Detalhes: Não consigo cadastrar novas condições de pagamento. Apresenta uma mensagem de erro.

Exemplo: Abrir o cadastro de condições de pagamento, clicar em novo, preencher os dados e clicar em Salvar.

Registrar Cancelar

Figura 16 – Registro de solicitações

Após o registro da solicitação ser efetuado com sucesso, o mesmo será visível através da tela de controle de solicitações que exibe todas que já foram registradas até o momento. Para acessar esta tela é necessário clicar no segundo botão da barra de atalhos do menu principal.

A tela para o controle de solicitações pode ser visualizada na Figura 17.

Assistente On-Line Dessis - Controle de Solicitações

Solicitações

Número	Tipo	Prioridade	Cliente	Status	Data solicitação
1	Inclusão	Importante (1)		Em aberto	27/02/2006 20:12:23
2	Alteração	Urgente (0)		Em aberto	15/05/2006 23:04:39
3	Alteração	Urgente (0)		Em aberto	15/05/2006 23:08:57
4	Sugestão	Urgente (0)		Em aberto	20/05/2006 11:36:23
5	Correção	Urgente (0)		Em aberto	21/05/2006 15:56:54
7	Sugestão	Urgente (0)		Em aberto	28/05/2006 10:51:53
8	Correção	Importante (1)	Teste	Em aberto	30/05/2006 21:43:56

Solicitante Programa Condições de Pagamento

Descrição

Detalhes

O que

Onde

Em que situação

Objetivo

Exemplo

Figura 17 – Controle de solicitações

Através da tela acima o atendente analisa as solicitações e pode fazer o atendimento instantâneo ao cliente. Para tanto, basta clicar no botão “Atender Solicitação” que abre uma nova tela para realizar o atendimento ao cliente. Nela são exibidas as informações da solicitação e a mesma permite a comunicação entre cliente e atendente a fim de se resolver o problema.

Caso se deseje conversar via VoIP com o cliente, basta pressionar o botão “Falar” para estabelecer uma conversação por voz entre cliente e atendente. Assim que for necessário terminar a conversa, basta pressionar o botão “Parar” que aparece no lugar do botão “Falar”.

A tela de atendimento à solicitação é demonstrada na Figura 18.

Assistente On-Line Dessim - Atendimento à Solicitação

Solicitação

1. Resumo 2. Detalhes

Número 8 Tipo Correção Prioridade Importante (1) Status Em aberto

Cliente Teste Solicitante Data 30/05/2006 21:43:56

Programa FIN04I020 Condições de Pagamento

Descrição Erro ao cadastrar condições de pagamento

Detalhes Não consigo cadastrar novas condições de pagamento. Apresenta uma mensagem de erro.

Atendimento

Histórico

Mensagem Olá... em que posso ajudá-lo?

Enviar Falar Encerrar atendimento

Figura 18 – Atendimento a solicitações

Ao fim do atendimento, basta clicar em Encerrar atendimento para marcar a hora de término e concluir este atendimento ao cliente.

Na parte do Servidor de Aplicação existe apenas uma tela com a lista de usuários conectados no momento que pode ser visualizada na Figura 19.

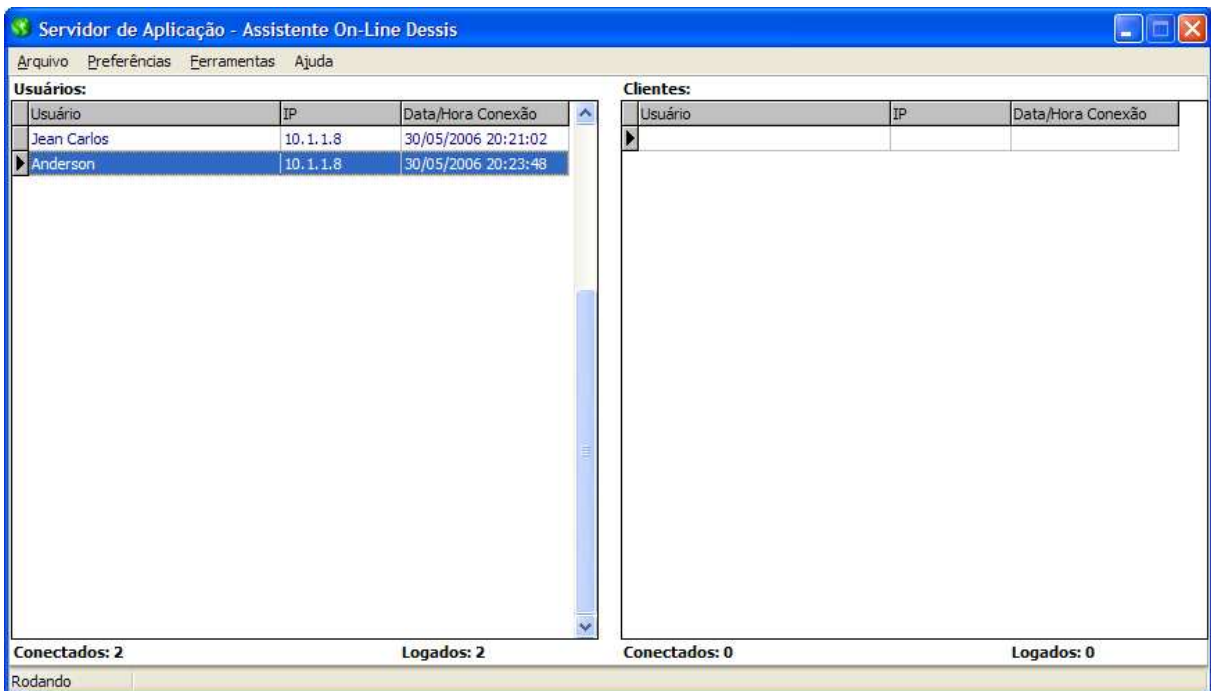


Figura 19 – Servidor de aplicação

Assim que os usuários desconectam da ferramenta, seu nome é retirado da lista de usuários logados e é registrada a sua desconexão no banco de dados.

3.5 RESULTADOS E DISCUSSÃO

Conseguiu-se com este trabalho aplicar conceitos de controle de acessos a regiões críticas, impedindo que uma área comum seja acessada por mais de uma *thread* ao mesmo tempo. Além disso, tornou-se possível implementar um sistema de comunicação textual entre usuários de forma mais adequada, sem o uso de *timers* para verificação de novas mensagens no servidor ou para atualização da lista de usuários conectados como é feito normalmente em aplicações comuns.

Os *timers* foram substituídos pela técnica de *Callback*, que possibilita que o servidor de aplicação execute procedimentos remotos no *frontend*, fato este que apresenta uma vantagem significativa já que só é utilizada a banda de conexão quando realmente existe alguma alteração, como por exemplo a conexão de um novo usuário.

Além disso, a estruturação em 3 camadas permite que os clientes se conectem ao servidor de aplicação remotamente pela internet através do protocolo TCP/IP.

Tentou-se implementar a comunicação por voz utilizando os protocolos padrões como o SIP e o RTP, no entanto a experiência com um protocolo proprietário demonstrou uma qualidade de transmissão e de voz inimagináveis, com o som da fala sem distorções e com baixa utilização de banda de conexão com a rede. As taxas giram em torno de 3 ou 4 Kb/s, possibilitando uma comunicação por voz até mesmo em conexões *dialup* com a internet.

O protocolo proprietário utilizado pertence à empresa Conaito *Technologies* e disponibiliza em um controle *ActiveX* a possibilidade de implementação de um ambiente com comunicação por voz.

4 CONCLUSÕES

Ao término deste trabalho foram alcançados parcialmente os objetivos propostos e foram descobertas novas possibilidades antes não previstas, como a utilização da técnica de *Callback* que permitiu realizar uma implementação mais perto do ideal sem *overhead*.

No entanto surgiram dificuldades não imaginadas inicialmente, como a concorrência entre *threads* que chegou a criar conflito quando conectou-se um grande número de usuários ao mesmo tempo e grande parte utilizando com frequência o recurso de envio de mensagens. Ocorreu de mensagens serem enviadas trocadas, pois inicialmente não foi prevista esta situação. Mas tão logo o problema foi encontrado, sua solução foi desenvolvida com o uso da classe *TThreadList* que possui uma lista de *threads* e permite efetuar um *lock* de modo a bloquear o acesso de outras *threads* a um mesmo trecho de código enquanto não seja executado o método *unlock*.

A implementação da comunicação por VoIP utilizando um protocolo padrão não foi realizada e foi levantada a causa. A utilização da linguagem de programação Delphi não se mostrou adequada, pois tornou o processo de busca por uma biblioteca gratuita para VoIP com SIP mais demorado, já que a maior parte das bibliotecas encontradas eram para a linguagem C++ e também para Visual Basic (VB).

No entanto foi utilizado o controle *ActiveX* da empresa Conaito para incluir na ferramenta a comunicação por voz entre os usuários. Desta forma, não foi utilizado um protocolo padrão, mas sim um protocolo proprietário que atendeu muito bem as expectativas e possibilitou atender um dos objetivos desta ferramenta.

O controle *ActiveX* não é gratuito e a versão paga não é liberada com os fontes para apreciação do funcionamento. Para este trabalho utilizou-se a versão *trial* com um mês para testes.

O assunto deste trabalho tornou-se mais complexo do que o esperado e resolveu-se não dispensar muito tempo com a busca de uma biblioteca para VoIP com SIP e adiantar outras questões importantes, como a estruturação da ferramenta em 3 camadas, a comunicação textual entre usuários e ainda a parte do registro de solicitações que são a base de um sistema para suporte.

O presente trabalho é de grande contribuição para a empresa, prova disto é a já utilização dele em fase de desenvolvimento e testes. Com ele é possível realizar comunicações entre atendente e cliente com o registro das conversas e sem distrações que são permitidas pelo uso do *MSN Messenger*, já que o mesmo possibilita a troca de *emoticons* ou carinhas animadas, cujo único objetivo em um ambiente de trabalho é distrair.

Outra ferramenta existente que está relacionada com a presente ferramenta é o Skype, que também possibilita a comunicação textual e por voz entre os usuários. No entanto a utilização do Skype em um ambiente de suporte a clientes apresenta algumas desvantagens, como a não contabilização das horas automaticamente e o não controle rigoroso do que é conversado por algum supervisor.

4.1 EXTENSÕES

Existem alguns pontos que podem ser melhorados em futuras extensões para este trabalho:

- a) implementar as rotinas que executam procedimentos ou funções remotas em *threads* isoladas, para não haver o congelamento de toda a aplicação caso os procedimentos remotos não estejam acessíveis por uma falha durante a comunicação;
- b) implementar o *broadcast* e o *multicast* de mensagens, ou seja, possibilitar o envio

de uma mesma mensagem a todos ou vários usuários ao mesmo tempo, dispensando então a necessidade de o usuário ter que enviar a mesma mensagem para cada destinatário individualmente;

- c) futuras implementações de ambientes de comunicação instantânea *online* poderão também ser realizadas utilizando-se como base o presente trabalho, que possui um modelo muito interessante e muito vantajoso, como já foi mencionado. E também, outros alunos poderão utilizar-se da mesma técnica de *Callback* implementada neste trabalho para otimizar suas aplicações, servindo-lhes como base para pesquisa e consulta;
- d) sugere-se como uma boa continuação para este trabalho também o desenvolvimento de uma comunicação por voz com SIP a fim de se fazer comparativos de qualidade e performance das comunicações entre o padrão proprietário e o padrão livre. Tais comparativos podem ser utilizados por futuros desenvolvedores de aplicações VoIP terem parâmetros no momento da escolha do protocolo a ser utilizado em seus trabalhos;
- e) localizar e utilizar uma biblioteca padrão livre para VoIP na implementação da comunicação por voz entre os usuários.

REFERÊNCIAS BIBLIOGRÁFICAS

CONAITO TECHNOLOGIES. Conaito VoIP SDK. [Alemanha], [2006]. Disponível em: <<http://www.conaito.com>>. Acesso em: 03 jun. 2006.

GUEDES, Gilleanes T. A. **UML: uma abordagem prática**. São Paulo: Novatec, 2004.

GURLE, David; HERSENT, Oliver; PETIT, Jean-Pierre. **Telefonia IP: comunicação multimídia baseada em pacotes**. Tradução Adriano Vilela Barbosa. São Paulo: Makron Books, 2002.

Internet Engineering Task Force. **RFC 2543: Session Initiation Protocol**. New York, 1999. 153p.

LIMA, Adilson da Silva. **UML 2.0: do requisito à solução**. São Paulo: Érica, 2005.

MARTINS, Joberto Sérgio Barbosa. Qualidade de serviço (QoS) em redes IP: princípios básicos, parâmetros e mecanismos. **JSMNet Networking Reviews**, Salvador, n. 1, set. 1999. Disponível em: <<http://scholar.google.com/url?sa=U&q=http://www.nuperc.unifacs.br/RT-NUPERC-2000-3p.pdf>>. Acesso em: 10 set. 2005.

NAZARIO, Débora Luciani. **Protótipo de um sistema de telefonia IP para lans baseado no padrão SIP**. 2003. 47 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PÉRICAS, Francisco A. **Redes de computadores: conceitos e a arquitetura internet**. Blumenau: Edifurb, 2003.