

## INTRODUÇÃO

O surgimento da Educação a Distância on-line, provocou uma grande mudança no ambiente educacional, promovendo novos métodos de ensino, maior difusão do conhecimento e ampliando horizontes na busca de novos adeptos e maior aceitação dessa metodologia de aprendizagem. Tem como objetivo aproximar alunos e professores localizados em lugares e regiões distintas, enfatizando o auto-aprendizado do aluno, através de ferramentas tecnológicas que facilitem essa interatividade.

A videoconferência é a tecnologia criada para transmissão de áudio e vídeo em tempo real. Tornou-se grande ferramenta de comunicação para ser integrada em diversas aplicabilidades e funcionalidade em diversos ambiente de ensino, como por exemplo, escolas, faculdades e outros.

Com a aplicação da videoconferência nas aulas do ensino médio, vai ocorrer uma diferenciação na metodologia de ensino, capaz de abranger grande potencial competitivo, criando aulas interdisciplinares envolvendo uma maior quantidade de alunos, garantindo uma maior interação e dinâmica do ensino. Com o auxílio de ferramentas tecnológicas que encurtem distancias, é possível proporcionar economias e vantagens para os integrantes deste modelo.

Foi realizada uma pesquisa no ambiente de ensino do colégio ISO – Instituto Social Objetivo, tendo como público os alunos e professores do ensino médio que participaram na avaliação da ferramenta chamada JMF (*Java Media Framework*), com intuito de analisar, comparar e disponibilizar os seus resultados, com o foco no desenvolvimento de novas aplicações multimídia customizadas a partir desse framework.

O JMF foi escolhido após a análise das vantagens, através de um estudo comparativo com outras ferramentas, onde sua principal funcionalidade é a captação, processamento e transmissão de áudio e vídeo em tempo real. O JMF foi utilizado nos testes para comunicação entre duas salas de aula e foi escolhida a conexão ponto-a-ponto para sua transmissão.

O JMF provê suporte aos principais formatos e codecs de áudio e vídeo, sendo que podemos com a API reutilizar as funcionalidades para o desenvolvimento de novas aplicações.

A ferramenta de videoconferência Ekiga foi escolhida a partir de análise comparativas com outros aplicativos, onde é uma solução gratuita e de boa portabilidade, com o objetivo de comparar com o aplicativo JmStudio, verificar todos os recursos úteis em EAD e com a finalidade idealizar futuras implementações e desenvolvimento de novas soluções com o apoio do framework JMF.

O JMF é o principal objeto de estudo deste projeto, nele podemos verificar os mecanismos capazes de realizar a videoconferência em Ensino a distancia, abordando os tipos de conexões, tráfegos de dados, padrões ou recomendações, protocolos de comunicação, além de toda estrutura e funcionamento do framework.

Com o uso do framework podemos obter as seguintes vantagens: redução de custos, maximização de reuso, menor manutenção, estabilização do código (menos defeitos), melhoria na compatibilidade e consistência.

## 1.1 Objetivo

O objetivo deste trabalho é testar conceitos e características da videoconferência - qualidade de áudio e vídeo, interface, usabilidade, aplicabilidade e outros - no Colégio ISO - Instituto Social Objetivo - com alunos do ensino médio, no auxílio das aulas dos professores.

O framework a ser estudado é o *Java Media Framework* (JMF) que possui boa portabilidade, além de ser adquirida gratuitamente. Será utilizada a ferramenta JmStudio desenvolvida com o JMF nos testes deste projeto.

Com o estudo e avaliação deste framework poderemos propor novas idéias para o desenvolvimento de novas aplicações para realização de videoconferência em EAD, que serão capazes de aperfeiçoar os recursos disponibilizados pelo JMF.

## 1.2 Justificativa

A EAD on-line tem por objetivo propiciar uma educação diferenciada, tanto como apoio às aulas presenciais quanto para suporte às aulas à distância, incentivando a interação e potencializando a aprendizagem.

Esse ambiente adquire importância na medida em que pode enriquecer as práticas atuais de ensino, promovendo um processo de ensino-aprendizagem mais dinâmico pelas novas formas de relação professor-aluno, ampliando a interação entre ambos para além da sala de aula.

Dessa forma, torna-se importante testar ferramentas tecnológicas de apoio à EAD.

### 1.3 Metodologia

Foi realizada uma pesquisa no Colégio ISO, uma entidade privada com duas filiais, localizada na cidade de Salvador, Bahia, no mês de outubro de 2009, com o objetivo de obter dados estatísticos sobre o framework JMF que serão importantes para o desenvolvimento de novas ferramentas. A amostra foi formada por alunos e professores do ensino médio.

Da população de 250 alunos do ensino médio do colégio ISO, foi selecionada uma amostra de 90 alunos e 4 professores que participaram de um processo avaliativo através de um questionário aplicado no final dos testes, contendo perguntas específicas no intuito de avaliar a ferramenta aplicada.

Estes questionários abordam pontos sobre a eficiência, qualidade de áudio e vídeo, desempenho, usabilidade, aplicabilidade, interface e outros sobre a ferramenta aplicada nos testes. Com esta pesquisa também foram coletadas algumas sugestões para melhorias desta ferramenta.

Após a coleta dos dados efetuou-se o tratamento estatístico e este resultado foi utilizado como base de conhecimento para o desenvolvimento deste trabalho. Os resultados desta pesquisa serão apresentados sob a forma de tabelas nesta monografia.

### 1.4 Organização

Esta monografia está distribuída em 6 capítulos, incluindo a introdução que tem por objetivo abordar o tema apresentado, bem como o desenvolvimento de uma solução que é a ferramenta sugerida para auxílio para geração de videoconferência em EAD.

No primeiro capítulo é realizada uma abordagem sobre a educação a distancia, com uma breve passagem dos seus histórico e conceitos até os dias atuais e a importância de seu surgimento.

O segundo capítulo tem como foco a videoconferência e sua estrutura, características, levantando os tipos de conexões, tráfegos, os padrões de recomendações para áudio e vídeo, codificação e decodificação com os codecs, a importância do desempenho com a qualidade de serviço e atenção quanto ao retardo de transmissão de dados e os protocolos de comunicação e transferência RTP (*REAL TIME PROTOCOL*) e RTCP (*PROTOCOLO DE CONTROLE DE TRANSPORTE EM TEMPO REAL*).

O terceiro capítulo apresenta as principais informações do framework JMF a ser trabalhado e pesquisado durante todo o projeto, detalhes das suas classes e toda sua estrutura, começando pela arquitetura, especificações da mídia de origem e destino, a forma que é feita a captura, localização, reprodução e transmissão da mídia.

O quarto capítulo aborda a ferramenta JMStudio, onde é disponibilizada como modelo de teste e apresentação do framework JMF.

No quinto capítulo serão mostrados os procedimentos dos testes realizados e os seus resultados detalhados.

No sexto capítulo serão mostrados os procedimentos dos testes e o estudo comparativo realizado seguido dos seus resultados detalhados com base na pesquisa desenvolvida no teste.

Por fim, temos a apresentação da conclusão e a proposta de trabalhos futuros como continuidade desta pesquisa.

## 2. EAD

A Educação a Distância on-line (EAD) consiste em uma modalidade de educação que utiliza a tecnologia de informação e de comunicação, tem a

finalidade de propor o auto-aprendizado, onde os alunos são incentivados a estudar e pesquisar de forma mais responsável e autônoma.

O surgimento da videoconferência aplicada em EAD acabou permitindo a interdisciplinaridade nas aulas e proporcionando uma maior interação entre professores e alunos. Também pode facilitar a troca de informações entre professores, alunos, pesquisadores e demais pessoas envolvidas no ensino à distância.

Com a aplicação da videoconferência nas escolas podem observar o aumento da interatividade na relação entre professores e alunos. A videoconferência realça a comunicação, criando um senso de presença física e permitindo uma comunicação diferenciada, podendo ser reconhecida até mesmo por gestos.

Hoje entendemos que o desenvolvimento atual da tecnologia favorece a criação e o enriquecimento das propostas na educação a distância na medida em que permite abordar de maneira ágil inúmeros tratamentos de temas, assim como gerar novas formas de aproximação entre docentes e alunos, e de alunos entre si. As modernas tecnologias resolvem o problema crucial da educação a distância, que é a interatividade (LITWIN, 2001, p.17).

Atualmente, devido à dificuldade em atender a ampla e diversa demanda de pessoas nas diferentes localidades foi necessário a adequação do processo educacional e o meio de comunicação dos professores para compartilharem o seu conhecimento para esta grande demanda.

Com o avanço tecnológico no sistema educacional, entre eles a videoconferência, utilizando da Internet como principal suporte para o ensino a distância, surgiram grandes serviços e foram criados critérios para sua implantação, dentro os quais: qualidade, velocidade e baixo custo.

A educação a distância, até então, caracterizada pela separação física entre o aluno e o professor ou a instituição de ensino, migra rapidamente para um conceito de aproximação ou mesmo integração virtual entre os agentes dos processos de ensino aprendizagem (VIANNEY, 2003).

Com isso, o futuro da educação a distância implicará em capacitar os professores, para que os mesmos possam usufruir da tecnologia de videoconferência, permitindo-se superar as dificuldades relacionadas ao tempo e o espaço físico, podendo dessa forma apresentar uma melhoria na qualidade de ensino.

### 3. VIDEOCONFERÊNCIA

Em abril de 1933, a sede da AT&T fez a primeira videoconferência pública para o laboratório Bell, na Cidade de Nova Iorque, com microfones e alto-falantes para transmissão de áudio, enquanto as imagens eram transmitidas e capturadas através de células fotoelétricas, sob luz azul. A partir disso, teve-se a importância da presença virtual a fim de encurtar as distâncias. (COOKBOOK, 2005).

Foram utilizados e apresentados sistemas de videoconferência ao público em 1964, pela AT&T, um produto chamado “*PicturePhone*”, onde se podia visualizar fotos sem movimento ao mesmo tempo em que se ouvia a voz do interlocutor. Porém este mesmo produto não foi bem aceito pelos seus clientes.

Na década de 70, Surgiram os sistemas do tipo “*freeze frame*” (congelamento da imagem da TV quadro a quadro) e “*slow motion*” (câmera lenta) com o intuito de trazer inovação, mas fracassou porque causava desconforto para os usuários.

Na década de 80, houveram mudanças significativas na videoconferência, com a entrada de técnicas de compressão de vídeo que possibilitaram um melhor gerenciamento da banda utilizada e diminuição do custo de processamento envolvido.

Já na década de 90, surgiram os sistemas de videoconferência utilizando computadores pessoais combinado com equipamentos, diminuindo assim as aparelhagens exclusivamente dedicadas à videoconferência.

Os sistemas de videoconferência realizam as transmissões através do tráfego de áudio e vídeo em tempo real com qualidade, permitindo a realização de videoconferência e trazendo uma comunicação interativa entre pessoas em locais diferentes.

A videoconferência tem sido utilizada para dar auxílio aos professores no suporte as aulas, reuniões e palestras, permitindo assim a integração entre os participantes.

Os recursos hoje incorporados aos sistemas de videoconferência permitem o compartilhamento de documentos (através de protocolos de transmissão de arquivos), transmissão de imagens gráficas (através de câmera

de documentos), utilização do Quadro Branco Eletrônico, oferecendo cada vez mais a esta tecnologia as condições necessárias para a realização das aplicações às quais ela se destina, com um nível elevado de interatividade e qualidade (GOMES, 2003).

Até pouco tempo atrás, os sistemas de videoconferência eram baseados em ISDN (*Integrated Services Digital Network*) que é um serviço de telefonia com mais recursos e digital, ou seja, suporta uma grande quantidade de dados, como áudio e vídeo. Atualmente, os mais utilizados são para os sistemas baseados em IP.

Os sistemas de videoconferência têm sido usados como apoio ao ensino e aprendizagem a distância. Os sistemas de videoconferência podem ser realizados em dois tipos de ambientes, segundo Leopoldino (2001):

- Videoconferência em estúdio: em salas especialmente preparadas com modernos equipamentos de áudio, vídeo e *codecs* (codificadores/decodificadores), para fornecer vídeo e áudio de alta qualidade para reuniões, palestras e cursos;

- Videoconferência em computador: em residência ou escritório, com computador pessoal equipado com hardware e software adequado. É mais barata que a videoconferência baseada em estúdio e mais apropriada para uso individual, ou para pequenos grupos.

Introduzir a videoconferência nos cenários educacionais tornou-se uma vantagem competitiva para as escolas, mas, o foco principal deve ser o aluno e não a tecnologia.

Os sistemas de videoconferência apresentam recursos bastante úteis, tais como:

- Áudio: é um recurso de comunicação essencial existente nos sistemas de videoconferência.
- Vídeo: utilizado pelos usuários da videoconferência para visualizar um ao outro, sendo comunicando-se também através de gestos e podendo exibir objetos a outros integrantes da conferência.
- Chat: permite que os usuários possam trocar mensagens.
- Whiteboard: é uma área de desenho que permite que os usuários possam importar imagens gráficas ou fazer anotações.

- Transferência de arquivos: é o recurso que permite aos usuários disponibilizar textos ou programas, possibilitando o envio e recebimento deste arquivo aos outros participantes da conferência.
- Compartilhar aplicações: permite que os participantes de uma conferência visualizem as aplicações de outro usuário, sendo possível a partir de seu computador controlar a execução dessa aplicação.

### 3.1 Tipos de conexão em videoconferência

Para a transferência dos dados foram criados os tipos de conexões, as conexões Ponto-a-Ponto e Multiponto. Na utilização da conexão ponto-a-ponto, quando houver comunicação entre dois terminais e a Multiponto entre dois ou mais terminais.

#### 3.1.1 Conexão Ponto-a-Ponto

A conexão é ponto-a-ponto, que também é chamada “um-para-um” é muito utilizada nas transmissões de dados, onde um terminal se conecta diretamente com outro e trocam dados entre si. Pode ser utilizada em reuniões, aulas, entre outros.

A figura 1 mostra como funciona a comunicação entre dois terminais com a conexão ponto-a-ponto, onde o fluxo de dados se torna único.

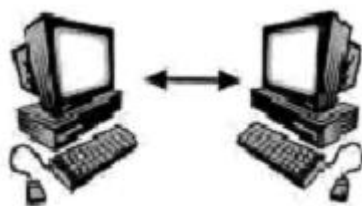


Figura1: Exemplo de Conexão Ponto-a-Ponto.

#### 3.1.2 Conexão Multiponto

A conexão multiponto permite que vários pontos sejam interligados ao mesmo tempo, também é conhecida como “conexões todos-para-todos”, ou



seja, onde todos os clientes em um mesmo grupo se comunicam e trocam dados entre si simultaneamente.

Com a figura 2, mostrada abaixo, podemos exemplificar o modelo de conexão multiponto, onde o relacionamento da rede ocorre entre vários terminais.

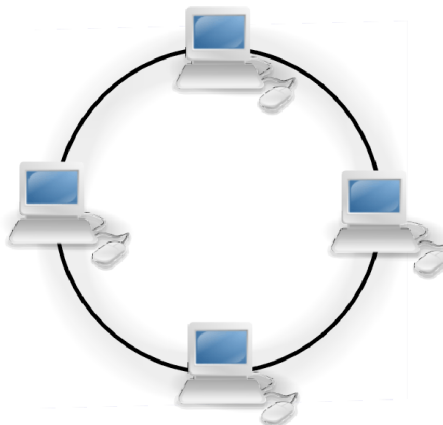


Figura 2: Exemplo de Conexão Multiponto.

O MCU (*Multipoint Control Unit*) é o componente que centraliza os fluxos de vídeo e áudio em uma sessão multiponto, determinando qual ponto pode transmitir e receber a cada instante de tempo (*multiponto-to-multiponto half duplex*) ou combinando as imagens de diversos pontos em um único fluxo (subdividindo a tela) e transmitindo a imagem composta para todos os pontos (*Continuous Presence, full duplex*) (BORDIGNON, 2001).

O MCU não manipula os dados codificados e compactados, ou seja, não implementa nenhuma função de CODEC. O componente apenas processa os dados de controle da aplicação. Esta funcionalidade é conhecida como MC (*Multipoint Controller*). O serviço (também conhecido como MP – *Multipoint Processor*) descompacta as imagens e as combina em um novo quadro, que é novamente compactado e retransmitido para os destinos.

### 3.2 Tipos de tráfego

Dentro das classificações dos tipos de conexões, podemos separar os tipos de tráfegos, ou seja, como esses dados ou pacotes são distribuídos na rede. Podemos utilizar o tráfego *unicast*, onde o fluxo de dados vai ser enviado

apenas para um receptor e o *multicast* ou *broadcast* que vai ser enviado para muitos receptores.

### 3.2.1 Tráfego unicast

O tráfego *unicast* é um tipo de conexão “ponto-a-ponto”, em que cada cliente recebe um fluxo distinto do servidor, onde o mesmo é enviado para um único cliente que efetuou a solicitação.

Podemos ilustrar com a figura 3 o esquema de tráfego unicast, em que os pacotes de dados são transferidos entre dois terminais.

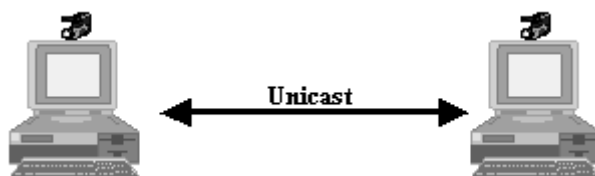


Figura3: Exemplo de Conexão Ponto-a-Ponto.

A utilização da conexão *unicast* para transferir dados em massa das aplicações de vídeo, poderá comprometer desempenho da rede, pois estes pacotes de dados serão recebidos pelo mesmo canal de comunicação da rede.

### 3.2.2 Tráfego multicast

O tráfego *multicast* é do tipo “um-para-muitos” sendo enviado os dados para os terminais que solicitarem os pacotes de dados, assim sendo possível a transmissão de áudio e vídeo para várias pessoas. Permite que um emissor envie dados para muitos receptores ou grupo simultaneamente.

Na figura 4, temos um exemplo de tráfego *multicast*, onde o servidor pode enviar os dados para todos os receptores da rede.

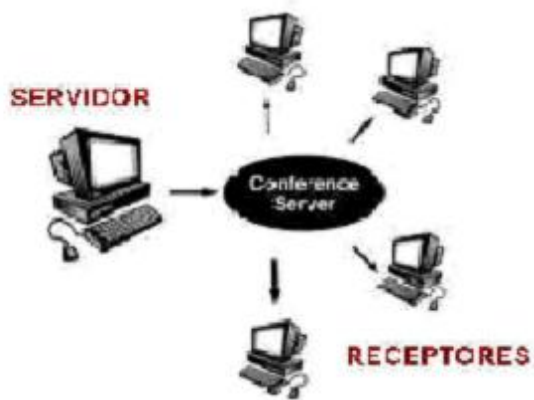


Figura 4: Exemplo de tráfego Multicast

Esta tecnologia foi criada com o objetivo de economizar largura de banda, fazendo com que apenas receptores interessados pelos pacotes que foram enviados recebam os dados, não sendo necessário o envio dos dados para destinatários não solicitantes.

### 3.2.3 Tráfego broadcast

O tráfego *broadcast* compreende o tráfego gerado por um nó de rede, cujo conteúdo é endereçado a todos os demais nós da rede, feito de forma não repetitiva. As mensagens *broadcast* são geradas de forma que todos os nós da rede simultaneamente recebam a informação enviada.

Na figura 5, observar-se a demonstração do tráfego broadcast, onde os dados são enviados apenas uma vez para toda a rede.

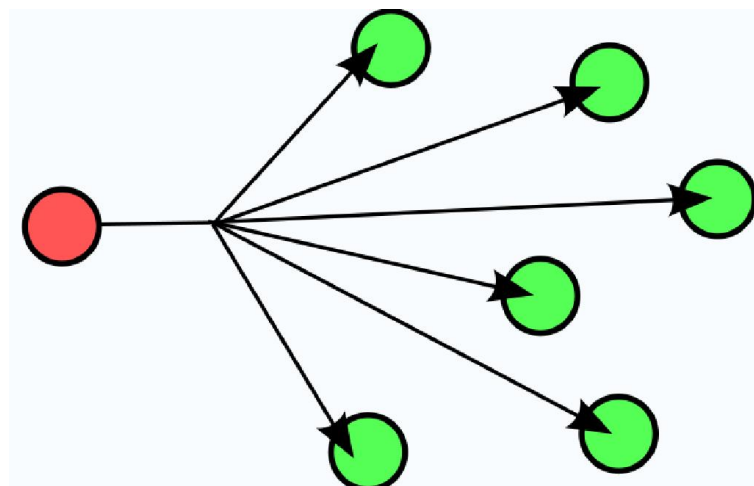


Figura 5: Exemplo de tráfego Broadcast

Uma das desvantagens do tráfego *broadcast* é o consumo de processamento de todos os computadores conectados à rede, pois, cada um destes receptores irá receber os pacotes que foram enviados.

### 3.3 Padrões de videoconferência

Nos primeiros sistemas de videoconferência, de soluções proprietárias, não existia muita integração entre esses sistemas, pois não havia recomendações ou normas estabelecidas, em vista deste problema, houve necessidade de estabelecer padrões para transmissão de videoconferência e desta maneira pode ocorrer uma interoperabilidade entre os diferentes sistemas.

A ITU-T (*International Telecommunication Union Telecommunication Standardization sector*) realiza as normalizações para a codificação de vídeo e áudio, sistema de controle das conexões, e especificações de recomendações para a transferência de dados numa sessão de videoconferência.

Na Figura 6 pode ser visto a compatibilidade entre os equipamentos e sistemas de videoconferência.

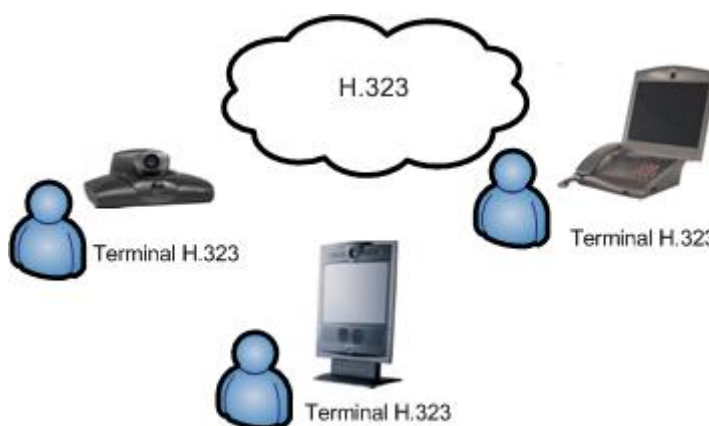
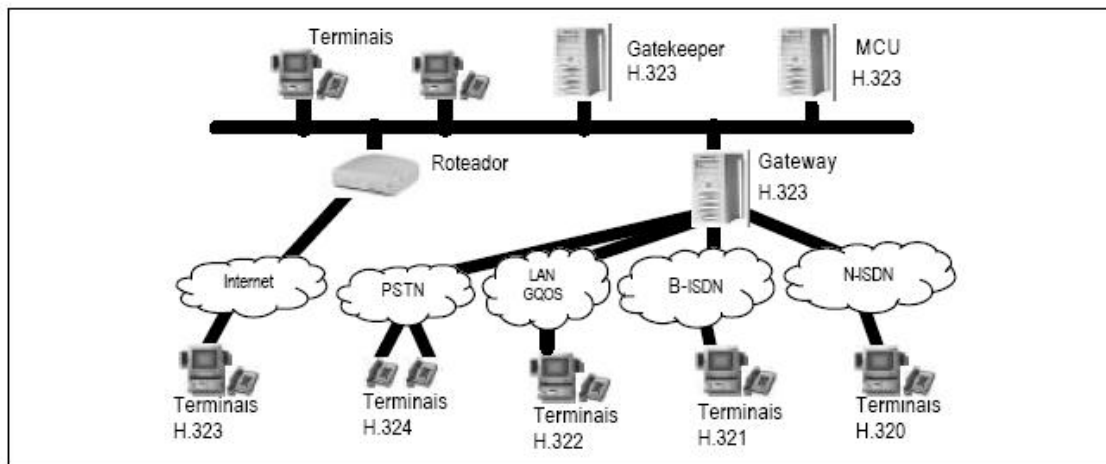


Figura 6: Modo de comunicação H.323

Os componentes utilizados pela arquitetura H.323, segundo ITU, na realização de serviços de comunicação multimídia sobre redes de pacotes, são os seguintes: os terminais, *gateways*, *gatekeepers* e MCUs (Figura 1).



**Figura 1:** Componentes da arquitetura H.323

- Terminais: um ponto de rede que consiste numa comunicação bidirecional com outro equipamento terminal, *gateway* ou MCU em tempo real.
- Gateway: Tem a função de converter diferentes formatos de mensagens e procedimentos de comunicação.
- Gatekeeper: É um elemento opcional que prover de serviços de controle de chamadas para os terminais.

Segundo LEOPOLDINO (2001), a adoção do padrão H.323 para aplicações multimídia em redes traz benefícios, entre os quais: Independência de rede, Interoperabilidade de equipamentos e aplicações, Independência de plataforma, representação padronizada de mídia, flexibilidade nas aplicações clientes, Interoperabilidade entre redes, suporte a gerenciamento de largura de banda, suporte as conferências multiponto e suporte a multicast.

SIP (Session Initiation Protocol) é um protocolo de camada de aplicação pode ser rodado sobre UDP ou TCP, passou a ser alternativa ao H.323, aprovado pelo IETF, pode integrar vários conteúdos através do gerenciamento de sessão e são representados como URLs, por exemplo sip:osmario@jmstudio.net.

### 3.3.1 Vídeo

As aplicações de videoconferências sempre foram tratadas como críticas, pois transmitem e recebem áudio e vídeo simultaneamente e em tempo real. Outra preocupação é manter a compatibilidade entre aplicações desenvolvidas por diferentes fabricantes. A interoperabilidade pode ser garantida através da definição de padrões, que basicamente definem o modo de transmissão dos diferentes tipos de mídia que são utilizadas em sessões de videoconferência (LOPES, 2004).

A Recomendação H.323 pode utilizar o protocolo TCP ( *Transport Control Protocol* ) que torna a comunicação confiável ou protocolo UDP ( *User Datagram Protocol* ) que realizar a comunicação não confiável.

Da família H.32x fazem parte as recomendações abaixo que também referenciam outras, como protocolos para codificação de áudio e vídeo, multiplexação, sinalização e controle (RNP, 2006).

- H.310 – Normaliza as transmissões de videoconferência em redes ATM e B-ISDN (para MPEG-2);
- H.320 – Transmissões de videoconferência em redes comutadas por circuitos (N-ISDN, SW56 e redes dedicadas);
- H.321 – Recomendação para transmissão de videoconferência em redes ATM e B-ISDN;
- H.323 – Recomendação para transmissão de videoconferência em redes que não provêm uma garantia de Qualidade de Serviço (QoS);
- H.324 – Recomendação para transmissão de videoconferência em redes telefônicas.

A figura 7 ilustra o modelo recomendado pela ITU e toda estrutura composta pela família H.32x.

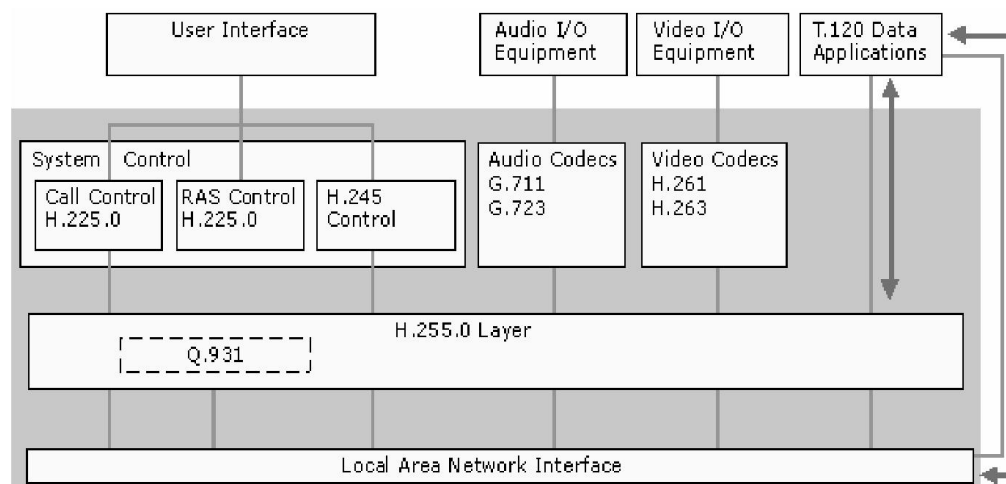


Figura 7: Estrutura do H.323.

### 3.3.2 Áudio

O áudio é usado de várias maneiras, muito especificamente para comunicação através da fala, por exemplo. No caso da videoconferência, os pacotes de áudio deverão ser encaminhados à camada de transporte periodicamente, dentro dos intervalos determinados pelo CODEC de áudio.

No caso da recomendação H.323 todos os terminais devem possuir um CODEC de áudio. A recomendação H.323 especifica que todos os CODECS deverão também ser capazes de operar no padrão G711 (GOMES, 2003).

A recomendação G.711 oferece uma gama de três faixas de transmissão, a saber: 48 kbps, 56 Kbps e 64 Kbps, sendo esta última a resultante de 8 Khz de frequência de amostragem e 8 bits de resolução digital, totalizando a taxa de 64 Kbps.

Os terminais H.323 para os serviços de videoconferência podem receber mais de um canal de áudio, provendo uma função de áudio *mixing*, com o propósito de entregar ao usuário um sinal de áudio composto, resultante de conversações realizadas nas inúmeras salas de videoconferência..

### 3.4 Codecs

Um codec é utilizado para compressão e descompressão de dados. A compressão é utilizada quando estes dados vão ser transmitidos ou

armazenados. No componente receptor estes dados são descompactados para um formato de apresentação (JORGE, 2008).

Dentro dos critérios que influencia de escolha do Codec temos o consumo de banda, nos números de canais simultâneos e principalmente na qualidade da voz para o usuário.

O codificador opera com o objetivo de entregar quadros sucessivos para a transmissão em um determinado meio, aplicando os processos necessários de compressão para redução da largura de banda para a efetiva transmissão (GOMES, 2003).

Em se tratando de transmissão de dados em rede de videoconferência, deve-se utilizar artifícios de compressão de vídeo para reduzir os fluxos de dados embora, exista ainda necessidade de codificação e decodificação do vídeo para exibição e isso pode levar a uma maior latência. A família MPEG (*Moving Picture Experts Group*), com distintas especificações, trata da compressão e transmissão de áudio e vídeo.

Atualmente existem três padrões relacionados com compressão de vídeo publicados pelo ISO-MPEG: MPEG1, MPEG2 e MPEG4. Os outros dois padrões que a MPEG está trabalhando são o MPEG7 (para descrever conteúdo multimídia – metadados) e MPEG21 (que é para definir um quadro multimídia). Atualmente, eles ainda não têm um papel importante no mundo da videoconferência. (COOKBOOK, 2005).

Na Tabela 1, segue a comparação entre os três padrões com relação à resolução da imagem e à largura de banda.

	<b>MPEG1</b>	<b>MPEG2</b>	<b>MPEG4</b>
<b>Resolução</b>	352x240	720x480	720x480
<b>Taxa típica</b>	1.5 Mbps	5 Mbps	2 Mbps
<b>Taxa máxima</b>	2.5 Mbps	15 Mbps	4 Mbps

Tabela 1: Resolução e largura de banda dos padrões MPEG.

No desenvolvimento de uma aplicação multimídia, os autores devem escolher que técnica de compressão utilizar. Esta escolha geralmente baseia-se nas classificações apresentadas anteriormente, nos parâmetros de



desempenho da técnica e nos requisitos da aplicação (WILLRICH, 2003). Os parâmetros de desempenho mais usados são:

- Taxa de compressão: razão entre o tamanho do dado original e o tamanho do dado após a compressão. No caso de técnicas sem perda, quanto maior a taxa de compressão melhor é a técnica de compressão. Para técnicas de compressão com perda deve-se considerar também a qualidade da mídia reconstituída;

- Qualidade da mídia reconstituída: medida em SNR (Razão Sinal/Ruído). Este parâmetro é aplicável apenas para técnicas com perda. Para a escolha de uma técnica de compressão com perdas, deve-se optar pelo compromisso entre uma alta taxa de compressão e a qualidade desejada para a aplicação em desenvolvimento;

- Complexidade de implementação e velocidade de compressão: geralmente quanto mais complexa a técnica menor é a velocidade de compressão. No caso de aplicações tempo-real, como videoconferência, estes parâmetros devem ser considerados.

Na tabela 2 são apresentados os tipos mais comuns de formatos de vídeo:

<b>Formato</b>	<i>Content Type</i>	<b>Qualidade</b>	<b>Requisitos de Processamento</b>	<b>Requisitos de Largura de Banda</b>
MPEG-1	AVI QuickTime	Alta	Alto	Alto
H.261	MPEG	Baixa	Médio	Médio
H.263	QuickTime AVI RTP	Média	Médio	Baixo
JPEG	QuickTime AVI RTP	Alta	Alto	Médio

Tabela 2: Formatos comuns de vídeo.

O sinal gerado é inicialmente digitalizado, para então passar por um processo de compressão, que diminui seu tamanho, tornando-o viável para ser transmitido na rede.

Na tabela 3 são apresentados os tipos mais comuns de formatos de áudio:

Formato	Content Type	Qualidade	Requisitos de Processamento	Requisitos de Largura de Banda
MPEG-1	MPEG	Alta	Alto	Alto
MPEG Layer 3 (mp3)	MPEG	Alta	Alto	Médio
GSM	WAV RTP	Baixa	Baixo	Baixo
G.731	WAV RTP	Média	Médio	Médio
PCM	AVI QuickTime WAV	Alta	Baixo	Alto

Tabela 3: Formatos comuns de áudio.

Temos os três fatores relevantes na qualidade de áudio e processo de digitalização que são: a taxa em que as amostras são realizadas, a quantidade de bits utilizada para representar cada valor amostrado e o número de canais utilizados.

### 3.5 Desempenho

Para melhorar o desempenho de uma sessão de videoconferência, o mecanismo de transmissão indicado seria o *Multicast*, que combina os mecanismos de *Unicast* e *Broadcast*. Através dele, um pacote é enviado simultaneamente para um grupo de computadores que, por exemplo, participam de uma sessão de videoconferência, e somente esses computadores recebem o pacote, minimizando a quantidade de largura de banda utilizada.

Podemos definir o desempenho como sendo um dos aspectos mais importante para qualquer projeto de sistema de videoconferência, no qual o

desenvolvedor terá que dar uma atenção maior na qualidade de serviço e evitar assim o retardo da transmissão.

Com os valores apresentados na Tabela 4 e 5, fica clara a necessidade da utilização de uma rede que disponibilize banda larga de transmissão para aplicações que tenham sido elaboradas para a transmissão de vídeo com boa qualidade.

<b>Qualidade de Áudio</b>	<b>Especificação</b>	<b>Taxa de Tx.</b>
Áudio com Qualidade de Voz	1 canal, amostras de 8-bit / 8kHz	64 Kbps
Voz Digitalizada	Padrão G.728, 3.4 kHz	16 Kbps
Áudio Monofônico	1 canal, amostras de 16-bit/44.1kHz	705.6Kbps
Áudio com qualidade de CD	2 canais, amostras de 16-bit/44.1kHz	1.411Mbps
Áudio codificado com MPGE	Equivalente a qualidade de CD	384 Kbps

Tabela 4 - Taxa de Transmissão para Áudio e Áudio Comprimido

	<b>Resolução</b>	<b>1 minuto</b>	<b>1 hora</b>
a	640 x 480	1.6 GB	97 GB
a	320 x 240	400 MB	24 GB
b	640 x 480	16 MB	970 MB
b	320 x 240	4 MB	240 MB

Tabela 5 - Valores Estimados do Volume do Sinal de Vídeo (a) Não Comprimido e (b) Comprimido, em Bytes

A transmissão de vídeo apresenta muitas propriedades em comum com a transmissão de outras mídias, por ser um grande volume de dados e a necessidade da transmissão síncrona e em tempo-real. Podemos separar os seguintes critérios:

- Atraso total, devido à busca no servidor e a transmissão;
- Variação do atraso;
- Taxa de transmissão;
- Taxa de erro de transmissão.

### 3.5.1 Qualidade de serviço

Qualidade de serviço é definida da seguinte forma “Qualidade de Serviço é um requisito que uma classe de aplicações possui de exigir que uma

série de parâmetros esteja dentro de uma faixa de valores bem definidos, para o seu perfeito funcionamento” (BORDIGNON, 2001).

A largura de banda é a capacidade de transferência de dados, ou seja, na quantidade de pacotes que podem ser transferidos de um ponto a outro num determinado intervalo de tempo. Sendo importante na qualidade de transmissão do áudio e do vídeo transmitido na rede.

Dentro da qualidade de serviço podem separar os parâmetros como, por exemplo, perdas, atrasos, vazão para que proporcione um desempenho aceitável de início e fim da transferência de pacotes.

O resultante da variação do atraso de propagação dos pacotes é chamado de *Jitter*. O atraso ou retardo é o tempo decorrido entre o momento em que o dado está pronto para ser transmitido e o momento em que chega ao seu destino.

Podemos diminuir a qualidade de vídeo para transmissão de baixa velocidade, reduzindo a resolução geométrica (número de *pixels*), reduzindo a quantidade de cores, o número de quadros por segundo.

A determinação de um parâmetro de QoS deve ser definido de acordo com o ambiente utilizado, pois cada aplicação possui sua particularidade. Esses parâmetros de QoS são negociados entre a aplicação e a rede, tanto no momento de estabelecimento de uma conexão, quanto após o estabelecimento da mesma, haja visto que os processos de sinalização permitam esse tipo de negociação (VILELA, 2007).

### 3.5.2 Retardo de transmissão

*Jitter* é uma variação estatística do retardo na entrega de dados em uma rede, ou seja, pode ser definida como a medida de variação do atraso entre os pacotes sucessivos de dados.

A consequência do *jitter* é que a aplicação no destino deve criar um buffer cujo tamanho vai depender da variação da entrega de pacotes, gerando mais atraso na transmissão de videoconferência. Esse buffer vai servir como uma reserva para manter a taxa de entrega constante no interlocutor. Daí a importância de latência e *jitter* baixos em determinadas aplicações sensíveis a esses fatores, como videoconferência.

A medida do atraso tornou-se um dos critérios básicos de avaliação da qualidade de serviço de uma rede, onde o mesmo pode comprometer as aplicações multimídia e resultar em perda do sincronismo.

Latência é o atraso de tempo entre o terminal remoto e a ação de um evento. A mesma ocorre em aplicações interativas e de tempo real, tais como videoconferência, por serem sensíveis ao atraso.

Sabendo que atraso de *codec* precisa ser o menor possível, pois tratar-se de aplicações multimídia interativas, todo processo de codificação e decodificação deve possuir a velocidade necessária que os olhos humanos não percebam a transmissão com paradas ou até mesmo falhas.

### 3.6 RTP

O RTP (*Real-Time Protocol*) é o protocolo de transporte de aplicações em tempo real, ou seja, promove serviço de envio e recebimento de dados como uma videoconferência.

O RTP é um protocolo que funciona no espaço de usuário do sistema operacional, vinculado a camada de aplicação, porém é um protocolo genérico e independente de aplicações, comum aos protocolos da camada de transporte (TANENBAUM, 2003).

Este protocolo de transporte (RTP) pode ser usado para uma grande variedade de aplicações, com diferentes requisitos e finalidades. Também provê mecanismo de sequenciamento, sincronismo, identificações e informações de qualidade de serviço chamado QoS (*quality of service*).

A Figura 8 mostra a relação do protocolo RTP com outros protocolos e aplicações.



Figura 8: Relação do protocolo RTP com outros protocolos e aplicações (LIMA, 2003)

O RTP é geralmente colocado sobre o UDP na pilha de protocolos. Assim, ambos colaboram para o serviço de transporte da aplicação. Pelo fato de poder trabalhar tanto sobre UDP quanto sobre TCP, o RTP não pode ser classificado como um protocolo orientado à conexão nem como não orientado à conexão (LOPES, 2004).

Atualmente, é aconselhável a utilização do protocolo RTP sobre UDP/IP, pois o mesmo provê maior qualidade às aplicações e possibilita a adaptação da transmissão de mídia em tempo real.

No diagrama da Figura 9, podemos ver que o emissor é responsável por capturar e transformar dados audiovisuais para a transmissão, bem como gerar os pacotes RTP. Ele poderá participar também da correção de erros e do controle de congestionamento ao adaptar o *media stream* em resposta a um *feedback* do receptor.

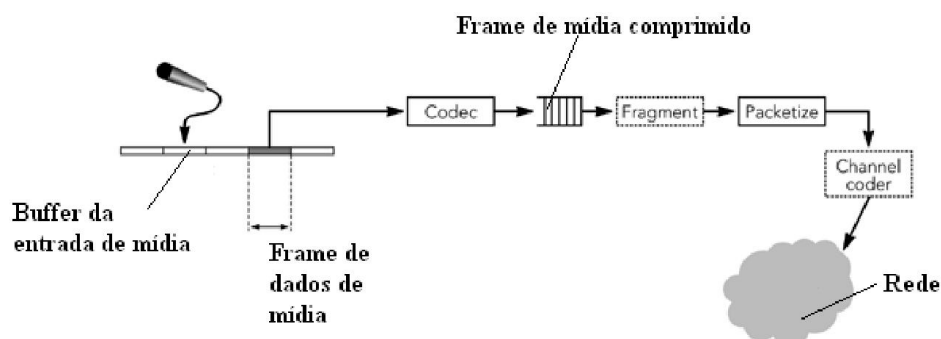


Figura 9: Diagrama de blocos de um emissor RTP.

Os dados de mídia não compactados (áudio ou vídeo) são capturados e colocados em um *buffer*, a partir dos quais *frames* comprimidos são produzidos. Os *frames* podem ser codificados de diversas formas dependendo do algoritmo de compressão utilizado, e os *frames* codificados poderão depender de dados enviados anteriormente ou a posteriori (QUELHO, 2006).

A fragmentação dos *frames* ocorrerá caso seu tamanho se torne grande, casos contrários estes poderão ser colocados num único pacote RTP.

O protocolo RTP não reserva recursos de rede e nem garante QoS, geralmente ele é utilizado em paralelo com o RTCP (*RTP Control Protocol*), com isso permite o monitoramento da comunicação e o transporte de informações sobre os participantes de uma sessão RTP.

### 3.7 RTPC

O RTPC é baseado na transmissão periódica de pacotes de controle para todos os participantes de uma sessão, usando os mesmos mecanismos de distribuição dos pacotes de dados [Lopes 2004].

O RTCP (Real-Time Transport Control Protocol), especificado também pela RFC 1889, por sua vez, monitora a entrega dos dados e provê funções mínimas de controle e identificação. Com o uso de dois protocolos para o transporte em tempo real, diminui-se o overhead que o uso de somente um único protocolo teria. (QUELHO, 2006, p.84).

Podemos utilizar o protocolo RTCP para monitorar a qualidade de serviço e transportar informações sobre os participantes de uma sessão do RTP, onde o mesmo também contém um nível de transporte de identificação, para uma fonte RTP que o destinatário utiliza para sincronização de áudio e vídeo.

O protocolo de controle RTCP permite que os receptores forneçam 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 transmissão, enquanto que outros receptores podem determinar se os problemas na qualidade do serviço são locais ou gerais.

## 4. JMF ( Java media framework )

O JMF (Java Media Framework) é um framework da Sun que foi desenvolvido em parceria com a Intel e IBM, que permite manipular recursos multimídia em aplicações Java. (JORGE, 2008).

Este framework possui independência de plataforma, garantindo portabilidade aos demais sistemas, além de provê suporte aos principais formatos e codecs de áudio e vídeo. Além disso, o JMF API possui as seguintes características:

- Manipulação integrada e uniforme de áudio e vídeo como objetos de mídia.
- Uma estrutura de processamento uniforme que suporta todas as operações com mídia.
- Captura de mídia de dispositivos tais como câmeras e microfones.
- Recepção de fluxos de mídia transmitidos através da rede.
- Transmissão de fluxos de mídia (através da rede).
- Extensível para suportar novos formatos e plug-ins.

O JMF utiliza o modelo da Figura 10, como idéia base do seu framework. Um objeto *DataSource* tem a função de encapsula o fluxo de mídia da mesma forma que a fita e um objeto *Player* para os mecanismos de processamento e controle similares a um VCR.

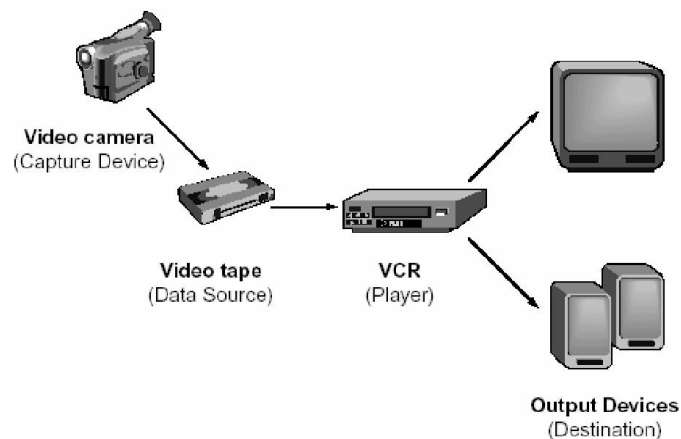


Figura 10: Modelo básico de processamento de mídia

A arquitetura de classes do JMF está baseada nos seguintes modelos: tempo, dados e eventos. A partir destes modelos, são concebidas as interfaces



e as classes do JMF. A figura 11 ilustra um diagrama de herança para algumas das interfaces relevantes.

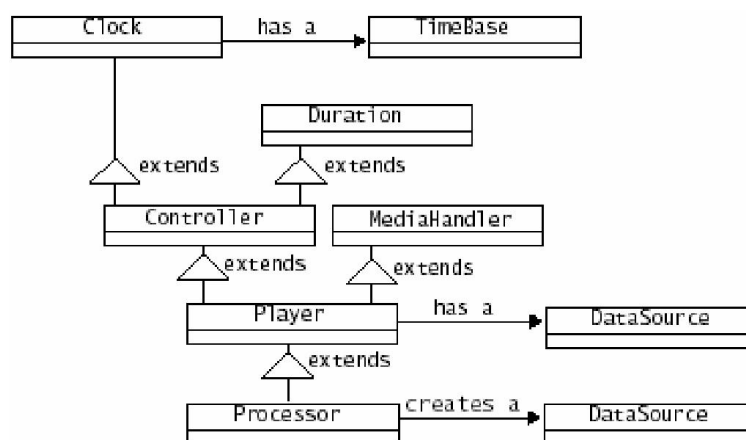


Figura 11: Diagrama de heranças de classes no JMF

O suporte oferecido pelo JMF aos principais formatos e codecs de áudio e vídeo e a portabilidade provida pela linguagem Java foram fatores decisivos para a adoção da API como base do framework. É possível criar aplicativos Java que reproduzem, editam e capturam muitos dos tipos de mídia através das reutilizações disponíveis pelo JMF.

#### 4.1 O framework

Para o desenvolvimento de aplicações de videoconferência será necessário muito esforço dos desenvolvedores, pois transmitir áudio e vídeo requer preocupação com diversos aspectos como a complexidade e o tempo de desenvolvimento destas aplicações, além das particularidades do sistema e domínio para a reutilização nas funcionalidades mais genéricas.

Diversos setores da sociedade demandam softwares que possuam recursos de áudio e vídeo, tais como: segurança, controle de acesso e entretenimento, motivados pela redução do custo das câmeras e ampliação da largura de banda da internet. Com o objetivo de agilizar e simplificar o desenvolvimento dessas aplicações tem-se buscado a reutilização com a adoção de frameworks. (JORGE, 2008, p.60)

Para se construir um framework é necessário ter experiência com o tipo de aplicação a ser desenvolvida, pois permitirá ao desenvolvedor definir a

estrutura do framework bem como as funcionalidades que deverão ser providas pelo mesmo (JACQUES, 2006).

JMF ainda oferece suporte ao protocolo de transmissão de mídia em tempo real RTP, permitindo ao desenvolvedor, criar aplicações que fazem transmissão de dados em tempo real através de uma rede [Jmfrtp 2006]. A utilização de um framework proporcionará um nível maior de reutilização tanto na etapa de análise quanto na fase codificação.

A principal desvantagem de implementações nativas de media players é que elas são dependentes de plataforma. Por esta razão elas não são portáteis entre as plataformas. Isto significa que as aplicações utilizadas, dependentes de plataforma e processadores, são inadequadas para web. O JMF provê uma plataforma neutra para manuseio de multimídia (SILAS, 2006).

A Figura 12 mostra a visão geral do ambiente do framework e de como o mesmo poderá ser utilizado por diversas aplicações. Pode-se notar que o novo framework ocupa uma posição intermediária entre as aplicações e a API JMF, encapsulando as diversas atividades de médio-baixo nível, como criação de seções RTP, detecção de dispositivos de captura, sincronização dos fluxos de dados entre as máquinas envolvidas, criação e sincronização de *threads*, entre outras.

Figura 12: Ambiente do framework

Existe uma camada de integração que permite facilitar a construção de novas aplicações de videoconferência ou realizar integração com as aplicações já existentes.

O JMF suporta diversos formatos de mídia popularmente conhecidos, tais como JPEG, MPEG-1, MPEG-2, QuickTime, AVI, WAV, MP3, GSM, G723, H263 e MIDI (SILAS, 2006).

## 4.2 Arquitetura e recursos do jmf

Para criação dos componentes providos pelo JMF para capturar, processar e apresentar os recursos de multimídia são utilizados os seguintes componentes gerentes (*Managers*) (JORGE, 2008):

- **Manager:** Responsável pela criação de DataSources e Players, entre outros, além de possibilitar a customização de componentes do framework;
- **PackageManager:** Mantém um registro dos pacotes que contém classes do JMF, tal como classes customizadas: Players, Processors e DataSources;

- `CaptureDeviceManager`: Contém uma base com os dispositivos de captura detectados na máquina;
- `PluginManager`: Responsável por manter a lista de plug-ins disponíveis para uso pelo JMF, sendo possível a instalação novos plug-ins.

O JMF configura uma API para a manipulação de mídia (tanto áudio quanto vídeo) em alto nível, especificando assim uma arquitetura unificada, protocolos de mensagens e uma interface de programação para captura, execução e sincronização de mídias baseadas no tempo (QUELHO, 2006, p. 90).

A API JMF possui a tarefa de especificar, controlar o acesso a fonte dos dados da mídia e prover com a classe `Java.net.URL`, a identificação de qualquer recurso na Internet. Sendo possível a localização da mídia através do objeto `MediaLocator` ou `URL`.

As classes do JMF relacionadas ao protocolo RTP são extensões a esse. Não substituem nem aperfeiçoam as funcionalidades do núcleo do JMF, como por exemplo, as encontradas nas classes do *Player*, do *Processor*, do *DataSource*, do *DataSink*, entre outras.

A fonte da mídia pode ser a captura de um dispositivo, ou um arquivo armazenado localmente ou remotamente na rede, ou um fluxo de mídia em tempo-real disponível na rede. Os gerenciadores da mídia a processam, e esse processo inclui a demultiplexação, ou multiplexação, ou codificação, ou decodificação. (SILAS, 2006, p.23)

A captura de dispositivos inclui câmeras, placas de som e todos os dispositivos capturados devem ser registrados com a classe `CaptureDeviceManager` em ordem para serem utilizados com o JMF. O `CaptureDeviceManager` pode selecionar uma lista de todos os dispositivos que suportam um formato de saída válido.

Como mostra a Figura 13, o modelo de processamento é basicamente dividido em três etapas: Entrada, Processamento e Saída.

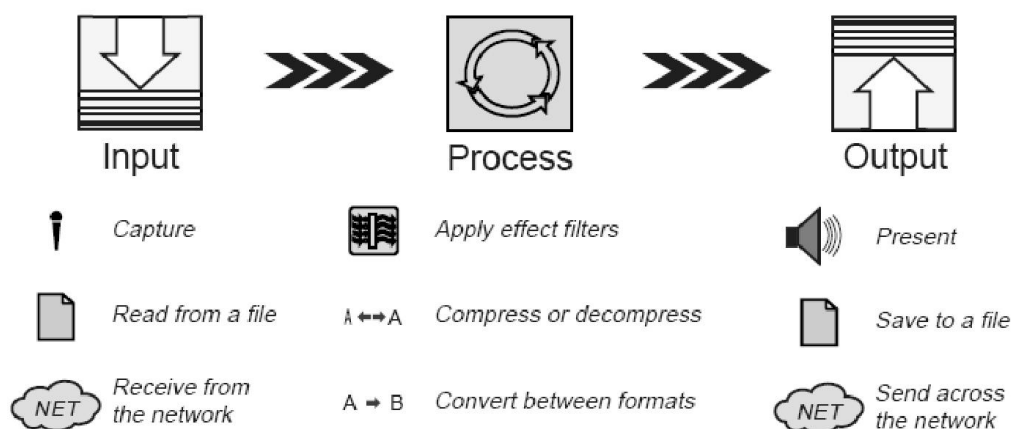


Figura 13: Modelo processamento de mídia [JmfGuide 1999].

O processo inicia com a captura dos sinais de áudio e vídeo a serem transmitidos. Estes dispositivos de captura são originados pelas câmeras e microfones, que podendo ser armazenado localmente ou rede.

Durante o processamento, são realizadas a filtragem, compressão, descompressão e a conversão entre formatos. Após o processamento dos dados é necessário dar um destino aos mesmos. Na etapa de saída esses dados podem ser salvos em um arquivo, enviados através da rede ou simplesmente apresentados em algum dispositivo, como tela e alto-falantes (FERNANDES, 2000).

A figura 14 mostra o relacionamento entre as principais classes da arquitetura JMF, demonstrando todo processo de relação entre as classes: Player, Processor e Datasource.

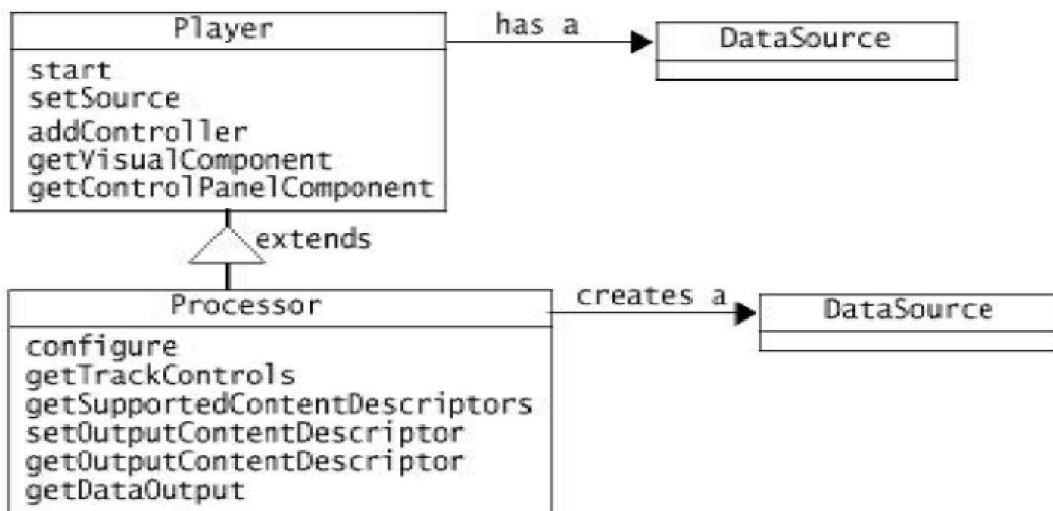


Figura 14: Relacionamento entre as classes DataSource, Processor e Player [JmfGuide 1999].

A arquitetura do JMF esta dividida em Data source, Capture device, Player, Processor, DataSink, Format e Manager:

- *Data source encapsula* os pacotes de dados das mídias de vídeo, áudio e passa a alimentar o player.
- *Capture descesse* realiza a captura dos dispositivos de áudio e vídeo, como por exemplo: webcam e microfone.
- Um *Player* recebe uma stream contendo as informações passada pelo datasource.
- *Processor* é a classe que é extendida pela classe player, por isso tem os mesmos controles que o player e pode redenrizar o datasource
- O *DataSink* tem como objetivo lê a mídia do DataSource e envia para algum destino.
- *Format* representa formato da mídia, não carregando parâmetros específicos de codificação.
- *Manager* é um objeto intermediário, que pertence a API JMF e que pode criar interface dois objetos diferentes.

#### 4.2.1 Players

O objeto *Player* tem a característica de apresentar a *stream* de áudio ou vídeo, ou ambos, após preparar a mídia pelo *DataSource* e utilizar os estados da classe *Controller*.

A figura 15 representa o diagrama de relacionamento com a classe *Player*, utilizando o objeto *Controller* para controlar a apresentação da mídia e também o objeto *Clock* para garantir a temporização.

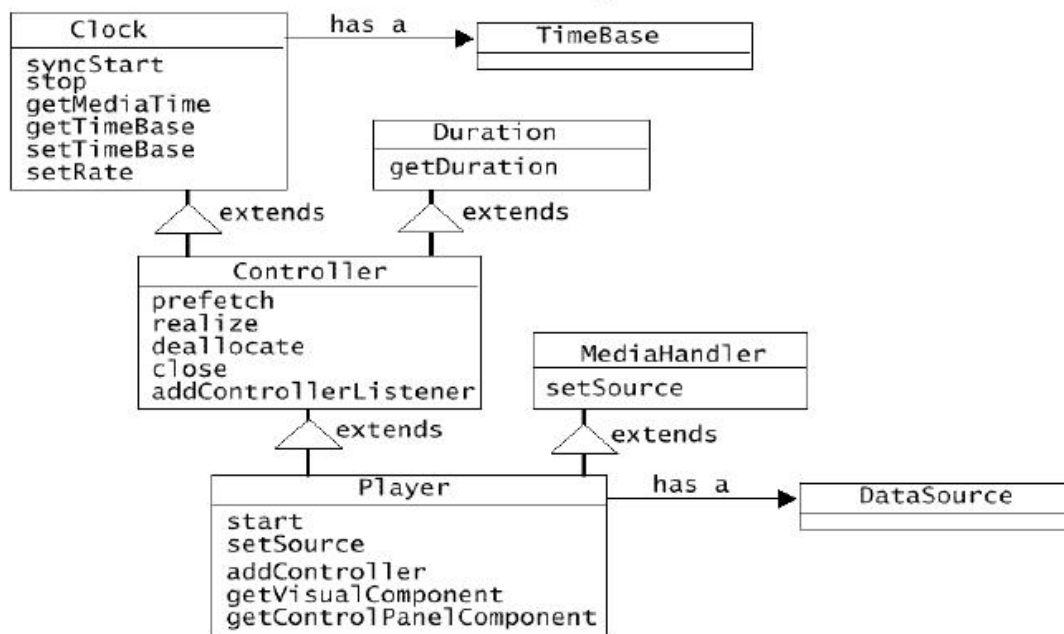


Figura 15: Diagrama de relacionamento entre classes do player [JmfGuide 1999].

Segundo [JmfGuide 1999], antes de ser executado, um *Player* pode estar em um dos seis estados definidos pela interface *Clock*. A Figura 16 mostra estes estados.

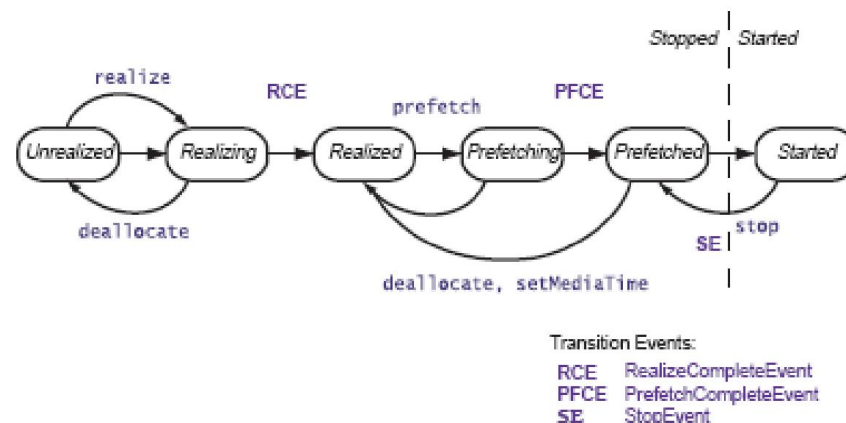


Figura 16: Estados de um Player [JmfGuide 1999].

Os cinco estados do objeto *Controller* que ampliam de *Clock* o conceito (estado) de “tempo parado” são:

- Unrealized – O Player fica nesse estado quando o objeto instanciado ainda não realizou a tarefa de alocar recursos.
- Realizing – O estado onde o Player obtendo informações sobre os recursos necessários ao seu funcionamento.
- Realized - o *Player* já tendo conhecido detalhes a respeito da mídia, ou seja, sobre todos os recursos necessários a execução de sua tarefa.
- Prefetching - o Player esta se preparando para inicia o carregamento da mídia no buffer ou adquirir recursos de hardware.
- Prefetched - o *Player* após adquirir todos os recursos necessários e estando pronto para ser iniciado.

#### 4.2.2 Processors

Os processors são utilizados também na apresentação de mídia, ou seja, é um tipo de Player, suportando todos os controles de apresentação do *Player*, o objeto *Processor* permite que o usuário defina qual processamento será realizado nos dados de entrada de fonte (*DataSource*).

#### 4.2.3 Managers

A classe Manager pode criar um player a partir de um datasource, além de poder fornecer métodos de acessos. Estes métodos iniciados pelo



ControllerEvents realizam a monitoração do processo de tratamento da mídia. A classe *PlayerEventHandler* fornece implementações vazias dos métodos da interface *ControllerListener*. A classe *ControllerAdapter* facilita a implementação de *ControllerListener*. Os *Players*, com base nesses estados, após transições durante o processamento da mídia, serão enviados a confirmação no final do processo.

Com o objeto manager pode-se acessar diversos recursos dos objetos instanciados, conforme demonstra o digrama de classe *Manager* na figura 17.

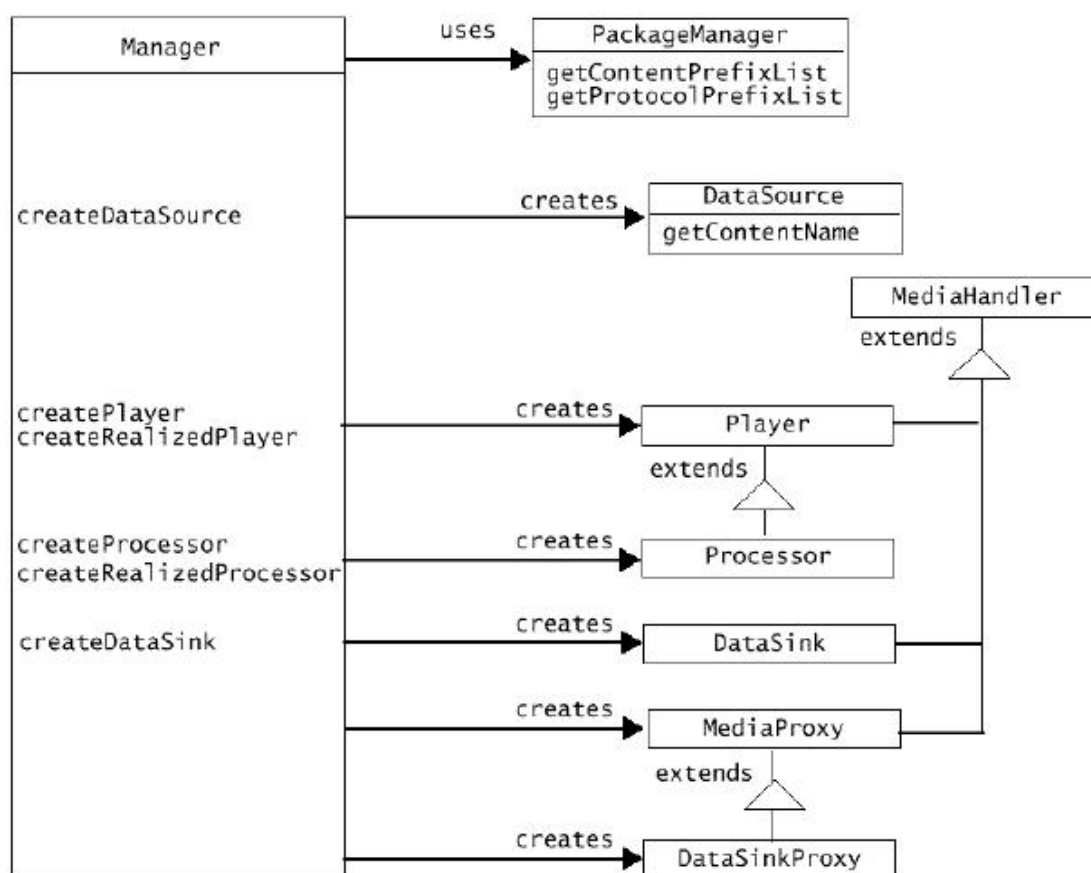


Figura 17: Diagrama de classe do objeto Manager [JmfGuide 1999].

#### 4.2.4 DataSource

Um *DataSource* é o meio pelo qual os objetos *Players*, *Processors*, ou *DataSinks* obtêm seus dados, que representa uma fonte de dados a ser processada, para ser transmitida ou apresentada. Este *DataSource* pode ser

localizado pelo objeto *MediaLocator*, possibilitando o acesso e a reprodução da mídia pelo objeto *Player*.

#### 4.2.5 DataSink

Um *DataSink* é usado para ler uma mídia de um *DataSource* e abstrai a localização do destino da mídia e fornecer um protocolo para execução da fonte da mídia em um destino, através da rede ou como fluxo (Stream).

#### 4.2.6 Formato de dados

*Format* é a classe mais especializada dentro das classes *AudioFormat* (realiza a especificação de atributos de trilhas de áudio) e *VideoFormat* (realiza a especificação de atributos de trilhas de vídeo).

A figura 18, apresenta o diagrama de classe *format* que realiza a especificação dos formatos da mídia.

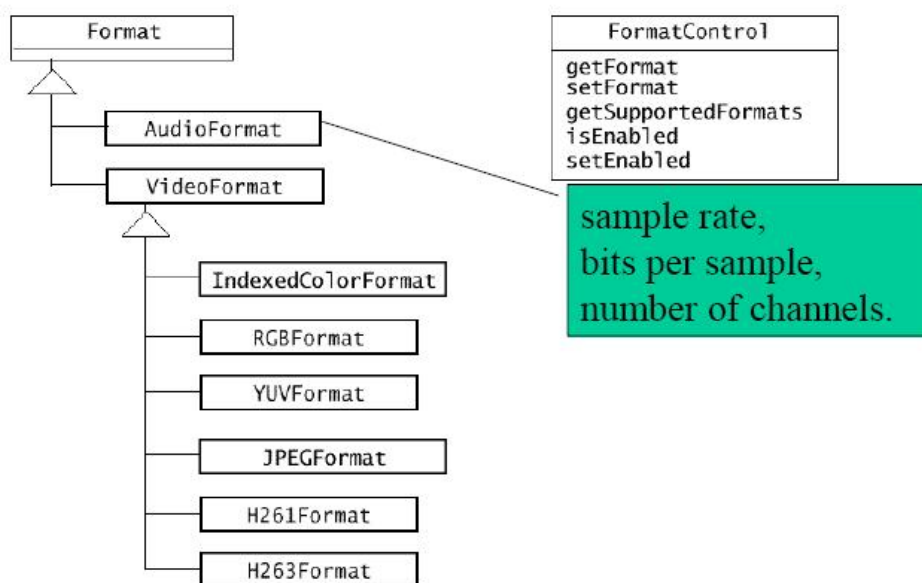


Figura 18: Diagrama da classe *Format* [JmfGuide 1999].

#### 4.2.7 Modelo de Dados

O *datasource* é um dos componentes principais do modelo de dados, pois encapsula o protocolo utilizado na transmissão e na localização da mídia escolhida, sendo identificado pelo objeto *MediaLocator* e URL.

Existem duas categorias de fonte de dados, definidas de acordo com a maneira em que o dado é transferido:

- Pull (puxar): o cliente inicializa a transferência dos dados e controle de fluxo.
- Push (empurrar): o cliente é responsável pela inicialização e o controle de fluxo.

A figura 19, ilustra o diagrama do modelo de dados com diversas classes relacionadas.

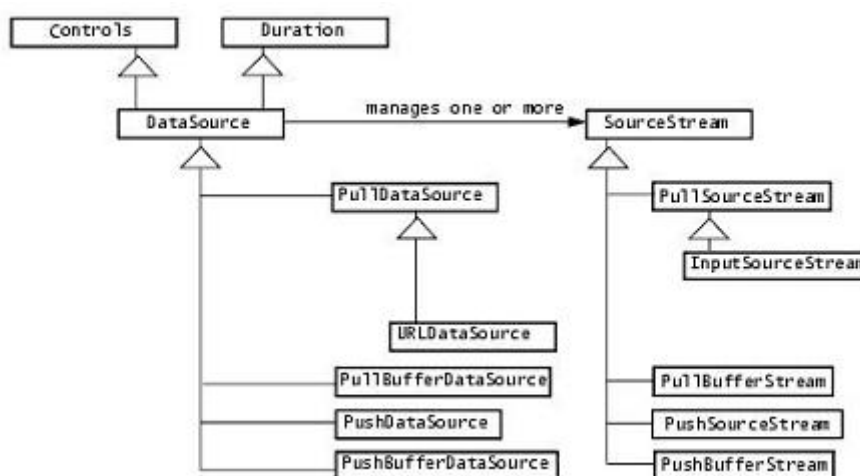


Figura 19 : Modelo de dados do JMF [JmfGuide 1999].

#### 4.2.8 Modelo de eventos

Quando um evento é disparado por um objeto, é necessário que informe seu estado atual, esse instancia um *MediaEvent*. São quatro as classes filhas de *MediaEvent*:

- *ControllerEvent*: a classe utilizada como base para os eventos disparados por um *Controller*;
- *DataSinkEvent*: a classe utilizada como base para os eventos disparados por um *DataSink*;
- *GainControlEvent* : quando é disparado por um *GainControl*, com o objetivo de sinalizar uma mudança de estado. Com a interface *GainControl* pode ser especificado as trilhas de áudio.
- *RTPEvent*: a classe utilizada como base para todas as notificações de um *SessionManager*.

Na Figura 20 pode-se notar o relacionamento da classe de eventos (ControllerEvent).

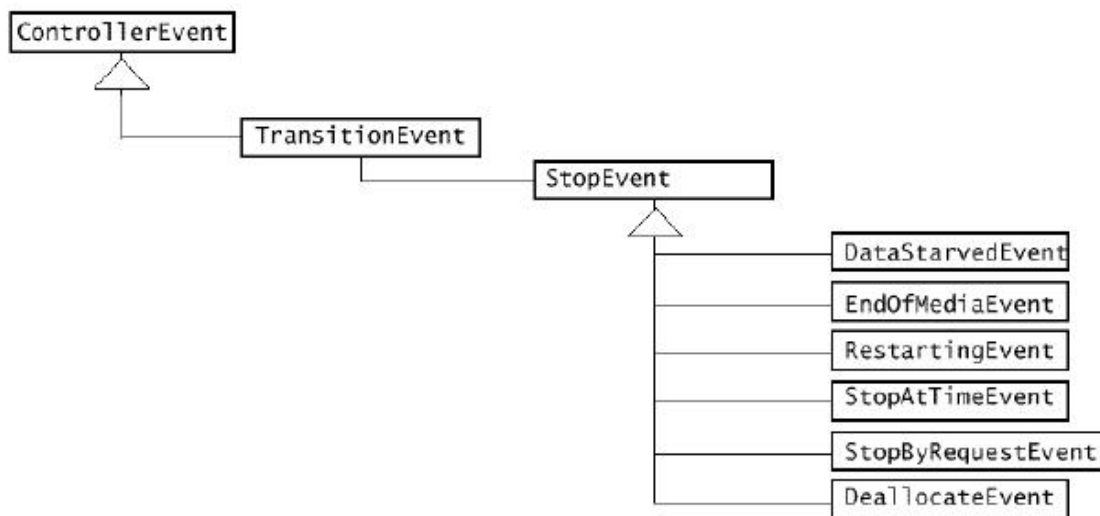


Figura 20: Diagrama de classe controle de eventos [JmfGuide 1999].

#### 4.2.9 Mídias baseado em tempo

Segundo [JmfGuide 1999], quaisquer dados que mudam significativamente com o passar do tempo podem ser caracterizadas como Mídias Baseadas no Tempo ou Mídias Temporizadas. Clipes de áudio, vídeo e animações em geral são os tipos mais comuns de mídias temporizadas. Os fluxos de mídia podem ser obtidos de diferentes fontes, como arquivos locais ou armazenados na rede, câmeras, microfones, entre outros.

A interface *Clock* define as operações de sincronização e temporização necessárias à apresentação dos dados de mídia.

A figura 21 mostra a classe *Clock* relacionando o objeto *TimeBase* e as demais classes que controlam o tempo da mídia.

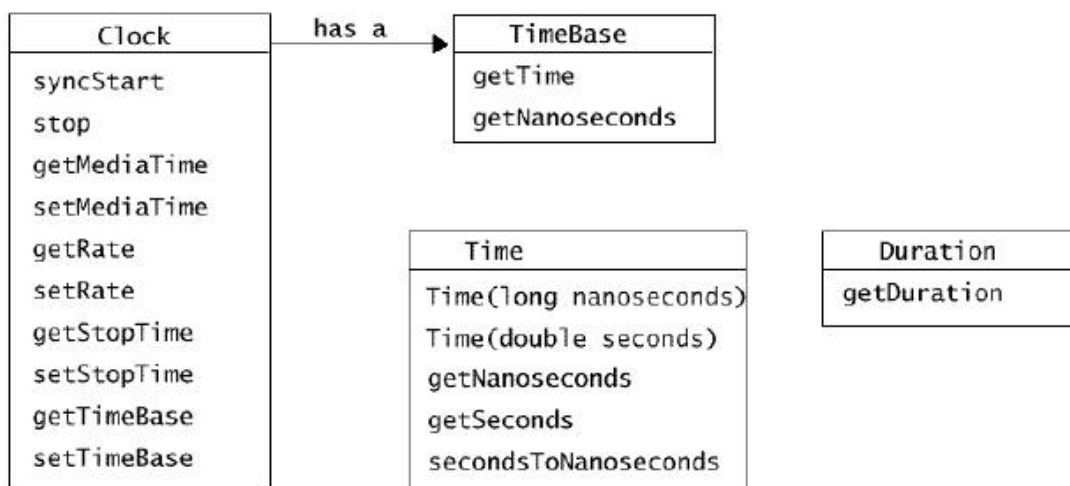


Figura 21: Diagrama de classe entre relacionamento da classe Clock [JmfGuide 1999].

A classe Clock provê uma transformação de tempo de TimeBase que está associado com o tempo de mídia pode ser determinado, através dos três parâmetros: a taxa, o tempo de início da mídia e o tempo inicial da base de tempo.

#### 4.2.10 Protocolo de tempo real

O JMF suporta um protocolo de transporte chamado Real-time Transport Protocol (RTP) para transmissão e recepção de mídia que tipicamente é utilizado sobre a camada UDP. O controle dos pacotes enviados é realizado pelo RTCP que envia dados de controle para os participantes da sessão.

Cada uma dessas aplicações é chamada de participante, onde todo participante utiliza um objeto chamado RTPManager para coordenar a sessão RTP a seu favor. O fluxo de mídias trocados em uma sessão RTP é chamado de RTPStreams. O RTPStreams pode ser de dois tipos, sendStream (Envio do fluxo) e ReceiveStream (Recebimento do fluxo) (SILAS, 2006, p.27).

A classe relacionada ao RTP proporciona as funcionalidades de gerenciamento de sessões, monitoração de eventos e fluxo de mídia.

## 5. JMSTUDIO

A SUN Microsystems desenvolveu o JM Studio, ferramenta que utiliza as funcionalidades do JMF permite a criação e gerenciamento de sessões RTP para áudio, vídeo ou ambos. Com esta ferramenta é possível capturar dispositivos como microfone ou webcam, podendo realizar transmissão e recepções na Internet ou rede local.

A figura 22 mostra a ferramenta do framework JMF sendo apresentada sua tela principal após sua execução.



Figura 22: Interface do JmStudio após ser executado.

O JMStudio é uma importante ferramenta que permite a reprodução, captura, armazenamento, transmissão ou recepção de mídias. Quando o JMF é instalado, um ícone do JMStudio é automaticamente criado no desktop.

Com as funções de reprodução do framework JMF no JmStudio, é possível abrir arquivos de áudio ou vídeo no formato suportado pelo aplicativo, pode ser visto na figura X.



Figura X: Mídia de vídeo sendo reproduzida no JmStudio.

Na figura X, é demonstrado como exportar a mídia em diferentes formatos e opções de configurações desejadas pelo usuário.

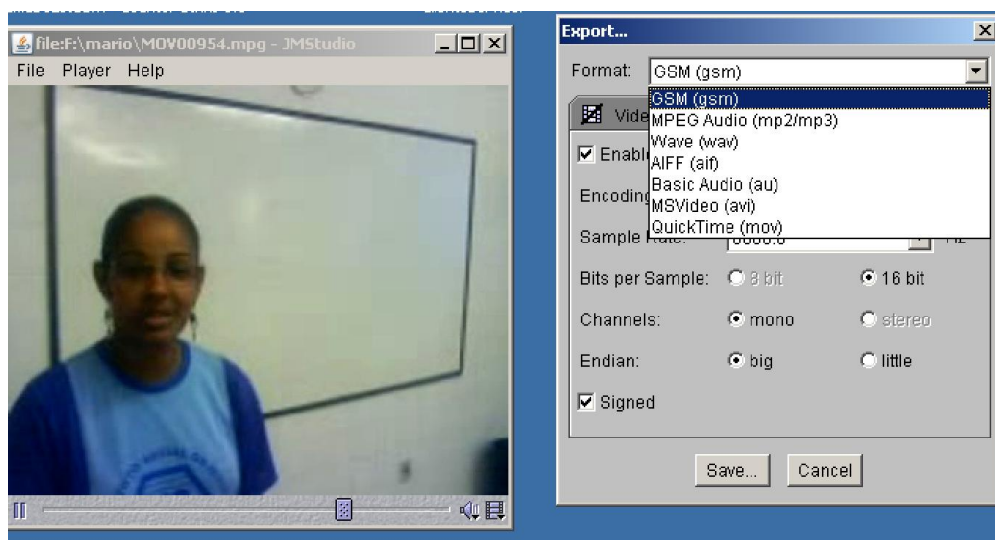


Figura X: Janela que mostra as opções de exportação da mídia.

No menu principal do aplicativo JmStudio, demonstrado na figura 23, pode-se selecionar a opção captura de dispositivos de mídia, abrir e exportar arquivos de áudio e vídeo, criar uma nova janela do programa (New Window), abrir um arquivo de mídia, URL e sessão por RTP, além da opção de transmissão (Transmit) dos dados de áudio, vídeo ou ambos e configurações gerais da ferramenta (Preferences).

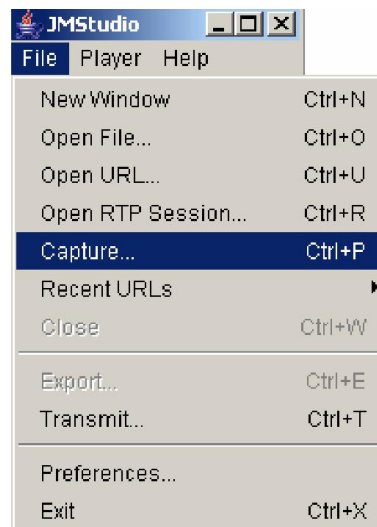


Figura 23 : Menu principal do JmStudio.

No framework JMF, também existe recursos de capturar de áudio e vídeo conforme mostra a figura 24 abaixo. Podem-se configurar os tipos de plug-ins suportados, pacotes e os tipos de dados para serem adicionados, retirados ou atualizados.

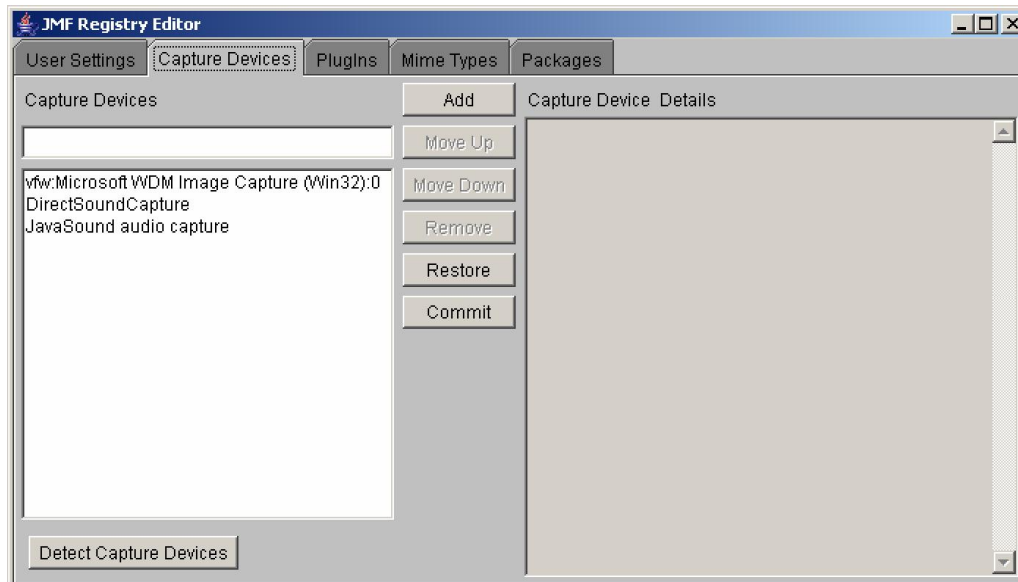


Figura 24: Visualização dos dispositivos de mídia.



A figura X mostra a janela de captura de dispositivos do JmStudio, tem a finalidade de detectar e preparar para ser utilizados.

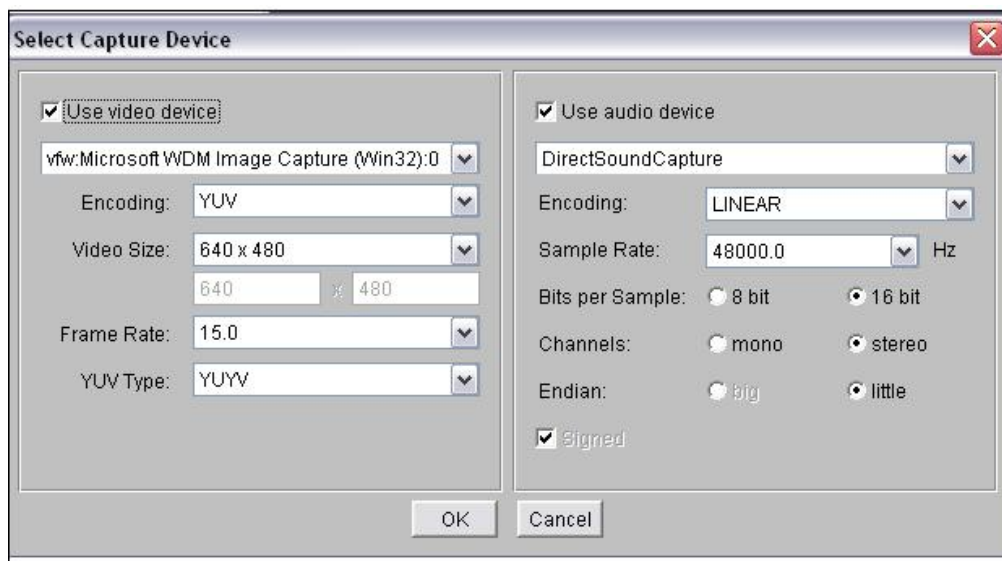


Figura X: Janela de seleção de captura de dispositivo de áudio e vídeo.

Após a capturar de dispositivos será selecionado e identificado a localização da mídia de áudio (dsound://) e vídeo (vfw://0) para ser exportado na transmissão da conferência, pode ser observado na figura X.

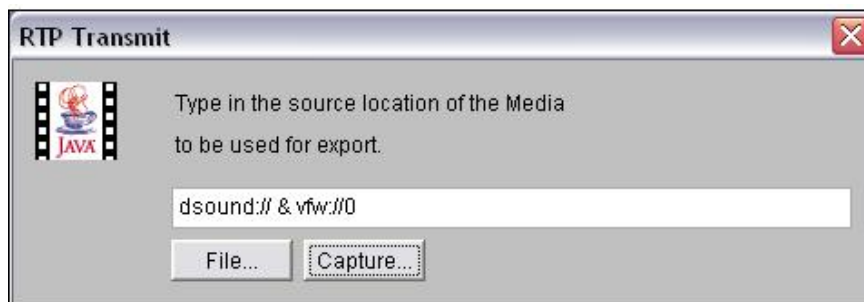


Figura X: Janela que selecionar a localização das Mídias.

Na figura X mostra a janela de abertura de sessão para receber a mídia que está sendo transmitida, poderá ser aberta uma para áudio e outra para vídeo.

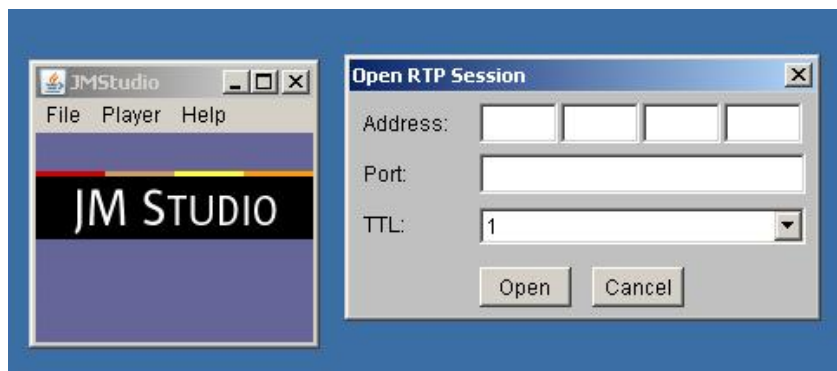


Figura X: Janela para inserção dos dados para abertura de sessão.

A figura 25 é a janela que o usuário vai digitar os endereços, porta e escolha TTL, para realizar o processo de transmissão de áudio e vídeo ao destinatário da conferência. Neste caso foi colocado o endereço IP 192.168.1.103, para ser transmitido as mídias, foi colocado a porta 7000 para vídeo, 2222 para áudio e TTL com valor 1 para ambas.

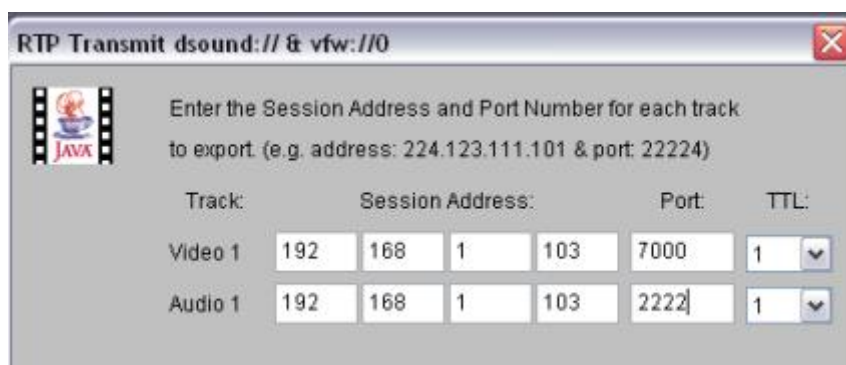


Figura 25: Entrada dos dados para a transmissão de Áudio e Vídeo utilizando o JMStudio.

A figura 27 e 22 apresenta os dados estatísticos da transmissão pelo recurso RTP ferramenta apresentada, dentre as quais: taxa de envio de pacotes, a quantidade em bytes sendo enviados, os pacotes por RTCP, locais de colisões, colisões remotas e falhas de transmissões.

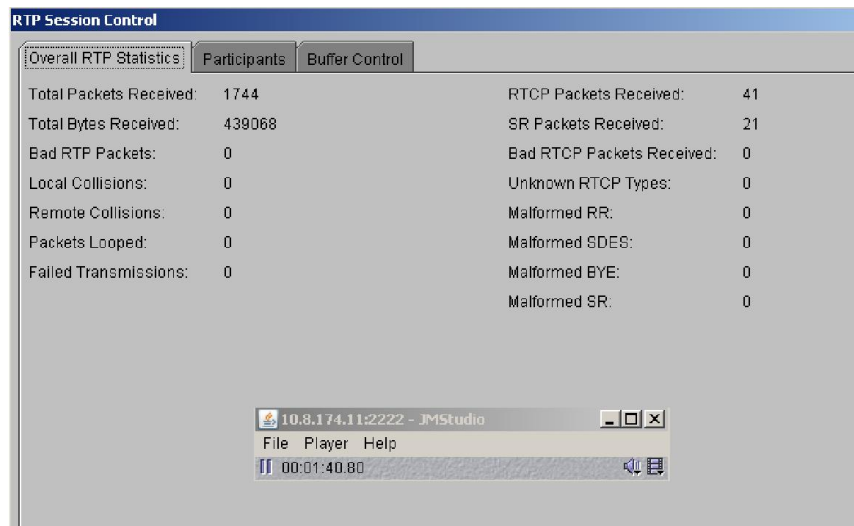


Figura 25: Dados de controle da sessão RTP.

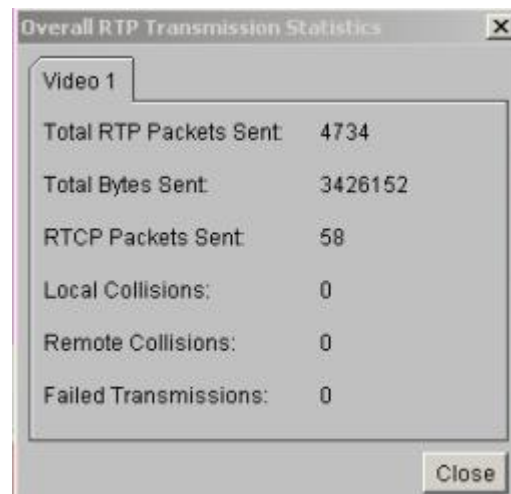


Figura 27: Transmissão de Áudio e Vídeo utilizando o JmStudio.

A figura 28 mostra o aplicativo de visualização dos plugins do Jmstudio sendo executado e visualizado todo seu processo de uma forma dinâmica.



Figura 28 : Visualização dos plug-ins em execução.

Foram realizados testes no laboratório de informática do colégio ISO com a conexão ponto-a-ponto e multicast, através de microfones, webcams e framework instalados nas máquinas.

A figura 29 mostra a realização da videoconferência, com as janelas de transmissão e recepção de áudio e vídeo utilizando o JMStudio.

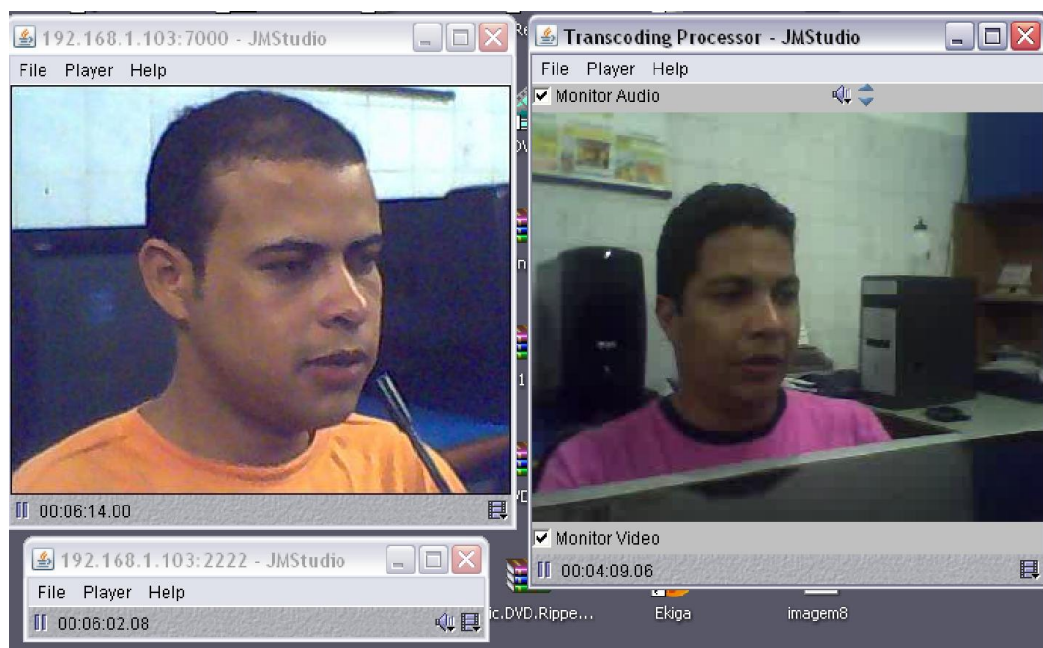


Figura 29 : Testes do transmissão e recepção da mídia no laboratório.

## 6. EKIGA

Ekiga é um aplicativo para efetuar ligações de áudio e vídeo com usuários remotos por meio dos protocolos H.323 e SIP, possui suporte para os principais recursos em VoIP e videoconferência.

O Ekiga é um projeto em Software Livre, possui suporte aos protocolos SIP (Session Initiation Protocol) e H.323 e aos atuais codecs livres de áudio e vídeo, também conhecido anteriormente como GnomeMeeting, seu criador e mantenedor do projeto, Damien Sandras, iniciou em 2000 como uma tese de mestrado na Universidade Católica de Louvain, na Bélgica, e o mantém até a atualidade.

Esta ferramenta permite que todas as características modernas de um videoconferência como suporte de provedor inteligente e fornecedor de chamadas de telefone no computador. É utilizado bastante como VoIP, onde é possível configurar números de acesso rápido, mensagens de "não perturbe" e respostas automáticas, videoconferência em tela cheia, entre outros.

A figura 30 mostra o aplicativo em execução e disponibilizando uma interface amigável aos usuários.

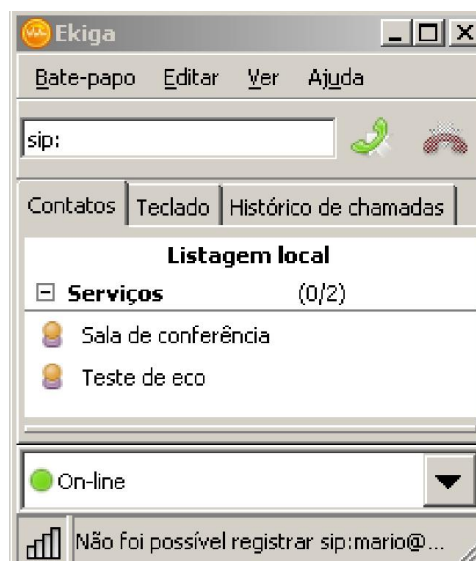


Figura 30: Interface do Ekiga em funcionamento.

O Ekiga disponibiliza um caixa de texto pra ser colocado o endereço de destino para qual vai ser conectado, neste caso foi utilizado 10.8.174.11, conforme mostra a figura X e após a confirmação será enviado uma caixa de diálogo com opções de aceitação por parte do destinatário.



Figura 30: Entrada de endereço para conexão.

A figura x mostra a caixa de dialogo enviada pelo transmissor para iniciar a conexão da videoconferência, onde poderá aceitar, rejeitar ou transferir a chamada.



Figura 30: Caixa de dialogo da chamada recebida.

No Ekiga após ser conectado com outro usuario será aparece no status conforme mostra na figura X.



Figura 30: Janela do Ekiga após se conecta com o destinatário.

No Ekiga é possível utilizar o recurso de chat para conversação, conforme mostra na figura X.



Figura 30: Interface do Ekiga do chat em funcionamento.

Existe na interface principal do Ekiga, nota-se na figura X, os itens de configuração de vídeo, áudio, teclado para discagem e estatísticas das informações dos pacotes de dados, assim como seu ordenamento e detalhes de atraso ou perda.

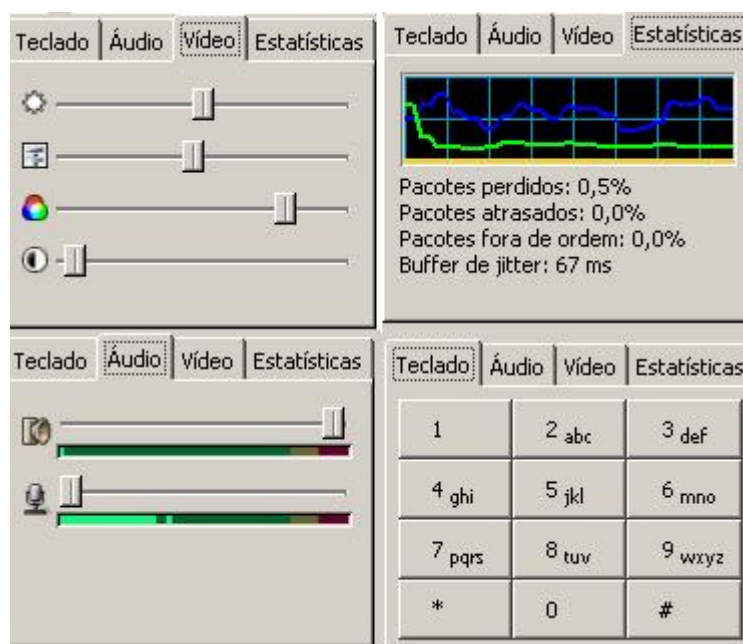


Figura 30: Janelas de configurações e acesso do Ekiga.

Com o aplicativo Ekiga pode-se adicionar contatos, função semelhante ao MSN, pode realizar as sessões de videoconferência, através da sala de conferencia e realizar chamadas com serviços de VoIP.

Através da Figura 31, pode-se notar a realização de uma videoconferência realizada pela ferramenta Ekiga.



Figura 31: Interface do Ekiga realizando a videoconferência ponto-a-ponto.



A ferramenta Ekiga possui uma interface leve e extremamente simples.

O Ekiga possui também diversos recursos como:

- Mensagens Instantâneas.
- Indicação de novas mensagens de voz.
- Algoritmo dinâmico para detecção de silêncio.
- Cancelamento de eco.
- Controle de transmissão e recepção de vídeo.
- Suporte e detecção automática de grande maioria das webcams existentes no mercado.
- Assistente de configuração.

A Figura 32 mostra a interface do Ekiga durante um teste com a ferramenta na transmissão de videoconferência, que possui várias formas de visualização das imagens que estão sendo transmitida e recebida.

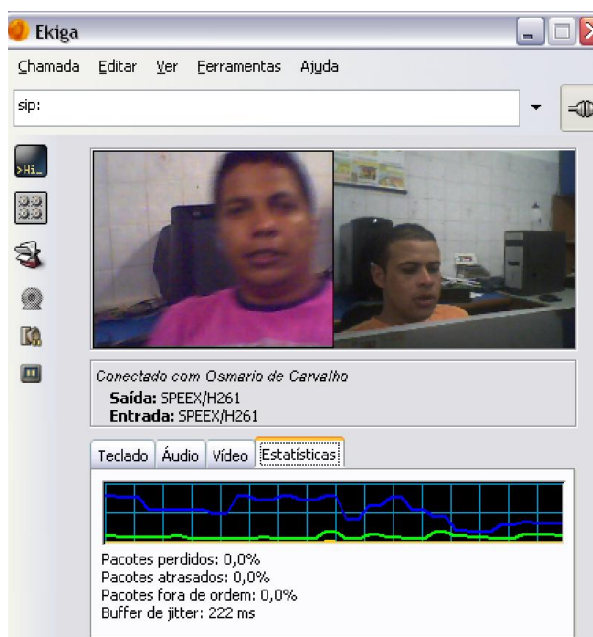


Figura 32: Ekiga realizando a videoconferência com imagens emparelhadas.

Na figura 34, pode-se visualizar a lista de histórico de chamada registrada pela ferramenta Ekiga, com o recurso de gravação automática da data e horário da chamada.

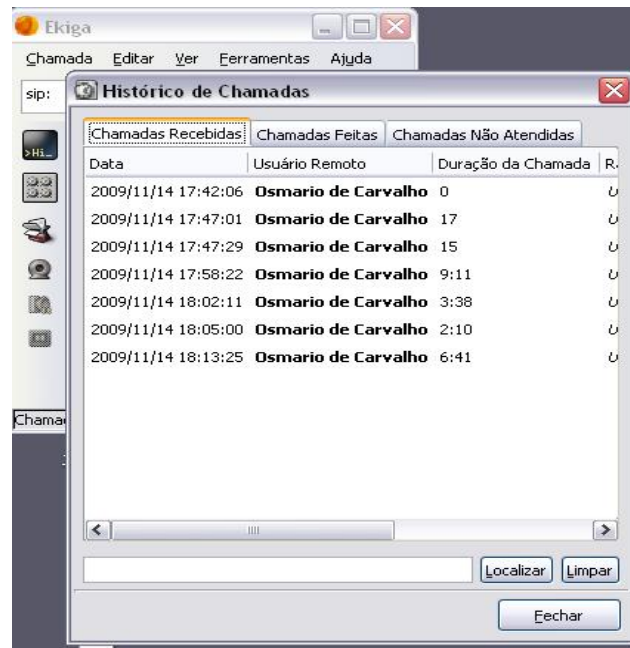


Figura 34: Tela de histórico de chamadas do Ekiga.

É possível realizar discagem para ligações com recursos VoIP, como pode ser visto na figura 35, esta função do aplicativo serve para ligar e se conectar com outras pessoas.



Figura 35: Interface do Ekiga para discagem.

Podemos ver na Figura 36, os tipos de configurações que podem ser realizados:

- Dados pessoais (Modificar o nome do usuário).
- Configurações gerais (configurar status e janela do usuário conectado).

- Opções de chamada (maneira como a chamada vai ser efetuada).
- Evento de som (configurar os sons das ações efetuados pelo programa).
- Protocolos (habilitar os protocolos de áudio e vídeo).
- Áudio (Configurar os dispositivos de áudio e seu suporte aos codecs).
- Vídeo (Configurar os dispositivos de vídeo e seu suporte aos codecs).

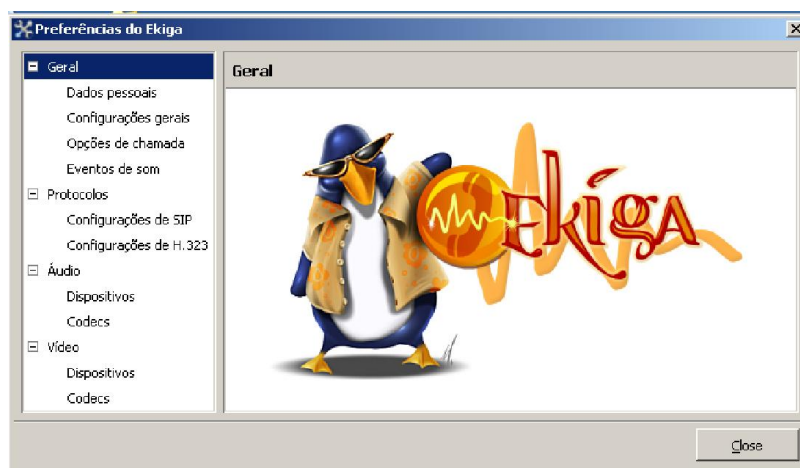


Figura 36: Tela de configuração do Ekiga.

Enfim, segundo a análise comparativa realizada de VILELA (2007), o Ekiga teve destaque entre softwares entre os demais comparados, apesar de que não possui portabilidade para o MAC.

## 7. TESTES

O teste do framework foi realizado através do JmStudio instalados em duas máquinas em rede local configuradas e preparadas para este evento. As máquinas com configuração Celeron 2.2 GHz e 1GB de RAM, além de duas webcam e dois microfones estando localizado em duas salas distintas durante os testes.

Na figura 37, pode ser visto os alunos assistindo a aula através da recepção dos dados de áudio e vídeo transmitido pelo professor em outro local.



Figura 37: Os alunos assistindo a aula pela transmissão de áudio e vídeo.

Na figura 38, temos a sala de aula com os alunos do ensino médio participando da aula e sendo transmitida em tempo real para os demais alunos.



Figura 38: Professor transmitindo a aula para os alunos.

A figura 39 mostra a interação do aluno durante o teste realizado e ocorrendo assim uma participação na aula utilizando a ferramenta JmStudio.

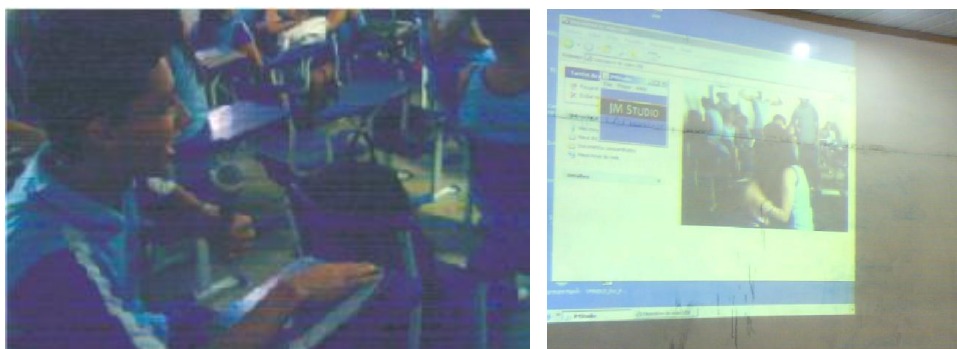


Figura 39: Aluno interagindo com o professor.

Na figura 40, mostra um dos professores que participou do teste realizado com o framework JMF, Prof. Marcus e na seqüência mostra Osmário, um dos idealizadores do teste junto ao professor.



Figura 40: O professor Marcus e o aluno graduando da Ucsal, Osmário.

A figura 41 ilustra a forma como foi executada a comunicação do teste, utilizando a conexão ponto-a-ponto entre duas salas de aula e todos os equipamentos e recursos aplicados nos testes.



Figura 41: Comunicação de grupo para grupo (sala de aula) utilizando chamada ponto-a-ponto (COOKBOOK, 2005).

O processo de instalação ocorreu da seguinte forma: Foi conectada a câmera USB e instalou-se o seu respectivo *driver*. Em seguida, foi efetuado o *download* e a instalação a última versão do Sun JDK. Na seqüência foi baixado e instalado a ultima versão do JMF em seu site oficial - <http://java.sun.com/products/java-media/jmf/>. Após o final da instalação do JMF foi iniciado automaticamente para realizar o processo de identificação dos dispositivos de áudio e vídeo instalados no computador.

## 7.1 ESTUDOS COMPARATIVOS

Segundo Mello (2006), o JMF apresentou um excelente desempenho no critério de transmissão de imagens dinâmicas, o usuário não necessita instalar nenhum software adicional, pois são incluídos arquivos “.jar ” que rodam junto ao applet para exibição de vídeos, simples de ser implementada e possui uma vasta documentação.

Na tabela 6, mostrar os resultados obtidos pela análise realizada por Mello (2006), onde podemos observar que o JMF recebeu nota máxima no critério de facilidade aos usuários do sistema e mais baixa em reprodução remota.

	JMF	Fobs4JMF	VLC	JFFMPEG	WMP
Facilidade na implementação	7	7	10	6	10
Interface agradável	7	7	6	7	10
Facilidade aos usuários do sistema	10	10	8	7	7
Reprodução remota dos formatos utilizados por parceiros	5	5	7	5	10

Tabela 6: Demonstração do resultado comparativo com JMF (MELLO, 2006).

Segundo CATUNDA (2005), o aplicativo Ekiga obteve o destaque na análise comparativa com várias ferramentas similares.

Conforme VILELA (2007), o Ekiga foi considerado destaque entre software Softphones (Aplicativos voltado para Voip), pelos recursos, estabilidade e facilidade de uso apresentada nos testes.

Na Tabela 6, podemos visualizar os dados avaliativos da ferramenta Ekiga conforme o estudo realizado por CATUNDA (2005). Definiu os seus parâmetros e avaliou a ferramenta e obteve o resultado apresentado na tabela 6.

<b>Recursos disponíveis</b>	Áudio, vídeo, <i>chat</i> , transferência de arquivos, <i>whitboard</i> (para desenhos), compartilhamento de aplicativos, cancelamento de eco. O bate papo e desenhos podem ser salvos em arquivos.
<b>Qualidade de áudio e vídeo</b>	Básica.
<b>Controle de acesso</b>	Por detecção de silêncio.
<b>Modelos de comunicação</b>	Modelo centralizado e descentralizado.
<b>Plataformas suportadas</b>	<i>Linux, freeBSD e Windows</i>
<b>Licença</b>	Gratuita.
<b>Comunicação e cenário</b>	Sistema <i>desktop</i> ponto-a-ponto e multiponto.

Tabela 6: Resultado de avaliação do Ekiga (CATUNDA, 2005).

Com base na tabela x e X, é importante considerar um dos objetivos deste projeto que é comparar a ferramenta JmStudio com o aplicativo Ekiga para separar os recursos ausentes no protótipo do framework do JMF e através das idéias fundamentada pelos testes realizada no colégio, levantar as necessidades destes recursos no ambiente EAD para o uso de videoconferência, com isso recomendar para futuras implementações dos novos aplicativos para atender as necessidade desta demanda.

Recursos disponíveis	Áudio, Vídeo, reprodução de mídia
Plataformas suportadas	Linux, Windows, Mac e outros
Licença	Gratuita
Comunicação e cenário	Sistema desktop ponto-a-ponto e multiponto

Na tabela X é demonstrado as informações sobre o JmStudio, mostrando os recursos disponíveis, plataformas suportadas, licença, comunicação e cenário.

Tabela X: Informações detalhada da ferramenta JmStudio.

## 7.2 ANÁLISE DOS RESULTADOS

Esta pesquisa realizada tem como objetivo testar os conceitos de videoconferência no JmStudio afim de verificar as funções do JMF na prática, os alunos e professores responderam os questionários e esses dados foram tratados e colados na forma de tabela para demonstrar o seu resultado.

No questionário voltado para os professores separamos 8 critérios como parâmetros de teste que foram: interface, usabilidade, qualidade de vídeo, qualidade áudio, erro/falha, participação, dinâmica e aplicabilidade. Com base nestes parâmetros a ferramenta JmStudio foi testada após a transmissão de áudio e vídeo aplicado na aula. Nos critérios interface e usabilidade obteve uma aceitação de 50% dos professores, com relação a qualidade de áudio, vídeo e participação teve 75%, e nos parâmetros dinâmica, aplicabilidade e não erro/Falha tem 100% de aprovação.

A tabela 7 representa os dados estatísticos das respostas obtidas do questionário aplicado aos professores participantes.

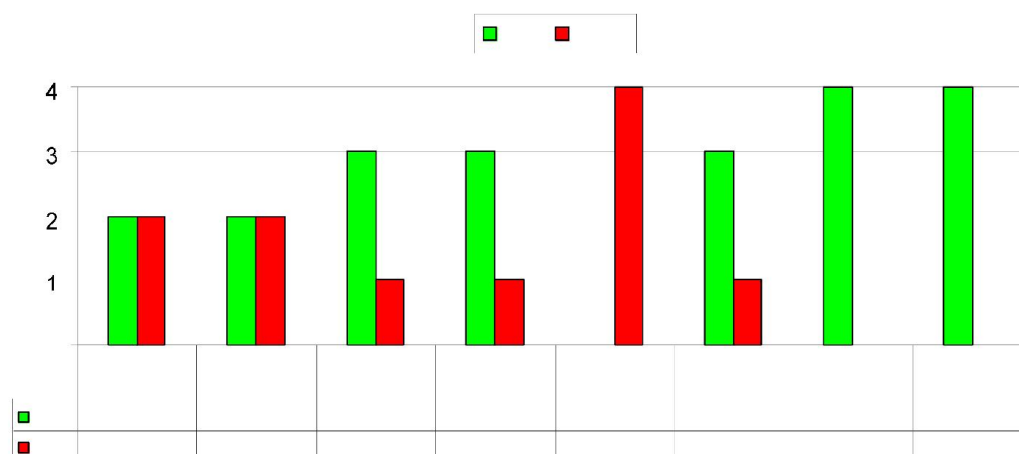


Tabela 7: Demonstração do resultado da pesquisa aos professores.

No questionário entregue aos alunos foi separado 6 critérios como parâmetros de teste que foram: qualidade de vídeo, qualidade áudio, erro/falha,



participação, dinâmica e aplicabilidade. Com base nestes parâmetros a ferramenta JmStudio foi testada após a transmissão de áudio e vídeo aplicado na aula. Nos critérios as seguintes aceitações 67% na qualidade de vídeo, 64 % qualidade de áudio, 52% participação, 54% dinâmica, 71% aplicabilidade e 100% de não erro/falha.

A tabela 8 representa os dados estatísticos das respostas obtidas do questionário aplicado aos alunos participantes.

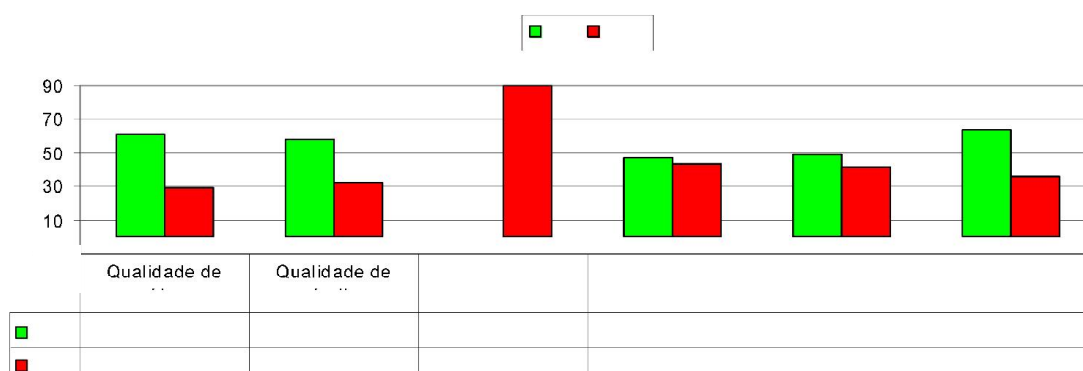


Tabela 8: Demonstração do resultado da pesquisa aos alunos.

Os resultados dos testes tiveram uma boa aceitação por partes professores e alunos do colégio ISO, sabendo que foram separados alguns pontos importantes que foram observados por parte dos professores durante os testes, a interface e usabilidade podem ser melhorados para facilidade de seu uso, com isso é preciso uma atenção especial no nestes critérios para futuras implementações para novas aplicações.

## CONCLUSÃO E TRABALHOS FUTUROS

A EAD tem se mostrado eficiente na disseminação da informação e socialização do conhecimento, oferecendo a oportunidade de mais pessoas terem acesso ao conhecimento. Com a utilização de videoconferência na Educação a distância diversas organizações e empresas estão aderindo esta tecnologia, que tem causado um impacto positivo no desenvolvimento de soluções para este fim.

Neste projeto foi possível aprender os conceitos que envolvem a videoconferência, assim como, possibilitando na prática com a utilização da ferramenta JmStudio. Com uma configuração simples e utilizando-se ferramenta gratuita de grande portabilidade e com os recursos necessários para realizar o teste de videoconferência em EAD.

Este trabalho apresentou o resultado de pesquisa realizada com o teste do framework JMF numa instituição de ensino, foi realizada a videoconferência entre duas salas distintas, com estes dados foi possível testar o funcionamento do framework JMF e orientar nas futuras implementações, desenvolvimento de aplicações multimídia e soluções de videoconferência.

No resultado do teste observou-se uma aprovação da maioria dos alunos e professores nos critérios de qualidade de vídeo e áudio, assim como participação, dinâmica e aplicabilidade. Também não houve erros ou falhas na conexão de voz/vídeo utilizando RTP, sabendo que a transmissão e recepção dos dados ocorreram com um bom desempenho com baixo atraso ou baixo jitter.

Ao término do desenvolvimento deste projeto, pode-se constatar com os estudos produzidos por esta monografia serão importantes para o desenvolvimento de novos sistemas de videoconferência em EAD mais customizada e completa.

As idéias e recomendações para futuras implementações e trabalho futuro com base nas necessidades do JmStudio encontrada e que também serão importante para o desenvolvimento do framework JMF, segue abaixo a lista de recursos a serem desenvolvido:

- Capacidade de alterar o tamanho da janela de vídeo.
- Capacidade de receber imagens sem o hardware de vídeo instalado localmente.
- Capacidade de copiar imagens de vídeo para a área de transferência.
- Capacidade de ajustar a qualidade do vídeo.
- Transferência de Arquivos em segundo plano nas conexões ponto-a-ponto ou multiponto.
- Permissões para interrupções ou participações dos usuários.
- Criação e gravação de uma lista ou histórico das sessões realizadas.
- Opção de gravação da sessão de áudio e vídeo.

- Emparelhamento das imagens de vídeo, sendo transmitido entre participantes da conferência.
- Ordenamento da lista de solicitação de participação.
- Whitboard, uma área de desenho que permite que os usuários possam importar imagens gráficas ou fazer anotações.
- Chat, onde se pode enviar e receber mensagens de texto livremente durante uma sessão.
- Uma interface amigável, para tornar as funcionalidades mais fáceis de ser executada.
- Acesso mais dinâmico para as funções a serem mais executadas, conexões de transmissão, abertura de sessão e outros.
- Compartilhamento de aplicativos entre os participantes da conferência.
- Supressão automática de ruído (ANS).
- Detecção de silêncio como controle de acesso.
- Implementação de sistema via web, para ser executado na plataforma web.

## Referências Bibliográficas

BORDIGNON, Marcio. VIDEOCONFERÊNCIA: Conceitos Tecnologias e Uso, RJ: Book Express – 2001.

CATUNDA, J. A. T. . Análise Comparativa de Sistemas de Videoconferência como Alternativa ao Sistema adotado pelo Instituto CENTEC. In: V ENPPG V Encontro de Pesquisa e Pos-Graduação., 2005, Fortaleza. V ENPPG. Fortaleza, 2005.

COOKBOOK. Videoconferencing Cookbook Version 4.1. Disponível em: <[http://www.videnet.gatech.edu/cookbook.en/list\\_page.php?topic=1&url=pf](http://www.videnet.gatech.edu/cookbook.en/list_page.php?topic=1&url=pf)

[intro.html&level=1&sequence=1&name=Introduction](#)>, março de 2005. Acesso em: outubro de 2007.

FERNANDES, Jorge (2006), Natal, Bob: Java Media Framework 2.1: Programação Multimídia em Java.

GOMES, Fábio Lúcio Soares. Videoconferência: sistemas e aplicações, SC: Visual Books, 2003.

JORGE, Eduardo; VINICÍCIUS, Marcus; MARTINS, Silvio. Aplicações multimídias em Java. Java magazine, Rio De Janeiro, RJ: DevMedia Group, v. 7, n. 57, p. 60-68, maio. 2008.

LEOPOLDINO, G.M.; MEDEIROS, R. C. M. H.323: um padrão para sistemas de comunicação multimídia baseado em pacotes, NewsGeneration, volume 5, número 6, de 05 de dezembro de 2001. Disponível em: <<http://www.rnp.br/newsgen/0111/h323.html>>. Acesso em: Agosto de 2009.

LITWIN, E. (org); Das tradições a virtualidade. In: Educação a Distância: temas para o debate de uma nova agenda educativa. Porto Alegre: Artmed Editora, 2001.

JmfGuide (1999), "Java Media Framework API Guide", Disponível em: <<http://java.sun.com/products/java-media/jmf/2.1.1/guide/index.html>> Acesso em: 11/09/2009.

LOPES, Celso Oviedo da Silva (2004), VideoChat: Uma Ferramenta de Videoconferência Pessoal. Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo para obtenção do título de Mestre em Engenharia de Computação.

MELLO, M. R. ; SILVA, Danilo Vasconcelos da ; NEVES, Laís de Mendonça; MADRUGA, M. E. ; NOVAES, Magdala de Araújo . Estudo Comparativo de Tecnologias Utilizadas para Exibição de Imagens Dinâmicas em um Sistema de

Cooperação em Saúde na Web. In: XI CBIS Congresso Brasileiro de Informática em Saúde, 2006, Florianópolis. Estudo Comparativo de Tecnologias Utilizadas para Exibição de Imagens Dinâmicas em um Sistema de Cooperação em Saúde na Web, 2006.

QUELHO, RAFAEL TONICELLI DE MELLO; CARVALHO, MARCELO DE AQUINO. (2006). Desenvolvimento de software VoIP e aplicativos multimídia

RNP. Serviços de videoconferência, 2004. Disponível em:

<<http://www.rnp.br/videoconferencia>>. Acesso em: setembro de 2009.

SILAS, Rafael S. Videoconferência e o JMF. Trabalho de Conclusão de Curso de Ciência da Computação. Universidade São Francisco. Itatiba, 2006.

TANENBAUM, Andrew C. "Redes de Computadores" - 3ª edição. Ed.Campus, Rio de Janeiro, 1997.

TANENBAUM, A., Redes de Computadores. Tradução da 4ª edição americana. Editora Campus, 2003.

VIANNEY, J. A terceira geração da educação a distância no Brasil. In: Universidade Virtual no Brasil: o ensino superior a distância no país. Tubarão: ed. Unisul, 2003.