

# ArCoLIVE: uma ferramenta de código livre baseada em componentes para videoconferência

Leandro Melo de Sales, Felipe Barros Pontes,  
Luiz Eugênio Fernandes Tenório, Márcio de Medeiros Ribeiro,  
Evandro de Barros Costa, Henrique Pacca Loureiro Luna

<sup>1</sup>Departamento de Tecnologia da Informação - TCI  
Universidade Federal de Alagoas - UFAL  
Campus A. C. Simões, BR 104 - Norte, Km 97.  
Tabuleiro dos Martins - Maceió/AL. CEP: 57072-970

{leandro, felipe, marcio}@labpesquisas.tci.ufal.br  
left@cluster.ufal.br, {evandro, pacca}@tci.ufal.br

**Abstract.** *The multimedia tools for videoconference and voice over IP systems, are getting increase in the distributed systems area, mainly by the newest technologies in network infra-structure. This paper proposes a free client/server environment written in Java, which uses the component-based approach adopted in many systems for develop robust applications with a high level code reuse.*

**Resumo.** *As ferramentas multimídia para sistemas de videoconferência e telefonia IP vêm ganhando espaço na área de sistemas distribuídos, principalmente pelos grandes avanços tecnológicos em infra-estrutura de rede. Este artigo propõe um ambiente cliente/servidor de código livre escrito em Java, a qual utiliza uma abordagem de desenvolvimento baseada em componentes, paradigma este amplamente adotado com o objetivo de construir aplicações com alto grau de reuso.*

## 1. Introdução

Nos últimos anos houve um extraordinário desenvolvimento e ampla disseminação das aplicações em rede que transmitem e recebem conteúdo de áudio e vídeo através da Internet. As novas aplicações de rede multimídia como vídeo de entretenimento, telefonia IP, rádio pela Internet, sites WWW multimídia, jogos massivamente online, videoconferência, mundos virtuais, aprendizado à distância e muitas outras, compartilham as mesmas preferências como tempo de resposta curto, escalabilidade e confiabilidade.

Com este explosivo crescimento, faz-se necessário o desenvolvimento de novas ferramentas, técnicas e protocolos Internet com o objetivo de suprir algumas carências e solucionar problemas motivados por este crescimento.

Mesmo com a crescente difusão de tais ferramentas, alguns métodos de desenvolvimento mostram-se ultrapassados ou inadequados, podendo ocasionar altos índices de custo na etapa de construção do sistema bem como dificuldades na evolução e manutenção do mesmo. O desenvolvimento baseado em componentes visa a construção de aplicações com características de sistematização de desenvolvimento, tais como escalabilidade e reuso.

Neste contexto, visando minimizar o problema supracitado, este artigo propõe um ambiente cliente/servidor de videoconferência baseado em componentes e de código livre, denominado **ArCoLIVE - Live Internet Videoconference Environment**, que tem como objetivo principal apoiar o uso de um sistema de Ensino à Distância desenvolvido utilizando um *framework* para a construção de comunidades virtuais chamado **ArCo**. A implementação dos componentes foi baseada numa extensão da linguagem Java - A Java Media Framework (**JMF**) [Microsystems, 2005] - que, embora tenha enfoque em aplicações *standalone*, dá suporte à transmissão de *stream* de mídia através da rede.

O restante do artigo está organizado da seguinte forma: na seção 2. é apresentado de forma resumida o cenário atual dos sistemas de videoconferência, relacionando a utilização desse tipo de recurso colaborativo em sistemas de comunidades virtuais. Em seguida, na seção 3., descreve-se o ambiente **ArCoLIVE**, destacando os componentes desenvolvidos e o grau de portabilidade atingido. Na seção 4. é descrito o servidor **ArCoLIVE**, cujo seu principal objetivo é controlar as requisições dos componentes implementados. Em 5. é dada uma breve descrição do **ArCo**, sendo discutida também, como estudo de caso, a integração do **ArCoLIVE** ao mesmo. Por fim, na seção 6. são apresentados os resultados do estudo e as considerações finais sobre a ferramenta desenvolvida.

## **2. Contextualização**

Nesta seção é apresentada uma visão geral tanto da situação atual das ferramentas de videoconferência quanto dos aspectos utilizados no desenvolvimento do **ArCoLIVE**.

### **2.1. Cenário atual**

Atualmente, a Internet é composta essencialmente por sistemas distribuídos. O modelo cliente-servidor predomina em aplicações para Web, correio eletrônico, DNS e videoconferência, onde a forma e a ordem das informações trocadas são baseadas em protocolos.

Segundo [Monteiro, 1997], sistemas multimídia, em particular os de videoconferência, vem ganhando espaço em aplicações voltadas para a Internet. No entanto, boa parte deles não se preocupa com algumas características que permitem o reuso, a escalabilidade e a fácil manutenção, diminuindo significativamente a vida útil destes sistemas.

### **2.2. Desenvolvimento baseado em componentes**

O desenvolvimento baseado em componentes trata da criação de componentes que possam ser reutilizados em outras aplicações. Várias vantagens são obtidas com essa abordagem, tais como: reutilização através da divisão do sistema em módulos, redução do tempo de desenvolvimento com a utilização de componentes pré-fabricados e gerenciamento das dependências dos módulos através da definição de componentes dependentes.

É importante salientar que arquiteturas baseadas em componentes têm sido usadas com sucesso para tratar necessidades específicas de aplicações em sistemas distribuídos, oferecendo um mecanismo efetivo de reutilização de software [Pasin, 2003].

### **2.3. Domínio da aplicação**

Segundo [Oudshoff et al., 2003], uma comunidade virtual pode ser entendida como “uma rede social *on-line* de um grupo de pessoas com interesses em comum”. No contexto de educação, as comunidades virtuais de aprendizagem têm recebido importante atenção no âmbito do ensino à distância. Um exemplo clássico para esta abordagem seria professores interagindo e compartilhando documentos com seus alunos e, estes, por sua vez, podendo tirar dúvidas com aqueles através de trocas de *e-mails*, por exemplo.

A interação supracitada, todavia, pode ser obtida através de diversas modalidades em um ambiente de comunidades virtuais. Ferramentas de fórum, correio eletrônico e *chat* são alguns dos exemplos de mecanismos mais comuns. Além destas, merece destaque a utilização de recursos de câmeras digitais, as quais proporcionam o que hoje se chama de videoconferência.

Como se percebe, o utensílio desta abordagem ratifica a importância que o ensino à distância vem recebendo. A videoconferência tem sido utilizada para dar suporte a aulas, seminários e reuniões, liberando seus espectadores de estarem fisicamente presentes ao local onde está ocorrendo o evento.

O estudo de caso deste artigo mostra os componentes da ferramenta de videoconferência **ArCoLIVE** aplicados ao **ArCo** - Arcabouço de Comunidades - um arcabouço para construção de comunidades virtuais concebido e desenvolvido no Departamento de Tecnologia da Informação (TCI) da Universidade Federal de Alagoas (UFAL).

O **ArCo** é um arcabouço extensível e de código aberto para a construção de ambientes de comunidades virtuais. O arcabouço dispõe de ferramentas para a interação e colaboração de atores, componentes de infra-estrutura e interface gráfica, além de dar suporte a integração com outros sistemas [Almeida et al., 2004].

A construção de ambientes utilizando o **ArCo** é baseada no conceito de montagem de componentes de prateleira, os quais encapsulam serviços de comunidades virtuais bem conhecidos, tais como *chat*, fórum, indexação de conteúdo, entre outros.

Desta forma, o desenvolvedor de um sistema de comunidades virtuais que utilizar este arcabouço terá à sua disposição componentes providos pelo mesmo, tornando o desenvolvimento do sistema fácil, rápido e confiável, pois pressupõe-se que tais componentes já foram previamente testados e utilizados.

#### **2.4. JMF: Java Media Framework**

Aplicações de tempo real, como as de videoconferência, trabalham com dados que dependem de maneira significativa do tempo. Fundamentalmente, a **JMF** é uma coleção de classes que permite o processamento da mídia baseada no tempo [Terrazas et al., 2002]. A **JMF** consiste em um conjunto de funcionalidades para incorporação de multimídia às aplicações Java que abstrai do usuário o processamento da mídia, facilitando operações de baixo nível. Ela fornece mecanismos para configurar os atributos de um formato de mídia, bem como para verificação da compatibilidade do formato de um arquivo, tendo sido desenvolvida com o objetivo de ser fácil de programar e permitir o desenvolvimento de *plug-ins*.

Essas características podem ser aplicadas em mídia originada de um arquivo, dispositivo de captura ou até de um sistema final remoto. Em uma rede IP pode-se transferir pacotes através de TCP ou UDP. Entretanto, por conta do comportamento do *stream* de multimídia com relação ao tempo, o uso de TCP torna-se inaceitável em muitos casos. Uma solução mais adequada é utilizar o protocolo UDP. Um modo geralmente usado para transmissão de *streams* de mídia é combinar a utilização do protocolo RTP (*Real-Time Protocol*) [RFC1889, 1996], que fragmenta a informação multimídia, adiciona a cada fragmento informações de seqüência e de tempo de entrega e o protocolo UDP, que é usado para o transporte deste *stream*. A **JMF** oferece suporte a estes protocolos para transmissão de fluxos multimídia pela rede.

Aproveitando a portabilidade da linguagem *Java*, ferramentas de videoconferência desenvolvidas com a **JMF** podem ser executadas em celulares ou *handhelds*. Atualmente, existe um vasto mercado de dispositivos móveis. A cada dia, estes dispositivos oferecem novas características aos usuários finais. Incremento na taxa do clock do processador e memória possibilitam a execução de tarefas, tais como: tocar arquivos MP3, criar slides, acessar a Internet, ler mensagens de correio eletrônico ou ainda participar de uma videoconferência [A. Belda and Cermeño, 2004].

### **3. A ferramenta ArCoLIVE**

Nesta seção é apresentado um conjunto de componentes que foram desenvolvidos para suportar as necessidades apresentadas anteriormente. Além disso, é descrito como tais componentes foram modelados e desenvolvidos. Por fim, são apresentados os componentes do conjunto **ArCoLIVE**, acompanhados de uma breve descrição.

Alguns exemplos de ferramentas que podem ser desenvolvidas utilizando o **ArCoLIVE** são: ferramentas para controle de segurança eletrônica, administração de ambientes

empresariais, reconhecimento de padrões em vídeos capturados (*face-track* por exemplo), salas de aulas virtuais, videoconferência multi-ponto ou qualquer outra ferramenta, onde os recursos de colaboração visual podem ser aplicados.

### 3.1. Arquitetura

A arquitetura do **ArCoLIVE** é dividida em três camadas principais: aplicação, servidor e persistência, como mostrado na Figura 1. Na camada de aplicação está implementado o **ArCoLIVE-T**, na qual os componentes interagem entre si. A camada **ArCoLIVE-S** (ver Seção 4.) é responsável por tratar as requisições e respostas dos componentes, autenticação de usuários e prover uma interface entre as camadas de aplicação e de persistência, ambas baseadas no **ArCoLIVE-P**. A camada de persistência é usada para armazenar perfis dos participantes, descrições e permissões de um serviço e algumas propriedades das mídias para vídeo sob demanda.

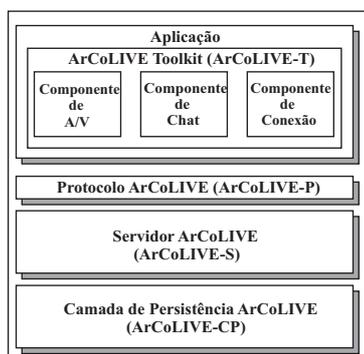


Figura 1. Arquitetura do ArCoLIVE

### 3.2. Desenvolvimento dos componentes

Os componentes foram desenvolvidos utilizando a linguagem de programação Java combinada com a extensão **JMF**. Com o objetivo de resolver problemas de padronização, a *Sun* criou o modelo de componentes JavaBeans [Szyperski et al., 2002]. O **ArCoLIVE** é baseado nesta tecnologia, onde a produção dos componentes obedece a uma especificação rigorosa. A idéia básica é que estes, como em qualquer outro modelo, sejam altamente coesos, independentes e configuráveis, tornando possível a livre comercialização e reutilização.

Como uma aplicação desenvolvida utilizando os componentes **ArCoLIVE** não possui conhecimento direto da implementação deles, é possível alterar o código de um componente sem afetar todo o sistema. Assim, desde que as interfaces sejam preservadas, novas funcionalidades podem ser adicionadas através da simples substituição dos componentes. A figura 2 mostra a tela do IDE<sup>1</sup> Netbeans com os diversos componentes do **ArCoLIVE** descritos logo a seguir.

### 3.3. Componentes

Os componentes (**ArCoLIVE Toolkit**) desenvolvidos estão listados a seguir:

- **ArCoLIVEConnection** - o componente de conexão suporta dois protocolos para transmissão de dados, o UDP (com RTP) e o TCP. Para transmissão de áudio e vídeo, utiliza-se o componente de conexão com o protocolo UDP selecionado. Uma das importantes características da atual implementação deste componente é o suporte a *multicast* quando o protocolo UDP é selecionado.

<sup>1</sup>Integrated Development Environment

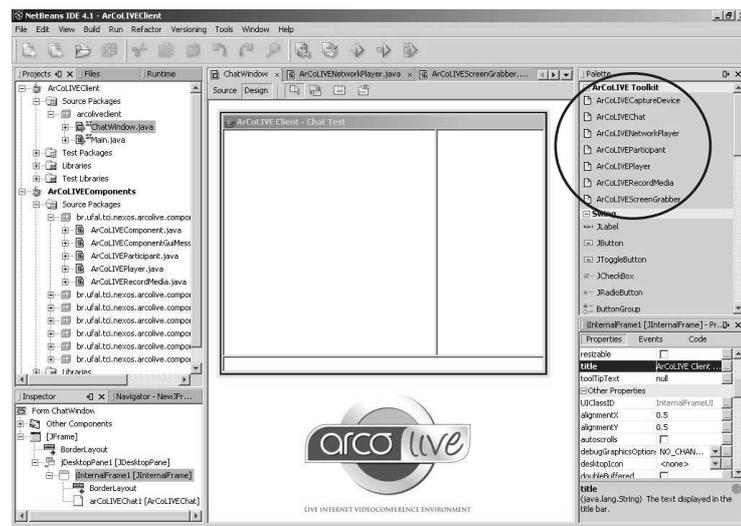


Figura 2. Conjunto de componentes do ArCoLIVE

- **ArCoLIVEConnectionManager** - utilizado quando se deseja obter informações da conexão em questão. As informações do estado da conexão são obtidas através do protocolo RTP. Este componente provê dados de quantos pacotes RTP foram enviados e recebidos, assim como a taxa de perda de pacotes.
- **ArCoLIVEPlayer** - é um componente para reprodução local de fluxos de áudio e vídeo. A principal característica deste componente é abstrair a fonte geradora do fluxo de dados, seja um arquivo, uma *webcam* conectada ao computador ou um microfone. Este componente pode interagir com outros, como o ArCoLIVERecordMedia, por exemplo.
- **ArCoLIVENetworkPlayer** - é um componente para reprodução de fluxos de áudio e vídeo oriundos de um servidor qualquer.
- **ArCoLIVEProcessor** - componente que trata alguns aspectos de controle e processamento, tais como conversão de formatos, inserção de imagens em vídeos etc.
- **ArCoLIVEChat** - este componente oferece recursos para conversação através de texto. Um recurso importante provido por este componente é a possibilidade de moderar uma sessão de *chat*.
- **ArCoLIVECaptureDevice** - componente responsável por gerenciar todos os dispositivos de captura de mídia do sistema local. Com este componente é possível obter a lista de todos os dispositivos multimídia instalados em um computador, assim como prover acesso a eles.
- **ArCoLIVEParticipant** - componente que representa o participante da conferência. Dados simples como nome, e-mail, login e outras informações que o descrevam podem ser obtidas através deste componente.
- **ArCoLIVERecordMedia** - ao obter uma mídia, o usuário pode salvar o fluxo de dados capturado através deste componente. É possível informar a localização do arquivo a ser gerado e seu formato.

#### 4. O Servidor ArCoLIVE

Com o objetivo de oferecer suporte aos componentes apresentados na seção 3.3. foi desenvolvido um servidor de componentes que possibilita a comunicação de grupos de usuários conectados na rede, em particular a Internet, através de serviços de áudio/vídeo, *chat* e outros recursos de colaboração visual assim como gerenciamento dos serviços disponibilizados.

O servidor oferece recursos com o objetivo de suportar as diversas requisições destes componentes. Os principais serviços oferecidos pelo servidor são: conferência, bate papo, compartilhamento de arquivo, quadro branco virtual e captura de tela.

O servidor possui um módulo de gerenciamento de serviços, o qual o administrador do sistema utiliza-o a fim de controlar o acesso aos usuários, criação de instância dos serviços e disponibilidade dos mesmos. As figuras 3(a) e 3(b) mostram as telas do gerenciador do servidor.

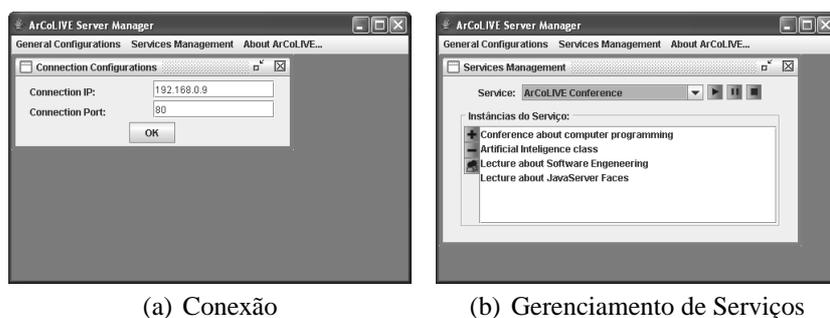


Figura 3. Módulo de Gerenciamento do Servidor

Como solução para comunicação padronizada, foi especificado e implementado um protocolo, o qual é utilizado para troca de informações e requisições entre cliente e servidor. Este protocolo pode ser apresentado como na figura 4.

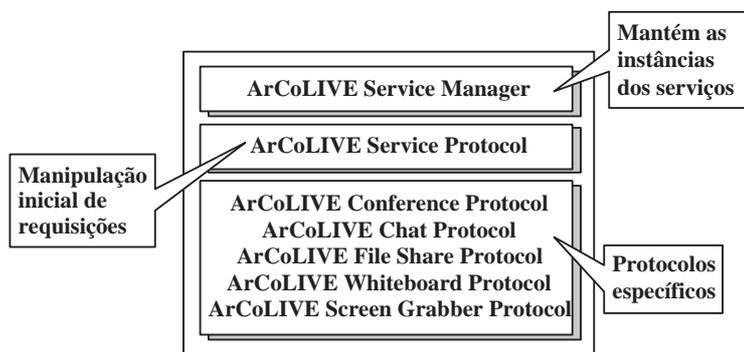


Figura 4. Protocolo de Comunicação do ArCoLIVE

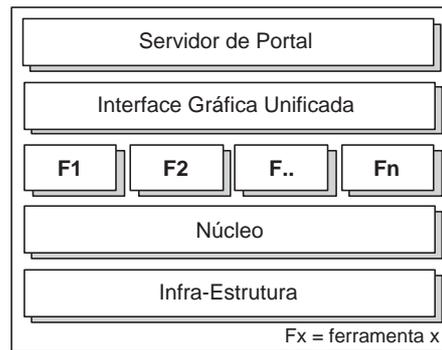
O servidor é baseado no padrão de projeto observer [Gama et al., 2000], que redireciona ao módulo de controle específico uma requisição de um usuário conectado a ele.

## 5. Estudo de caso: aplicação do ArCoLIVE a um ambiente de comunidades virtuais

### 5.1. A arquitetura do ArCo

A arquitetura do arcabouço **ArCo** consta de cinco camadas, conforme apresentada na figura 5.

- **Servidor de Portal:** as ferramentas de uma comunidade são disponibilizadas através de portais implementados pelo arcabouço *JetSpeed*, onde os usuários podem personalizar seus ambientes, assim como fornecer diferentes visões para as comunidades.



**Figura 5. Arquitetura do ArCo**

- **Interface Gráfica Unificada:** garante a identidade visual do ambiente. Esta camada utiliza a arquitetura *Model-View-Controller*, a qual é implementada pelos arcabouços *Struts* e *JSF*;
- **Ferramentas:** ferramentas como fórum são representadas nesta camada. Esta é a camada de enfoque deste artigo, pois o **ArCoLIVE** foi implementado como uma ferramenta para o **ArCo** provendo serviços de videoconferência para os usuários das comunidades virtuais;
- **Núcleo:** gerencia as atividades entre os entes básicos da comunidade virtual, tais como ator (usuário), comunidade (grupo de usuários com interesses em comum) e ferramentas de gerenciamento;
- **Infra-estrutura:** atende aos requisitos não funcionais do sistema, tais como persistência de objetos e autenticação de usuários;

## 5.2. Integrando o ArCoLIVE ao ArCo

Para ser integrada ao arcabouço, a ferramenta **ArCoLIVE** necessitou apenas implementar interfaces para serem utilizadas no gerenciamento do ciclo de vida da ferramenta e notificação dos eventos ocorridos no núcleo. Esses eventos, por sua vez, são repassados ao **ArCoLIVE** que os tratará de forma personalizada.

O **ArCo** não determina qual deve ser a arquitetura ou estrutura das ferramentas e, desta forma, qualquer ferramenta existente pode ser facilmente integrada, flexibilizando a adição de novas funcionalidades.

A figura 6(a) mostra o componente **ArCoLIVEPlayer** para reprodução de mídia, ao passo que a figura 6(b) mostra o componente **ArCoLIVEChat**, utilizado para comunicação textual entre usuários.



(a) ARCO.VIDEO

(b) ARCO.CHAT

## 6. Considerações Finais

Neste artigo foi mostrada uma solução portátil e de fácil utilização de um conjunto de componentes para a implementação de ferramentas para videoconferência.

A ausência de recursos multimídia em ambientes de comunidades virtuais, motivou significativamente a integração do **ArCoLIVE** ao arcabouço **ArCo**. Devido ao alto grau de independência dos componentes desenvolvidos, é possível utilizá-los em qualquer outro ambiente extensível que necessitem de tais recursos.

Portanto, a ferramenta discutida neste artigo oferece uma solução de baixo custo, principalmente por se tratar de uma ferramenta de código livre. Assim sendo, aplicações que utilizam o **ArCoLIVE** podem melhorar a comunicação entre as pessoas, contribuindo de modo significativo para empresas ou instituições que necessitam de soluções de comunicação multimídia em seus ambientes.

## Referências

- A. Belda, J. C. Guerri, A. P. and Cermeño, J. J. (2004). JMF-Mobile: A new system to access multimedia contents from handheld devices through internet.
- Almeida, H. O., Tenório, L. E. F., Costa, E. B., Barbosa, N. M., Bublitz, F. M., and Barbosa, A. A. (2004). Um arcabouço de software livre baseado em componentes para a construção de ambientes de comunidades virtuais de aprendizagem na web. In *XV Simpósio Brasileiro de Informática na Educação*, pages 188–197, Manaus-AM.
- Gama, E., Helm, R., Johnson, R., and Vlissides, J. (2000). *Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objeto*. Bookman, 1st edition.
- Microsystems, S. (2005). JMF - Java Media Framework. Site oficial do projeto JMF/SUN. <http://java.sun.com/products/java-media/jmf/>. Último acesso, Janeiro de 2005.
- Monteiro, J. A. S. (1997). O estado atual da pesquisa e desenvolvimento em sistemas distribuídos no brasil. *Laboratório Nacional de Redes de Computadores - Publicação eletrônica*. Último acesso, Junho de 2005.
- Oudshoff, A. M., I. E. Bosloper, T. B. K., and Spaanenburg, L. (2003). Knowledge discovery in virtual community texts: clustering virtual communities. *Journal of Intelligent & Fuzzy Systems*, 14(1):13–24. IOS Press.
- Pasin, M. (2003). *Réplicas para Alta Disponibilidade em Arquiteturas Orientadas a Componentes com Suporte de Comunicação de Grupo*. PhD thesis, Universidade Federal do Rio Grande do Sul - UFRS.
- RFC1889 (1996). RTP: A Transport Protocol for Real-Time Applications. <http://www.ietf.org/rfc/rfc1889.txt>. Último acesso, Janeiro de 2005.
- Szyperski, C., Gruntz, D., and Murer, S. (2002). *Component Software, Beyond Object-Oriented Programming*. ACM Press, 2nd edition.
- Terrazas, A., Ostuni, J., and Barlow, M. (2002). *Java Media APIs, Cross-plataform Imaging, Media and Visualization*. SAMS, 1st edition.