

# UM SISTEMA DE BAIXO CUSTO PARA CONTROLE DE CÂMERA IP VIA INTERNET

**Diego BOMFIM ANDRADE (1); Leonardo BARRETO CAMPOS (2)**

(1) Instituto Federal de Educação Tecnológica da Bahia – Campus de Vitória da Conquista, Av. Amazonas 3150 - Zabelê - Vitória da Conquista, e-mail: diegoandrade.eng@gmail.com

(2) Instituto Federal de Educação Tecnológica da Bahia – Campus de Vitória da Conquista, Av. Amazonas 3150 - Zabelê - Vitória da Conquista, e-mail: leonardobcampos@ifba.edu.br

## RESUMO

O monitoramento remoto de ambientes utilizando câmeras tornou-se indispensável no setor de segurança. Porém, para obter um sistema com câmeras controláveis, acessível pela Web, escalável e eficaz, significará alto custo. Nesse sentido, o presente artigo apresenta um sistema de monitoramento de imagens e vídeos que possui como principal característica o acesso remoto, eficiente e de baixo custo. O principal objetivo do projeto é especificar uma arquitetura baseada no modelo cliente/servidor que comunique via Web e possibilite, aos aplicativos cliente, controlar motores de passo responsáveis pelo movimento das câmeras IPs conectadas ao servidor. Dessa forma, o sistema é capaz de provê movimento às câmeras projetadas inicialmente para permanecerem fixas.

**Palavras-chave:** Sistemas Multimídia, Web, JMF, RTP.

## 1 INTRODUÇÃO

As tecnologias que dão suporte a Web permitem o desenvolvimento de sistemas cada vez mais poderosos e utilitários como, por exemplo, os sistemas de instrumentação e medida baseados em computador. A instrumentação já é realidade no cotidiano das pessoas, pois está integrada aos aparelhos eletro-eletrônicos de fácil acesso, em forma tal que propicia a criação de ambientes inteligentes.

Os ambientes inteligentes vêm consolidando-se rapidamente e avança, sobretudo, em residências, onde o uso dessa tecnologia destaca-se por fornecer aos seus usuários mais conforto, segurança e economia. Todas as vantagens decorrem da forma de intervenção sobre o ambiente que é bastante simples, basta estar com um computador ou qualquer outro controle com capacidade de acessar o ambiente para exercer controle sobre ele, não exigindo assim, a presença física do operador.

Um aliado dos ambientes inteligentes controlados remotamente é a possibilidade desses ambientes serem assistidos vinte e quatro horas por dia através de câmeras. Praticamente, todas as organizações, municípios, instituições e negócios consideram a segurança e o monitoramento de um ambiente como uma de suas principais prioridades. Nesse contexto, este artigo apresenta um sistema baseado no modelo cliente/servidor que controla remotamente dois motores de passo responsáveis pelos graus de liberdade das câmeras IPs instaladas no servidor.

Para obter uma melhor compreensão do sistema apresentado, este artigo está dividido da seguinte forma: a seção 2 faz uma análise crítica sobre dois sistemas de monitoramento existentes, apresentando seus pontos positivos e as brechas deixadas pelos autores para projetos futuros. Nas seções 3, 4 e 5 serão apresentados os principais conceitos compreendido pelo modelo cliente/servidor, a API *Java Media Framework* (JMF) e o protocolo *Real-Time Transfer Protocol* (RTP), respectivamente. Na seção 6 será descrita a arquitetura padrão do sistema. Em seguida, na seção 7, serão apresentados os aplicativos cliente e servidor bem como o funcionamento do sistema. Finalmente, a seção 8 apresenta as considerações finais e os trabalhos futuros.

## 2 SISTEMAS DE MONITORAMENTO

Com os avanços tecnológicos na infraestrutura das redes de computadores, experimentos que antes eram controlados localmente, começaram a ser disponibilizados na Internet, possibilitando acesso a serviços e recursos independente da posição geográfica dos seus usuários. Nas seções a seguir será apresentada uma

análise crítica dos sistemas que permitem acesso remoto, seus pontos positivos e as brechas que podem ser exploradas para melhorias.

## 2.1 RobWebCam

O projeto RobWebCam é uma das primeiras aplicações desenvolvidas pelo Grupo de Automação e Controle (GRACO) da Universidade de Brasília. Este projeto consiste num sistema com a finalidade de manipular uma câmera de vídeo com dois graus de liberdade (Norte-Sul; Leste-Oeste) e enviar as imagens capturadas por essa câmera através da Internet para um cliente conectado via browser. O sistema RobWebCam contém uma placa de circuito impresso responsável por controlar os movimentos da câmera e por passar o status do hardware para o servidor através da interface paralela (GRACO 1998). A câmera possui alimentação própria de energia e é interligada ao servidor através de um cabo de sinal de vídeo conectado na placa de captura de vídeo do Servidor. No servidor estão alocados o *drivers* para receber as imagens e as páginas WWW(World Wide Web) utilizadas para teleoperar o sistema. O cliente, via browser, recebe as imagens e os comando deste servidor via Internet.

Apesar de apresentar inúmeras características inovadoras o RobWebCam não é escalável, ou seja, não possibilita que novas câmeras sejam associadas ao servidor. Outra falha apresentada pelo projeto é a falta de perfis de utilização sobre o hardware, dessa forma, todos os usuários são tratados de forma igual não havendo, portanto, distinção entre usuários comuns e o administrador do sistema.

## 2.2 O Automated Internet Measurement Laboratory (AIM-Lab)

O Laboratório da Universidade Norueguesa de Ciência e Tecnologia é voltado para alunos de graduação, pós-graduação e pesquisadores e contém, entre outros módulos, um curso de característica de dispositivos e semicondutores com acesso remoto via Web (FIELDLY 2000) bastante simples de usar, eficiente e independente de plataforma. O AIM-Lab foi baseado em um sistema de arquitetura cliente/servidor e utiliza a linguagem Visual C++ para implementação do servidor e Java para o cliente. Os comandos gerados pelo cliente, de acordo com o conjunto de parâmetros especificados pelo usuário, são enviados ao servidor através do *socket* TCP/IP (*Transmission Control Protocol/ Internet Protocol*) implementados nos dois softwares, cliente e servidor, para que a comunicação via Internet seja estabelecida.

Uma boa prática adotada nesse projeto é a utilização do Java no cliente, dessa forma, além de contar com uma interface amigável, diversos recursos de conexão e transmissão de dados pela Internet são explorados. Porém, as câmeras utilizadas no Laboratório de Medidas da Universidade Norueguesa de Ciência e Tecnologia possuem graus de liberdade oriundos de fábrica, sem que exista uma redução no custo do projeto.

## 3 MODELO CLIENTE/SERVIDOR

O principal padrão de interação entre os aplicativos de cooperação de rede é conhecido como paradigma cliente/servidor (KUROSE, 2003)(TANEMBAUM, 2001). O termo servidor aplica-se a qualquer programa que ofereça um serviço passível de ser alcançado através de uma rede. Um servidor aceita uma solicitação efetivada da rede, executa seu serviço e retorna o resultado ao solicitante (cliente). A Figura 1 apresenta uma arquitetura simplificada do modelo cliente/servidor.

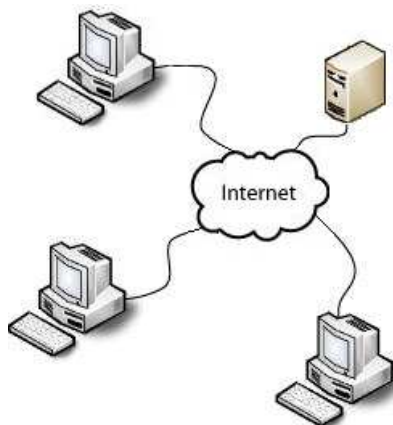


Figura 1 - Exemplo de Modelo Cliente/Servidor

Para implementar uma solução cliente/servidor em Java é comum o uso de soquetes. Segundo a Sun (1998), um soquete “é um ponto de extremidade de um link de comunicação bidirecional entre dois programas em execução na rede. Um soquete é ligado a um número de porta para que a camada TCP (*Transmission Control Protocol*) possa identificar o aplicativo e que dados estão destinados a ser enviado”. O Java fornece classes de soquetes que são usadas para representar a conexão entre um programa cliente e um programa servidor.

#### 4 API JAVA MEDIA FRAMEWORK – JMF

De acordo com Deitel et. al. (2003) os programas desenvolvidos na linguagem de programação Java consistem em partes chamadas classes. As bibliotecas de classes são também conhecidas como Java APIs (*Applications Programming Interfaces* – interfaces de programas aplicativos). Programas em Java que utilizam APIs podem melhorar o desempenho do programa, uma vez que essas classes e métodos são cuidadosamente escritos para rodar de maneira eficiente. Esta técnica também melhora a velocidade de prototipagem no desenvolvimento de programas, ou seja, o tempo necessário para desenvolver um programa novo e fazer funcionar a sua primeira versão.

Entre as APIs do Java o projeto apresentado neste artigo utiliza a API JMF (*Java Media Framework*). Segundo Horstmann et al. (2002), JMF “é uma coleção de classes que permitem a captura e visualização de dados multimídias em aplicações e *applets* Java”. Usando a API JMF é possível criar aplicativos Java que capturaram, produzem e editam os tipos de mídias de áudio e vídeo mais populares do mercado, por exemplo: MPEG, AVI, MP3, etc.

Além dessas vantagens a API JMF também permite “a construção de sistemas de programação em diferentes plataformas” (SUN MICROSYSTEMS 1998), ou seja, os usuários conectados ao servidor (fonte das mídias) podem ter configurações distintas de hardware e software. Esse tipo de comunicação conhecida como cliente/servidor, permite ao usuário que a transmissão de dados seja feita remotamente pelas arquiteturas distribuídas ao longo das redes de computadores. Por esses motivos, a API JMF foi definida como tecnologia de suporte para o tratamento das mídias no sistema apresentado neste artigo.

A API JMF fornece dois principais componentes de usuário padrão, são eles: (i) Componente Visual: com o qual será exibido os arquivos de mídia (vídeos capturados através da câmera IP) e (ii) Componente de Controle: que controla a apresentação do arquivo de mídia (*player*).

Para transmissões de mídias através da rede, a API JMF especifica um conjunto de instruções que permitem transmissões de dados em tempo real. O protocolo de comunicação utilizado para essas transmissões é o RTP (*Real-time Transport Protocol*). Segundo a Sun(1998), RTP oferece serviços de entrega de rede fim a fim para transmissões de dados em tempo real, o que pode ser considerado uma característica essencial no que diz respeito a monitoramento remoto como o apresentado neste artigo.

#### 5 REAL-TIME TRONSPORT PROTOCOL – RTP

De acordo com a RFC (1889), “o Protocolo de Transporte de Tempo Real (*Real-Time Transpot Protocol – RTP*) fornece funções de transporte em rede fim a fim para aplicações que transmitem dados de tempo-real como, por exemplo, áudio, vídeo ou dados de simulação”. Para iniciar uma transmissão de dados em tempo real é necessário abrir uma sessão RTP. Segundo a Sun (1998), uma sessão RTP é uma associação entre um conjunto de aplicativos que se comunicam com a RTP, sendo que esta sessão é identificada por um endereço de redes e um par de portas. Uma porta é usada para os dados de mídia e a outra é usada para dados de controle.

Ainda segundo a Sun (1998), cada tipo de mídia é transmitido em uma sessão diferente, por exemplo, se em uma conferência, uma sessão é utilizada para transmitir os dados de áudio e uma sessão separada é usada para transmitir os dados de vídeo. Como o trabalho apresentado neste artigo trata-se de um sistema de monitoramento de imagens, apenas uma sessão RTP foi criada para tornar possível a transmissão de vídeo pela Internet.

Os aplicativos mais comuns que utilizam esse tipo de protocolo são divididos entre aqueles que recebem dados (RTP Cliente) e aqueles que enviam dados (Servidores de RTP), como por exemplo:

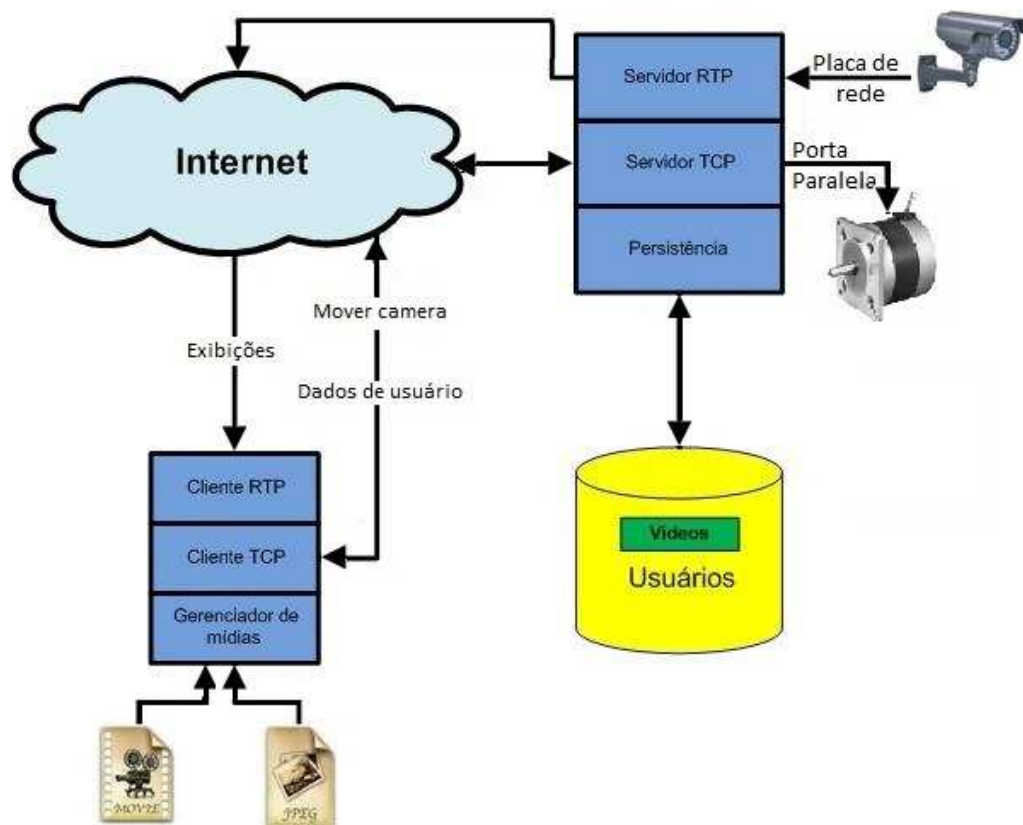
- Aplicativos de conferência: devem ser capazes de receber um fluxo de mídia de uma sessão RTP e processá-lo no console;

- Um aplicativo de secretária eletrônica de telefone: deve ser capaz de receber um fluxo de mídia de uma sessão RTP e armazená-lo em um arquivo;

Para um sistema de monitoramento os aplicativos RTP Cliente e Servidores RTP devem receber um fluxo de mídia de uma sessão RTP, processar e armazenar esse fluxo.

## 6 ARQUITETURA DO SISTEMA

Através da arquitetura do sistema é possível identificar os relacionamentos entre os módulos do sistema bem como compreender a estrutura e os componentes envolvidos em determinadas tarefas. Dessa forma, a arquitetura apresentado na Figura 2 refere-se ao sistema de monitoramento de baixo custo especificado neste artigo.



**Figura 2 – Arquitetura do sistema de baixo custo**

Os principais componentes desta arquitetura são:

- Aplicação Cliente: software desenvolvido na linguagem de programação Java que tem como objetivos: estabelecer conexão com o servidor, receber as imagens vindas do servidor, enviar comandos através de uma conexão TCP para o hardware que controla os movimentos da câmera IP e armazenar, tratar e exibir arquivos de áudio ou vídeo. Para isso a aplicação cliente é dividida em três módulos, o Cliente RTP, o Cliente TCP e o Gerenciador de Mídias.
- Aplicação Servidora: software desenvolvido na linguagem de programação Java que tem como objetivos: aceitar e gerenciar pedidos de conexões oriundos de aplicações cliente, obter e processar as imagens das câmeras IPs, enviar as imagens capturadas pelas câmeras para a aplicação cliente através de uma conexão RTP e interpretar os comandos de movimento enviados pela aplicação cliente e encaminhá-los, via porta paralela, para os motores de passo. A aplicação servidora é definida em três módulos, o Servidor RTP, o Servidor TCP e o Módulo de Persistência dos Dados.

## 7 RESULTADOS

O sistema apresentado neste artigo é formado por uma arquitetura distribuída, através de um aplicativo cliente/servidor conforme apresentado na seção 6.

## 7.1 Aplicativo Servidor

O aplicativo servidor deve ser instalado em um computador na mesma rede da câmera IP. Através das instruções contidas no aplicativo servidor as seguintes funções são executadas, não simultaneamente: API JMF será responsável por reconhecer a câmera IP; capturar as imagens da câmera IP; salvar as imagens em um banco de dados na máquina servidora. Por outro lado o protocolo RTP deverá: viabilizar o envio das imagens capturadas por meio da câmera IP para o aplicativo cliente. Enquanto isso o protocolo TCP estará apto para: reconhecer e gerenciar a conexões com os aplicativos clientes; permitir que um cliente habilitado possa mover a câmera. A Figura 3 mostra a interface do aplicativo servidor, veja:



**Figura 3 – Interface de acompanhamento das tarefas realizadas pelo aplicativo servidor**

No momento em que o aplicativo servidor é iniciado e conectado à câmera IP, é feito um clone da fonte de mídia permitindo assim que o vídeo seja ao mesmo tempo enviado através da rede para o cliente e gravado em um banco de dados no servidor. O servidor apresentado neste sistema é capaz de atender as solicitações simultâneas de clientes, de modo que vários clientes podem receber ao mesmo tempo o fluxo de vídeo enviado pelo servidor. Por outro lado, para não haver conflitos, o aplicativo servidor permite que apenas um cliente por vez possa mover a câmera IP.

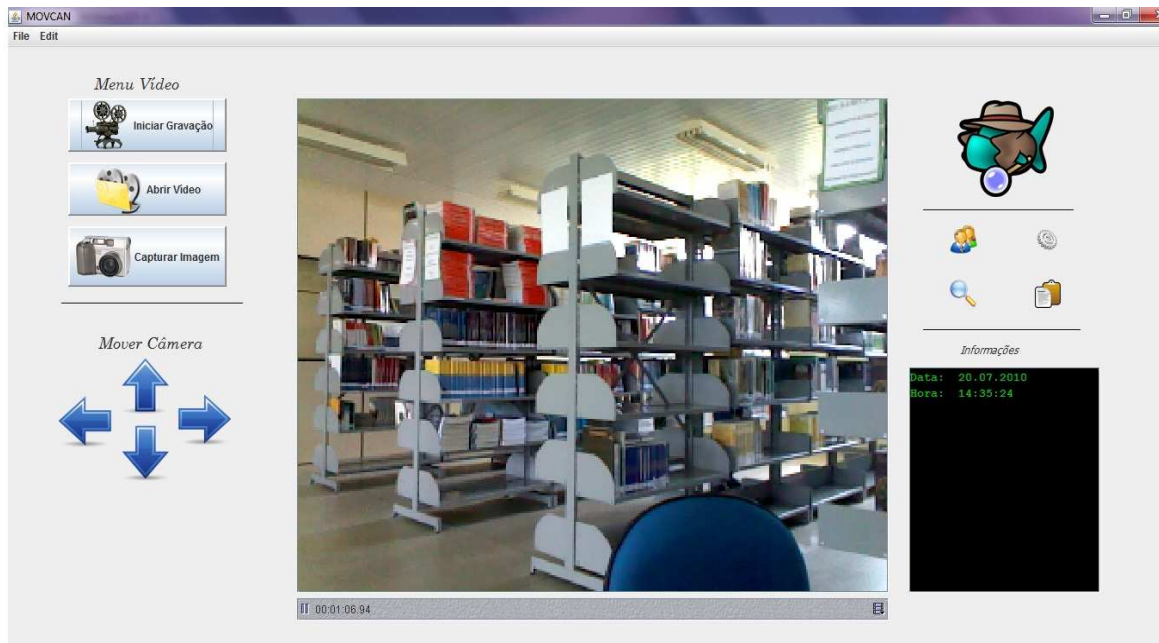
## 7.2 Aplicativo Cliente

Ao executar o aplicativo cliente, o sistema solicita que o usuário faça a autenticação no sistema através de interface mostrada na Figura 4. Caso o usuário ainda não seja cadastrado no sistema, ele poderá fazer uma solicitação de cadastramento que será encaminhada através de uma conexão TCP ao aplicativo servidor.



**Figura 4 – Tela inicial do aplicativo cliente para login de usuários**

Para os usuários cadastrados, o sistema permitirá que a tela principal do aplicativo cliente seja visualizada o usuário tenha acesso às principais funções do sistema. Através do aplicativo cliente, o sistema permite que o usuário realize as seguintes funções: (i) Gravar vídeo: esta tarefa implica gravar o fluxo de mídia que chega do servidor através da rede em arquivo; (ii) Capturar imagem: esta tarefa implica na captura momentânea da tela que está sendo exibida no monitor ; (iii) Abrir um vídeo local: permite ao usuário abrir um vídeo com formato de mídia válido que se encontra na máquina cliente; (iv) Mover câmera: através do menu mover câmera, um usuário habilitado tem o poder de controlar o movimento da câmera enviando sinais para o servidor através de uma conexão TCP. Essas tarefas podem ser identificadas na Figura 5:



**Figura 5 – Interface do Cliente**

### 7.3 Hardware

Além de toda programação usar software livre, outro importante fator que reduz o custo do projeto é o hardware que manipula as câmeras IP. Este hardware é um suporte composto de componentes eletrônicos simples e de fácil acesso. Sua confecção originou-se através de motores de passo retirados de sucatas de impressoras e de metais reciclados para a criação do braço de suporte, como pode ser observado na Figura 06.



**Figura 06: Hardware de controle do movimento da câmera IP**

A título de comparação, a Tabela 01 (MERCADO LIVRE 2010), mostra a diferença de investimento entre o sistema objeto deste artigo e algumas soluções similares disponíveis no mercado.

**Tabela 01: Relação de custos**

<b>Sistema</b>	<b>Preço (aproximado)</b>
D-LINK DCS-5220 (permite movimento de até dois graus de liberdade)	R\$ 600,00
CÂMERA IP COM PAN TILT VIVOTEK PT7315	R\$ 699,00
Sistema apresentado neste artigo	R\$150,00

## **8 CONSIDERAÇÕES FINAIS**

O Sistema apresentado neste artigo mostra a importância de projetar programas de monitoramento de acesso remoto eficiente, escalável e que possibilite o baixo custo. Como trabalhos futuros as pesquisas avançam na especificação de projetos que dêem suporte aos dispositivos móveis. Finalmente, os autores agradecem à Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) pela bolsa fornecida ao primeiro autor.

## **REFERÊNCIAS**

DEITEL, H.M., DEITEL, P.J. **Java, como programar**. 4. ed. Porto Alegre: Bookman, 2003

FIELDLY, T.A., SHUR, H.S., SHEN, H., YTTERDAL, T., **AIM-Lab: a system for remote characterization of electronic devices and circuits over the Internet**. Divesces, Circuits and Systems, 2000. pp. 143/1 – 143/6.

GRACO, "Desenvolvimento de um Manipulador com dois Graus de Liberdade Controlado Remotamente Via Internet". V Congresso de Engenharia Mecânica – Norte e Nordeste 1998 (V CEM–NNE98), Fortaleza, Ceara, 1998.

HORSTMANN, C. S.; CORNELL G., **Core Java 2 – Recursos Avançados**. São Paulo: Mafkron Books, 2002.

KUROSE, J. F; KEITH W., Ross, **Redes de Computadores E A Internet - Uma Nova Abordagem**, Editora Addison Wesley, 2003.

MERCADO LIVRE, 2010, Disponível em: <<http://lista.mercadolivre.com.br/CAMERA-IP-PAN-TILT-ZOOM>> Acesso em: 15 de outubro de 2010.

RFC 1889, 1996.

Sun Microsystems, Inc. **Guia do desenvolvedor Java Media Framework**. 1998, Disponível em: <<http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/guide/>> Acesso em: 18 julho 2010.

TANENBAUM, Andrew S. **Redes de computadores**. Rio de Janeiro: Campus, 4ª Edição, 2003. TORRES, Gabriel, **Redes de Computadores: curso completo**. Rio de Janeiro: Axcel Books, 2001.