

Solução Weblab para Ensino e Trabalho Colaborativo em Sistemas Digitais

Veruska R. Moreira¹, Fabbryccio A. C. M. Carodoso¹, Dalton S. Arantes¹

¹Faculdade de Engenharia Elétrica e Computação (FEEC) – Universidade Estadual de Campinas (UNICAMP) - Campinas– SP – Brasil

{veruska, cardoso, dalton}@decom.fee.unicamp.br

Abstract. *This work presents a reconfigurable platform designed for teaching and collaborative research over digital system laboratories. The platform provides a real-time, generic and flexible environment for laboratory practices in distance learning modes. The proposed architecture enables the development of new experiments in digital systems, focusing on the functionalities which are the subject of the research.*

Resumo. *Este trabalho apresenta uma plataforma reconfigurável aplicada ao ensino e pesquisa colaborativa em laboratórios de sistemas digitais. A plataforma proporciona um ambiente genérico e flexível em tempo real para prática de laboratório no modo remoto. A arquitetura proposta possibilita o desenvolvimento de novos experimentos em sistemas digitais, com foco na implementação das funcionalidades que se deseja investigar.*

1. Introdução

O conceito de laboratório remoto surgiu da necessidade de disponibilizar laboratórios reais a qualquer momento, sem restrição de tempo e espaço. Este conceito exige que seja adotada uma nova abordagem de ensino, permitindo a troca de informações, interação entre alunos, professores e tutores e a flexibilização de local e de tempo de estudo. Idealmente, o objetivo é minimizar os problemas de logística, facilitar o compartilhamento de recursos e possibilitar o trabalho colaborativo. Deseja-se que o meio de desenvolvimento de experiências de laboratório remoto seja um ambiente de aprendizado no qual o aluno possa se envolver remotamente em atividades laboratoriais convencionais, com a mesma eficiência que teria se estivesse presente fisicamente no laboratório real.

Uma possível abordagem para a prática de laboratório é disponibilizar o *hardware* necessário para que o aluno ou pesquisador desenvolva o experimento fora do ambiente da universidade (Oliver, et al., 2008). De modo geral, pode-se utilizar uma arquitetura de *software* cliente-servidor no contexto de um laboratório remoto com foco no gerenciamento do usuário e nas metodologias que podem ser empregadas no aprendizado (Cruz, et al., 2007). Entretanto, essa abordagem possui limitações no que diz respeito à prática de laboratório remoto. Essas limitações podem ser percebidas através da dificuldade de compartilhar recursos, fornecer o acesso ao experimento em tempo real, integrar novos experimentos ou equipamentos ou alterar os já existentes.

Além disso, os experimentos que não são concebidos para serem compartilhados requerem automatização externa para que os usuários possam ter acesso.

A proposta deste trabalho é disponibilizar uma plataforma de *hardware-software* que possibilite a experiência de execução de WebLabs, em condições semelhantes às encontradas em um laboratório presencial. Os WebLabs são laboratórios remotos onde experimentos, equipamentos e sistemas reais podem ser controlados a distância através da Internet (KyaTera FAPESP, 2003). O usuário deverá ter a liberdade de criar ou alterar os experimentos e executá-los sem restrição de tempo ou espaço.

A arquitetura da plataforma foi concebida para que, além de possibilitar o controle e a monitoração, também possibilite a distribuição do experimento em rede. Distribuir um experimento em rede significa implementar partes do mesmo em locais remotos e integrá-los automaticamente através da comunicação por fluxos de dados de tempo real em redes de computadores. Esse conceito é explorado neste trabalho para viabilizar a colaboração em pesquisa aplicada.

Um dos aspectos da plataforma de *hardware* é a capacidade de permitir o acesso em tempo real por meio do suporte a fluxos de dados (*streaming*), seja para monitoração do experimento como para comunicação entre partes do mesmo experimento. No entanto, há limitações no que se refere ao gerenciamento de usuários e de experimentos, assim como há limitações principalmente no suporte a aplicações de metodologias de ensino. Tais características são típicas de plataformas de *software*. Portanto, uma solução WebLab completa também deve contemplar uma proposta de plataforma de *software* que possibilite o gerenciamento do usuário e do experimento e que forneça uma interface de usuário que seja acessível para a execução do experimento. A plataforma de *software* compõe o laboratório remoto e fornece um ambiente de fácil uso para possibilitar a interação entre alunos, tutores e professores.

2. Arquitetura proposta para aplicações de laboratórios remotos

A concepção do sistema possibilita a interação de usuários e experimentos remotos através de uma plataforma de *software* integrada a uma plataforma de *hardware* reconfigurável, que executa em tempo real e permite o controle e o monitoramento de experimentos laboratoriais. A arquitetura proposta descreve o sistema com base nas funcionalidades a serem fornecidas, evitando a redundância de implementação e a perda de foco na missão do sistema. A missão do sistema é prover acesso remoto em tempo real a professores, alunos ou pesquisadores para a execução de experimentos laboratoriais de sistemas digitais.

As macro-funcionalidades do sistema são modularizadas e apresentadas como serviços. Os serviços são módulos com baixo acoplamento que possuem uma interface independente de sua implementação. A descrição da arquitetura em módulos permitiu diminuir o nível de complexidade do sistema, facilitou a tomada de decisões sobre o desenvolvimento e possibilitou identificar a sinergia entre os módulos que compõem o sistema. A definição da arquitetura ajudou na escolha das soluções técnicas para desenvolver o sistema. A Figura 1 mostra uma visão em camada da arquitetura proposta.

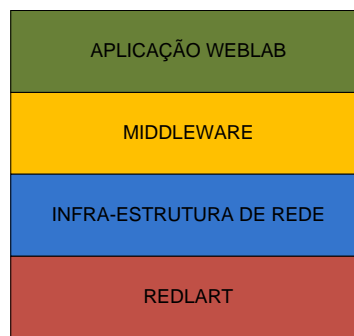


Figura 1: A arquitetura proposta para aplicações de laboratório remoto.

A arquitetura é composta por quatro camadas: Aplicação WebLab, *Middleware*, Infra-estrutura de Rede e REDLART. Cada camada possui uma função específica e executa sua função independentemente das outras camadas. Cada camada possui uma interface de programação de aplicação (*Application Programming Interface* - API) que estabelece o contrato ou a maneira como uma camada pode comunicar-se com a outra.

2.1. Aplicação WebLab

A camada aplicação WebLab representa o experimento do laboratório que será executado. Experimentos de filtros digitais, de processamento de áudio, de codificação/decodificação de vídeo, entre outros. Por exemplo, suponha que um usuário esteja executando um experimento que envolva a captura e a reprodução de fluxo de sinais de áudio. O usuário pode selecionar a fonte do sinal, o tipo de processamento, controlar o volume e determinar onde o sinal de áudio processado deve ser reproduzido. Essa camada é responsável pela ordem da execução.

A camada de aplicação WebLab possui o conhecimento do contexto e da ordem de execução. Nessa camada são definidos os processos ou as regras de negócio do experimento. O usuário interage diretamente com esta camada. A execução dessa camada pode exigir dados do usuário para iniciar ou continuar o processo. Conseqüentemente, a natureza da execução não é atômica, ou seja, a execução pode parar, retroceder e continuar.

2.2. A camada de middleware

A camada de *middleware* representa a plataforma de *software* que é responsável por fazer a comunicação entre a camada de aplicação WebLab e a camada REDLART através da rede local, da rede KyaTera (KyaTera FAPESP, 2003) ou da Internet convencional. Além disso, realiza o gerenciamento do usuário, o gerenciamento de controle da aplicação WebLab e o gerenciamento dos serviços necessários para executar o experimento de laboratório. Essas funcionalidades utilizam soluções consolidadas de segurança para executar seus processos como, por exemplo, fazer o gerenciamento de contas do usuário e acessar a aplicação, respectivamente.

2.3. A camada de infra-estrutura de rede

A camada de infra-estrutura de rede é responsável por estabelecer a troca de informações entre as camadas de *middleware* e de aplicação WebLab. Esta camada representa a infra-estrutura de rede que pode ser utilizada para estabelecer tal comunicação. Atualmente, além da Internet é possível utilizar redes de computadores

dedicadas ao ensino e pesquisa como, por exemplo, as redes do Projeto Kyatera (KyaTera FAPESP, 2003) e do Projeto Giga (Projeto GIGA, 2004). Essas redes acadêmicas são mais robustas porque sua infra-estrutura pode prover extensa largura de banda, baixa variação de atraso e de taxa de perda de pacotes (Teitelman, et al., 1998).

A FAPESP define o projeto KyaTera como um projeto cooperativo que consiste de uma rede de fibras ópticas projetada para a pesquisa e desenvolvimento de conexões de alta velocidade, integrando laboratórios de pesquisa que focam o estudo, desenvolvimento e demonstração de tecnologia e aplicações da Internet avançada (KyaTera FAPESP, 2003).

2.4. REDLART (*Reconfigurable Digital Laboratory for Advanced Research and Teaching*)

A camada REDLART consiste na plataforma de *hardware* da solução WebLab proposta neste trabalho. Uma das aplicações dessa plataforma é suportar a colaboração no desenvolvimento de um experimento. O conceito de colaboração é definido aqui pela possibilidade de se desenvolver partes do experimento em locais diferentes, integrados a partir de uma infra-estrutura de rede. Esse requisito impacta diretamente a forma de desenvolvimento do experimento remoto.

A arquitetura da REDLART é constituída por componentes de *hardware* padrão como controladores, *drivers*, *codecs*, conversores, interface Ethernet, entre outros. O acesso a essa infra-estrutura de componentes de *hardware* é permitido através de um módulo de interface de dados e de comunicação responsável pela abstração e pela comunicação com esses dispositivos.

Em linhas gerais, a plataforma REDLART deve ser constituída por uma placa de circuito impresso contendo, obrigatoriamente, um chip FPGA (*Field Programmable Gate Array*) da Xilinx (Xilinx Inc, 2009) e uma interface PHY Ethernet 10 Mbps/100 Mbps. O *hardware* da plataforma pode incluir ainda dispositivos de aquisição e reprodução de dados como o *codec* de áudio AC97 e conversores AD (Analogico Digital) e DA (Digital Analogico). A especificação dos conversores AD e DA depende do experimento a ser desenvolvido. A arquitetura da plataforma disponibiliza componentes funcionais para realizar a mudança de domínio de *clock* entre os conversores e o módulo de processamento. É importante destacar que devido à implementação atual da arquitetura, há uma restrição na taxa dos conversores em até 100 MHz. Os conversores AD e DA também podem ser externos à placa, conectados a partir de pinos de expansão.

Esses requisitos de *hardware* são atendidos na prova de conceito deste trabalho através dos kits de desenvolvimento XUP da Digilent com uma FPGA Virtex II Pro 2VP30 da Xilinx, um chip LM4550 da National (*codec* de áudio AC97), um chip PHY Ethernet 10/100 da Intel (modelo LXT972A) e pinos de I/O (Input/Output). Uma vantagem desse kit, além de possuir componentes de *hardware* de alto desempenho é o seu custo reduzido de apenas U\$ 299,00 (não inclusos os impostos) pelo programa universitário da Xilinx (XUP – *Xilinx University Program*) (Xilinx Inc, 2009).

2.4.1. Modelagem da plataforma REDLART

A arquitetura da plataforma de *hardware* foi estruturada em três módulos principais: Interfaces de Dados, Comunicação e Processamento. Uma visão geral da arquitetura da plataforma REDLART é apresentada na Figura 2.

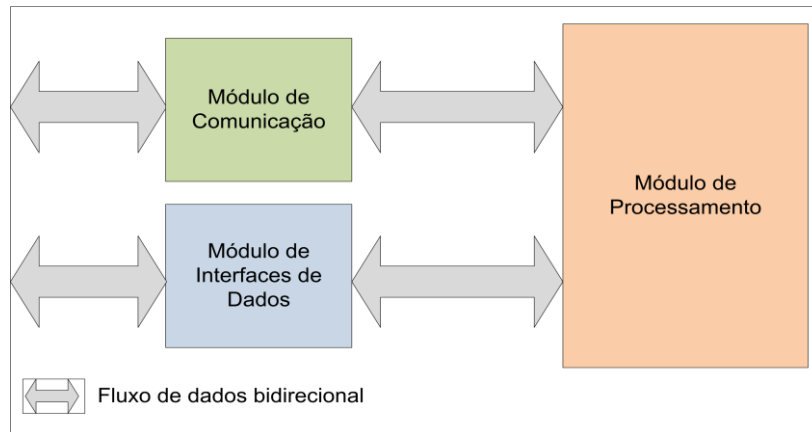


Figura 2: Visão geral da plataforma REDLART

O módulo de interface de dados é responsável por realizar toda a captura e reprodução local de sinais. Esse módulo também é responsável pela mudança de domínio de *clock* necessária para disponibilização do sinal em rajada para o módulo de processamento e para a reprodução do sinal obtido pelo módulo de processamento. Portanto, o fluxo de dados entre os módulos de processamento e interface de dados é bidirecional. Toda abstração e comunicação com dispositivos de captura e reprodução é realizada por esse módulo.

O módulo de comunicação é responsável por gerenciar todo fluxo de pacotes em tempo real via rede IP (*Internet Protocol*) (RFC 1180, 1991) e disponibilizar em rajada para o módulo de processamento. Esse módulo também é responsável por gerenciar a comunicação entre a plataforma de *hardware* (REDLART) e a plataforma de *software* cliente-servidor (*middleware*) para possibilitar a monitoração e o controle remoto do experimento. Nesse módulo são definidos os procedimentos e os protocolos para que tal comunicação aconteça.

De maneira geral, o módulo de comunicação envia pacotes de monitoração para o *middleware* para que o mesmo possa monitorar parâmetros de variação lenta do experimento desenvolvido no módulo de processamento, como temperatura e pressão. O *middleware*, por sua vez, sinaliza a alteração de valor do parâmetro gerando pacotes de controle para o módulo de comunicação. Para experimentos que geram fluxos de dados de tempo real com variação rápida, como áudio e sinais eletrônicos, o módulo de comunicação gera pacotes de sinais para transporte do mesmo entre plataformas remotas. Neste caso, o sinal é processado em locais diferentes ou o mesmo é requisitado para visualização por um nó remoto da rede. O módulo de comunicação fornece suporte à troca de pacotes de monitoração, controle e fluxo de dados através do protocolo MCP (*Monitoring and Control Protocol*) (Cardoso, et.al., 2007). O protocolo MCP foi

desenvolvido para transportar e identificar os fluxos de dados possíveis da plataforma, ou seja, de controle, de monitoração e de sinal.

3. Cenário de utilização da plataforma

Para ilustrar como os módulos de comunicação, de interfaces e de processamento são acionados em uma aplicação, um cenário de uso é apresentado na Figura 3. Esse cenário ilustra a possibilidade de se executar partes do experimento em locais distintos. Ambas as partes são implementadas na plataforma REDLART e integradas através do módulo de comunicação e pela camada de infra-estrutura de rede para operar como um único experimento.

Na primeira parte do experimento, o fluxo de sinal é capturado de uma fonte geradora de sinal utilizando o módulo de interface de dados. O fluxo de sinal é disponibilizado para o módulo de processamento que apenas encaminha o sinal para o módulo de comunicação. O módulo de comunicação gera pacotes de sinais e envia esses pacotes, através da Internet, diretamente para a parte número dois do experimento. O módulo de comunicação também pode gerar pacotes de monitoração e receber pacotes de controle para monitorar e controlar, respectivamente, os parâmetros do experimento.

Na segunda parte do experimento, os pacotes de sinais são recebidos, desencapsulados e o fluxo de sinal obtido são entregues para o módulo de processamento. O módulo de processamento, por sua vez, realiza o processamento do fluxo de sinal em tempo real e encaminha o sinal processado para reprodução local através do módulo de interface de dados. É importante destacar que esse mesmo sinal poderia também ser transmitido pela rede através do módulo de comunicação para reprodução e visualização remota no local da primeira parte do experimento.

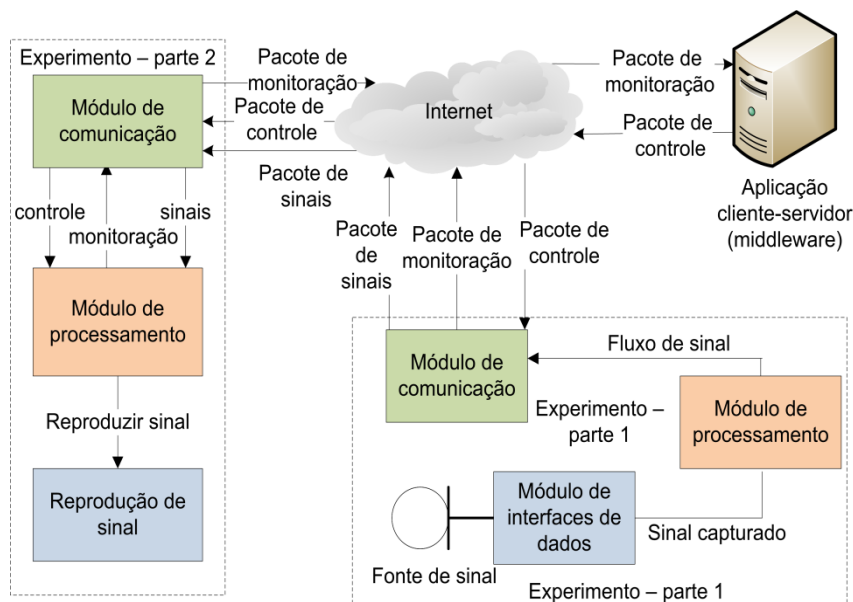


Figura 3: Um possível cenário de utilização da plataforma

4. O uso da arquitetura no contexto educacional

A arquitetura proposta pode ser utilizada como uma ferramenta adicional para auxiliar no processo de ensino-aprendizagem de práticas de laboratório de engenharia. A intenção não é substituir as aulas práticas de laboratório convencionais, mas fornecer uma ferramenta que possibilite estender o conceito de laboratório real para que as experiências possam ser realizadas a distância.

Uma forma eficiente de capturar a atenção do aluno e motivá-lo é demonstrando que o conhecimento adquirido nas aulas teóricas é efetivamente empregado em uma série de tecnologias presentes em seu dia a dia (Boruchovitch, et al., 2009). Isso pode ser conseguido de forma prática através de aulas laboratoriais, uso de recursos computacionais e através de ferramentas de ensino a distância. É nesse cenário que a plataforma REDLART pode ser introduzida para auxiliar no processo de ensino de laboratório a distância.

Por exemplo, nas aulas práticas de laboratório de sistemas digitais são abordados problemas reais referente a processamento digital de sinais (PDS). Um conceito bastante difundido em PDS é a teoria dos filtros digitais. Esse conceito pode ser aplicado em vários problemas reais (Nebeker, 1998). Após uma aula sobre filtros digitais, suponha que a prática de laboratório envolva o problema descrito a seguir. O controle remoto por rádio-freqüência é comum em portas de garagens e alarmes de carros. O princípio de funcionamento consiste em transmitir ondas de rádio que correspondam a um comando binário referente ao botão que o usuário pressionou (Senda, et al., 2005). Com base no conhecimento adquirido sobre filtros digitais pode ser requerido que se projete e teste um sistema para detecção do sinal de controle para acionamento do dispositivo alvo.

A plataforma REDLART pode ser utilizada para aplicar na prática os conceitos adquiridos sobre PDS, como no exemplo de filtros digitais citado acima. A plataforma é flexível o suficiente para permitir que novos experimentos possam ser incorporados a ela, abstraindo-se os detalhes de implementação que não são o foco do problema. Para a resolução de um problema como o citado acima, podem-se desenvolver experimentos que utilizem, por exemplo, filtros digitais diversos.

Após a integração do experimento na plataforma, os alunos podem ter acesso a dados reais, executados em tempo real na plataforma de *hardware* REDLART para aferir medidas, analisar os resultados e executar o experimento sem restrição de tempo e espaço físico. Esse é apenas um cenário no qual a plataforma REDLART pode ser utilizada. Outros cenários poderão ser explorados posteriormente.

5. Validação da plataforma, resultados e considerações finais

Para validar a plataforma, foi realizado um teste de distribuição que teve por base um experimento de processamento de sinais. Esse experimento consiste na captura de um sinal de áudio, no processamento, e na reprodução do sinal resultante. O processamento realizado consiste em filtros passa-baixa e passa-alta e filtro para gerar o efeito de reverberação. Nos testes também são realizados controles de *leds* e de volume de áudio.

O experimento foi configurado para utilizar duas plataformas REDLART e a infra-estrutura de rede do projeto KyaTera. A plataforma REDLART1 permaneceu na Faculdade de Engenharia Elétrica e Computação e a plataforma REDLART2 foi colocada no Instituto de Física, ambos da Unicamp. A Figura 4 mostra uma visão geral

do experimento. Essa demonstração foi realizada no Workshop Kyatera 2008 (Fapesp projeto Tidia-KyaTera, 2008).

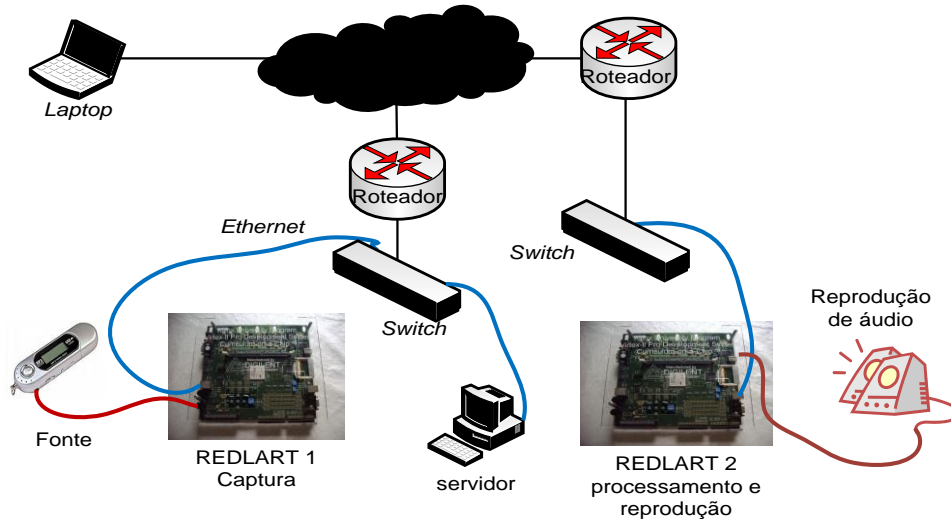


Figura 4: Experimento acessado remotamente via rede KyaTera.

Para cada plataforma foi atribuído um endereço IP estático, uma porta UDP (*User Datagram Protocol*) e um endereço MAC (*Media Access Control*) para que de fato pudessem participar da rede como um nó que envia e recebe pacotes de dados. Uma fonte de áudio estéreo, como um MP3 *player*, foi conectada na entrada *line in* na plataforma REDLART1. O nó REDLART1 capturou o fluxo de sinal de áudio da entrada *line in* e o disponibilizou em rajada para o módulo de processamento. O módulo de processamento encaminhou o sinal resultante para reprodução local, através do módulo interface de dados e para o nó REDLART2, através do módulo de comunicação e da rede KyaTera.

A plataforma REDLART2 recebeu os pacotes de sinal enviados pela plataforma REDLART1, desempacotou-os e verificou se os pacotes recebidos eram de controle ou de fluxo de sinais. O usuário pode escolher reproduzir o áudio processado ou reproduzi-lo na forma original sem processamento. Quando o usuário selecionava o tipo de processamento, o parâmetro do experimento que fazia referência ao tipo de reprodução era alterado na memória. Cada parâmetro está associado ao par endereço e valor na memória, portanto cada parâmetro de controle do experimento é mapeado para uma posição na memória. O módulo de processamento faz a leitura desse parâmetro e verifica a escolha do usuário. Caso seja reverberação, o módulo de processamento realiza a reverberação do fluxo de sinais e encaminha o sinal resultante para o módulo de interfaces de dados que executa a reprodução. Paralelamente aos processos de captura, transmissão, recepção, processamento e reprodução do áudio, o usuário pôde controlar o acendimento de *leds* e o volume do áudio.

O resultado do acesso ao experimento remoto foi satisfatório. Não ocorreu atraso significativo e a variação no atraso foi praticamente imperceptível, possibilitando que a plataforma operasse em tempo real. Isso foi confirmado utilizando o Wireshark (Wireshark Foundation) para verificar o envio e a recepção de pacotes na plataforma e no servidor. Não houve congestionamento na rede, portanto não aconteceu descarte de pacotes. Os pacotes foram reproduzidos a uma taxa média de 48 kHz, com *clock* de 24

MHz, e processados em um domínio de *clock* de 100 MHz. Os testes preliminares realizados demonstraram o funcionamento adequado da plataforma.

6. Conclusão

Este trabalho apresentou de forma resumida o conceito de ensino de laboratório a distância. O intuito do laboratório a distância é idealmente alcançar um número muito maior de pessoas e oferecer educação de qualidade a um custo bem menor em infraestrutura e recursos humanos. Além disso, se bem projetado, o ensino a distância pode incentivar o aluno a ter uma participação bem mais ativa no processo de ensino e aprendizagem do que no curso presencial.

Este trabalho propôs uma arquitetura de sistema composta por uma plataforma de *hardware-software* que possibilita a criação de laboratórios remotos de sistemas digitais. A arquitetura da plataforma proposta possibilita disponibilizar o desenvolvimento e a execução de experimentos remotos e distribuídos com controle e acesso pela *Web*. Além disso, foram apresentados os resultados iniciais de um WebLab distribuído utilizando a plataforma REDLART. Avaliações preliminares nas condições propostas do laboratório remoto demonstraram a funcionalidade, confiabilidade e flexibilidade da plataforma. Os resultados obtidos foram bastante satisfatórios, mostrando a viabilidade do sistema proposto.

A seqüência deste trabalho poderá envolver o desenvolvimento de novos WebLabs que serão incorporados a plataforma. O objetivo é que esses WebLabs sejam avaliados quanto ao desempenho, escalabilidade e usabilidade por grupos de alunos e de pesquisadores como usuários do sistema. As informações obtidas servirão como referência para o desenvolvimento de futuras versões da plataforma e dos experimentos.

Referências

Boruchovitch, E. e Bzuneck, J. A. "A Motivação do Aluno - Contribuições da Psicologia Contemporânea", Petrópolis - RJ, Editora Vozes, 2009, 4ª edição.

Cruz, A. A., et al., "Development of a Robust and Flexible WebLab Framework based on AJAX and Design Patterns." Proceedings of 7th IEEE International Symposium on Custer Computing and the Grid (CCGrid 07), p. 811-816, Rio de Janeiro, Brasil, 2007.

Fapesp projeto Tidia-KyaTera, Workshop KyaTera (2008), acessado em 23 de novembro de 2008, - <http://kyatera.incubadora.fapesp.br/portal/eventos/workshop-kyatera-2008>.

KyaTera FAPESP "Rede experimental do Projeto Kyatera - FAPESP" acessado em 15 de julho de 2010. <http://kyatera.incubadora.fapesp.br/index.php/br/sobre-o-kyatera/plataforma-optica/rede-experimental>.

Nebeker, F. , "Signal Processing - The Emergence of a Discipline: 1948 to 1998". IEEE History Center. - New Jersey, 1998.

Oliver, J. P., and Haim, F., "Lab at Home: Hardware Kits for a Digital Lab". IEEE Transaction on Education, p. 46-51, vol. 52, 2008.

Projeto GIGA (2004), acessado em 15 de fevereiro de 2011. <http://www.rnp.br/pd/giga/>.

RFC 1180 - "A TCP/IP Tutorial Network Working Group", Internet Engineering Task Force (IETF), 1991, acessado em 17 de fevereiro de 2011. <http://tools.ietf.org/html/rfc1180>.

RFC 826 - "An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses", Internet Engineering Task Force (IETF). - Network Working Group, 1982, acessado em 23 de novembro de 2009, <http://www.ietf.org/rfc/rfc0826.txt>.

Schafer, T., Seigneur, J. M. and Donnelly, A., "PEARL: A Generic Architecture for Live Experiments in a Remote Lab", International Conference on Simulation and Multimedia in Engineering Education, 2003.

Senda, A., et al., (2005) "Aplicações de Processamento Digital de Sinais em Engenharia Elétrica", XXXIII Congresso Brasileiro de Ensino de Engenharia, 2005.

Teitelman, B. and Hanss, T. (1998), "Qos Requirements for Internet2", Internet2 QoS Work Group Draft, 1998.

Wireshark Foundation, Wireshark, acessado em 17 de fevereiro de 2011, <http://www.wireshark.org/>.

Xilinx Inc., "Programmable Logic Design - Quick Start Handbook", 2009, acessado em 17 de fevereiro de 2011, http://www.xilinx.com/publications/products/cpld/logic_handbook.pdf.

Cardoso, F. C. M. A. e Arantes, D. S. Curso IE344 B - Simulação, Co-simulação e Prototipagem de Sistemas de Comunicações Digitais [Online]. - 2007. - acessado em 05/11/2007. - <http://www.decom.fee.unicamp.br/~cardoso/ie344b.html>