

LABORATÓRIOS VIRTUAIS DE ENSINO À DISTÂNCIA COM
COMPARTILHAMENTO DE RECURSOS

William Roger Salabert da Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. José Manoel de Seixas, D.Sc.

Prof. Carmen Lucia Lodi Maidantchik, D.Sc.

Prof. Ana Regina Cavalcante da Rocha, D.Sc.

Prof. Luiz Pereira Calôba, Dr.Ing.

Prof. Marley Maria Bernardes Rebuzzi Vellasco, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

JULHO DE 2005

SILVA, WILLIAM ROGER SALABERT DA

Laboratórios Virtuais de Ensino à Distância com Compartilhamento de Recursos [Rio de Janeiro] 2005

X, 84 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2005)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Laboratório virtual 2. Grid 3. DSP 4. Instrumentação 5. Ensino à distância

I. COPPE/UFRJ II. Título (série)

Agradecimentos

- Aos professores José Manoel de Seixas e Carmen Maidantchik, pela orientação deste trabalho.
- Aos meus amigos do LPS, que sempre estiveram dispostos a me ajudar nos momentos de dificuldade, em especial aos colegas Neimar Machado e Rodrigo Coura Torres pelo auxílio na implementação do laboratório virtual.
- À Thaís Gomes Pereira Salabert pela inspiração e compreensão das ausências necessárias à elaboração desse trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

LABORATÓRIOS VIRTUAIS DE ENSINO À DISTÂNCIA COM
COMPARTILHAMENTO DE RECURSOS

William Roger Salabert da Silva

Julho/2005

Orientadores: José Manoel de Seixas

Carmen Lucia Lodi Maidantchik

Programa: Engenharia Elétrica

Em engenharia elétrica, laboratórios de bancada com ambientes de aquisição e análise de dados são fundamentais em diferentes disciplinas do curso de graduação e de pós, bem como para pesquisa na área. Entretanto, estes laboratórios são bastante custosos, devido à sofisticação dos equipamentos.

Neste cenário, com crescente interesse de ampliação de vagas em cursos de engenharia, é notável a influência das novas tecnologias de rede sobre os métodos de educação, em relação aos laboratórios. Esta mudança incentiva a criação de laboratórios virtuais e de seu compartilhamento entre diferentes instituições. O cenário colocado pela computação Grid realça este compartilhamento e propõe novas perspectivas para soluções eficazes de ensino e pesquisa. Os laboratórios virtuais podem ainda ser disponibilizados para os cursos técnicos de ensino médio, tema que vem tendo bastante realce no esforço de universalização do ensino médio no país.

Este trabalho tem por objetivo propor um modelo para o compartilhamento de experimentos remotos de hardware programável em laboratórios virtuais via Web. A solução envolveu conceitos de computação Grid, uso da tecnologia Web e projeto de um protótipo em linguagem gráfica orientada à componentes.

A bancada virtual foi implementada no âmbito de uma teia de laboratórios virtuais, possibilitando a realização de experimentos remotos e integrando hardware e software para o laboratório virtual PAPROS, na área de processamento de sinais, como elemento de atualização deste laboratório.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

VIRTUAL DISTANCE LEARNING LABORATORIES WITH RESOURCES
SHARING

William Roger Salabert da Silva

July/2005

Advisors: José Manoel de Seixas

Carmen Lucia Lodi Maidantchik

Department: Electrical Engineering

In the electrical engineering environment, traditional laboratories workbenches for data acquisition and analysis are indispensable in various undergraduate and graduate disciplines, as well as for research projects in the area. However, these laboratories are quite expensive due to the use of sophisticated equipments.

With an increasing interest on vacant increase in engineering courses, it is remarkable the influence of networks technologies on the education methods, related to laboratories. This change encourages the creation of virtual laboratories and their sharing among different institutions. The scene placed for Grid computing enhances this sharing and considers new perspectives to efficient solutions of education and research. Virtual laboratories may also be accessible by high school level technical courses, subject that has been lengthily discussed when related to the effort of including every person in education in our country.

The present work aims at defining a sharing model to remote experiments that make use of programmable hardware within Web virtual laboratories. The solution involved Grid computing concepts, the use of the Web technology and the design of a prototype developed with a graphical component oriented language.

The virtual workbench was implemented within a net of virtual laboratories that allows the accomplishment of remote experiments, integrating hardware and software, for the active virtual laboratory PAPROS, in the signal processing field domain.

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivação	2
1.3	Objetivos	3
1.4	Organização do Trabalho	4
2	O Ensino à Distância	5
2.1	Definições e Conceitos em Ensino à Distância	5
2.2	Laboratórios Virtuais	6
2.2.1	Laboratório Virtual do “Blekinge Institute of Technology”	6
2.2.2	Laboratório Remoto Aplicado ao Ensino de Engenharia Eletrônica	8
2.2.3	Laboratório Virtual Utilizando Sistema Operacional Aberto	10
2.2.4	Laboratório Virtual do Departamento de Biofísica e Engenharia Eletrônica da Universidade de Gênova	13
2.2.5	A Teia de Laboratórios Virtuais	16
2.2.6	Páginas de Apoio a Sistemas Lineares na UFRJ	18
2.2.7	Páginas de Apoio ao Processamento de Sinais na UFRJ	19
2.3	Considerações Finais	20
3	A Computação Grid	21
3.1	Definições e Conceitos em Computação Grid	21
3.2	Categorização em Computação Grid	22
3.3	Requisitos	22
3.4	Arquitetura	23

3.5	Funcionalidades	24
3.6	O Projeto Globus	25
3.7	O Projeto Legion	26
3.8	Considerações Finais	27
4	O Modelo Proposto	28
4.1	Modelo Conceitual	28
4.2	Requisitos	29
4.3	Arquitetura	30
4.4	Componentes da Bancada Virtual	32
4.4.1	O Servidor <i>Web Services</i>	32
4.4.2	Interface com o PC	34
4.4.3	Arranjo Experimental	35
4.4.4	O Aplicativo	36
4.5	Integração com a Grid	36
4.6	Considerações Finais	37
5	O Protótipo da Bancada Virtual	38
5.1	Visão Geral do Protótipo	38
5.2	Arquitetura	41
5.2.1	Módulo de Aplicação Web Services	42
5.2.2	Módulo Autenticador do Usuário	42
5.2.3	Módulo Alocador de Recurso de Experimento	43
5.2.4	Módulo Configurador de Experimento	44
5.2.5	Módulo Controlador de Instrumentação Remota	45
5.3	Implementação	46
5.3.1	A Placa de Avaliação ADSP 21160M EZKIT Lite	46
5.3.2	ADSP21160M	47
5.3.2.1	Núcleo de Processamento	48
5.3.2.2	Memória Interna SRAM com Portas Duais	48
5.3.3	Programa em C++ para o DSP	49
5.3.4	Barramento GPIB	51
5.3.5	Osciloscópio	52

5.3.6	Gerador de Sinais	53
5.3.7	O LabVIEW	54
5.3.8	O aplicativo LabVIEW	55
5.3.8.1	Módulo Autenticador do Usuário	56
5.3.8.2	Módulo Alocador de Recursos de Experimento	56
5.3.8.3	Módulo Configurador de Experimento	57
5.3.8.4	Módulo Controlador de Instrumentação Remota	60
5.3.9	Interface com a Teia	62
5.4	Detalhamento da Sessão	64
5.4.1	Visão Geral das Etapas	64
5.4.2	Autenticação	66
5.4.3	Configuração do Experimento	67
5.4.4	Manipulação dos Instrumentos Remotos	69
5.4.5	Encerramento da Sessão	69
5.5	Considerações Finais	70
6	Conclusões	71
6.1	Trabalhos Futuros	73
A	Noções Básicas de Programação LabVIEW	79
A.1	Componentes do Instrumento Virtual no LabVIEW	79
A.2	Estruturas de Controle de Fluxo do Programa	81

Lista de Figuras

2.1	<i>Arranjo experimental 1 (extraído de [8]).</i>	7
2.2	<i>Arranjo experimental 2 (extraído de [8]).</i>	8
2.3	<i>Arquitetura do Laboratório Virtual (extraído de [18]).</i>	10
2.4	<i>Arranjo experimental (extraído de [18]).</i>	11
2.5	<i>IPv6 sobre IPv4 (extraído de [6]).</i>	11
2.6	<i>Arquitetura do Laboratório Virtual (extraído de [6]).</i>	13
2.7	<i>Arquitetura do ISILab (extraído de [20]).</i>	14
2.8	<i>Interface do ISILab (extraído de [20]).</i>	15
4.1	<i>Diagrama de Camadas da Arquitetura Proposta</i>	31
4.2	<i>Arquitetura SOA (extraído de [38]).</i>	33
4.3	<i>Aplicativo Web Services (extraído de [38]).</i>	34
4.4	<i>Componentes da “Service-Oriented Architecture” (SOA) (extraído de [39]).</i>	35
5.1	<i>Arquitetura Modular do Protótipo</i>	40
5.2	<i>Diagrama em blocos da placa ADSP-21160M EZKIT Lite.</i>	48
5.3	<i>Arranjo Experimental no LPS</i>	53
5.4	<i>Diagrama de Blocos do Módulo Autenticador do Usuário.</i>	57
5.5	<i>Diagrama de Blocos do Módulo Alocador de Recursos de Experimento.</i>	57
5.6	<i>Controle do Módulo Configurador do Experimento FIR.</i>	58
5.7	<i>Diagrama de Blocos do Módulo Configurador de Experimento, nível 1</i>	59
5.8	<i>Diagrama de Blocos do Módulo Configurador de Experimento, nível 2</i>	60
5.9	<i>Geração de código no Módulo Controlador de Instrumentação Remota</i>	61
5.10	<i>Geração de código no Módulo Controlador de Instrumentação Remota</i>	62
5.11	<i>Painel de controle do Módulo Controlador de Instrumentação Remota</i>	62

5.12	<i>Fluxograma da Sessão de Bancada Virtual</i>	66
5.13	<i>Janela de Autenticação do Usuário</i>	67
5.14	<i>Janela de Configuração do Experimento</i>	68
5.15	<i>Osciloscópio e Gerador de Sinais da Bancada Virtual</i>	69
A.1	<i>Exemplo de Painel de Controle do LabVIEW (extraído de [16])</i>	80
A.2	<i>Exemplo de Diagrama de Blocos do LabVIEW (extraído de [16])</i>	80
A.3	<i>Exemplo da estrutura While Repeat do LabVIEW (extraído de [16])</i>	82
A.4	<i>Exemplo da estrutura For do LabVIEW (extraído de [16])</i>	82
A.5	<i>Exemplo da estrutura Case do LabVIEW (extraído de [16])</i>	83
A.6	<i>Exemplo da estrutura Sequenced do LabVIEW (extraído de [16])</i>	84

Capítulo 1

Introdução

O presente capítulo descreve o contexto no qual está inserido o tema desta tese, motivações que levaram ao desenvolvimento deste trabalho, os objetivos do modelo proposto e do protótipo implementado. Por fim, a organização do texto é apresentada.

1.1 Contexto

A rápida evolução tecnológica tem provocado mudanças em diversos aspectos do modo de vida humano. Uma das principais mudanças ocorre no campo da educação [1]. Os métodos de educação têm sido influenciados, principalmente, pelos meios de transmissão, busca e interatividade, que são proporcionados pelas novas tecnologias de rede [2]. Em relação à interatividade, uma verdadeira mudança de eixo nos modelos de educação tem ocorrido com o crescente deslocamento do aluno, de uma posição de assistente passivo à agente da construção de seu próprio conhecimento, pela maior interação com o objeto de estudo [3].

No âmbito das engenharias e das ciências da natureza, os avanços tecnológicos das telecomunicações permitiram a criação de laboratórios virtuais [4]. No domínio da engenharia elétrica, a literatura revela que diversas iniciativas têm surgido no sentido de viabilizar a realização de experimentos remotos via Web [5, 6, 7, 8, 9]. Também há uma crescente demanda por laboratórios com dispositivos programáveis, que integram hardware e software em uma mesma plataforma, para atender às exigências da evolução tecnológica dos dispositivos de processamento digital de

sinais, microcontroladores, FPGAs, entre outros.

1.2 Motivação

A utilização de laboratórios com equipamentos de bancada é uma demanda do ensino e pesquisa das ciências exatas e da tecnologia. Os laboratórios são necessários aos centros de pesquisa, cursos de formação do ensino médio, técnico, graduação, especialização, pós-graduação e empresas de base tecnológica. As aplicações são em treinamento, educação continuada ou demonstração de produtos [2].

Em geral, um laboratório pressupõe a existência de um local estruturado, como uma sala ou edifício [10]. Isto implica na necessidade de infra-estrutura adequada, pessoal de apoio, ambientação e espaço para os equipamentos e pessoas, além de uma localização condizente com sua aplicação. No caso de muitas cidades, a segurança do local e do acesso torna-se mais um problema, sendo um limitante de horários e da disponibilidade. Em consequência disso, surgem os horários de pico de utilização em que o laboratório pode estar aquém da demanda e horários onde a ociosidade prevalece [8].

A capacidade de atendimento dos laboratórios tradicionais é um dos fatores limitantes ao aumento do número de vagas dos cursos que necessitam desse tipo infra-estrutura. Os custos de manter uma estrutura somados aos custos de instrumentos e equipamentos, sujeitos à rápida obsolescência, principalmente aqueles necessários ao estudo de temas referentes a tecnologias emergentes, tornam imperativa a pesquisa por soluções alternativas [6]. Entre estas estão os laboratórios virtuais. Porém, mesmo os laboratórios virtuais, com instrumentação remota, necessitam de atualização constante.

A tecnologia emergente de computação Grid tem despertado interesse em projetos que envolvem o compartilhamento e coordenação de recursos computacionais, sobre diferentes domínios administrativos, essa tecnologia não está direcionada a aplicações em EAD, contudo seus elementos fornecem um ambiente capaz de possibilitar o compartilhamento, inclusive, dos recursos necessários aos laboratórios virtuais.

1.3 Objetivos

Este trabalho tem por objetivo apresentar um modelo para a implementação e possível compartilhamento de bancadas virtuais, que integram hardware (HW) e software (SW) aos experimentos remotos, nos laboratórios virtuais PAPROS e PALAS, componentes da *Teia de Interconexão de Laboratórios Virtuais segundo uma Concepção Multidisciplinar e Colaborativa* (Teia) [3, 4]. O projeto Teia corresponde a um conjunto de laboratórios virtuais, que abrange temas de cadeiras básicas e específicas dentro dos cursos de engenharia. Tais laboratórios virtuais são integrados por ligações hipertextuais numa teia, onde o aluno pode estabelecer pontes de forma multidisciplinar entre tópicos avançados em engenharia e ciências básicas.

A estrutura dos laboratórios do projeto Teia corresponde a um conjunto de módulos tutoriais apoiados por ferramentas para o desenvolvimento de projetos cooperativos, comunicação, simulação e bancada virtual para emulação de experimentos em DSP [4]. Dentre as ferramentas citadas, as bancadas virtuais ocorrem apenas no laboratório PAPROS, como elementos somente de emulação, não sendo possível trabalhar com dados físicos reais. Além disso a implementação existente possui uma estrutura de bancada virtual que não permite o compartilhamento de seus recursos entre os diferentes laboratórios virtuais ou domínios administrativos de rede componentes da Teia.

Na extensão proposta à Teia por este trabalho, o compartilhamento das bancadas virtuais ocorre em um ambiente multi-institucional de computação Grid, com compartilhamento de recursos e acesso via Web, por todos os membros, independente dos domínios administrativos de rede. Com esses recursos o aluno de engenharia elétrica pode, ao acessar a Teia, realizar experimentos remotos em processamento digital de sinais e sistemas lineares. Tudo isto em um ambiente de ensino à distância assíncrono, através de uma bancada de instrumentos virtuais associados a experimentos remotos que lhe permite obter dados experimentais, como se estivesse em um laboratório presencial [1].

Para validação do modelo, foi implementado um protótipo de laboratório remoto aplicado ao processamento digital de sinais e sistemas lineares, que utiliza “*Digital Signal Processor*” (DSP), que é uma plataforma necessária ao processamento dos experimentos implementados em filtros digitais e análise espectral. Este pro-

tótipo foi integrado aos laboratórios virtuais PAPROS e PALAS, como mais uma ferramenta de suporte à Teia, para a adição de bancadas virtuais com compartilhamento de recursos.

1.4 Organização do Trabalho

O próximo capítulo inicia com a apresentação dos conceitos e requisitos das aplicações de ensino à distância e laboratórios virtuais. Em seguida, são descritos exemplos de laboratórios virtuais aplicados ao ensino de engenharia elétrica.

No capítulo 3 se desenvolve um estudo sobre a computação Grid, seus principais conceitos e definições, requisitos, a arquitetura básica e a descrição de projetos Grid. O capítulo 4 possui a descrição do modelo conceitual proposto e dos principais requisitos necessários à implementação da solução de instrumentação virtual associada aos experimentos remotos com compartilhamento de recursos entre diferentes instituições. O capítulo 5 é descrito o protótipo realizado, sua arquitetura, componentes e implementação. Por fim, no capítulo 6 são apresentadas as conclusões e propostas futuras sobre este trabalho.

Capítulo 2

O Ensino à Distância

Este capítulo apresenta um conjunto de definições relativas ao ensino à distância e laboratórios virtuais, são abordados alguns projetos de *Laboratório Virtual*, pesquisados como referência do atual estado-da-arte.

2.1 Definições e Conceitos em Ensino à Distância

A evolução dos métodos de ensino à distância (EAD) está intrinsecamente relacionada com o tipo de mídia envolvido. O EAD iniciou-se com material impresso, passando pelos cursos televisivos, culminando nos recursos interativos, possibilitados pelos avanços das tecnologias de rede. Registros históricos comprovam que as práticas de EAD, provavelmente, devem ter se iniciado com os cursos de correspondência na Europa do século XVIII [11].

O conceito de EAD se contrapõe ao do ensino tradicional em relação às condições de horário e local da aula. Este se caracteriza pela presença simultânea da audiência e do professor no local das aulas. Já no EAD ocorrem duas práticas diferentes. No primeiro caso, definido como *síncrono*, o aluno assiste à aula apenas em um local diferente. No segundo caso, os horários e o local variam conforme as necessidades do aluno, uma condição definida na literatura como *assíncrona* [12].

A inclusão das tecnologias de informação e comunicação no processo de ensino criou o conceito de educação *baseada em Web*. Neste caso, o foco do modelo pedagógico está localizado no processo interativo, ou seja, naquilo que o aluno realiza enquanto aprende. Daí surgem séries de aplicações e ferramentas baseadas nas tec-

nologias da Web, tais como: fóruns de discussão, grupos de email, livros e bibliotecas digitais, museus e galerias on-line, aplicativos de mensagens instantâneas, seminários on-line, laboratórios virtuais, quadros negros virtuais, aulas virtuais, etc [1, 7].

2.2 Laboratórios Virtuais

Os laboratórios virtuais são ambientes de ensino disponibilizados via Internet, compostos de uma série de recursos multimídia capazes de integrarem tutoriais, exercícios, aplicativos de simulação, testes, meios de comunicação com instrutores e colegas, bancadas de acesso a experimentos remotos, entre outros [4]. Esses recursos são oferecidos pela Web de forma assíncrona, ao contrário do que ocorre com os laboratórios tradicionais.

O principal conceito relacionado aos laboratórios virtuais é o de instrumento virtual. O instrumento virtual é um aplicativo, “driver”, que se utiliza de uma interface gráfica em computador, como painel frontal de um instrumento, que pode ou não existir fisicamente. O instrumento virtual faz parte do domínio da aplicação, sendo uma interface com um instrumento remoto ou um aplicativo de emulação oferecido por um computador Servidor [13]. Um conjunto de instrumentos virtuais forma uma bancada virtual [4].

Entre as tecnologias emergentes de rede Internet e instrumentação remota, a linguagem gráfica orientada a componentes, obtida comercialmente, minimiza o tempo de desenvolvimento da virtualização dos painéis dos instrumentos remotos e construção dos seus “device drivers”, que são fornecidos pelos fabricantes dos equipamentos, sendo comum na literatura a referência ao uso dessas ferramentas [14].

2.2.1 Laboratório Virtual do “Blekinge Institute of Technology”

Com o objetivo de apoiar as cadeiras de laboratório, que envolvem circuitos elétricos nos cursos de graduação em engenharia, o projeto de *Laboratório Virtual do Blekinge Institute of Technology* (BTH) na Suécia baseou-se na premissa de que a prática de experimentos de laboratório é indispensável ao ensino de engenharia.

Na visão do BTH, as práticas de laboratório representam a oportunidade de

desenvolver habilidades e percepção em atividades que envolvem processos físicos e instrumentação, além de servirem de ambiente para trabalhos a serem realizados em grupo [8]. Entretanto, no modelo tradicional de laboratório, a aquisição de instrumentação e equipamentos envolve custos elevados. O aluno perde uma parte significativa de seu tempo montando os experimentos. Além disso, a estrutura de laboratório limita a realização das experiências a um local e horário determinados.

O projeto do BTH construiu um *Laboratório Virtual* para emulação de dois experimentos envolvendo as *Leis de Kirchoff's* para circuitos elétricos em regime de tensão AC ou DC e outros três experimentos envolvendo a teoria de medição de filtros elétricos. Para acessar o *Laboratório Virtual*, o aluno deve instalar um programa cliente na máquina local e realizar as tarefas via Web. A parte didática é composta pelo manual de tarefas do laboratório tradicional e por um manual de instruções sobre o programa cliente, ambos obtidos no mesmo site [15].

Foram realizados dois tipos de arranjo de hardware. No primeiro laboratório, os equipamentos são instrumentos de bancada, controlados por um *instrumento virtual* na forma de um painel de controle. Utilizou-se um painel de instrumentação da *National Instruments* padrão “*Peripheral Component Interconnect Extensions for Instrumentation*” (PXI) contendo, um multímetro digital, uma unidade de matriz de chaveamento, um gerador de funções, um osciloscópio e uma fonte controlada de tensão contínua [16]. Nesta mesma ordenação, da esquerda para a direita, os instrumentos podem ser observados na pelas indicações na foto da figura 2.1.



Figura 2.1: *Arranjo experimental 1 (extraído de [8]).*

O segundo arranjo utilizou equipamentos de bancada, controlados via barramento padrão GPIB pelo aplicativo escrito em *LabVIEW*. O osciloscópio *HP 54600B*

e o gerador de funções estão representados na parte esquerda da figura 2.2 com o gerador de sinais acima do osciloscópio e ambos interligados ao PC por meio do barramento GPIB compondo o bloco do *Servidor de Laboratório*. À direita do *Servidor de Laboratório* o diagrama indica que o *Programa Cliente* acessa o servidor de laboratório via Internet através do protocolo TCP-IP.

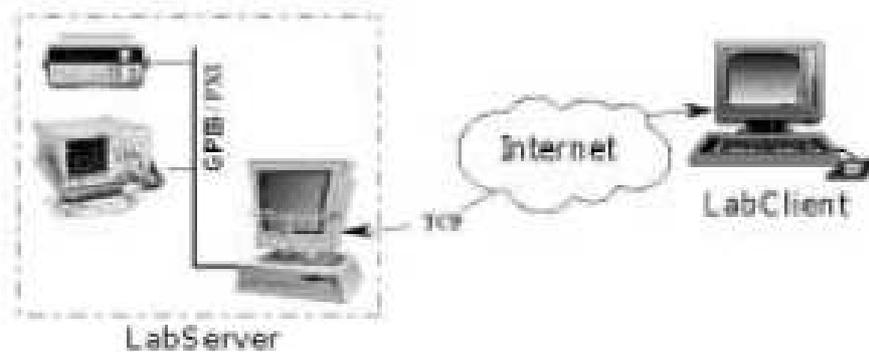


Figura 2.2: *Arranjo experimental 2 (extraído de [8]).*

O autor menciona que o tempo de resposta é significativo, se comparado ao de experimentos tradicionais, dando o exemplo de que uma mudança na sensibilidade vertical do osciloscópio pode levar alguns segundos para ser percebida pelo usuário.

2.2.2 Laboratório Remoto Aplicado ao Ensino de Engenharia Eletrônica

Este laboratório foi desenvolvido por um grupo de estudos na *Universidade Estadual de Campinas* (UNICAMP) formado por integrantes da *Faculdade de Engenharia Elétrica e Computação* (FEEC) e do *Departamento de Microeletrônica* (DEMIC). A partir do estudo dos serviços da Web, foi observado o potencial desses serviços para o desenvolvimento de ambientes de laboratórios virtuais como apoio a cursos presenciais e de EAD [17].

Os laboratórios presenciais, desde a infra-estrutura, passando pela manutenção, até se chegar às necessidades de atualização, representam uma fonte de custos significativa para os cursos de base tecnológica. Isto fortalece o enfoque didático, que visa a associar as tecnologias da Internet com as tecnologias de instrumenta-

ção virtual em um ambiente de aprendizado, que permita realizar remotamente as atividades de laboratório, podendo-se atingir um custo menor por aluno, quando comparados com os laboratórios presenciais.

Uma das oportunidades destacadas foi a ampliação da faixa de utilização de alguns equipamentos de custo elevado, que devido à maior liberdade de horários e locais para o acesso serão necessários em menor quantidade, diminuindo-se com isso os custos da atualização dos laboratórios.

A proposta envolve a realização de uma plataforma multi-experimentos à distância na área de eletrônica, capaz de atender aos seguintes requisitos:

- flexibilização dos horários para execução das tarefas de laboratório;
- acesso compartilhado entre diferentes instituições para equipamentos de custo elevado;
- apoio aos cursos à distância ou presenciais;
- execução de experimentos em tempo-real;
- aproveitamento dos procedimentos experimentais já existentes nos laboratórios convencionais;
- modularidade para agregar novos experimentos;
- custo menor, por estudante, com relação a um laboratório convencional.

O sistema é composto de três elementos principais: o software servidor, o software cliente e o hardware, representados nos três blocos da figura 2.3. A comunicação entre o cliente e o servidor é feita via protocolo *Socket*, já o hardware é controlado via interface serial padrão *USB*. A interface gráfica do cliente corresponde ao ambiente do *Laboratório Virtual* e permite o acesso aos controles gerais, aos materiais didáticos, procedimentos experimentais e observações do professor. Do lado servidor, a interface serve para o monitoramento e diagnóstico do sistema.

Na parte de hardware, a *Placa de Controle* suporta até oito *Placas de Experimentos* simultaneamente. Existe uma *Placa de Barramento* que serve para interconectar as demais placas do sistema. A figura 2.4 traz a ilustração da *Placa de Barramentos*, conectada perpendicularmente às *Placas de Experimentos* e à *Placa*

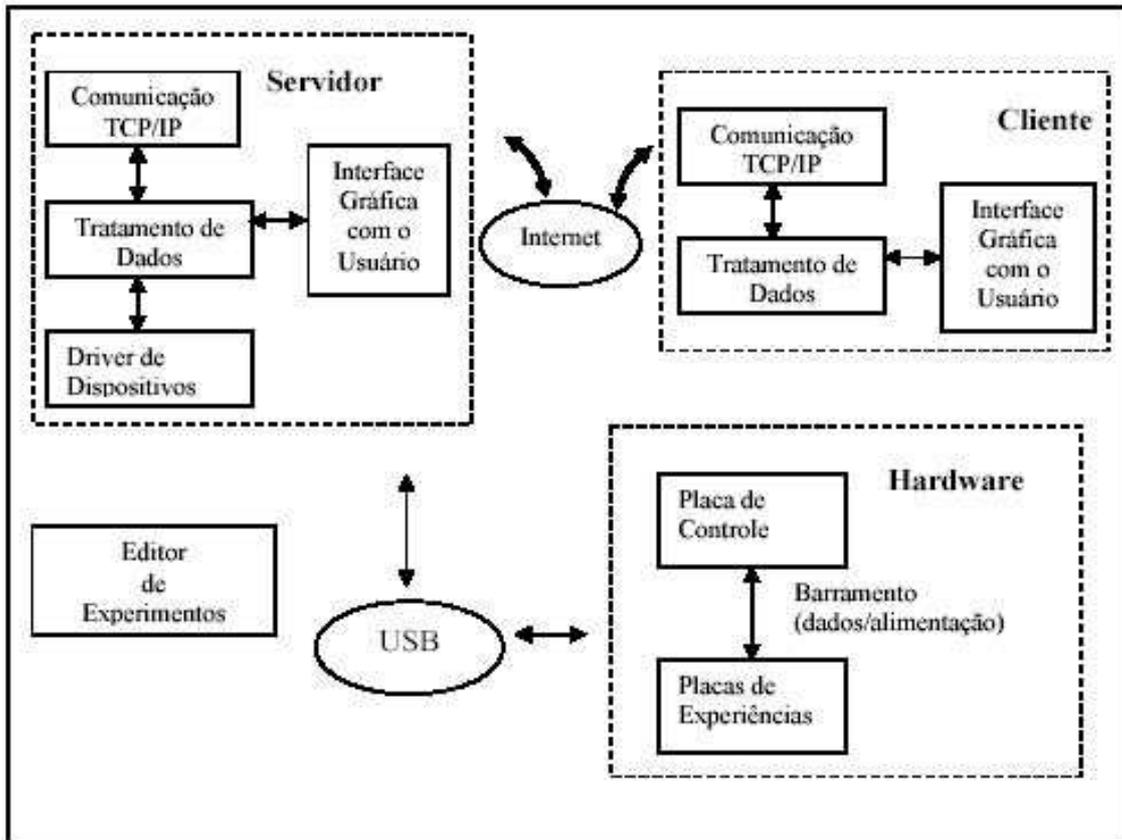


Figura 2.3: *Arquitetura do Laboratório Virtual (extraído de [18]).*

de Controle, esta última é quem gerencia a interface serial USB com o *Computador Servidor de Experimento* e interliga o *Instrumento de Medição*, representado por uma caixa abaixo das placas, com as *Placas de Experimento*. Também existe uma conexão entre o *Instrumento de Medição* e o *Computador Servidor de Experimento*, não representada na figura, para a transferência dos dados experimentais obtidos.

As *Placas de Experimento* permitem a associação de componentes eletrônicos por meio de relés e a observação de curvas de comportamento elétrico nos nós do circuito, geradas pela variação de potenciômetros digitais. A aquisição dos resultados é obtida por um osciloscópio interligado ao computador e à placa de controle.

2.2.3 Laboratório Virtual Utilizando Sistema Operacional Aberto

A proposta apresentada pelo grupo de pesquisas do *Laboratório de Microeletrônica da Escola Politécnica da USP* em São Paulo é a implementação de laboratórios virtuais, com a utilização de placas de aquisição de dados e controle

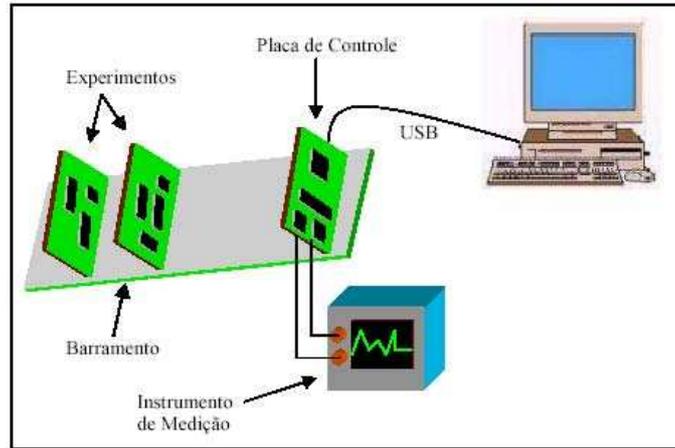


Figura 2.4: Arranjo experimental (extraído de [18]).

dos experimentos via Internet em uma arquitetura de rede baseada no protocolo TCP/IP versão 6 (IPv6) para o domínio da rede local, com comunicação sobre a infra-estrutura da Internet em TCP/IP versão 4 (IPv4). A figura 2.5 apresenta um diagrama da arquitetura proposta, indicando a existência de domínios locais fixos, representados pelos prédios, e móveis, representados pelos dois corredores, com o protocolo de rede IPv6, interligados à Internet através de um elemento Gateway que implementa a tradução entre os protocolos IPv4 e IPv6.

IPv6 sobre infraestrutura IPv4

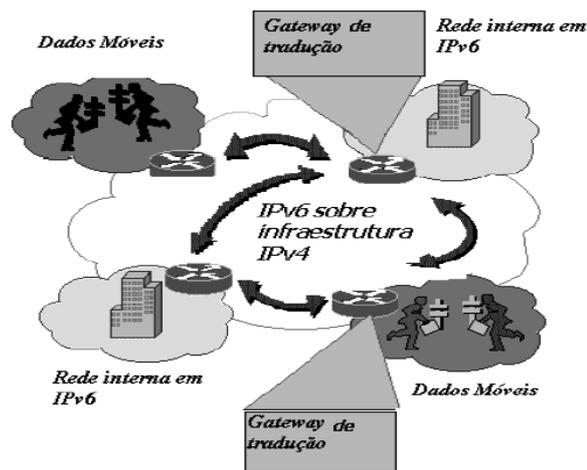


Figura 2.5: IPv6 sobre IPv4 (extraído de [6]).

O projeto considera a dificuldade dos custos envolvidos na constante atualização tecnológica, que é necessária aos cursos técnicos e superiores de base tecnológica,

no cenário nacional. Dificuldade esta que impacta diretamente nos indicadores de qualidade baseados na relação *alunos X posto de trabalho* estabelecidos pelo *Ministério da Educação* (MEC) e pela *Secretaria de Educação Média e Tecnológica* (SEMTEC) [6]. Os custos da constante atualização de equipamentos e licenças de software dos laboratórios tradicionais se refletem, em geral, na oferta de vagas e nos preços das mensalidades para as instituições de ensino privadas.

Sob outro prisma, o projeto foi orientado à aplicação em larga escala via Internet. Assim, consideraram-se condições como a futura exaustão do espaço de endereçamento, o uso de políticas de roteamento, o controle de fluxo, a garantia da qualidade de serviço (QoS) e segurança, funcionalidades nas quais o projeto IPv6 foi desenvolvido o que definiu a sua escolha como protocolo de rede e transporte para este projeto [19].

Como elemento de validação da proposta, foi implementado um *Laboratório Virtual* conectado à Internet pelo Servidor Web da rede local. Na figura 2.6, encontra-se o diagrama com a interligação do Servidor de Experimento do Laboratório Virtual representado pelo *Microcomputador para Aquisição de Dados* seus possíveis clientes no domínio local, *Estações Locais de 1 a n*, e seus possíveis clientes externos à rede local, representados pelo *Micro Remoto* e *Laptop*, interligados ao laboratório virtual via Internet e através do *Servidor Web*. O papel do *Microcomputador para Aquisição de Dados* é controlar os instrumentos remotos, osciloscópio e gerador de sinais, via GPIB e interagir com a placa de experimento por uma placa de aquisição e geração de dados da *National Instruments* (DAQ). O conjunto oferece até cinco experimentos que podem ser aplicados aos cursos introdutórios de eletrônica.

O experimento é selecionado pelo usuário no programa cliente. Um conjunto de relés, controlados por uma placa de aquisição e saída de dados digitais, configura a interligação de componentes para um dado experimento. A interação com o experimento é feita por gerador e osciloscópio remotos interfaceados ao PC servidor via GPIB. Esses elementos são controlados pelo usuário por meio das interfaces de instrumento virtual do programa cliente.

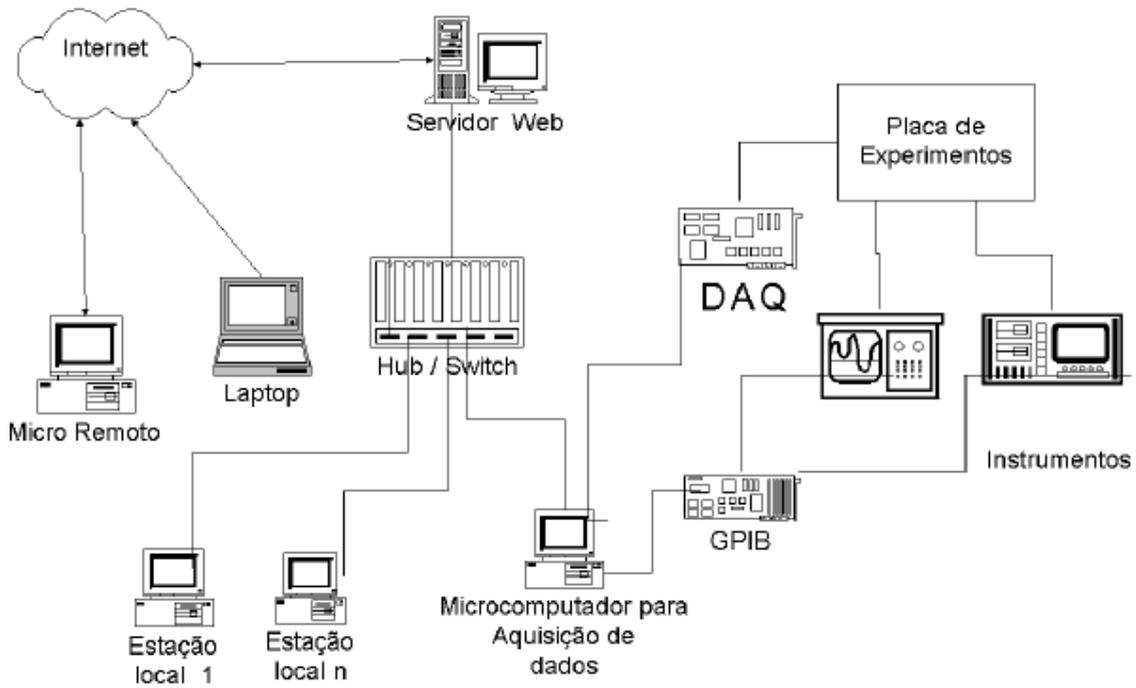


Figura 2.6: *Arquitetura do Laboratório Virtual (extraído de [6]).*

2.2.4 Laboratório Virtual do Departamento de Biofísica e Engenharia Eletrônica da Universidade de Gênova

O “Department of Biophysical and Electronic Engineering” (DIBE) da Universidade de Gênova, Itália, desenvolveu um modelo para compartilhamento de laboratórios de eletrônica digital via Internet chamado o *ISILab* [20].

Em sua concepção, o projeto *ISILab* considera importante a aplicação prática dos modelos teóricos no ensino técnico e científico. Em suas motivações, o projeto *ISILab* leva em conta as dificuldades associadas à atividade presencial nos laboratórios, tais como: a restrição de horários e locais, a necessidade de acesso a recursos caros (que poucas instituições possuem), condições perigosas de operação de determinados instrumentos, entre outras.

Considerando a necessidade e as restrições existentes, a proposta do projeto se orientou para uma solução alternativa aos laboratórios presenciais, na forma de uma plataforma de *Laboratório Virtual* com instrumentos e equipamentos remotos controlados por um PC via rede. Somando-se a isso, foi realizada a formulação de um modelo de arquitetura Grid para o compartilhamento dos recursos.

Foi implementado um *Laboratório Virtual* com arquitetura de software em

camadas, constituídas de dois tipos de servidores, o *Servidor de Laboratório Real* (RLS) e o *Servidor de Laboratório Virtual* (VLS), implementados com *LabVIEW 7.0*, conectados à Internet.

O VLS é o Servidor principal que gerencia a autenticação das máquinas, usuários e a alocação dos recursos dos RLSs entre os clientes. Os RLSs controlam os equipamentos e instrumentos remotos, podendo estar distribuídos geograficamente. Os clientes operam num ambiente multi-usuário concorrente e enxergam os recursos da rede de forma transparente, não interessando sua localização.

Na figura 2.7 está o diagrama do modelo *ISILab*, com todos os computadores clientes e servidores interligados via Internet e o arranjo experimental conectado ao “*Real Laboratory Server*”, também é indicada a possibilidade de acesso à rede via cliente discado pelo ícone do computador portátil. Neste modelo, não há uma gerência local, ou seja, todas as máquinas são interligadas diretamente na Internet, tornando necessário que cada cliente, RLS e VLS seja um servidor com IP fixo e que possua firewall próprio.

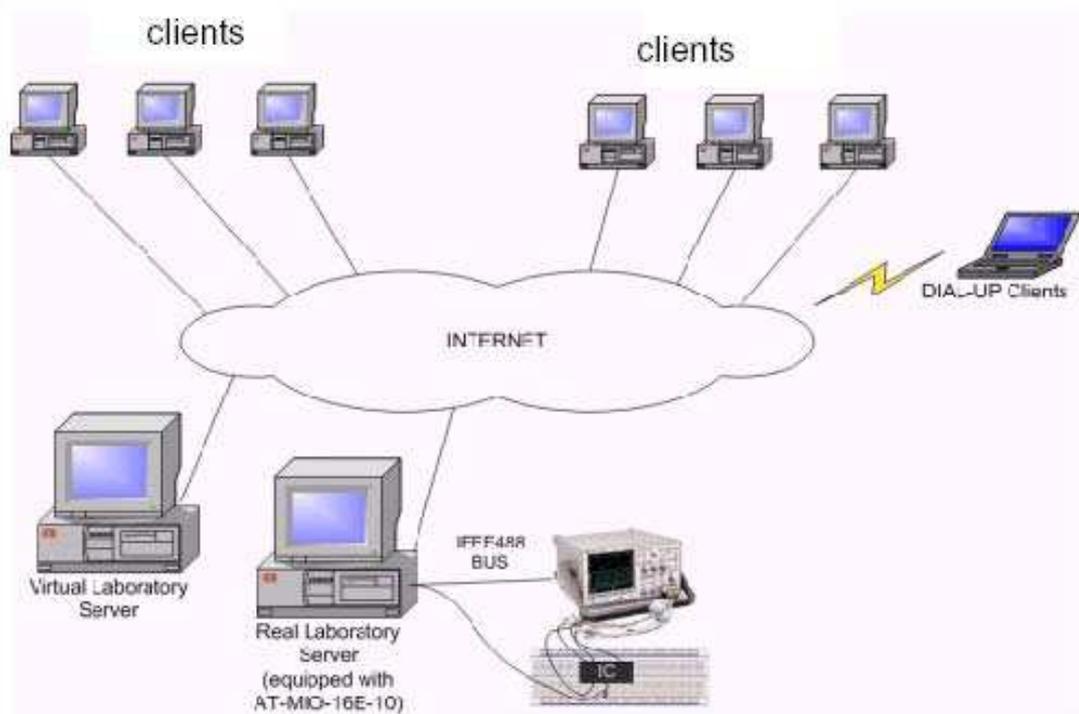


Figura 2.7: *Arquitetura do ISILab (extraído de [20]).*

Os experimentos podem ser realizados de dois modos: direcionados e não direcionados. No modo direcionado, um usuário privilegiado pode alterar as condi-

ções operacionais e atuar no controle dos instrumentos. Os demais usuários podem apenas visualizar a resposta do sistema na tela de seus computadores. Este modo é útil para o apoio a aulas via Internet. No modo não direcionado, todos os usuários podem, em paralelo, controlar as condições de operação e os instrumentos virtuais, recebendo em suas telas apenas os resultados dos próprios comandos. O próprio RLS se encarrega de escalonar as tarefas e processar as sessões de cada usuário.

No experimento, o usuário possui interfaces gráficas de instrumentação virtual para o controle dos instrumentos remotos. A figura 2.8 uma foto das principais interfaces gráficas utilizadas no ISILab, que são: à esquerda um analisador de estados lógicos (“Digiscope”), à direita o gerador de pulsos (“Clock Generator”) e o gerador de sinais (“Angilent”) na parte inferior. Todos criados de forma realística possuindo grande semelhança com os instrumentos reais.

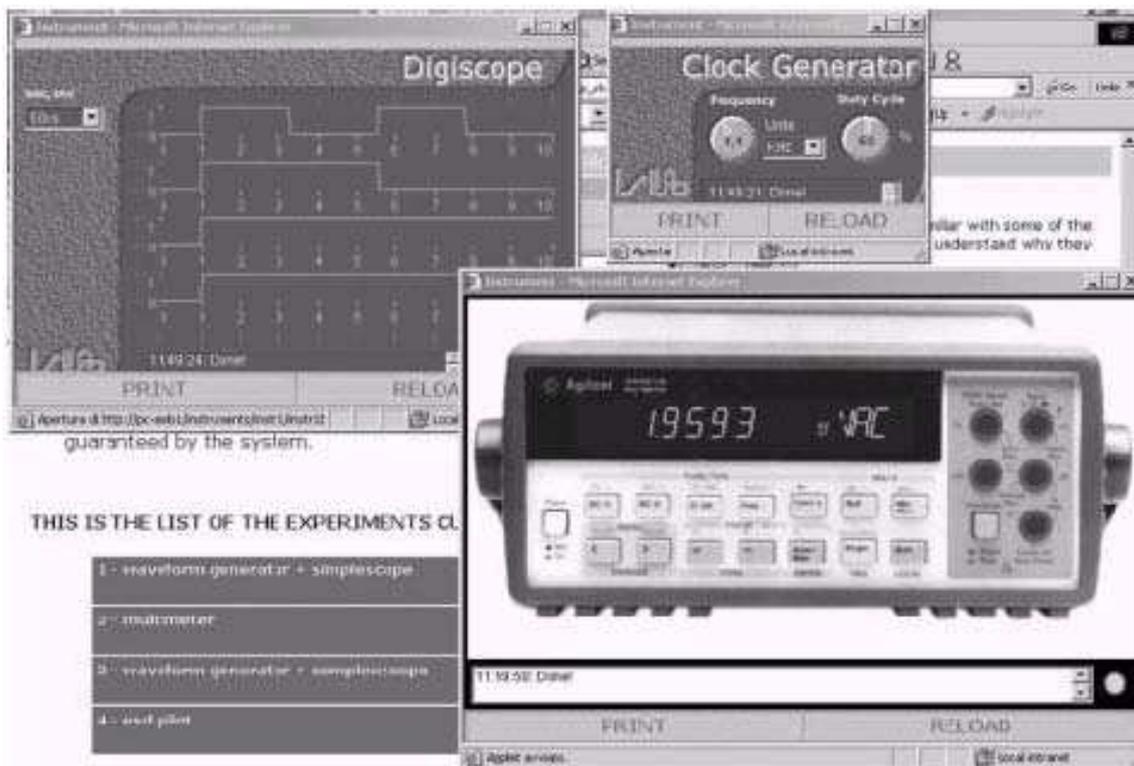


Figura 2.8: Interface do ISILab (extraído de [20]).

O Servidor é um *Pentium II* 350 MHz com 192 MB de memória RAM, sistema operacional *Windows 2000*, equipado com uma placa multifuncional de entrada e saída de *National Instruments AT-MIO-16E-10*, interface *IEEE 488 National Instrument AT-GPIB*, conectando um osciloscópio digital *HP 54645D* com canais

analógicos e digitais [16].

As autoras mencionam que objetivam criar uma rede entre diferentes instituições para o compartilhamento dos recursos computacionais numa arquitetura Grid, mas não especificam os requisitos dos serviços necessários para suportar a sua aplicação nem como será administrada a heterogeneidade de dispositivos e laboratórios.

O modelo de laboratório proposto serve para processamento de experiências em circuitos digitais, com escalonamento de usuários por janela de tempo.

2.2.5 A Teia de Laboratórios Virtuais

A *Teia de Interconexão de Laboratórios Virtuais segundo uma Concepção Multidisciplinar e Colaborativa* (Teia) é uma estrutura baseada em tecnologia da informação, que disponibiliza um ambiente virtual de ensino cooperativo e multidisciplinar em engenharia elétrica agregando um conjunto de módulos didáticos em tópicos de ciência básica e engenharia.

A Teia tem sua origem ligada ao *Laboratório de Processamento de Sinais* (LPS) da *Coordenação de Programas de Pós-Graduação em Engenharia* (COPPE) da *Universidade Federal do Rio de Janeiro* (UFRJ) [21]. Hoje conta com uma teia composta pelos seguintes laboratórios:

- *Páginas de Apoio para o Ensino de Cálculo* (PACUS), com tópicos sobre limite, diferenciação, integração, seqüências e séries;
- *Páginas de Apoio para o Ensino de Álgebra Linear* (PALIN), que aborda operações vetoriais de dimensão finita e espaço de funções;
- *Páginas de Apoio para o Ensino de Sistemas de Controle* (PATROL), que inclui aspectos do controle clássico e moderno;
- *Páginas de Apoio para o Ensino de Processamento de Sinais* (PAPROS), que aborda tópicos sobre Processamento de Sinais de sistemas discretos e analógicos;
- *Páginas de Apoio para o Ensino de Sistemas Lineares* (PALAS), que agrega tópicos da cadeira de sistemas lineares analógicos e discretos;

- *Páginas de Apoio para o ensino de Redes Neurais (PANN)*, que inclui aspectos sobre treinamento supervisionado e não supervisionado além de aplicações em física experimental e engenharia;
- O *WebEAD* que é um módulo de ferramentas de apoio ao desenvolvimento técnico de laboratórios virtuais para Web.

Cada um dos laboratórios da Teia é composto de módulos didáticos que se utilizam de ligações hipertextuais para navegação entre módulos ou laboratórios do Teia, para que o aluno navegue de acordo com seus interesses, necessidades ou orientação do curso. Na Teia o aluno pode seguir seu próprio referencial de aprendizado, consultando material de revisão ou buscando informação à frente para conhecer aplicações e tópicos mais avançados sobre seu tema de estudo. A interligação entre os ambientes didáticos permite que os alunos ligados a outros cursos, como matemática, também possam navegar e conhecer tópicos de engenharia, numa rede multidisciplinar que possibilita o aprofundamento da relação entre ciência básica e engenharia.

Os laboratórios da Teia são compostos por: tutoriais associados aos principais assuntos e técnicas relacionados ao tema do laboratório em questão. Nestes laboratórios também existem atividades de simulação com a utilização de pacotes de software comerciais de uso comum em engenharia elétrica como: (*Matlab, LabVIEW, PSIPICE*)

Ao longo dos módulos, podem ser encontrados testes de avaliação para acompanhar o desenvolvimento do aluno. Em caso de dúvidas o aluno dispõe de comunicação via correio eletrônico como meio de apoio e atendimento didático.

Ao final dos módulos existe uma seção de desafio, na qual o aluno é exposto a um problema que extrapola o tema abordado pelo tópico, para a verificação da capacidade do aluno de criar sobre um aspecto não abordado, mas que se relaciona diretamente com o conteúdo do tópico estudado.

O módulo WebEAD possibilita a criação e a integração dos laboratórios virtuais da Teia, com várias ferramentas que foram desenvolvidas para aplicações em projetos cooperativos possibilitando a documentação dos módulos e tópicos, além de disponibilizar as facilidades de comunicação entre os alunos e os professores, entre

outras funcionalidades. Como principais ferramentas estão: a ferramenta de cadastro, que realiza controle do cadastro e acesso dos usuários, sua classificação quanto ao conhecimento adquirido e criação de uma lista de correio eletrônico para estimular a comunicação interna e a ferramenta de busca por palavra-chave, que facilita a tarefa de busca de determinada informação no sítio de um *Laboratório Virtual*, uma vez que percorre todas as páginas do diretório de arquivos dos módulos didáticos, apresentando como resultado da busca, uma página de saída com um conjunto de ligações hipertextuais direcionados aos resultados da busca.

Os laboratórios da Teia servem de suporte aos cursos presenciais ou no apoio a projetos de EAD. Seus módulos didáticos permitem o acesso a repositórios de informação, planejamento do desenvolvimento dos cursos, estabelecimento de fóruns de discussão, integração entre hardware e software por meio de simulação e emulação de experimentos remotos, entre outras funcionalidades. As suas facilidades de acesso remoto também permitem que seja utilizado como meio para atender aos alunos temporariamente incapacitados de acompanhar o ensino presencial.

2.2.6 Páginas de Apoio a Sistemas Lineares na UFRJ

A proposta do laboratório virtual *Páginas de Apoio a Sistemas Lineares* (PALAS), se baseia no fato que a teoria de sistemas lineares é uma das mais importantes para o ensino de engenharia elétrica, a partir do fato de que a maior parte dos sistemas em engenharia comportam-se de forma linear, ou podem ter seu comportamento modelado como um sistema linear [22]. No contexto da engenharia elétrica, a teoria de sistemas lineares é de fundamental importância em áreas como controle de sistemas, sistemas de comunicação, processamento de sinais, entre outras.

Por outro lado, apesar de sua importância, verificou-se que uma parte significativa dos alunos de graduação possui dificuldades na assimilação dos conteúdos de Sistemas Lineares, dificuldades estas que podem ser associadas à sutileza dos conceitos e à matemática avançada necessária ao tema [23]. Havendo a necessidade de uma ferramenta auxiliar aos cursos de Graduação em Engenharia, foi elaborado o *Laboratório Virtual* PALAS, que também se caracteriza como plataforma para laboratórios virtuais em projetos EAD.

Na sua concepção, considerou-se que, para o melhor aproveitamento dos avan-

ços tecnológicos na educação, são necessários atualizações na forma de educar e maior foco no deslocamento do aluno para um papel de agente, não apenas mero espectador do processo de aprendizagem. Portanto, o PALAS integrou ao laboratório, aspectos teóricos organizados em módulos, com cada módulo contendo tarefas de simulação, tarefas experimentais com circuitos, exercícios, desafios de aprofundamento de conteúdo, avaliações e ferramentas de comunicação com os instrutores [23].

O PALAS foi desenvolvido com o uso da *Web*, permitindo sua integração a laboratórios de assuntos correlatos como parte de uma Teia [4]. Esta estrutura funciona com cada página HTML contendo um módulo e ligações hipertextuais para os outros módulos. Após a parte teórica o aluno se depara com as tarefas de simulação de cada módulo, endereçadas a um determinado pacote de software, de uso comum entre os alunos. Não há instrumentação virtual integrada ao ambiente. Ao término de cada módulo o aluno se depara com um desafio e testes de avaliação de conhecimento.

2.2.7 Páginas de Apoio ao Processamento de Sinais na UFRJ

Denominado como *Páginas de Apoio ao Processamento de Sinais* (PAPROS), este laboratório virtual foi desenvolvido para o ensino e apoio ao desenvolvimento de aplicações em filtros digitais como mais um laboratório virtual do projeto Teia [24].

A integração com outros laboratórios da Teia é realizada por ligações hipertextuais, possibilitando o acesso a laboratórios de temas correlatos. Porém, sua arquitetura difere do PALAS, pela proposta do uso de experimentos remotos em suas atividades [25].

O PAPROS possui o módulo de filtros digitais, abrangendo tópicos sobre a teoria de filtros elétricos e a realização de filtros digitais de resposta ao impulso (FIR) e resposta infinita ao impulso (IIR) [3]. O material didático está distribuído por oito capítulos na forma de tutoriais, exercícios teóricos, simulações e experimentos virtuais além de um *problema desafio* por capítulo.

A interação com os professores e monitores do curso é apoiado por aplicativos comuns para cooperação em rede, tais como correio eletrônico, salas de bate-papo, listas de discussão e grupos de notícias.

Em sua atual implementação, o usuário do PAPROS, por meio da instalação

de um programa cliente, pode testar o projeto de um filtro digital específico em DSP, com a limitação de estar enviando e recebendo o vetor de dados de entrada e saída diretamente para a memória da placa de DSP, sem a interação com sinais elétricos reais [26].

2.3 Considerações Finais

Este capítulo abrangeu os principais conceitos e definições relacionados à EAD e exemplos de laboratórios virtuais representativos do atual estado-da-arte, que descrevem propostas e protótipos de laboratórios virtuais baseados em tecnologias da Web e instrumentação remota com pacote comercial de software LabVIEW e Barramento GPIB. A preocupação com a integração entre os laboratórios virtuais pode ser destacada nos laboratórios do projeto Teia, e a proposta de utilização de tecnologias de rede Grid como parte da infra-estrutura dos laboratórios virtuais é o diferencial da proposta do ISILab.

Capítulo 3

A Computação Grid

Este capítulo inicia com a apresentação de um conjunto de definições e conceitos relacionados à computação Grid abrangendo as origens do conceito, nomenclatura, descrição e definições dos requisitos, arquitetura e funcionalidades de uma rede Grid. Em seguida são estudados os projetos *Globus* e *Legion*, suas origens evolução e relação com as arquiteturas de *Grid Services*.

3.1 Definições e Conceitos em Computação Grid

Com analogia à infra-estrutura da rede de distribuição elétrica, que forma um sistema interligado entre as diversas usinas geradoras, com alta capilaridade, confiabilidade e de simples acesso ao serviço de distribuição, cientistas da área de computação em meados dos anos 90 idealizaram o conceito de Grid de potência computacional, ou simplesmente Grid [27].

Subjacente ao conceito de Grid, reside o problema de descoberta e coordenação do compartilhamento de recursos em uma *organização virtual* (OV). Uma OV é formada por um conjunto dinâmico de indivíduos e/ou instituições, seguindo regras definidas em cada domínio administrativo para o compartilhamento de seus *recursos*. Por recursos entende-se processamento, armazenamento, largura de banda, SW, dados demais dispositivos de hardware que possam estar associados a um dos *nós* da rede. Cada máquina componente do Grid é considerada um nó da rede [28, 29, 30, 31, 32].

3.2 Categorização em Computação Grid

Alguns autores consideram o Grid mais do que uma tecnologia, e sim um paradigma para o fornecimento de informações, recursos de HW e SW e serviços em rede. A partir da pesquisa de diversos conceitos e técnicas relacionados ao Grid, é possível inferir três tipos de Grid, sejam: o *Grid de Informação*; o *Grid de Recursos*; e o *Grid de Serviços* [28].

O *Grid de Informação* fornece acesso à informação de qualquer tipo que esteja em qualquer computador interligado à Grid, de forma transparente em relação à sua localização. Entre suas principais tecnologias está a Web.

O *Grid de Recursos* fornece acesso, de forma transparente em relação à sua localização, a recursos computacionais como: processamento, armazenamento de dados, banda passante ou algum hardware especial. Tal proposta, dá a impressão da existência de um único computador repleto de recursos. O *Grid de Recursos* refere-se ao conceito de Grid surgido em meados dos anos 90 [33].

O *Grid de Serviço* fornece serviços e aplicações, independente de sua localização, implementação ou plataforma de hardware. Os serviços são construídos sobre os recursos de um *Grid de Recursos*. A principal diferenciação entre os *Grids de Serviço e Recursos* está nos níveis de abstração entre eles. *Grid de Serviço* provê abstração de serviços de rede, enquanto o *Grid de Recursos* permite o acesso a recursos de hardware oferecidos pelas máquinas da rede.

As três categorias de Grid possuem diferentes características e linguagens de representação. No *Grid de Informação* a principal é o *Hypertext Markup Language* (HTML) utilizado para o armazenamento estruturado de informações. Para o *Grid de Recursos e de Serviços* não existe um padrão, onde cada projeto implementa sua própria linguagem para especificações e busca de informação de recursos ou serviços.

3.3 Requisitos

Os principais requisitos que caracterizam a Grid são [27]:

- Autonomia dos múltiplos domínios administrativos: significa que são os proprietários de cada domínio administrativo, e não a OV, que decidem sobre o gerenciamento e a política de utilização de seus recursos;

- Suporte à heterogeneidade de recursos, uma vez que os recursos disponibilizados pelos nós do Grid podem ser de natureza heterogênic;
- A Escalabilidade, pois o Grid pode englobar um rede da ordem numérica de milhões de nós;
- Capacidade de adaptação dinâmica, pois em caso de falha de recurso deve haver um remanejamento orientado pela aplicação ou pelo gerenciador de recursos.

3.4 Arquitetura

A Grid é composta basicamente de dois componentes, a infra-estrutura e o middleware. Por infra-estrutura pode-se considerar os elementos, hardware e software necessários à conexão dos nós da rede que podem ser compostos de diferentes plataformas de software e equipamentos de hardware. O *Grid Middleware* corresponde aos componentes que irão realizar os serviços de rede [34]. No âmbito da infra-estrutura, os componentes de hardware e software mais comuns são:

- as autoridades certificadoras e *firewalls* para a segurança;
- os roteadores, servidores e *switchs* para a rede;
- ferramentas de monitoramento e gerenciamento de recursos, para o sistema de armazenamento de dados e compartilhamento dos demais recursos de HW.

A infra-estrutura de rede deve fornecer garantia de taxa de transmissão de dados conforme os requisitos de cada aplicação. As ferramentas de gerenciamento servem para a monitoração do consumo e estado dos recursos envolvidos com as aplicações Grid. O sistema de armazenamento de dados serve como suporte à utilização de recursos em aplicações de uso intensivo de dados [35].

A parte de *Middleware* pode ser considerada como um conjunto de serviços de rede. Essa parte corresponde ao núcleo da Grid e o principal diferencial entre os projetos existentes. As funções envolvidas com o middleware são:

- segurança, oferecendo serviços de autenticação, autorização e comunicação segura, entre os recursos da Grid;

- gerenciamento de recursos, que corresponde ao conjunto de serviços responsáveis por tornar transparente a origem dos recursos da Grid;
- serviço de informações, que controlam a disponibilidade dos recursos consolidando o estado dos recursos disponíveis para as aplicações;
- serviços de movimentação de dados, que gerenciam a movimentação de dados na rede e oferecem serviços como compartimento virtual de dados.

Os principais componentes da *Grid Middleware* são:

- *A Fábrica*, composta de todos os recursos distribuídos nos nós da Grid;
- O *Core Middleware*, que oferece serviços como gerenciamento remoto de processos, alocação de recursos, segurança, busca e registro das informações sobre os recursos do sistema e controles ligados a qualidade de serviço (QoS) como reserva de recursos e priorização;
- *User-Level Middleware*, que oferece um ambiente de desenvolvimento de aplicações com ferramentas de programação e ferramentas para o gerenciamento de recursos e agendamento de tarefas.

A Camada de Aplicações e/ou Portais, onde as aplicações em Grid são desenvolvidas utilizando-se linguagem de programação e utilitários próprios do Grid. Os portais Grid oferecem serviços baseados em Web, através destes os usuários podem submeter tarefas à Grid via Web.

3.5 Funcionalidades

Os sistemas Grid integram e coordenam recursos e usuários de diferentes domínios administrativos [35]. Para o compartilhamento de recursos, existe a necessidade da criação das OV permitindo o compartilhamento transparente de recursos entre os usuários da Grid .

Quanto à gerência de recursos e usuários, na Grid, diferentemente de um sistema cliente-servidor, não há um único nó concentrador de informação, sobre os recursos compartilhados. Esta tarefa é distribuída entre todos os nós do sistema independente do domínio administrativo [30].

As plataformas Grid prevêem a necessidade de um sistema de autenticação única, independente do domínio administrativo do usuário, permitindo que ele possa acessar todos os recursos da Grid a partir de qualquer um dos domínios administrativos membros [31].

3.6 O Projeto Globus

O Globus é um projeto baseado em arquiteturas e códigos abertos, desenvolvido pelo “*Argonne National Laboratory*” (ANL) juntamente com a “*University of Southern California*” e mantido por uma comunidade internacional de programadores [36].

Seus objetivos abrangem a criação de uma arquitetura Grid capaz de implementar segurança, busca e gerenciamento de informações, recursos e dados, comunicação, detecção de falhas e portabilidade entre diferentes plataformas.

O *Globus Toolkit 4* especifica um conjunto de ferramentas e bibliotecas de software que dão suporte à arquitetura e às aplicações do Globus Grid. Este conjunto é baseado no conceito de *Grid Services*, uma expansão do conceito de *Web Services* pela arquitetura da *Open Grid Services Architecture* (OGSA), onde cada recurso, como processamento, memória, disco, banco de dados, entre outros dispositivos, é entendido como um serviço de rede.

O projeto Globus OGSA busca justamente implementar um *Middleware* onde os recursos sejam disponibilizados como serviços e que possam ser acessados de forma transparente à heterogeneidade de recursos subjacentes. A OGSA é baseada nos padrões já definidos para os *Web Services* abrangendo a utilização intensiva do *Simple Object Access Protocol* (SOAP), *Web Service Description Language* (WDSL) e *Extended Markup Language XML*. Com a utilização de uma Grid baseada em *Web Services*, também chamados de *Grid Services* é possível a simplificação da migração de aplicações, atualmente projetadas em *Web Services* para aplicações em *Grid Services* [31].

3.7 O Projeto Legion

O Legion é um projeto Grid, da *Universidade de Virgínia nos Estados Unidos da América* (EUA), originado em 1993, cuja proposta é baseada na defesa de uma infra-estrutura orientada a objetos para a construção do núcleo Grid. O princípio de desenvolvimento do Legion teve como objetivo tornar transparente o hardware do Grid, em uma espécie de máquina virtual, dando a impressão de um sistema único, provendo processamento e capacidade.

A máquina virtual agrega um grande número de computadores, com diferentes sistemas operacionais, em uma única abstração, simplificando a tarefa de escrever aplicações para sistemas distribuídos e heterogêneos. Todo o sistema e componentes desenvolvidos no Legion são objetos. A rede provê o gerenciamento de processos, a comunicação inter-processos, o armazenamento de dados, o sistema de arquivos e os serviços de segurança e suporte a linguagens de programação. Uma das principais metas do Legion foi implementar um sistema Grid que possibilitasse, com a orientação a objeto, a simplificação no desenvolvimento de aplicações para o Grid.

Durante o início de sua distribuição, o Globus também experimentou uma certa visibilidade. Foi então que iniciou-se uma série de comparações entre as duas plataformas e seus problemas surgindo a idéia de que poderiam ser complementares. O Globus com sua infra-estrutura de segurança, gerenciamento de recursos parecia complementar o Legion, permitindo às aplicações o acesso a uma Grid baseada em objetos, ferramentas gráficas, aplicativos, etc.

Após surgiu o *Legion G* que, na prática, corresponde ao Legion rodando sobre o Globus, com a promessa de ser integrado em uma única plataforma. Uma série de pesquisas e desenvolvimentos foram realizados nessa linha, até que surgiu o OGSA baseado em *Web Services*. A partir de então houve um redirecionamento na pesquisa do Legion por soluções Grid para plataformas que permitissem a utilização do OGSA, tornando obsoleto o modelo do projeto Legion G após verificada a sua incompatibilidade em suportar a OGSA. O projeto Legion foi então direcionado para uma arquitetura capaz de suportar os *Grid Services* chamada de *OGSI.NET*, utilizando a plataforma .NET, da Microsoft, e seu suporte a *Web Services* para a construção do *Middleware Grid*, numa arquitetura que viabiliza os *Grid Services*.

3.8 Considerações Finais

Este capítulo apresentou conceitos e definições relacionados à Grid e dentre eles a própria origem da nomenclatura advinda da analogia com a infra-estrutura de rede de distribuição elétrica. Também foram abordados de forma geral os requisitos, a arquitetura, as funcionalidades de uma rede Grid e a *Grid Middleware* como camada intermediária entre a aplicação e o núcleo da Grid, e elemento central na arquitetura das *Aplicações Grid*.

Com relação aos projetos existentes, este capítulo tratou especificamente dos projetos Globus e Legion, sua evolução e a convergência das últimas implementações para um *Middleware* baseado nos protocolos *Web Services* gerando os chamados *Grid Services*, ou OGSA e OGSI, onde a cada recurso, como processamento, memória, disco, banco de dados entre outros, é entendido como um serviço de rede.

Capítulo 4

O Modelo Proposto

Este capítulo descreve o modelo conceitual, os requisitos, a arquitetura e os principais elementos e tecnologias necessários a implementação das funcionalidades da *Bancada Virtual* associada à um *Experimento Remoto*, como recurso compartilhado na Teia de laboratórios virtuais

Neste modelo a computação Grid acrescenta os requisitos necessários ao compartilhamento de recursos heterogêneos entre os diferentes domínios administrativos componentes da Teia.

4.1 Modelo Conceitual

O modelo conceitual apresenta um modelo de solução para a implementação de *Bancadas Virtuais* associadas a *Experimentos Remotos* nos *Laboratórios Virtuais* do projeto Teia. As *Bancadas Virtuais* são compostas de um conjunto de *Instrumentos Virtuais*, que são *interfaces de usuário* associadas a instrumentos reais, como parte de um arranjo de *Experimento Remoto Real*, ou associadas a um *Experimento Remoto Virtual* gerado por algum programa simulador.

Os *Experimentos Remotos* são executados sobre dispositivos eletrônicos programáveis pelo usuário, que realizam algum processamento de sinais elétricos. Os sinais elétricos são injetados e observados pela instrumentação eletrônica controlada por computador, associada ao arranjo experimental. Com isso os alunos que acessam os *Experimentos Remotos* através das *Bancadas Virtuais* obtêm respostas equivalentes as que obteriam na realização do mesmo experimento em uma *Bancada*

Real de laboratório.

A *Bancada Virtual* proposta é um recurso compartilhado entre os diversos *Laboratórios Virtuais na Teia*. Como recurso, a *Bancada Virtual* deve possibilitar o acesso remoto autenticado e seguro, além da gerência de adição, exclusão e classificação de todos os usuários da Teia, independente da instituição ou do domínio administrativo de rede a que pertençam.

As *Bancadas Virtuais* podem ser compartilhadas e estarem distribuídas sob diferentes domínios administrativos de rede e sobre plataformas computacionais, tal como os próprios *Laboratórios Virtuais da Teia*. Como cada domínio de rede possui seu próprio gerenciamento de recursos e neste modelo, as *Bancadas Virtuais* podem ser adicionadas ou retiradas dinamicamente da rede, sem a necessidade de reconfiguração de todos os elementos da rede. No programa de aplicação, a localização do recurso compartilhado é transparente ao usuário, ou seja, os próprios serviços da rede se encarregam de procurar e alocar para ele o recurso que atenda a sua necessidade. Para isso, existe um sistema de informação de recursos hierárquico, distribuído e acessível via Web.

O usuário de uma *Bancada Virtual* pode realizar um ou mais tipos de experimento e projetar as suas principais configurações. A partir do projeto, o aplicativo gera um programa que é carregado no dispositivo programável e, em seguida, o usuário tem acesso aos *Instrumentos Virtuais* do arranjo experimental para o controle do experimento e a aquisição dos dados experimentais.

A integração da aplicação de *Bancada Virtual* com a Teia é realizada através da inserção das *interfaces gráficas* das *Bancadas Virtuais* nas páginas hipertextuais dos *Laboratórios Virtuais* da Teia. Isto permite a realização dos experimentos apenas com o uso de um programa navegador.

4.2 Requisitos

Os requisitos para a construção das *Bancadas Virtuais* abrangem necessidades em hardware, software, serviços de rede e integração com a Teia. Para a implementação do modelo conceitual proposto, os seguintes requisitos devem ser atendidos:

Requisitos de Hardware

- servidor baseado em plataforma de computador pessoal, PC, devido aos seus custos e uso comum em instituições de ensino;
- arranjo experimental com capacidade de controle e monitoramento remoto via PC.

Requisitos de Software

- acesso remoto multiplataforma, pois o programa cliente poder ser executado em diferentes plataformas computacionais independente do hardware e sistema operacional;
- autenticação e gerência de usuários independente dos seus domínios administrativos;
- busca e alocação de recursos associados aos experimentos remotos, de forma transparente quanto à localização e independente do domínio administrativo de rede;
- estrutura de software baseada em módulos, que permita a execução distribuída dos módulos e facilite o reaproveitamento dos *device drivers* utilizados para o controle dos instrumentos e equipamentos remotos.

4.3 Arquitetura

A arquitetura proposta é uma estrutura capaz de atender aos requisitos da *Teia de Interconexão de Laboratórios Virtuais* de forma altamente escalável e de gerenciamento distribuído com o compartilhamento das *Bancadas Virtuais* associadas a *Experimentos Remotos*.

O núcleo da proposta é transformar o compartilhamento de aplicativos e recursos necessários aos laboratórios virtuais em serviços de rede. Os serviços presentes na arquitetura estão aqui distribuídos em seis camadas. A figura 4.1 ilustra a estruturas das camadas de serviço, nela pode-se observar que a camada *Localizador*

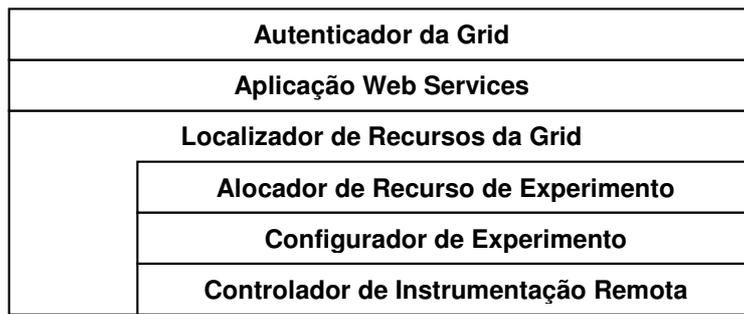


Figura 4.1: *Diagrama de Camadas da Arquitetura Proposta*

de Recursos da Grid é a única que se comunica com mais de uma camada inferior, possuindo interface com as três últimas camadas inferiores.

A camada *Autenticador da Grid*, atende ao requisito de autenticação única para acesso aos recursos de todos os domínios administrativos da Grid. Esta camada também provê segurança na troca de informações de autenticação.

Após a autenticação, o usuário tem acesso à camada de *Aplicação Web Services*. Nela o usuário requisita remotamente o serviço de uma Bancada Virtual, num ambiente multi-plataforma integrado via navegador, atendendo ao requisito de execução em diferentes plataformas computacionais.

A camada *Localizador de Recursos da Grid* se encarrega de pesquisar todos os servidores que possam atender aos serviços de *Configurador de Experimento* e de *Controlador de Instrumentação Remota*, repassando essa informação à camada de *Alocador de Recurso de Experimento*.

A camada *Alocador de Recurso de Experimento*, de posse das informações sobre os recursos disponíveis na Grid e da requisição do usuário, reserva o recurso na camada *Controlador de Instrumentação Remota* e inicia o serviço da camada *Configurador de Experimento*. Esta camada, juntamente com a camada *Localizador de Recursos da Grid*, atende ao requisito de busca e gerenciamento de recursos.

A camada *Configurador de Experimento* realiza um serviço que permite ao usuário projetar ou configurar sua experiência. Após a ação do usuário a camada *Configurador de Experimento* repassa os parâmetros do experimento, gerados pelo usuário e inicia o serviço da camada *Módulo Controlador de Instrumentação Remota* reservado anteriormente.

Finalmente, o usuário tem acesso ao controle do experimento, com os ser-

viços da camada *Controlador de Instrumentação Remota*, estes serviços iniciam o experimento conforme os parâmetros recebidos da camada *Configurador de Experimento* e permite ao usuário, controlar e observar os estímulos de entrada e saída na *Bancada Virtual*. Assim a camada *Controlador de Instrumentação Remota*, atende, em conjunto com a camada *Configurador de Experimento*, os requisitos de controle e monitoramento remoto.

4.4 Componentes da Bancada Virtual

Uma das funcionalidades necessárias à *Bancada Virtual* é o acesso multiplataforma via rede local ou Web. Para tanto é preciso disponibilizar um computador servidor de aplicações que possam ser executadas através de navegadores.

Algumas soluções têm sido desenvolvidas nessa área, entre elas o uso dos protocolos *Web Services* (WS) vem se destacando como uma importante fonte de interoperabilidade entre plataformas computacionais [37].

4.4.1 O Servidor *Web Services*

Algumas aplicações de *Bancadas Virtuais* associadas a *Experimentos Remotos*, utilizam *Web Services* em conjunto com aplicações *Java*, *Ativex* ou *CGIs*, o que possibilita a visualização e o controle do painel de instrumento virtual via navegador[20][17].

A aplicação *Web Services* é identificada por um *Universal Resource Locator* (URL) e que pode ser acessada remotamente pela troca de mensagens baseadas em XML, escritas conforme as regras especificadas no *Simple Object Access Protocol* (SOAP).

De modo a entender melhor o funcionamento das *Web Services*, considere o modelo teórico descrito a seguir, na qual ela é formada por três entidades e três operações, o modelo teórico chama-se *Service-Oriented Architecture* (SOA). Na figura 4.2, o *Consumidor* é uma aplicação que deseja receber o serviço de outra aplicação, para encontrá-la o usuário recorre, por meio da operação *Localizar*, ao Diretório de *Web Services*. Este, por sua vez, retorna as informações de contato e detalhes da conexão para um provedor daquele *Web Services*, registrado no seu *Diretório Web*

Services.

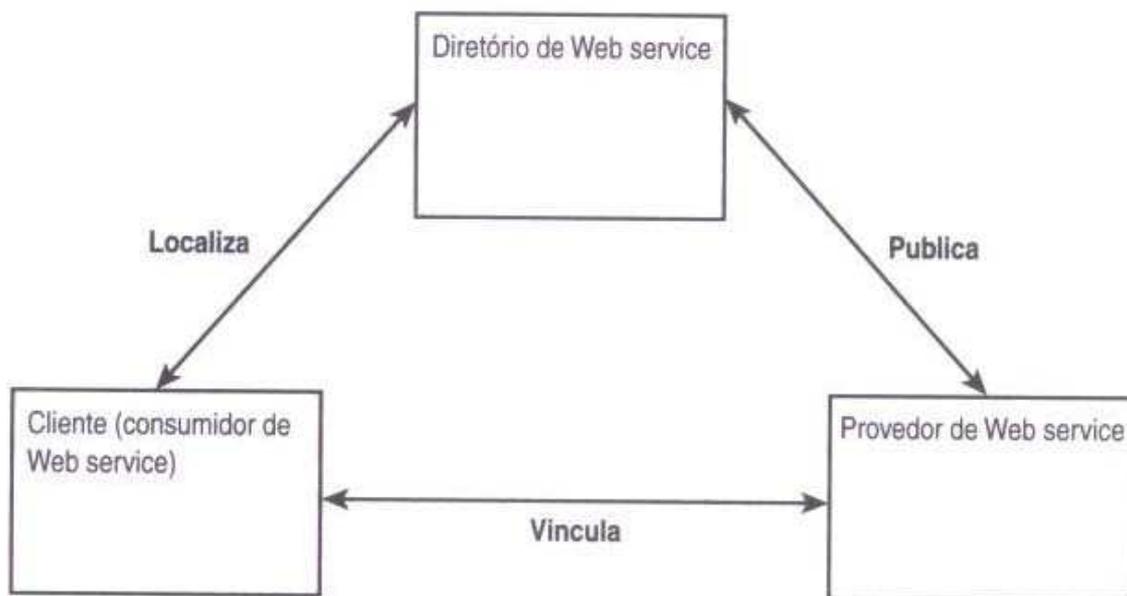


Figura 4.2: *Arquitetura SOA (extraído de [38]).*

Após o processamento, o aplicativo *Web Services* repassa uma resposta em formato SOAP ao servidor Web que a encaminha ao aplicativo cliente. Na figura 4.3 o navegador cliente em HTTP (o SOAP também suporta o uso dos formatos de transporte FTP, SMTP e JMS) repassa o documento ao mecanismo SOAP que, após a análise e validação do arquivo, envia seus dados para o aplicativo cliente que originou a chamada.

A figura 4.4 contém componentes utilizados na implementação do modelo SOA, que são: a linguagem XML, o protocolo SOAP, o *Web Service Description Language* (WSDL), que é uma especificação para a descrição em XML do aplicativo quanto às chamadas e métodos a que ele responde e o *Universal Discovery, Description and Integration* (UDDI), um mecanismo de procura de serviços, que armazena os URLs e WSDL dos *Web Services* publicados para informação dos requisitores de serviços. O protocolo de transporte é o HTTP entretanto não aparece nesta figura.

Posto que um servidor WS é ao mesmo tempo um servidor Web, podendo também ser utilizado para oferecer os serviços e prover conteúdo adicional em HTML, para viabilizar o acesso ao *Experimento Remoto*, além do servidor WS é necessário um servidor de *Experimento Remoto* na camada do *Controlador da Ins-*

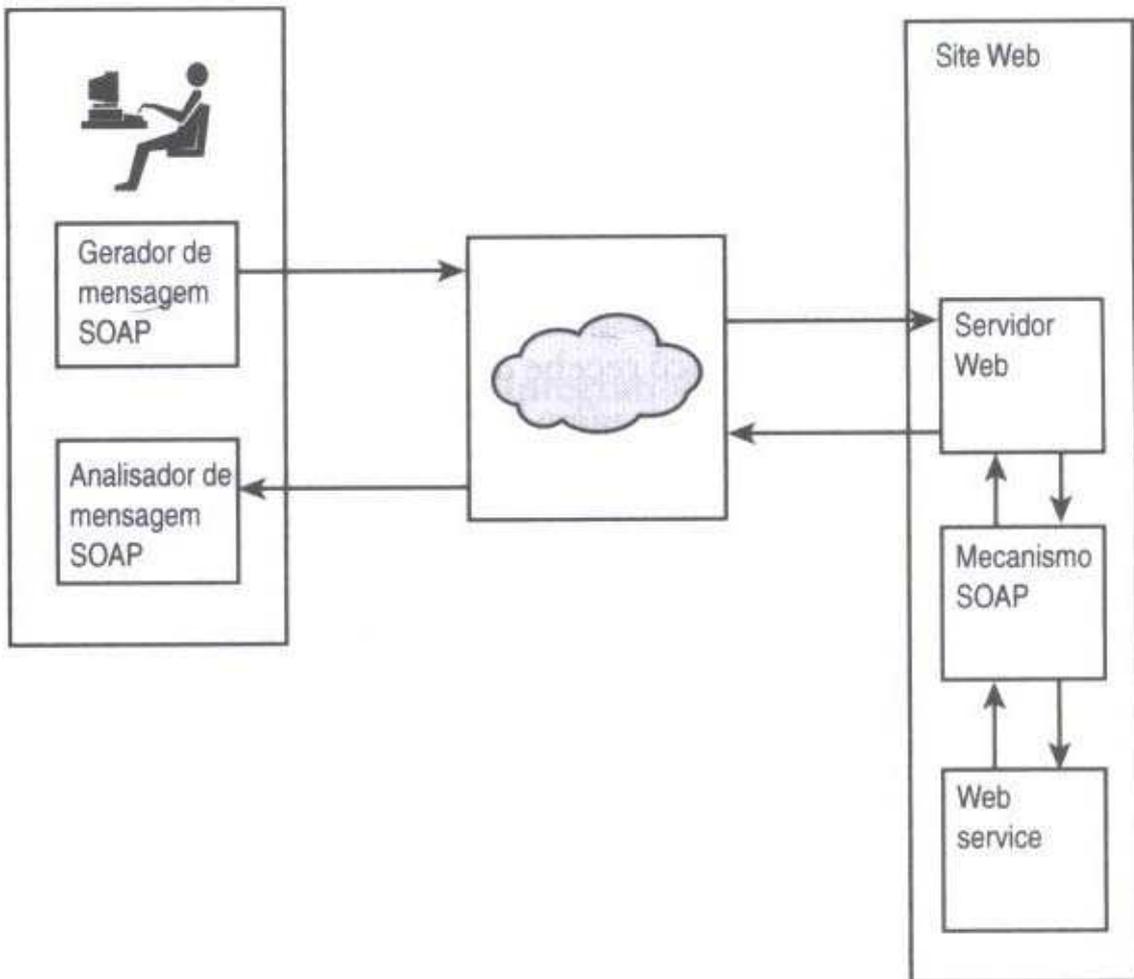


Figura 4.3: *Aplicativo Web Services (extraído de [38]).*

trumentação Remota, que pode ou não estar na mesma máquina do servidor WS. Para construir o servidor de *Experimento Remoto*, os equipamentos e os instrumentos reais ou o sistema de aquisição devem estar conectados a um computador servidor com acesso à rede. Essas interfaces variam em quantidade e padrões conforme o arranjo experimental.

4.4.2 Interface com o PC

Para a interface entre os equipamentos, instrumentos e o PC são necessárias vias distintas, uma para o controle da entrada e aquisição de dados, e outra para a configuração do experimento.

Na configuração do experimento, a interface serial se encarrega de transferir os comandos para a placa de configuração. Esses comandos variam desde um rote-

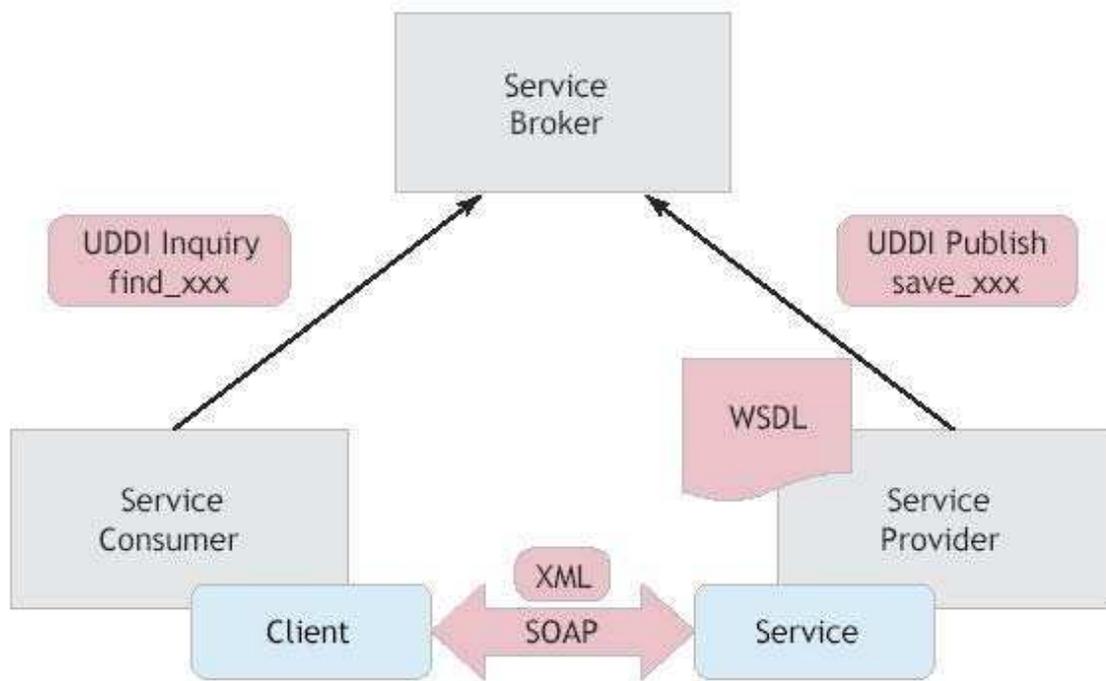


Figura 4.4: Componentes da “Service-Oriented Architecture” (SOA) (extraído de [39]).

amento para uma matriz de interconexão até um programa em código de máquina para execução em algum dispositivo programável.

Para o controle dos instrumentos reais a utilização de um barramento de controle industrial de uso geral elimina a necessidade de uma conexão por instrumento, economizando a quantidade de cabos e portas seriais. E possibilita, a um mesmo servidor, controlar os instrumentos de um ou mais, *Experimentos Remotos* através de um único barramento.

4.4.3 Arranjo Experimental

É na *Aplicação* que o usuário define qual experimento o arranjo experimental irá realizar [6, 8, 18, 20]. Ao receber essas informações, o servidor de *Experimento Remoto* as traduz como parâmetros de configurações e as envia serialmente para o arranjo experimental que, a partir dessas informações, passa a realizar a função do experimento esperado. As placas de experimento podem ser do tipo matriz de contatos ou placas avaliação de dispositivos programáveis

No arranjo experimental a geração e aquisição de sinais elétricos aplicados à placa de experimentos é realizada por um gerador de sinais e um osciloscópio digital, interfaceados com o PC através de um barramento de controle industrial. O software servidor, utilizando-se de *device drivers* específicos, possibilita o envio de comandos e a captação de dados dos instrumentos remotos.

4.4.4 O Aplicativo

O desenvolvimento da aplicação baseada em *módulos* permite a distribuição destes entre diversas tarefas ou máquinas da rede. Esses módulos podem comunicar-se entre si e distribuir a carga computacional pelos elementos da rede ou configurar servidores independentes de determinadas etapas da sessão de experimento [20].

De forma geral a autenticação e gerência dos usuários é realizada a partir de um *módulo* da aplicação através de simples validação de cadastro e senha.

A programação em linguagem visual orientada a componentes facilita a implementação da modularidade, sendo comum a utilização da XML para a troca de informações entre elementos que podem estar distribuídos sobre diferentes plataformas de computação. Para as *Bancadas Virtuais* estes elementos podem ser aplicativo cliente e aplicativo servidor ou módulos do aplicativo servidor distribuídos por várias máquinas [20].

4.5 Integração com a Grid

Para a integração com a Grid é preciso que os servidores da Teia e os servidores de *Bancadas Virtuais*, sejam nós de uma rede Grid de *Middleware* baseado em *Grid Services* OGSA ou OGSI. Com a Teia de *Laboratórios Virtuais* sobre a arquitetura Grid, cada servidor de *Laboratório Virtual* ou *Bancada Virtual* passa a ser também um nó da rede, e como nó compartilha seus recursos como *serviços de rede* com os demais nós do Grid.

O aplicativo da *Bancada Virtual* deve ser modelado sobre a arquitetura de protocolos *Grid Services* do projeto Grid escolhido. Quando o aplicativo da *Bancada Virtual* for realizado antes da implementação da Grid, este deve ser modelado sobre a arquitetura de protocolos *Web Services* para que e em seguida, quando a rede

Grid estiver ativa, não haja a necessidade de profundas alterações no projeto, sendo principais as seguintes:

- publicação do WSDL da aplicação *Web Services*, no formato requerido pelo mecanismo de publicação e descoberta do projeto Grid utilizado;
- alteração no formato de mensagens de busca ao UDDI do aplicativo cliente para que este utilize o formato de mensagem próprio do diretório de busca do Grid Services;
- caso a Grid tenha incorporado um formato de segurança na troca de mensagens SOAP, tanto o aplicativo cliente quanto o aplicativo servidor, devem acrescentar a funcionalidade de chamada do serviço de decifragem da mensagem antes de repassá-la para o mecanismo SOAP.

4.6 Considerações Finais

Este capítulo abordou o modelo conceitual proposto que serve de base para a implementação da *Bancada Virtual* com compartilhamento de recursos na Teia. Na seqüência está a descrição dos requisitos do modelo e de sua arquitetura em camadas de serviços de rede.

Os componentes da *Bancada Virtual*: Servidor *Web Services*, Interface entre o experimento e o PC, o *Arranjo Experimental* e o *Programa Aplicativo*, são descritos e relacionados às camadas de serviço de rede do modelo proposto.

Por fim, para possibilitar o compartilhamento de recursos de *Bancadas Virtuais* entre os laboratórios da Teia é descrita uma orientação sobre forma de integração destas *Bancadas Virtuais* a rede Grid com *Middleware* baseado em *Grid Services*.

Capítulo 5

O Protótipo da Bancada Virtual

O presente capítulo apresenta o *Instrumento Virtual* associado a um *Experimento Remoto* doravante denominado *Bancada Virtual*, implementado com base no modelo proposto. Aqui será abordada a arquitetura do protótipo da *Bancada Virtual*, suas limitações, a integração do protótipo com o *Laboratório Virtual* PAPROS e suas condições operacionais.

5.1 Visão Geral do Protótipo

O protótipo apresentado neste capítulo foi projetado e implementado para atender parte dos objetivos e requisitos do modelo teórico proposto e verificar a viabilidade da integração dos componentes da *Bancada Virtual*, em um ambiente de pré-existente de *Laboratório Virtual*.

Para o entendimento do protótipo, alguns de conceitos tornam-se necessários, o primeiro deles é a definição de *sessão*. É entendido como *sessão*, a interação do usuário com a *Bancada Virtual*, a partir do momento em que ele recebe a interface de autenticação do usuário, até o momento em que ele comanda o término da interface de controle da instrumentação virtual.

Outro conceito importante é o de projeto do *Experimento Remoto*, por projeto entenda-se a escolha das configurações e parâmetros do experimento, como exemplo, a configuração dos parâmetros da interface de configuração do *Experimento Remoto* de filtros digitais implementada no protótipo, onde é possível *projetar* o tipo de filtro, a ordem do filtro, o janelamento do filtro etc.

Em fim, o conceito de *reserva* está relacionado literalmente à reserva de um arranjo experimental a um usuário que esteja realizando o projeto do *Experimento Remoto*, como forma de garantir que este usuário, não perca tempo projetando um *Experimento Remoto*, para o qual não haja recursos disponíveis no momento de sua execução.

O protótipo é uma *Bancada Virtual* composta por vários módulos, capazes de, em conjunto, possibilitarem a realização de *Experimentos Remotos*, com acesso via *navegador* Web, o gerenciamento e cadastro de usuários e recursos, e realizar o controle de instrumentos remotos. Ele foi implementado em estrutura modular, com interfaces entre os módulos, baseados em documentos na linguagem XML. A codificação foi realizada em *Linguagem Visual Orientada a Componentes LabVIEW*, possibilitando a integração dos módulos em um único aplicativo.

A figura 5.1 representa a estrutura de módulos no qual foi baseado o protótipo e sua interação. Nela as trocas de mensagens entre os módulos possuem letras para auxiliar a sua descrição. A *sessão* inicia com o aplicativo com o aplicativo cliente efetuando a autenticação para ter acesso aos recursos da *Bancada Virtual*. Reconhecido o registro e a autenticidade do usuário, sua requisição é enviada a um nóduo capaz de procurar e alocar os recursos disponíveis para o projeto e realização do *Experimento Remoto*. Uma vez iniciado o projeto a sua realização é assegurada pela reserva de recursos do arranjo experimental para a realização remota do experimento.

A descrição das letras que ilustram a figura 5.1 é a seguinte:

- “A” significa a troca de mensagens entre a aplicação cliente e o *Módulo Autenticador do Usuário* para validação de cadastro e senha do usuário;
- “B” indica a consulta do *Módulo Autenticador do Usuário* à TGU como forma de verificar se ele existe para o sistema;
- “C” indica a passagem das requisições das *Aplicações Cliente* do *Módulo Autenticador do Usuário* do experimento remoto tipo1 ao *Módulo Alocador de Recursos de Experimento*;
- “D” representa a consulta que o *Módulo Alocador de Recursos de Experimento* faz à Tabela TFBV no sentido de suprir as requisições das aplicações clientes

com *Bancadas Virtuais* disponíveis;

- “E” representa a consulta que o *Módulo Alocador de Recursos de Experimento* faz à Tabela BVU para verificar o tipo de *Bancada Virtual* que deverá ser alocado ao usuário;
- “F” representa a mensagem de acionamento do *Módulo Alocador de Recursos de Experimento* ao *Módulo Configurador de Experimento Remoto*;
- “G” representa a mensagem de reserva de *Bancada Virtual* feita para o *Módulo Configurador de Experimento Remoto*;
- “H” representa a mensagem de inicialização do *Módulo Configurador de Experimento Remoto* repassando os parâmetros de projeto do *Experimento Remoto* para o *Módulo Controlador de Instrumentação Remota*.

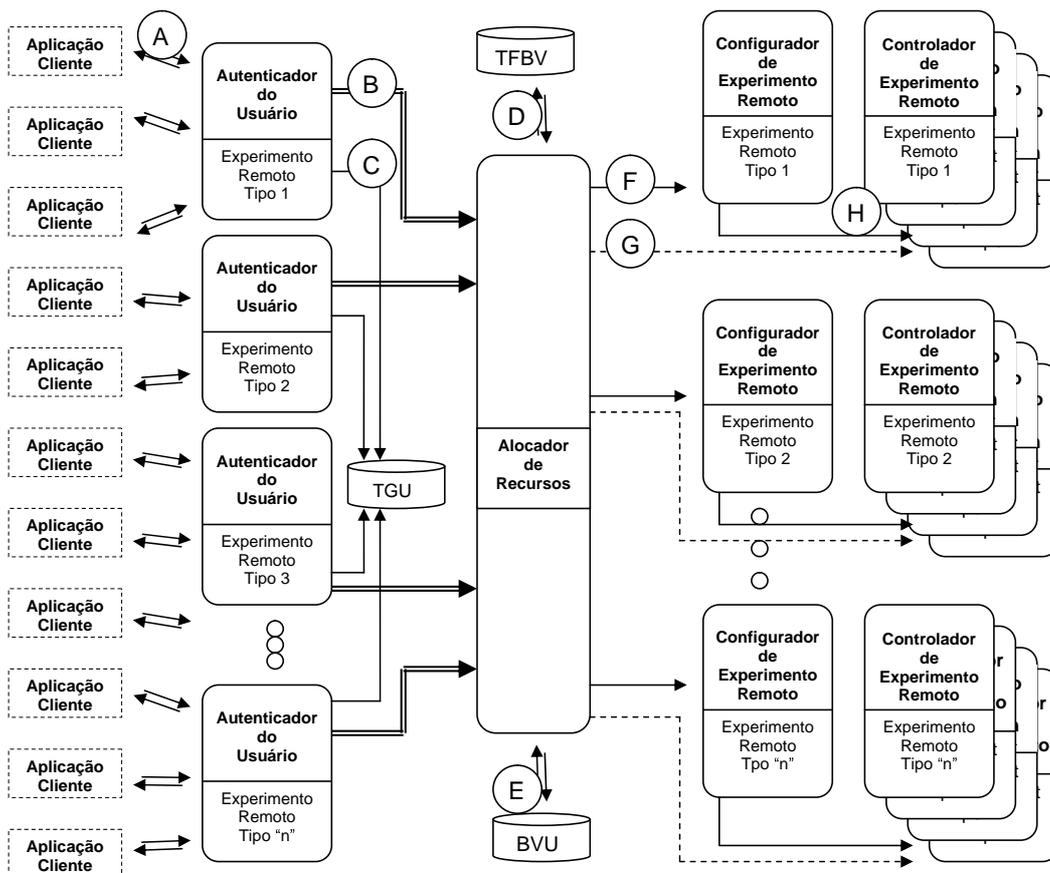


Figura 5.1: Arquitetura Modular do Protótipo

5.2 Arquitetura

A arquitetura do protótipo foi desenvolvida em módulos, baseando se na arquitetura do modelo teórico e considerando a necessidade de atender ao requisito de estrutura de software baseada em módulos, que permita a execução distribuída dos módulos e facilite o reaproveitamento dos *device drivers* utilizados para o controle dos instrumentos e equipamentos remotos.

O atendimento aos requisitos do modelo foi mais próximo nas camadas *Controlador da Instrumentação Remota*, *Configurador do Experimento* e *Alocador de Recurso de Experimento*. Em substituição às camadas *Autenticador do Grid* e *Buscador de Recursos*, foi implementado o *Módulo Autenticador do Usuário*, associado a uma única máquina, portanto não havendo necessidade da camada *Buscador de Recursos* para a *Bancada Virtual* em outra máquina da rede.

- Aplicação *Web Services*, responsável pela interação entre o aplicativos cliente e o servidor, que atende ao requisito de acesso remoto multiplataforma;
- Autenticador do Usuário, responsável pela verificação de cadastro e senha dos usuários, que atende parcialmente ao requisito de autenticação e gerência de usuários independente dos seus domínios administrativos, atende parcialmente pois a gerência dos usuários fica restrita ao domínio administrativo do servidor da *Bancada Virtual*;
- Alocador de Recurso de Experimento, chama o *Módulo Configurador de Experimento* e reserva uma sessão de *Experimento Remoto* no *Módulo Controlador de Instrumentação Remota*, que atende parcialmente ao requisito de busca e alocação de recursos associados aos experimentos remotos, de forma transparente quanto à localização e independente do domínio administrativo de rede, parcialmente pois este *Módulo* só opera dentro do próprio domínio administrativo de rede;
- Configurador do Experimento, serve para o usuário projetar as principais especificações do experimento que irá realizar, que atende ao requisito de acesso remoto multiplataforma;

- Controlador da Instrumentação Remota, provê acesso e controle aos instrumentos remotos do arranjo experimental, que atende ao requisito de acesso remoto multiplataforma.

5.2.1 Módulo de Aplicação Web Services

O *Módulo de Aplicação Web Services* se comunica com todos os demais módulos, isso porque é ele quem provê a comunicação com o aplicativo cliente. O *Módulo de Aplicação Web Services* atua como um servidor de *Web Services* e engloba o conjunto de serviços de rede necessários à configuração do servidor de *Bancada Virtual*, que são: um *Servidor Web*, o mecanismo SOAP entre o aplicativo cliente e o aplicativo *Web Services* protótipo. A integração do protótipo com os *Laboratórios Virtuais da Teia*, que se dá pela publicação do experimento desejado nas páginas HTML do Servidor. Essa publicação chama, um programa cliente em *ActiveX* que inicia o “*Plug-in*” *LabVIEW* e procede a comunicação com o Servidor *Web Services*.

O aplicativo cliente *Web Services* solicita ao servidor *Web Services* a interação com o *Módulo Autenticador do Usuário*. O servidor então envia uma mensagem SOAP com todos os dados necessários para que o aplicativo local reconstrua a interface gráfica deste e dos demais *Módulos* que se seguirão após a autenticação.

5.2.2 Módulo Autenticador do Usuário

Este é o primeiro módulo que interage com a aplicação cliente. A função do *Módulo Autenticador do Usuário* é gerenciar o cadastro de usuários, senhas, permissões de uso e o acesso de requisições ao *Módulo Alocador de Recursos de Experimento*.

Suas interfaces gráficas com o usuário, são as de acesso e administração. A interface de acesso recebe do aplicativo cliente as informações de usuário e senha. A interface de administração permite a edição dos cadastros de usuários e senhas na *Tabela Global de Usuários* (TGU), sendo permitida sua utilização apenas na máquina local ao servidor da (TGU), para evitar o trânsito de mensagens que contenham a senha do administrador, sem criptografia, através da rede.

Existe um *Módulo Autenticador do Usuário* por tipo de *Experimento Remoto*. Fica a cargo do integrador definir o tipo de *Experimento Remoto* que irá associar ao

seu *Laboratório Virtual*. A criação de um *Módulo Autenticador do Usuário* por tipo de *Experimento Remoto*, simplifica a estrutura da aplicação cliente, que se torna independente da heterogeneidade dinâmica de *Experimentos Remotos* existentes na rede, e simplifica a implementação do *Módulo Alocador de Experimento Remoto*.

5.2.3 Módulo Alocador de Recurso de Experimento

O *Módulo Alocador de Recurso de Experimento* se encarrega de verificar a disponibilidade, reservar a sessão, período pelo qual o usuário irá dispor de uma *Bancada Virtual*, e administrar a heterogeneidade das *Bancadas Virtuais* existentes na rede. Todas as informações necessárias ao seu funcionamento estão em duas tabelas em formato XML, chamadas: *Tabela de Bancadas Virtuais e Usuários* (BVU); e *Tabela Tipos e Freqüência de Bancadas Virtuais* (TFBV). A BVU contém a relação entre usuário, *Experimento Remoto* e tipo de *Bancada Virtual*, onde este usuário pode realizar o experimento selecionado; e a TFBV é uma tabela contendo o cadastro das *Bancadas Virtuais* existentes, seu tipo, quantas vezes este tipo de ocorre fisicamente na rede e a disponibilidade de cada bancada.

Os tipos de *Bancada Virtual* existem devido à possível variedade de arranjos experimentais. Os arranjos experimentais variam conforme o modelo e características da instrumentação e da placa de experimento utilizada. Esta diversidade de equipamentos pode causar variações na resposta de um mesmo experimento realizado em arranjos experimentais diferentes, ou seja, Bancadas Virtuais de tipos diferentes podem gerar diferentes resultados para um mesmo *Experimento Remoto*.

O *Módulo Alocador de Recursos de Experimento* é acionado por uma mensagem do *Módulo Autenticador do Usuário* contendo o cadastro do usuário e o experimento que este usuário deseja realizar. A partir dessas informações, o módulo verifica na BVU, se este usuário já realizou alguma vez o experimento solicitado. Caso positivo, na própria BVU existirá um registro contendo o usuário e o tipo de *Bancada Virtual* que o *Módulo Alocador de Recurso de Experimento* deve procurar para este usuário realizar o *Experimento Remoto* de seu *Laboratório Virtual*. Dessa forma, o usuário sempre realizará determinado *Experimento Remoto* num mesmo ambiente de equipamentos e instrumentos. Com isso, evita-se que haja diferenças nos resultados experimentais entre sessões de um mesmo usuário, devido variações

nas respostas dos diferentes tipos de instrumentos e dispositivos dos arranjos experimentais que a aplicação deve suportar.

Todavia, se o usuário estiver realizando o experimento solicitado pela primeira vez, o *Módulo Alocador de Recursos de Experimento* inicialmente busca uma *Bancada Virtual* disponível na TFBV, ao encontrá-la cadastra na BVU um registro associando o usuário, o tipo de *Experimento Remoto* solicitado e o tipo de *Bancada Virtual* disponibilizado pela primeira vez neste experimento. Em seguida procede a alocação de recursos. A busca da *Bancada Virtual* na TFBV é feita preferencialmente pelo tipo que existe em maior quantidade nos registros. Este método incentiva a demanda futura pelos tipos de *Bancadas Virtuais* mais freqüentes e, por conseqüência, reduz aos tipos existentes em menor quantidade.

O *Módulo Alocador de Recursos de Experimento* é apenas um agente de processamento e encaminhamento de requisições e não possui interface com o usuário. Uma vez realizada a busca de recursos, o *Módulo Alocador de Recursos de Experimento* reserva, para a aplicação cliente, uma sessão no *Módulo Controlador de Instrumentação Remota* e disponibiliza na interface gráfica do usuário, o controle do *Módulo Configurador de Experimento* associado ao *Experimento Remoto* requisitado no *Laboratório Virtual*.

5.2.4 Módulo Configurador de Experimento

A inicialização do *Módulo Configurador de Experimento* ocorre com uma mensagem de ativação vinda do *Módulo Alocador de Recursos de Experimento*, o qual, a partir da informação sobre o tipo de *Experimento Remoto* a ser realizado, seleciona e ativa o tipo de *Módulo Configurador de Experimento* a ser inicializado. Na mensagem do *Módulo Alocador de Recursos de Experimento* também está a informação sobre qual tipo de *Bancada Virtual* será iniciada e sobre qual sessão de *Experimento Remoto* o usuário irá realizar suas tarefas.

Para cada tipo de *Experimento Remoto* existe um *Módulo Configurador de Experimento* associado. No *Módulo Configurador de Experimento*, o usuário pode projetar as principais especificações do experimento que deseja realizar, limitado somente pelas faixas de valores permitidas na interface gráfica que o módulo disponibilizar.

O projeto do experimento gera um conjunto de parâmetros que são enviados na mensagem de ativação que o *Módulo Configurador de Experimento* repassa para o *Módulo Controlador de Instrumentação Remota*. Este módulo por sua vez utiliza-se do conjunto de parâmetros recebidos na mensagem de ativação para gerar, compilar e carregar o código em C++ na memória do DSP na placa do *Experimento Remoto*.

Esta etapa de configuração do *Experimento Remoto* é semelhante à etapa de seleção do experimento em laboratórios de circuitos elétricos [8, 6], nos quais o aplicativo cliente envia para o servidor de experimento uma mensagem indicando qual roteamento a matriz de contatos deve realizar para configurar determinado circuito.

5.2.5 Módulo Controlador de Instrumentação Remota

O *Módulo Controlador de Instrumentação Remota* tem a função controlar os elementos do arranjo experimental da *Bancada Virtual* associada ao *Experimento Remoto* em operação. Na atual versão do protótipo, ele controla dois *Instrumentos Virtuais* associados a dois *Instrumentos Remotos*, um osciloscópio e um gerador de sinais, interfaceados ao PC por um barramento GPIB, e a placa *ADSP 21160M EZKIT Lite* onde é realizado o *Experimento Remoto*.

Este módulo é iniciado por uma mensagem do *Módulo Configurador de Experimento*. Ele recebe uma mensagem contendo um conjunto de parâmetros necessários para gerar, compilar e carregar o código em C++ na memória do DSP da placa de *Experimento Remoto*. Portanto, ao depender do DSP e do tipo de instrumentação utilizada pelo conjunto, este módulo é dependente do hardware existindo um tipo de *Módulo Controlador de Instrumentação Remota* para cada tipo arranjo experimental existente. Ou seja, para cada tipo de *Bancada Virtual* existe somente um tipo de *Módulo Controlador de Instrumentação Remota* associado.

Em sua interface gráfica o *Módulo Controlador de Instrumentação Remota* disponibiliza um osciloscópio, um gerador de sinais e um botão que encerra a sessão. O encerramento da sessão é pode ser realizado pelo usuário ou por um temporizador interno à aplicação.

5.3 Implementação

Nesta seção será realizado o detalhamento dos componentes do protótipo e suas funcionalidades, acrescentando-se à descrição funcional, algumas informações específicas sobre os equipamentos e software utilizados no desenvolvimento do instrumento virtual.

O computador utilizado como servidor do laboratório virtual é um PC com processador Pentium 1GHz, sistema operacional Windows2000 e 120MB de memória RAM. O PC está conectado à rede local do LPS, com o *Servidor de Web* que faz parte do software LabVIEW 7.1.

Para o controle dos instrumentos, o servidor possui uma placa PCI de interface com o barramento GPIB que conecta o osciloscópio Tektronix TDS1012 e o gerador de sinais HP 33120A ao PC. O equipamento de teste é uma placa de avaliação de DSP - ADSP 21160M EZKIT Lite - interligada ao osciloscópio e ao gerador de sinais pelas suas entrada e saída analógicas, no padrão de áudio *LINE*, e conectada ao PC via USB.

5.3.1 A Placa de Avaliação ADSP 21160M EZKIT Lite

O protótipo implementado possui um arranjo experimental capaz de permitir a realização de experimentos em processamento digital de sinais, o elemento central deste arranjo é a placa de testes e avaliação *ADSP 21160M EZKIT Lite* comercializada pela Analog Devices. Esta placa é um circuito baseado no DSP *ADSP 21160M* e serve de interface entre o mesmo e um conjunto de dispositivos adicionais de uso comum em sistemas baseados em DSP.

A *Placa* é conectada ao servidor de *Experimento Remoto* por uma interface serial USB, de onde recebe a carga de software e os sinais e controle do *Módulo Controlador de Instrumentação Remota*. Os sinais elétricos de entrada são produzidos no gerador de sinais e chegam à *Placa* pela entrada de áudio, onde são digitalizados pelo CODEC e enviados como amostras ao DSP, que executa o processamento destas amostras e envia as amostras processadas para a saída CODEC. Este as converte novamente para sinal elétrico que é observado pelo osciloscópio.

A descrição completa dos recursos oferecidos pela *Placa* está disponível no

manual e no site da *Analog Devices*, mas não será transcrita aqui por se tornar uma enumeração demasiadamente extensa [40, 41]. Entre esses recursos destacam-se pela utilização no arranjo experimental, os seguintes:

- 4 MBits em módulos de memória RAM externa para o armazenamento das variáveis do programa;
- 4 MBits na unidade de memória Flash ROM para o armazenamento do programa principal;
- CODEC de áudio para processamento dos sinais elétricos de entrada e saída;
- interface USB para o recebimento dos sinais de controle, carga de depuração de software via PC.
- oscilador local único para a sincronizar de todos os elementos da *Placa*.

A figura 5.2 mostra as interligações entre o DSP *ADSP21160M* e os demais dispositivos da *Placa* em forma de diagrama de blocos.

5.3.2 ADSP21160M

No DSP *ADSP21160M* ocorre a execução do software do arranjo experimental. O DSP possui um núcleo de processamento com arquitetura de otimizada para a realização de soma de produtos, que são as operações básicas para o cálculo de produtos internos entre vetores. Por sua vez, os produtos internos entre vetores são as operações matemáticas básicas dos algoritmos de filtros digitais, tema principal dos experimentos a serem realizados na *Bancada Virtual* desse protótipo, o que justifica a escolha de um DSP para o núcleo de processamento do arranjo experimental.

Atualmente, a ampliação da capacidade de processamento, a massificação da produção e a conseqüente redução dos custos dos processadores digitais de sinais (DSP), tornou a utilização de DSPs e das técnicas de processamento digital de sinais comum e essencial em diversas áreas da eletrônica, tais como: telecomunicações, controle industrial, biomédica, áudio, instrumentação, entre outras [42].

A escolha do *ADSP21160M* se baseou na disponibilidade e no histórico de outros projetos realizados no LPS sobre as placas ADSP-21160M EZKIT Lite. O

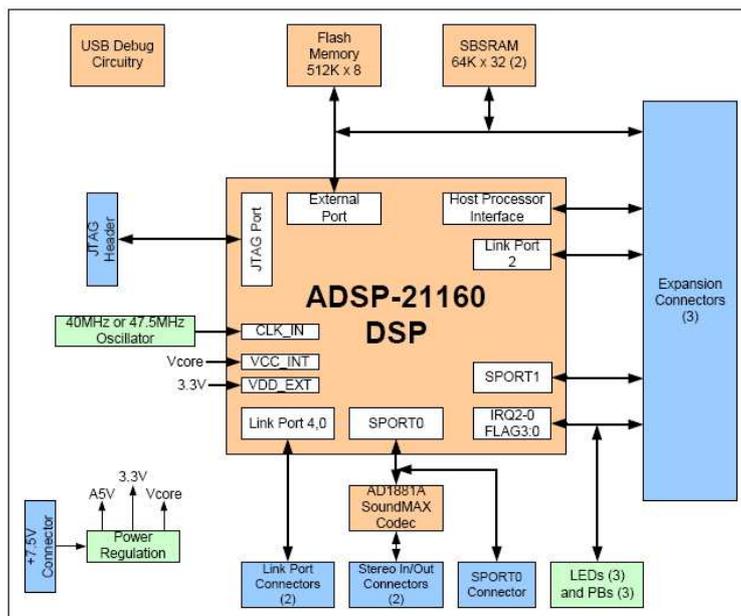


Figura 5.2:

Diagrama em blocos da placa ADSP-21160M EZKIT Lite (extraído de [40]).

ADSP-21160M é um processador digital de sinais que integra: núcleo de processamento, memórias e periféricos de entrada e saída em um único encapsulamento. A execução do algoritmo de filtros digitais ocorre no núcleo de processamento e na memória, cujas características técnicas são descritas nas próximas duas subseções.

5.3.2.1 Núcleo de Processamento

O núcleo de processamento do ADSP-21160M consiste de dois elementos de processamento. Cada um com uma unidade lógica e aritmética (ALU), um multiplicador, um deslocador e um arquivo de registro de dados. Além disso, o núcleo de processamento ainda contém um seqüenciador de programa, dois geradores de endereços, um temporizador e um *cache* de instruções. Todo o processamento digital de sinais ocorre no núcleo de processamento.

5.3.2.2 Memória Interna SRAM com Portas Duais

O ADSP-21160M possui 4 MBits de memória SRAM integrada, organizada em dois blocos de 2 MBits cada, que podem ser configurados para diferentes combinações de armazenamento de código e dados. Cada bloco de memória possui duas

portas para acesso independente, e em um único ciclo, pelo núcleo do processador e pelo processador de *I/O* ou controlador de DMA. Assim, num único ciclo, pode-se transferir dois dados para o núcleo do processador e um dado para o processador de *I/O*.

5.3.3 Programa em C++ para o DSP

O arranjo experimental do protótipo implementado é baseado no dispositivo programável ADSP21160M, que conforme o projeto do *Experimento Remoto* recebe do *Módulo Controlador de Instrumentação Remota* um programa, em código de máquina com todas as instruções e dados necessários à realização da sessão de *Bancada Virtual*.

Antes da carga do programa em código de máquina na memória do DSP, é gerado e compilado um programa em C++ no *Módulo Configurador de Experimento*. Nesta compilação é utilizado o conjunto de ferramentas do programa *VisualDSP++ 3.0* fornecido pelo próprio fabricante, *Analog Devices*, como parte do *ADSP 21160M EZKIT Lite* [41].

No exemplo, a seguir, de código C++ do programa que implementa um filtro FIR, passa-baixas com dez coeficientes em DSP, pode-se destacar como principais partes: o cabeçalho contendo os arquivos de definição e os parâmetros de experimento, a classe *Csound.h* e o programa principal, contendo o algoritmo do *Filtro Digital de Resposta Impulsiva Finita* (FIR).

No cabeçalho os “headers” *21160.h* e *def21160.h* são necessários para que a geração do código de máquina seja direcionada ao ADSP 21160M. Já os arquivos *defines.h* e *csound.h* são “headers” de definições e classes necessários ao funcionamento do CODEC. Por fim a constante *TAPS* recebe sempre o valor do número de coeficientes do filtro FIR, valores que são carregados na variável *coeffs[]*.

A Classe CODEC opera em duas etapas, na aquisição da amostra vinda do gerador de sinas para o DSP e no envio da amostra processada pelo DSP para a entrada do osciloscópio.

```
///prog.cpp
```

```
//Cabeçalho.
```

```

#include <21160.h>
asm("#include <def21160.h>");
#include "defines.h"

//Arquivo de definição da Classe csound extraído de

#include "csound.h"

//Variável que define o número de coeficientes do filtro FIR.
#define TAPS 10

float left, right;

//Coeficientes calculados no Módulo Configurador de Experimento Remoto.
float pm coeffs[10]= {1.103990E-3, 1.282880E-2, 5.845910E-2,
1.358600E-1, 1.989600E-1, 1.989600E-1, 1.358600E-1,
5.845910E-2, 1.282880E-2, 1.103990E-3};
//Programa Principal.
void main()
{ CSound codec;
float s;
float state[TAPS];
int i;
int ci=0;

for (i = 0; i < TAPS; i++) state[i] = 0;
while (true)
{ asm("idle;");

//Obtenção de amostra do sinal de entrada pelo CODEC.
codec.getSamples(left, right);
state[ci] = left;

```

```

for (i = 0, s = 0; i < TAPS; i++) {
    s += coeffs[i] * state[ci];
ci = circindex(ci, -1, TAPS);
}
ci = circindex(ci, 1, TAPS);
left = s;

//Envio da amostra processada para o CODEC.
codec.setSamples(left, left);
}
}

```

A comunicação entre o CODEC e o DSP é realizada serialmente, de forma transparente ao usuário. Existem dois modos de realizar a aquisição dos dados, via entrada de áudio padrão MIC (microfone) ou via entrada de áudio padrão *LINE*. No protótipo a entrada se deu via *LINE* [40].

5.3.4 Barramento GPIB

No protótipo, os instrumentos da *Bancada Virtual* são painéis de controle de Instrumentos Virtuais que operam remotamente os instrumentos reais, osciloscópio e gerador de sinais, do arranjo experimental. Para viabilizar esse controle, realizado pelo *Módulo Controlador de Instrumentação Remota*, foi necessário interconectar esses instrumentos ao computador servidor de *Experimento Remoto*.

Em vista dos experimentos de *Laboratório Virtual* descritos na literatura, da compatibilidade dos equipamentos disponíveis no LPS e no suporte que o programa LabVIEW oferece, o GPIB foi a interace escolhida para o controle da instrumentação do arranjo experimental.

A Interface GPIB é um barramento que foi desenvolvido no final dos anos 60 para facilitar a comunicação entre computadores e instrumentos externos. Um barramento é simplesmente o meio pelo qual o computador e os instrumentos transferem informações entre si. O GPIB possui um protocolo para gerenciar a comunicação, padronizado pelo *Instituto de Engenharia Elétrica e Eletrônica* (IEEE) em 1975, que passou a ser conhecida como o padrão IEEE 488 [43].

O padrão GPIB é um barramento digital paralelo com 24 pinos. Ele consiste de 8 linhas de dados, 5 linhas de gerenciamento, três linhas de *handshake* e oito linhas de terra. O padrão GPIB utiliza um esquema assíncrono de transferência de 8 *bits* de dados em paralelo. Ou seja, todos os *bytes* são movidos seqüencialmente através do barramento em uma velocidade determinada pelo participante mais lento da transferência. A sua principal característica é permitir o uso de um mesmo barramento simultaneamente por vários computadores e instrumentos, integrados em um único sistema de instrumentação eletrônica [44].

Os equipamentos necessários para usar o GPIB como barramento do sistema de instrumentação virtual do protótipo foram: uma placa GPIB-PCI instalada internamente ao computador servidor, um cabo GPIB e instrumentos com capacidade de interfaceamento para o padrão IEEE 488. O protótipo está implementado com barramento GPIB atuando como meio de comunicação entre o PC o gerador de sinais e o osciloscópio.

5.3.5 Osciloscópio

Para a observação dos sinais elétricos de saída do *Experimento Remoto*, foi utilizado um osciloscópio real controlado pelo *Módulo Controlador de Instrumentação Remota* via barramento GPIB com o PC.

O osciloscópio escolhido é um equipamento da marca Tektronix modelo TDS 1012 de dois canais, com banda passante de 100MHz e interface GPIB. Para a observação da saída, foi construído um cabo *jumper*, com uma das extremidades possuindo um conector padrão *BNC* e outra com conector padrão *P2*. A extremidade BNC fica conectada ao canal 1 do osciloscópio e a extremidade P2 fica conectada à saída de áudio padrão *LINE* do CODEC da placa de avaliação *ADSP 21160M EZKIT Lite*.

No início do programa o controle do osciloscópio é repassado para o *Módulo Controlador de Instrumentação Remota*. Internamente o osciloscópio se auto-ajusta em escala e base de tempo. A partir de então inicia a aquisição e envio de dados sobre a tensão da saída de áudio do DSP.

No arranjo experimental proposto, não é necessário observar o sinal na saída do gerador de sinais, por se tratar de um instrumento calibrado. Com isso, é obser-

vado apenas um canal de entrada por osciloscópio no arranjo experimental. Como o osciloscópio utilizado possui dois canais de entrada é possível a observação de até dois experimentos por osciloscópio. A figura 5.3 apresenta uma foto onde está ilustrado a disposição dos equipamentos, do PC servidor e de suas interligações no arranjo experimental.

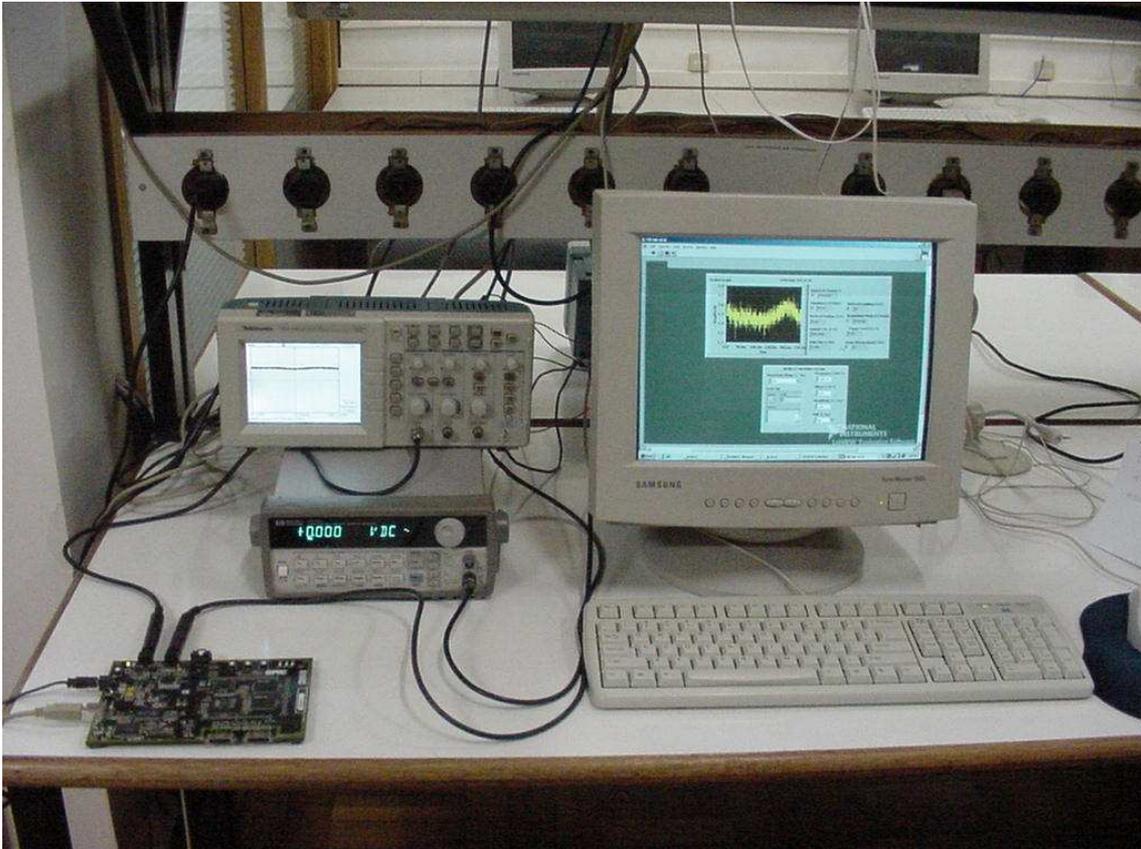


Figura 5.3: *Arranjo Experimental no LPS*

5.3.6 Gerador de Sinais

Uma vez que o DSP, núcleo do arranjo experimental, já esteja com o programa do *Experimento Remoto* em execução, para que seja verificado seu funcionamento, sinais elétricos são injetados na placa *ADSP 21160M EZKIT Lite* possibilitando a observação do comportamento dos sinais de saída e a constatação do funcionamento da rotina de processamento de sinais do DSP. Esses estímulos elétricos provêm do gerador de sinais.

O gerador utilizado é um equipamento da marca HP modelo 33120A com

interface GPIB. De forma semelhante ao que foi realizado no osciloscópio, para a interface com a placa de DSP, construiu-se um cabo *jumper*, com uma das extremidades possuindo um conector padrão *BNC* e outra com conector padrão *P2*. A extremidade *BNC* fica conectada à saída do gerador de sinais e a extremidade *P2* fica conectada à entrada de áudio padrão *LINE* da placa de DSP.

Após o início da execução da rotina de processamento de sinais no DSP, o controle do gerador de sinais é assumido pelo *Painel de Controle do Módulo Controlador de Instrumentação Remota*, a partir de então o usuário pode escolher o *perfil*, a *frequência* e a *intensidade* da forma de onda a ser aplicada na entrada da placa *ADSP 21160M EZKIT Lite*.

5.3.7 O LabVIEW

O modelo de implementação do software com estrutura modular, foi realizada em LabVIEW. A escolha do software LabVIEW 7.1 se deu, principalmente, pelo fato do LPS, que forneceu suporte ao desenvolvimento da Teia, já possuir histórico de outros projetos desenvolvidos sob a mesma plataforma e pela freqüente referência a esse pacote comercial em projetos de *Laboratórios Virtuais* citados na literatura.

Os programas em LabVIEW possuem um conjunto de conceitos que definem a sua estrutura de programação. Um programa LabVIEW é chamado de *Instrumento Virtual* (VI) devido à sua aparência e operação que imita instrumentos reais como osciloscópios e multímetros. Cada VI é formada por duas partes o *Diagrama de Blocos*, equivalente às linhas de código em uma linguagem textual e o *Painel de Controle*, equivalente à interface gráfica do usuário.

O LabVIEW é um ambiente de programação independente de plataforma, possibilitando o utilização das suas VIs em diferentes sistemas computacionais como Windows, Linux, Unix, Solaris entre outros [16]. Ele também possui recursos de comunicação com hardware para GPIB, PXI, RS-232, RS-485 e diversas placas de aquisição de dados comerciais. Essas características simplificam o desenvolvimento de aplicações multi-plataforma para o controle de instrumentação.

O LabVIEW possui uma série de facilidades que se demonstraram adequadas aos objetivos do protótipo implementado. O fato de ser uma *Linguagem de Programação Orientada a Componentes* facilita a visualização e implementação dos

Módulos.

A facilidade de interface com a instrumentação, que em muitos casos possui *device drivers* LabVIEW, desenvolvidos pelos próprios fabricantes e disponibilizados de forma gratuita, é utilizada como parte do *Painel de Controle do Módulo Controlador de Instrumentação Remota*, em que os *Instrumentos Virtuais* do osciloscópio e do gerador de sinais são *device drivers* fornecidos pelos fabricantes.

A quantidade de programas exemplos disponíveis também é outro recurso disponibilizado pelo LabVIEW. No protótipo de *Bancadas Virtuais*, foi utilizado como base para a implementação do *Módulo Configurador de Experimento* o programa exemplo *FIR Windowed Filter Desing.vi*.

Para comunicação pela Internet o LabVIEW possui recursos integrados de controle de aplicações através da Web, utilizando um Servidor Web próprio, o que permite o desenvolvimento integrado de aplicações sobre tecnologias de rede que operam com TCP/IP versão 4, o que é utilizado no protótipo para possibilitar o acesso do usuário às *Bancadas Virtuais* integradas no navegador Web.

5.3.8 O aplicativo LabVIEW

O aplicativo LabVIEW é o elemento integrador do sistema, ele agrupa o controle do arranjo experimental, as funcionalidades dos *Módulos* da arquitetura do protótipo implementado e o serviço do acesso ao cliente via Web em um único aplicativo, também chamado VI de LabVIEW.

O *Diagrama de Blocos* equivale em LabVIEW às linhas de código de uma linguagem textual, a análise dos *Diagramas de Blocos* da aplicação desenvolvida no protótipo é o objetivo desta subseção. Para simplificar o desenvolvimento, os módulos foram alocados em um único *Diagrama de Blocos*, sendo acionados de forma seqüencial, com a comunicação entre os módulos através da troca de mensagens em arquivos XML.

O código foi gerado com o pacote comercial de software LabVIEW 7.1 em *Windows 2000*. O fato de ter sido desenvolvido em *Windows 2000* não impede que o aplicativo funcione em outra plataforma de sistema operacional que possua o LabVIEW 7.1 instalado [41].

O detalhamento da implementação de cada *Módulo* está descrito nas subse-

ções a seguir. Para o melhor acompanhamento destas subseções, sugere-se que seja lido o apêndice, onde se encontra um breve tutorial sobre as estruturas básicas em programação LabVIEW.

5.3.8.1 Módulo Autenticador do Usuário

A autenticação do usuário é realizada pela comparação das informações de cadastro e senha. Após essa validação, o usuário obtém uma reserva no *Módulo Controlador de Instrumentação Remota* para o tipo de arranjo experimental associado ao seu cadastro. Como o protótipo possui um único arranjo experimental com a placa de avaliação ADSP 21160M EZKIT Lite, todas as chamadas ao experimento são alocadas no mesmo arranjo. O *Painel de Controle* do *Módulo Autenticador do Usuário* é composto de duas caixas de texto, uma para o nome e outra para a senha de usuário, e dois botões, um para confirmação e outro para desistência de início da sessão.

O *Diagrama de Blocos* da figura 5.4 apresenta o bloco autenticador do usuário. Na posição superior à esquerda da aludida figura estão os *Diagramas* dos *Controles* das caixas de texto que transferem o valor das informações de usuário e senha para o bloco de validação. Logo abaixo, o controle do botão de confirmação, que realiza uma operação lógica E com a saída do bloco de validação para chamar a execução do *Módulo Alocador de Recursos de Experimento*. Por último, o elemento mais abaixo é o controle do botão de desistência que aciona a saída da sessão.

5.3.8.2 Módulo Alocador de Recursos de Experimento

O *Módulo Alocador de Recursos de Experimento* compara as informações de cadastro e senha válidos, com as informações de arranjos experimentais disponíveis. Assim é gerada uma saída de habilitação para a reserva do arranjo experimental associado ao usuário, este *Módulo* não possui *Painel de Controle*.

O diagrama de blocos da figura 5.5, na posição superior à esquerda, contém a tabela de comparação entre cadastros de usuários e arranjos associados a esses usuários por experimento. Logo, à direita, encontra-se o controle de comparação do cadastro de usuários e a saída de habilitação.

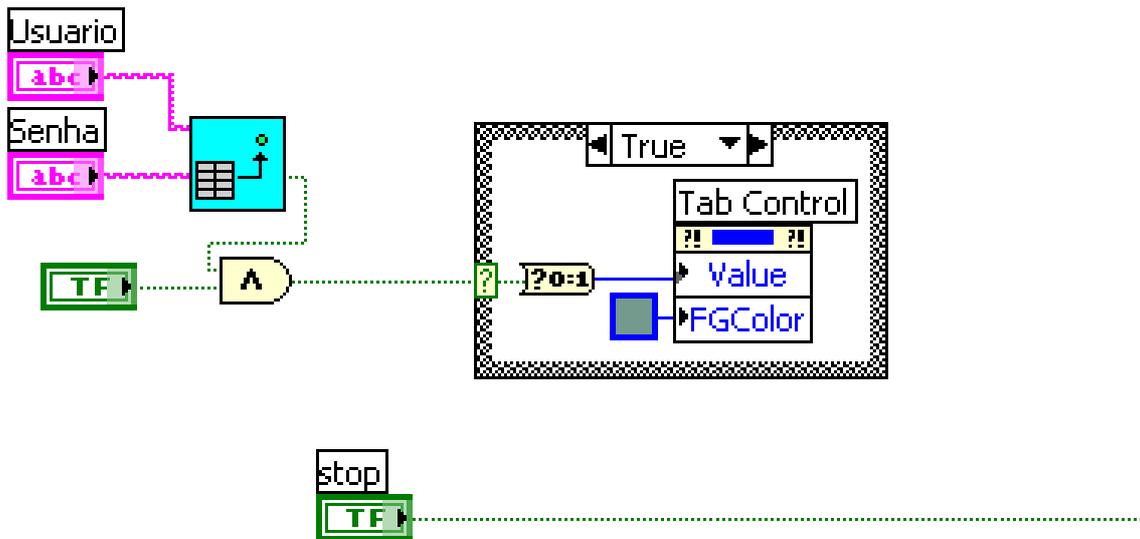


Figura 5.4: Diagrama de Blocos do Módulo Autenticador do Usuário.

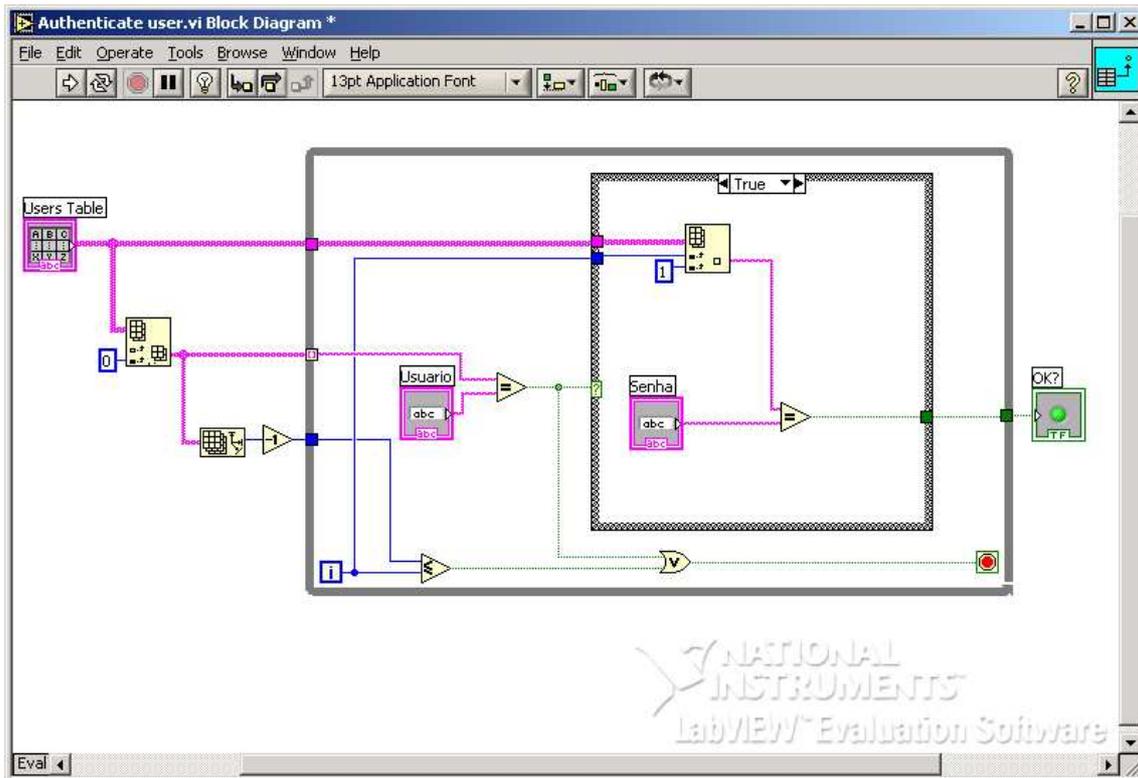


Figura 5.5: Diagrama de Blocos do Módulo Alocador de Recursos de Experimento.

5.3.8.3 Módulo Configurador de Experimento

Existe um *Módulo Configurador de Experimento* para cada tipo de *Experimento Remoto*. O *Experimento Remoto* possui seu próprio conjunto de parâmetros

de projeto a ser configurado pelo usuário. Estes parâmetros serão repassados para o *Módulo Controlador de Instrumentação Remota* por um arquivo XML.

O *Módulo Configurador de Experimento Remoto* possui um *Painel de Controle*, onde o usuário pode projetar os principais parâmetros de seu experimento e observar os gráficos teóricos de resposta em fase e resposta em frequência. Para integração com a Teia, considerou-se a realização de experimento sobre filtros digitais FIR.

O painel de controle da figura 5.6 representa a interface do usuário com o experimento de filtro digital FIR. Na lateral esquerda, de cima para baixo, encontram-se os controles para seleção das especificações típicas de um filtro FIR; ao centro dois gráficos representam as respostas em magnitude e fase do filtro projetado; a direita acima os coeficientes do filtro projetado podem ser acessados numa caixa de texto; e, logo abaixo da caixa o botão com a inscrição *SALVAR*, envia os coeficientes calculados em arquivo XML para o próximo módulo.

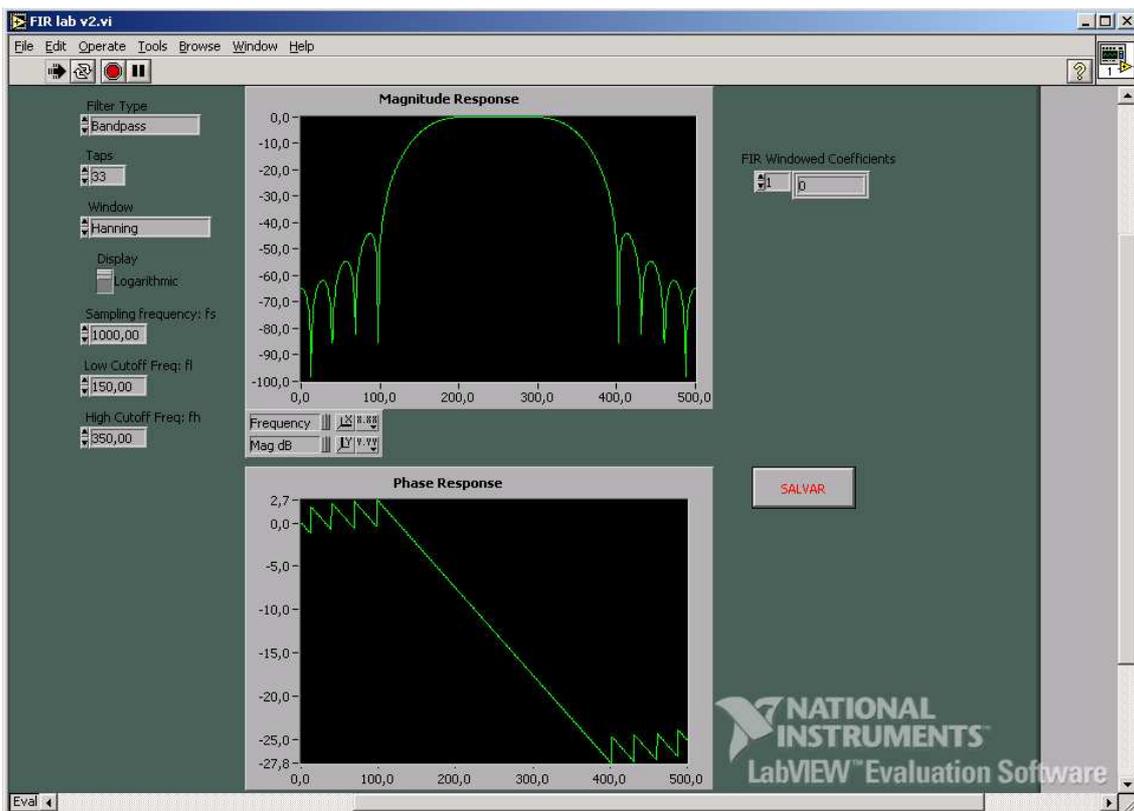


Figura 5.6: *Controle do Módulo Configurador do Experimento FIR.*

O diagrama de blocos foi implementado em dois níveis. O primeiro nível,

representado na figura 5.7, com os blocos de entrada do *Painel de Controle* à esquerda, desvia o fluxo de informações para um bloco central. Por sua vez, este realiza os cálculos necessários para a construção dos gráficos de resposta em fase e resposta em magnetude para o *Painel de Controle*, da extração dos coeficientes do filtro FIR e do controle de chamada do *Módulo Controlador de Instrumentação Remota*, representados nesta ordem, de cima para baixo, pelos blocos na parte direita da figura 5.7.

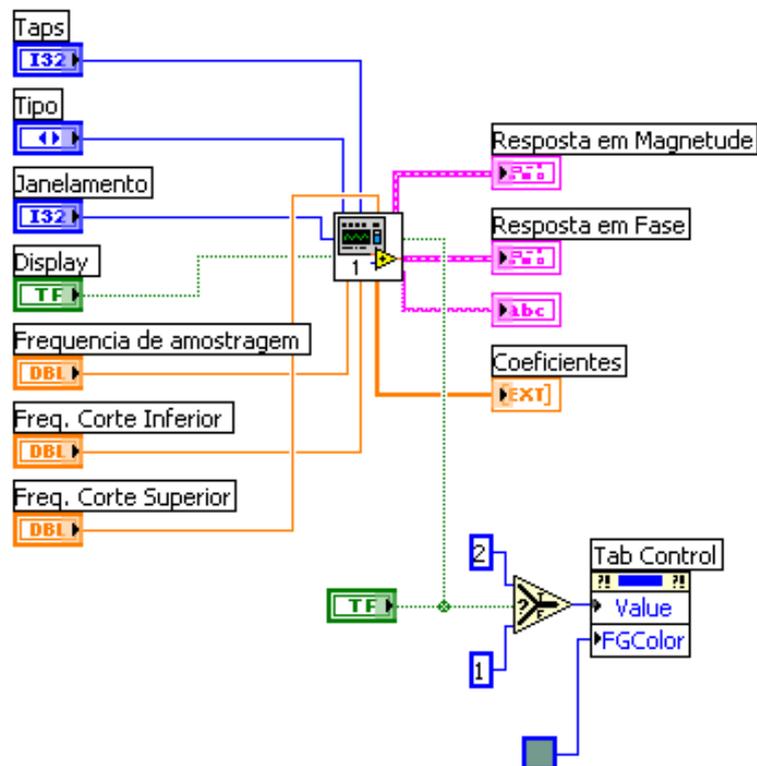


Figura 5.7: Diagrama de Blocos do Módulo Configurador de Experimento, nível 1

No segundo nível, o bloco principal do *Módulo Configurador de Experimento* realiza os cálculos dos coeficientes do filtro FIR, encaminha o fluxo de dados para o *Painel de Controle* e os coeficientes do filtro para o arquivo XML a ser carregado pelo *Módulo Controlador de Instrumentação Remota*. Na figura 5.8, os cálculos, a entrada e a saída de informações, são realizados pelos blocos da parte superior. No canto inferior direito está o bloco que cria o arquivo XML.

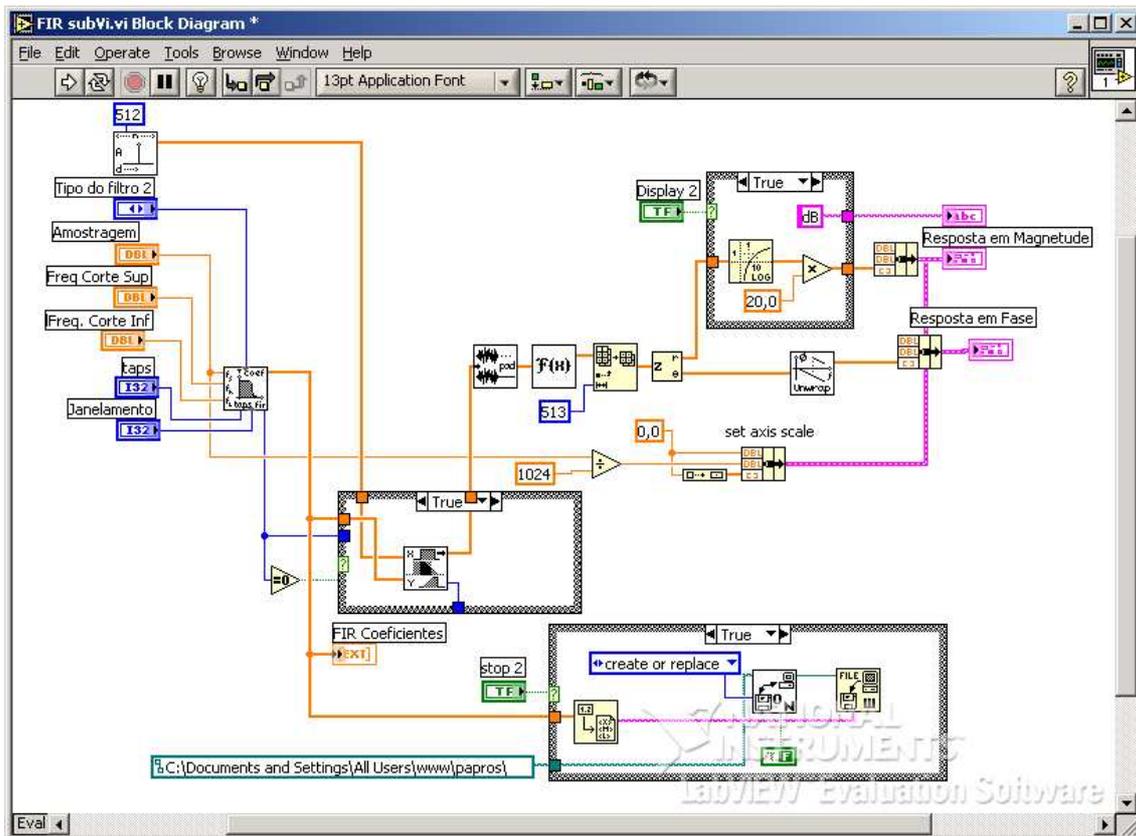


Figura 5.8: Diagrama de Blocos do Módulo Configurador de Experimento, nível 2

5.3.8.4 Módulo Controlador de Instrumentação Remota

O *Módulo Controlador da Instrumentação Remota*, inicia o processamento pela geração automática do programa em C++, seguindo as configurações escolhidas pelo usuário no projeto do experimento. Antes de ativar a interface gráfica este módulo compila e carrega o programa na memória do DSP.

Para compilar o programa, esse modulo lê os parâmetros repassados pelo arquivo XML e cria o código em C++ pela concatenação de diversos blocos de texto. Na figura 5.9 pode ser visto, na parte esquerda, o conjunto de blocos concatenadores de texto e, na parte direita, os blocos de leitura do arquivo XML e criação do programa fonte em C++.

O conjunto de blocos para criação do programa C++ compartilha uma estrutura de execução seqüencial com outros blocos responsáveis pela compilação e carregamento do programa em C++ no DSP. A figura 5.10 apresenta essa seqüência, da esquerda para a direita: a primeira estrutura compila o programa com um

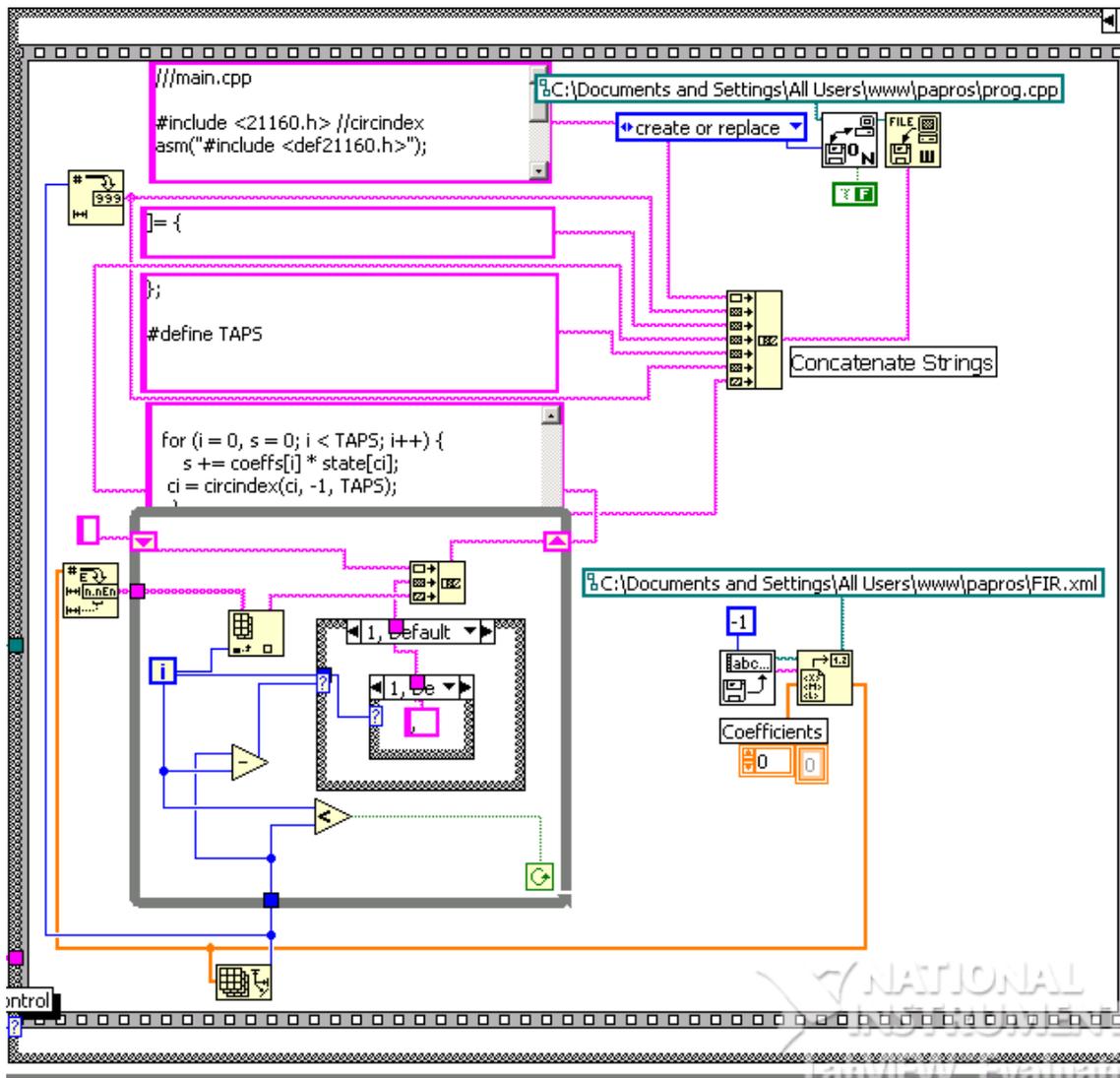


Figura 5.9: Geração de código no Módulo Controlador de Instrumentação Remota

arquivo de execução em lote chamado *compile.bat*. A segunda estrutura utiliza blocos semelhantes para realizar o carregamento e inicialização do programa no DSP. Em seguida, um bloco cria um atraso para que a instrumentação só se inicie com o programa carregado. O último bloco à direita, ativa o *Painel de Controle*.

Após o carregamento, o programa inicia o *Painel de Controle* do *Módulo Controlador de Instrumentação Remota*. Através dele o usuário controla os sinais de entrada e saída do osciloscópio e do gerador de sinais sobre o DSP. A figura 5.11 apresenta essa interface, nesta figura os controles associados ao osciloscópio estão no *Instrumento Virtual* da parte superior e os controles associados ao gerador de sinais no *Instrumento Virtual* localizado abaixo do osciloscópio.

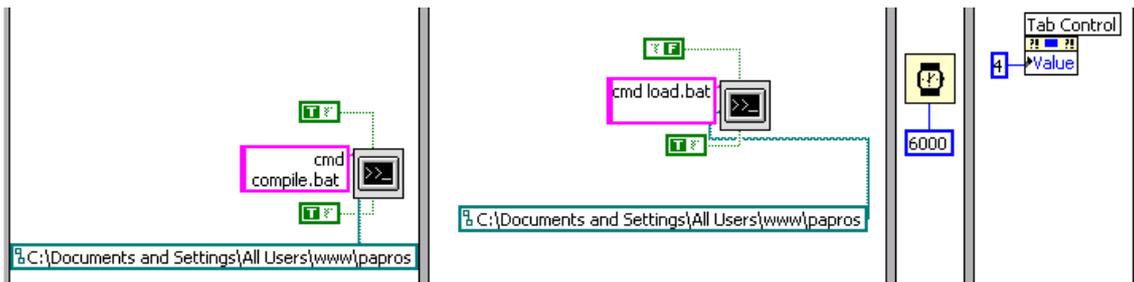


Figura 5.10: Geração de código no Módulo Controlador de Instrumentação Remota

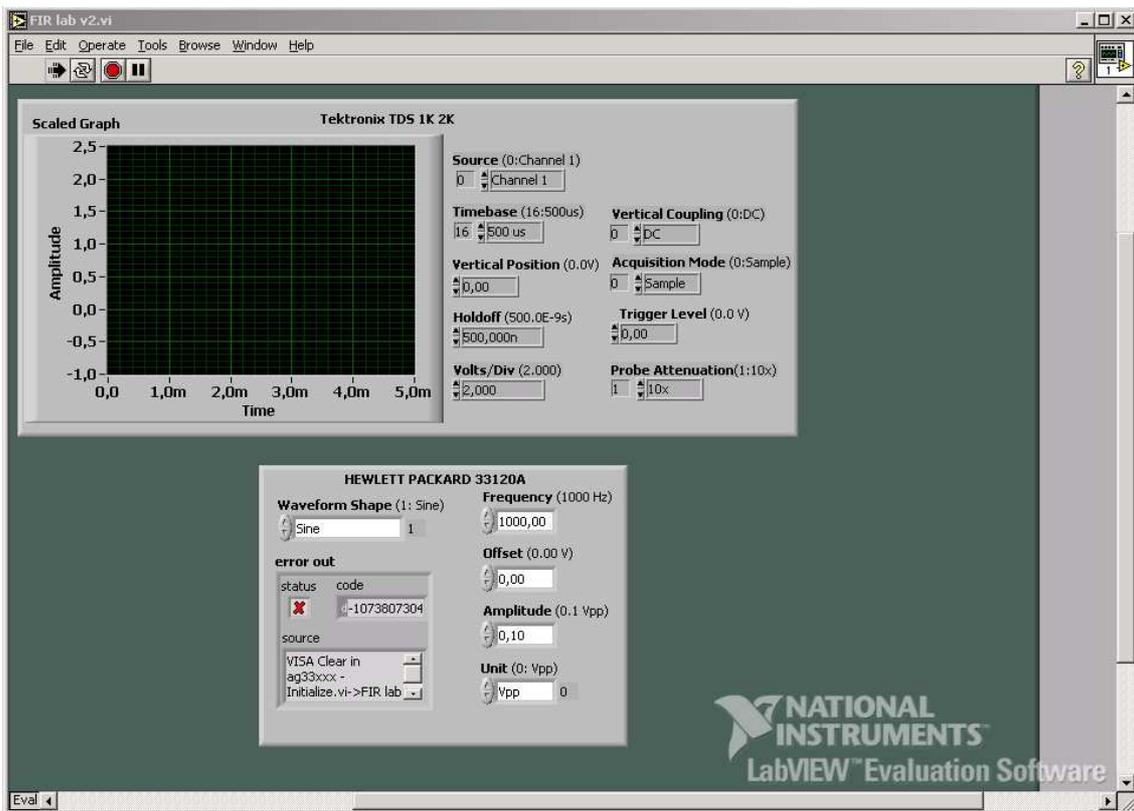


Figura 5.11: Painel de controle do Módulo Controlador de Instrumentação Remota

O diagrama de blocos desses instrumentos representa a interface entre o instrumento real e o virtual.

5.3.9 Interface com a Teia

No projeto Teia existem diversos *Laboratórios Virtuais* com conteúdo predominante em HTML. A forma escolhida para a integração das *Bancadas Virtuais* ao Teia, foi integrá-las diretamente na página HTML como uma chamada ao *Plug-in*

LabVIEW.

Para que o cliente possa utilizar o *Instrumento Virtual* a partir de seu navegador, o computador cliente de ter instalado o programa *LabVIEW RunTime*, fornecido gratuitamente pelo site da *National Instruments*, nas plataformas *Windows*, *Linux*, *Solaris* e *Mac*.

A integração se deu pela introdução do objeto *LabVIEWControl*, representado a seguir, no código HTML. No caso deste protótipo a VI chamada pelo Plug-in é a *Bancada Virtual* encontrada ao final do capítulo 8 do laboratório PAPROS. Dessa forma, o usuário após navegar pelo tutorial de filtros digitais, pode testar e experimentar o conhecimento que adquiriu sobre o tema.

Automaticamente um *Plug-in* integra-se ao navegador e, quando é acionado, se comunica com o servidor Web e inicia a sessão. As linhas abaixo foram geradas a partir do menu *Tools*, submenu *Web Publishing Tool* na VI da *Bancada Virtual*.

```
<OBJECT ID="LabVIEWControl"CLASSID="CLSID:A40B0AD4-B50E-4E58-8A1D-8544233807AB" WIDTH=880 HEIGHT=530 CODEBASE="ftp://ftp.ni.com/support/labview/runtime/windows/7.0/LVRunTimeEng.exe">
<PARAM name="LVFPPVINAME" value="FIR lab v2.vi">
<PARAM name="REQCTRL" value=false>
<EMBED SRC=".LV_FrontPanelProtocol.rpvi7"
LVFPPVINAME="FIR lab v2.vi" REQCTRL=false
TYPE="application/x-labviewrpvi7" WIDTH=880 HEIGHT=530
PLUGINSOURCE="http://www.ni.com/webappdemos/lv"></EMBED>
</OBJECT>
```

Para a reconstrução do *Painel de Controle* do módulo ativo na *Bancada Virtual*, o *LabVIEW Run Time*, utiliza uma combinação das tecnologias *ActiveX* e *.NET*, esta última baseada em *Web Services*, e reconstrói o *Painel de Controle* dentro da página HTML.

Pelo visto, o *Plug-In* do LabVIEW executa o papel do *Módulo de Aplicação Web Services*, logo não foi necessário implementá-lo pois sua funcionalidade já está integrado ao conjunto de recursos do LabVIEW.

5.4 Detalhamento da Sessão

Esta seção detalha as etapas necessárias à realização de uma sessão de *Bancada Virtual* via Web. O usuário do PAPROS, inicia sua atividade de aprendizado, navegando pelos capítulos do módulo de filtros digitais e, capítulo por capítulo, vai adquirindo conhecimento sobre o assunto, através dos tutoriais, exercícios e simulações disponíveis no ambiente de *Laboratório Virtual*.

O módulo de filtros digitais do PAPROS é composto de oito capítulos, quando o usuário chega ao capítulo 8, já possui uma base de conhecimentos suficiente para projetar e entender as especificações de um filtro digital. Então, ao fim do capítulo 8, ele encontra a *Bancada Virtual* integrada no ambiente do navegador. É neste ponto que se inicia a sessão, com a disponibilização do *Painel de Controle do Módulo Autenticador do Usuário* ativo no *navegador* do usuário.

Cada Módulo da arquitetura do protótipo é composto por uma VI e cada VI pode ser acessada, via Web, por no máximo 50 usuários simultâneos devido a limitações do próprio programa.

A quantidade de arranjos experimentais também limita o número de usuários, tendo em vista que o arranjo experimental deste protótipo não suporta mais de um usuário por sessão. Portanto, o protótipo pode trabalhar com até 50 usuários simultâneos nos módulos *Autenticador do Usuário* e *Configurador do Experimento Remoto* mas apenas com um usuário por vez no *Módulo Controlador de Instrumentação Remota*. Como forma de auxiliar a visualização das etapas, no final desta sessão é apresentado um fluxograma contendo as possíveis seqüências de operação de uma sessão de *Bancada Virtual*.

5.4.1 Visão Geral das Etapas

O diagrama da figura 5.12 representa as etapas possíveis de uma sessão, nele, o início é marcado pela requisição de autenticação por parte do usuário para a realização de um determinado experimento. Logo, o *Módulo Autenticador do Usuário* verifica na TGU se existe um cadastro, que permita a este usuário ocupar uma *Bancada Virtual* para a realização do experimento solicitado, caso não exista a *Bancada Virtual* retorna à ela inicial.

Após a verificação de existência de cadastro, o sistema consulta a tabela BVU para descobrir o do tipo de *Bancada Virtual* que o usuário pode utilizar, para o *Experimento Remoto* solicitado. Se não houver cadastro na BVU o sistema verifica na TFBV se existem *Bancadas Virtuais* compatíveis com o experimento solicitado, que estejam livres. Então, entre as opções de *Bancadas Virtuais* livres, é escolhida aquela cujo o tipo ocorra com menor frequência na rede, essa escolha é registrada na tabela BVU, para que da próxima vez que o usuário for realizar o mesmo experimento, receba o mesmo tipo de *Bancada Virtual*.

Para o caso em que já exista cadastro na BVU, o sistema procura diretamente na TFBV a *Bancada Virtual* do tipo cadastrado que esteja livre e realiza a reserva de um *Controlador de Instrumentação Remota* para o usuário, se não houver *Bancada Virtual* do tipo cadastrado na BVU livre, o sistema envia uma mensagem de erro e retorna à interface de autenticação.

Após essa etapa, a interface do *Módulo Configurador de Experimento Remoto* é iniciada e o usuário projeta os principais parâmetros do seu experimento. Em seguida a *Bancada Virtual*, disponibiliza ao usuário o acesso aos *Instrumentos Virtuais* do *Módulo Controlador de Instrumentação Remota*, onde efetivamente o usuário irá interagir com o arranjo experimental.

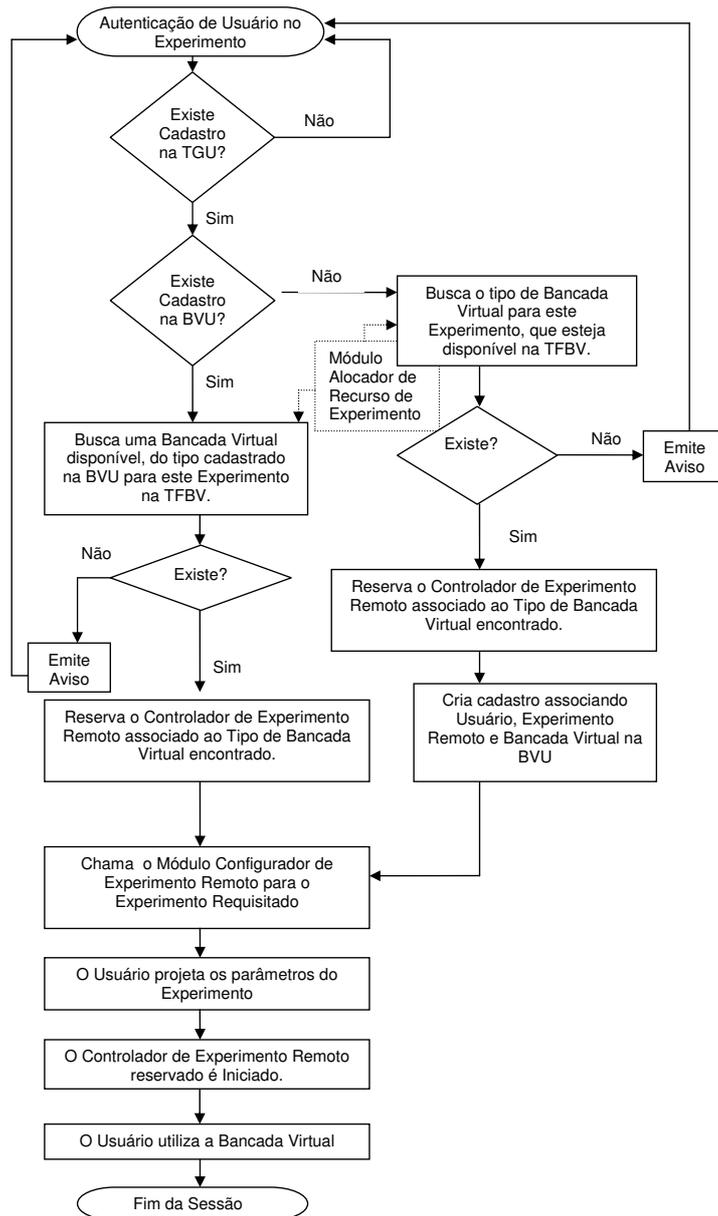


Figura 5.12: Fluxograma da Sessão de Bancada Virtual

5.4.2 Autenticação

O instrumento virtual possui duas categorias de usuário: o comum e o administrador. O usuário comum acessa o sistema para utilizar o laboratório virtual. O usuário administrador acessa o sistema para gerenciar o cadastro de usuários.

O processo de autenticação de usuário se dá por uma interface gráfica no navegador que solicita nome, senha e envia essa informação para o servidor Web em uma mensagem SOAP que será analisada pelo *Módulo Autenticador do Usuário*.

rio. Esse módulo irá comparar a informação recebida com a TGU e tomará uma determinação ação.

Caso o usuário não esteja cadastrado, o módulo retorna à interface de solicitação de nome e senha. Caso o usuário seja administrador e tenha se autenticado com a senha correta, inicia-se a interface de cadastro de usuários.

A figura 5.13 apresenta a tela de autenticação da *Bancada Virtual* integrada à página do *Laboratório Virtual* PAPROS. Nela o usuário insere o seu cadastro, senha e ao acionar o botão “OK” prosegue para a interface do *Módulo Configurador de Experimento*, caso acione o botão “STOP” ele desativa a aplicação.



Figura 5.13: *Janela de Autenticação do Usuário*

5.4.3 Configuração do Experimento

Com a autenticação o usuário tem acesso à interface de configuração do experimento. Nela ele irá definir os principais parâmetros de projeto do experimento a ser realizado. O *Módulo Configurador de Experimento* controla a interface gráfica e, após término desta etapa, repassa ao *Módulo Controlador de Instrumentação Remota* a interface com o usuário e um arquivo em XML contendo os parâmetros de

projeto solicitados pelo usuário.

Os parâmetros XML são utilizados para a geração automática do código em C, que será em seguida, compilado e carregado na memória de programa do DSP na placa de teste.

A figura 5.14 apresenta a tela do *Módulo Configurador de Experimento* da *Bancada Virtual* integrado à página do *Laboratório Virtual* PAPROS. Nela o usuário especifica os principais parâmetros de projeto do filtro FIR através da coluna de controles à esquerda das janelas dos gráficos e verifica a resposta teórica do seu projeto nas telas de *RESPOSTA EM MAGNETUDE* e *RESPOSTA EM FASE*, os coeficientes calculados podem ser acessados através do controle *COEFICIENTES DO FILTRO FIR*. Após a decisão sobre qual projeto de filtro será implementado, o usuário aciona o botão “SALVAR” e recebe a interface do *Módulo Controlador de Instrumentação Remota*, onde pode operar os *Instrumentos Virtuais* de sua *Bancada Virtual*.

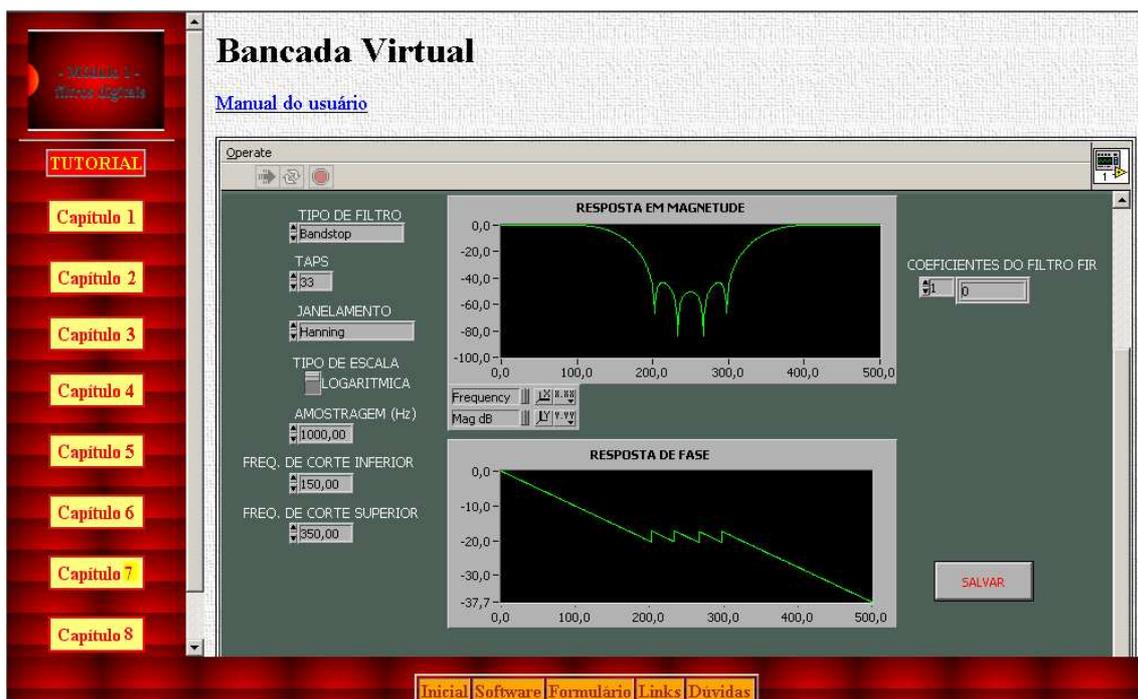


Figura 5.14: *Janela de Configuração do Experimento*

5.4.4 Manipulação dos Instrumentos Remotos

Nesta etapa o *Módulo Controlador de Instrumentação Remota*, após haver recebido o arquivo de XML com os parâmetros projeto e ter iniciado o experimento, disponibiliza uma interface gráfica, onde o usuário pode controlar o osciloscópio e o gerador de sinais, para realizar o conjunto de testes necessário ao estudo do experimento em DSP. O usuário pode observar, pelo canal 1, os sinais elétricos processados e enviados à saída da placa de DSP.

A figura 5.15 apresenta os instrumentos virtuais da *Bancada Virtual*, integrada à página do *Laboratório Virtual PAPROS*. Nela o usuário pode acessar as interfaces dos *drivers* de *Instrumento Virtual* fornecidos pelos fabricantes para o controle do osciloscópio, na parte superior da tela, e do gerador de sinais logo abaixo no arranjo experimental.

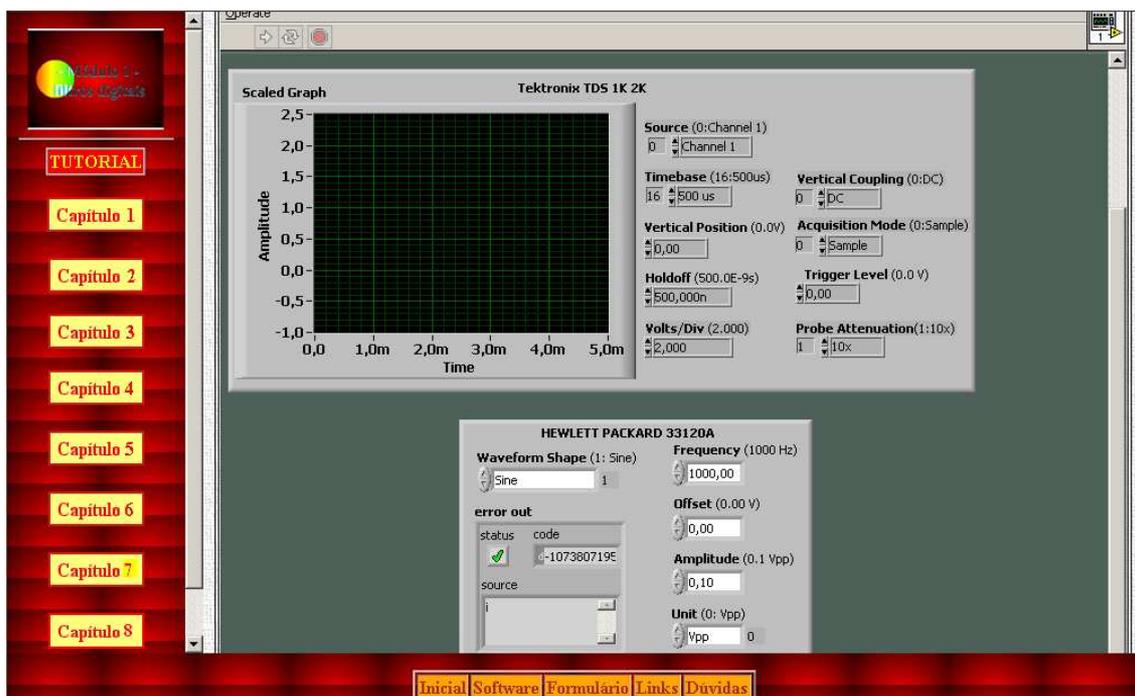


Figura 5.15: *Osciloscópio e Gerador de Sinais da Bancada Virtual*

5.4.5 Encerramento da Sessão

A sessão é encerrada quando o usuário seleciona o botão de parar, em formato de um círculo vermelho, na barra superior de controles da *Bancada Virtual*. Com

isso sua interface via navegador volta para a tela de autenticação e o *Instrumento Virtual* associado ao *Experimento Remoto* que estava ocupado, fica disponível para ser novamente disputado entre outros usuários. O aplicativo permite que seja configurado um tempo máximo de sessão, evitando, assim, que algum usuário monopolize os recursos de uma dada *Bancada Virtual*.

5.5 Considerações Finais

O presente capítulo abordou o detalhamento do sistema protótipo de *Bancada Virtual* com compartilhamento de recursos baseado no modelo conceitual apresentado no Capítulo 4. Foi apresentado o diagrama da arquitetura dos *Módulos* e enumerada correspondência entre a função de cada *Módulo* e o requisito do modelo teórico de *Bancada Virtual* atendido.

Sob o prisma de atendimento aos requisitos de software do modelo teórico, pode-se constatar que o protótipo atendeu a todos os requisitos, mas atendeu parcialmente à necessidade de gerenciamento de usuários em diferentes domínios administrativos e da busca e localização de recursos em diferentes domínios administrativos de rede. Isso porque, para atuar com diferentes domínios administrativos é necessário que esteja implementada uma infra-estrutura Grid de Serviços.

A arquitetura proposta para o protótipo não incorporou os serviços de compartilhamento de recursos da Grid, previstos no modelo conceitual, mas tornou possível avaliar a eficácia das *Bancadas Virtuais*, quanto à integração com o Laboratório Virtual PAPROS, ao acesso e controle via Web do *Experimento Remoto*, a busca e alocação de recursos e quanto a arquitetura baseada em *Web Services* com troca de mensagens XML.

Os elementos do sistema foram detalhados desde o arranjo experimental, até as estruturas básicas de programação LabVIEW. Com a descrição do funcionamento de cada *Módulo* e suas interações com os demais elementos do sistema.

Finalmente, a sessão foi vista sob o ponto de vista do usuário, com o detalhamento de toda a seqüência de passos necessária à realização de uma *sessão* de *Experimento Remoto* através da *Bancada Virtual*, implementada por este protótipo.

Capítulo 6

Conclusões

Este trabalho originou-se das necessidades e oportunidades advindas da rápida evolução tecnológica que, ao mesmo tempo, impõem a necessidade de constante atualização aos profissionais da área de tecnologia e oferece novas possibilidades no campo da educação.

Entre as novas tecnologias associadas ao ensino, a Web tem exercido papel importante no contexto da educação à distância. Neste contexto, o uso de *Laboratórios Virtuais*, acessíveis via Web, assume grande relevância no apoio de cursos e pesquisas em engenharia e ciências exatas.

Os *Laboratórios Virtuais* constituem uma importante ferramenta de apoio para cursos presenciais, onde oferecem a possibilidade de uma utilização bastante eficiente dos recursos de instrumentação. Adicionalmente, para os ambientes educacionais baseados em EAD, onde a presença no laboratório real possa ser inviável ou difícil, oferecem flexibilidade e uso ampliado, independente do local e sem a necessidade de adesão ao horário regular de trabalho nas instituições provedoras. Além disso, o apoio à pesquisa, na forma de conteúdo informativo para alunos e membros aderentes de equipes de pesquisadores, pode agilizar a inserção dos componentes e a absorção de novas linhas de pesquisa.

Os estudos realizados durante o desenvolvimento deste trabalho, indicam que o problema do compartilhamento de recursos entre *Laboratórios Virtuais* ainda encontra-se em aberto. Em particular, o problema do compartilhamento de *Bancadas Virtuais* foi centro do modelo conceitual de solução, desenvolvida neste trabalho.

Os requisitos do modelo conceitual abrangeram a necessidade já verificada nos

Laboratórios Virtuais constituente da Teia de laboratórios em engenharia elétrica, que apresenta uma arquitetura inter-disciplinar e multi-institucional, de se compartilhar recursos entre as diferentes instituições membros do *Grid de Informação*, através de um *Grid de Serviços*. Este que seria capaz de oferecer o gerenciamento de usuários e os recursos de *Bancada Virtual* como forma de serviços de rede acessíveis para todos os domínios administrativos da Grid.

A computação Grid serve de referência ao modelo de compartilhamento de *Laboratórios Virtuais*. Ela possibilita o gerenciamento e controle de recursos e usuários entre diferentes domínios administrativos de rede, além da busca e oferta de recursos de forma descentralizada e a autenticação segura dos usuários. Tudo isso em um ambiente onde os recursos variam, desde capacidade de armazenamento, ciclos de processamento até o próprio arranjo experimental, conectado ao servidor de *Experimento Remoto* via placa GPIB.

Após esses estudos, foi possível inferir a arquitetura dos componentes necessários à implementação de um *Laboratório Virtual* capaz de oferecer a autenticação segura, um sistema de busca, gerenciamento e compartilhamento de recursos e de usuários entre diversas instituições de diferentes domínios administrativos, a viabilização de experimentos com sistemas digitais programáveis, neste caso o DSP, e possibilidade de integração com outros sistemas de EAD existentes.

Destacam-se nesta proposta em relação aos projetos pesquisados na literatura: a possibilidade de autenticação segura e gerenciamento distribuído dos usuários dos diferentes domínios administrativos de rede; o compartilhamento de recursos de *Bancada Virtual* entre instituições de diferentes domínios administrativos; a viabilização de um arranjo experimental para o processamento de sinais elétricos reais com dispositivo programável (DSP) e possibilidade de integração da *Bancada Virtual* via página HTML sem a necessidade de instalação de algum aplicativo específico para determinado *Experimento Remoto*.

Para a validação desse modelo, foi construído um protótipo de *Bancada Virtual*, capaz de permitir a realização de *Experimentos Remotos* via Web, e de fácil integração a outros ambientes EAD baseados em HTTP. O protótipo foi integrado à Teia através do laboratório PAPROS, voltado para a área de processamento de sinais. Essa integração atualizou a condição anterior do PAPROS, que permitia ape-

nas a emulação de um filtro em DSP, para possibilitar experimentos com processamento de sinais reais. O protótipo foi desenvolvido sobre uma arquitetura extensível ao atendimento de outros conteúdos, apenas com a atualização de software, devido à versatilidade da plataforma de hardware programável em DSP. Através de carga de software no DSP o arranjo experimental sugerido pode assumir diversos papéis e servir de base para outros tipos de *Experimento Remoto*.

6.1 Trabalhos Futuros

Como proposta fica a implementação do laboratório num ambiente Grid multi-institucional, onde seja possível atender aos requisitos de autenticação, gerência e busca de recursos não abrangidos pelo atual protótipo.

Outra proposta é a disponibilização desses laboratórios para uma turma de formação técnica do ensino médio, oferecendo acesso aos recursos que muitas escolas não tem condições de adquirir. Mas, ao mesmo tempo, algumas escolas poderiam contribuir para a Teia, com o compartilhamento de experimentos de microcontroladores e laboratórios associados ao ensino desses dispositivos.

Também, como proposta, segue a necessidade de atualizar a plataforma de DSPs para uma configuração que permita o multiprocessamento e a criação de novos instrumentos virtuais associados a experimentos remotos nos laboratórios PALAS e PAPROS. Existe a necessidade de contemplar todos os módulos do PALAS com uma bancada virtual.

Dentro dos objetivos didáticos das bancadas virtuais, também são esperados aplicativos para o atendimento de usuários com necessidades especiais. Estes aplicativos incluiriam a possibilidade de navegação nas páginas da Teia e controle dos instrumentos virtuais, através de comandos de voz. Os resultados dos experimentos também poderiam ser enviados aos usuários por meio de síntese de voz.

Bibliografia

- [1] T, H., M, X., *Overview and Classification of Web-Based Education (Systems, Tools and Practices)*, Report, Hellenic Open University, 2005.
- [2] VALLE, R., *Educação do Trabalhador para Além dos Consensos Fáceis*. Editora Mauad, 2003.
- [3] SEIXAS, J. M. D., BITTENCOURT, M. C., JANDRE, F. C., “A Virtual Laboratory for Teaching and Develop Applications on Digital Filtering”, *Proceedings of the VII International Conference on Engineering and Technology Education*, 2001.
- [4] SEIXAS, J. M. D., MAIDANTCHIK, C., “Teia de interconexão de laboratórios virtuais segundo uma concepção multidisciplinar e colaborativa”, *Cobenge*, pp. 52–58, agosto 2001.
- [5] ANDO, B., GIANNONE, P., PECORA, E., *et al.*, “Development in Web-based Laboratory Sessions”, *Instrumentation and Measurement Technology conference IMTC2004*, maio 2004.
- [6] VARELLA, W. A., RAMIREZ-FERNANDEZ, F. J., *Laboratório Virtual utilizando sistema operacional aberto*, Report, Escola Politécnica - USP, 2005.
- [7] AMBIKAI RAJAH, E., EPPS, J., SHENG, M., *et al.*, “Evaluation of a Virtual Teaching Laboratory for Signal Processing Education”, *Proceedings of the IEEE Instrumentation and Measurement Technology conference ICASSP 2003*, abril 2003.
- [8] GUSTAVISSON, I., “Laboratory experiments in distance learning”, *International Conference on Engineering Education*, pp. 6B1–1 6B1–5, agosto 2001.

- [9] TYML, P., SLEPICKA, D., ROZTOCIL, J., “Virtual Measurement System for Distance Education and Training”, *Instrumentation and Measurement Technology conference IMTC2004*, maio 2004.
- [10] FERREIRA, A. B. D. H., *Minidicionário da Língua Portuguesa*. Editora Nova Fronteira, 1993.
- [11] RADENKOVIC, B., STEFANOVIC, D., MALISIC, A. K., *et al.*, “An Application of Distance Learning as Support for Traditional Education at the University”, *Proceedings of 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, TELSIKS 2003*, outubro 2003.
- [12] OLIVER, R., OMARI, A., HERRINGTON, J., “Investigating Implementation Strategies for WWW-Based Learning Environments”, *International Journal of Instructional Media*, pp. 121–128, Fevereiro 1998.
- [13] RIO, J. D., TRULLOLS, E., MANUEL, A., *et al.*, “ELVIS. A New Tool for Teaching and Training”, *Instrumentation and Measurement Technology conference IMTC2004*, maio 2004.
- [14] NEDIC, Z., MACHOTKA, J., NAFALSKI, A., “Remote Laboratories Versus Virtual and Real Laboratories”, *33rd ASEE/IEEE Frontiers in education Conference*, novembro 2003.
- [15] “Projeto EAD do Blekinge Institute of Technology BTH na Suécia”, <http://www.its.bht.se/courses/skt106/distanslabbar/english/>, último acesso em 20 junho de 2005.
- [16] “National Instruments”, www.ni.com, último acesso em 20 de junho de 2005.
- [17] MARCHEZAN, A. R., CHELLA, M. T., FERREIRA, E. C., “Laboratório Remoto Aplicado ao Ensino de Engenharia Eletrônica”, *I Workshop de Ciências da Computação e Sistemas da Informação da Região Sul*, maio 2004.
- [18] CHELLA, M. T., FERREIRA, E. C., “Ferramenta para o desenvolvimento de experimentos remotos com aplicações educacionais”, *VII Congresso Iberoamericano de Informática Educativa*, setembro 2004.

- [19] “The Internet Engineering Task Force”, www.ietf.org, último acesso em 20 de junho de 2005.
- [20] BAGNASCO, A., SCAPOLLA, A. M., “A Grid of Remote Laboratory for Teaching Electronics”, *2nd International LeGE-WG Workshop on e-Learning and Grid Technologies : a fundamental challenge for Europe*, junho 2003.
- [21] “Laboratório de Processamento de Sinais - LPS”, www.lps.ufrj.br, último acesso em 20 de junho de 2005.
- [22] “Páginas de Apoio ao Laboratório de Sistemas Lineares”, www.lps.ufrj.br/palas, último acesso em 20 de junho de 2005.
- [23] SEIXAS, J. M. D., JANDRE, F. C., “PALAS An Integrated WWW Based Laboratory for Supporting the Teaching of Linear Systems”, *International Conference on Engineerig Education*, agosto 1998.
- [24] “Páginas de Apoio em Processamento de Sinais”, www.lps.ufrj.br/papros, último acesso em 20 de junho de 2005.
- [25] SEIXAS, J. M. D., MAIDANTCHIK, C., ABRAHAO, G. C. R., *et al.*, “Teaching and Learning Signal Processing Techniques Using Web Technology”, *Proceedings of the Ibero-American Summit on Engineering Education*, março 2003.
- [26] BITENCOURT, M. C., *Desenvolvimento de um ambiente de ensino de filtros digitais na WEB usando processadores digitais de sinais (DSPs)*, Projeto final, UFRJ, dezembro 2000.
- [27] BUYYA, R., *Economic-based Distributed Resource Management and Scheduling for Grid Computing*. Tese de doutorado, Monash University, abril 2002.
- [28] REINEFELD, A., SCHINTKE, F., “Concepts and Technologies for a Worldwide Grid Infrastructure”, *Proceedings of the 8th International Euro-Par Conference on Parallel Processing - Euro-Par 02*, pp. 62–72, agosto 2002.
- [29] FOSTER, I., “The Grid: A New Infrastructure for 21st Century Science”, *Physics Today*, fevereiro 2002.

- [30] MILEY, M., “The Grid Bringing Computing Power to the Masses”, *Oracle Magazine*, v. 17, n. 5, pp. 39–44, 2003.
- [31] FOSTER, I., KESSELMAN, C., NICK, J., *et al.*, “Grid Services for Distributed System Integration”, *IEEE Computer Magazine*, junho 2002.
- [32] B, S. D., “Motivating Computational Grids”, *In Proceedings of the 2nd IEEE/ACM International Symposium on cluster Computing and the Grid CC-Grid 2002*, pp. 401–406, maio 2002.
- [33] FOSTER, I., KESSELMAN, C., *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.
- [34] NASH, M., “Infrastruture for Grid Computing”, *Oracle 10g*, , 2004.
- [35] FOSTER, I., KESSELMAN, C., TUECKE, S., *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, Report, Globus C., 2005.
- [36] “Globus Aliance”, www.globus.org, último acesso em 20 de junho de 2005.
- [37] BASIURA, R., BATONGACAL, M., BOHLING, B., *et al.*, *Professional ASP.NET Web Services*. Makron Books, 2003.
- [38] POTTS, S., KOPACK, M., *Sams Teach Yourself Web Services in 24 Hours*. Sams, 2003.
- [39] ROBINS, J., *Introduction to Web Services Architecture*, Report, Systinet Corp., 2002.
- [40] ANALOG DEVICES, *ADSP-21160 EZ-KIT Lite: Evaluation System Manual*. 3 ed., janeiro 2003.
- [41] “Analog Devices”, www.analog.com, último acesso em 20 de junho de 2005.
- [42] OPPENHEIM, A. V., SCHAFER, R. W., *Discrete-Time Signal Processing*. Prentice Hall, 1989.
- [43] CHICKAMENAHALLI, S. A., HALL, A., “Interfacing a digital oscilloscope to a personal computer using GPIB”, *Frontiers in Education Conference, 1997. 27th*

Annual Conference. 'Teaching and Learning in an Era of Change'. Proceedings.,
v. 2, pp. 904, novembre 1997.

- [44] CHEIJ, D., "Instrument connectivity through mixed I/O test systems", *AU-TOTESTCON 2003. IEEE Systems Readiness Technology Conference. Proceedings*, pp. 502– 504, settembre 2003.

Apêndice A

Noções Básicas de Programação

LabVIEW

Este apêndice resume os principais componentes e estruturas de um VI.

A.1 Componentes do Instrumento Virtual no LabVIEW

Em LabVIEW a interface do usuário chama-se *Painel de Controle*, ele é construído pelo usuário a partir de vários controles e indicadores disponíveis no LabVIEW. Na figura A.1 está um exemplo de *Painel de Controle* LabVIEW, sobre o painel existe uma janela de controles que é um menu de onde são retirados os controles e indicadores para a criação do *Painel de Controle*.

Este *Painel de Controle* é um VI que realiza a conversão de graus Celsius para graus Fahrenheit, possui um controle sob o título *degC*, e um indicador de resposta sob o título *degF*. Na parte superior da *Janela* estão os menus principais : *File*, *Edit*, etc. Logo abaixo do menu principal estão os menus de execução e formatação, na seguinte ordem: os quatro primeiros botões são responsáveis, respectivamente para acionar o programa, acionar recursivamente o programa, parar o programa e pausar o programa. Em seguida encontram-se algumas barras de formatação e edição do formato dos controles, seguidas pelo botão indicador de “Help”.

No LabVIEW, código é adicionado ao VI através de objetos gráficos representando as funções que irão interagir com os objetos do *Painel de Controle*. O

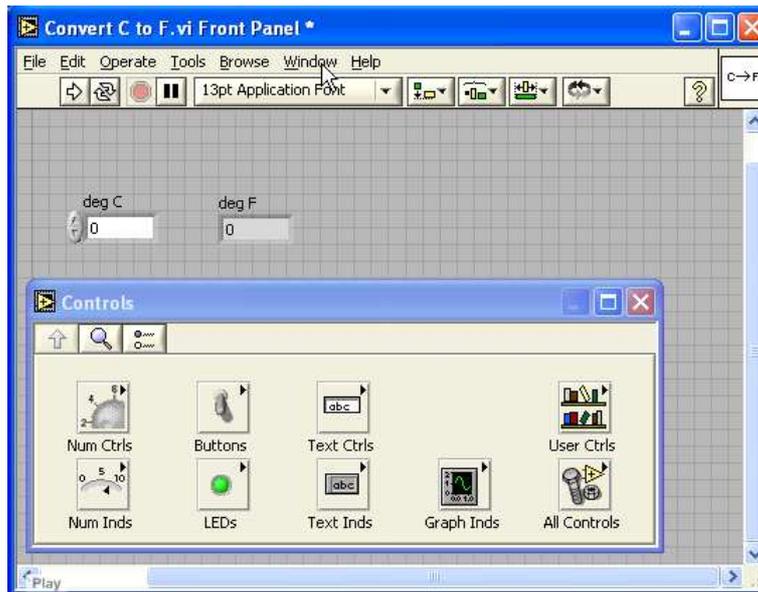


Figura A.1: *Exemplo de Painel de Controle do LabVIEW (extraído de [16])*

Diagrama de Blocos equivale ao código fonte da VI, enquanto o *Painel de Controle* equivale à interface gráfica com o usuário. Na figura A.2 encontra-se um diagrama de blocos típico, no qual os controles e indicadores são representados pelos blocos com contorno mais grosso. Sobre o diagrama está a janela de controle com os menus de elementos de controle e fluxo de dados.

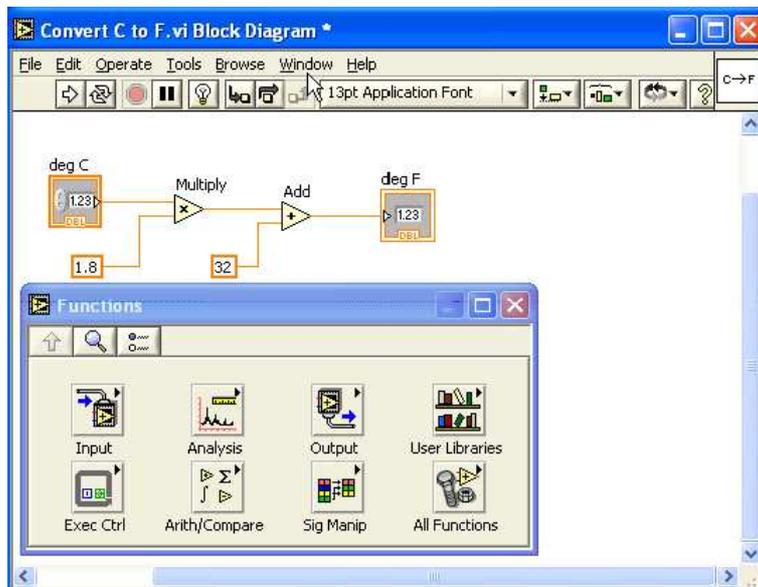


Figura A.2: *Exemplo de Diagrama de Blocos do LabVIEW (extraído de [16])*

Em resumo, uma VI consiste em três partes principais:

- o *Painel de Controle*, que é a interface gráfica do VI, criado a partir dos controles existentes no menu de controle e em seus submenus, com controles que simulam as entradas e saídas de instrumentos reais, e fornecem dados para o diagrama de blocos da VI;
- os *Indicadores*, que simulam a saída de instrumentos reais e mostram os dados de saída dos diagrama de blocos para o usuário;
- o *Diagrama de Blocos*, que é composto por terminais que permitem a troca de dados com o *Painel de Controle*, os diagramas dos objetos do *Painel de Controle* podem ser reconhecidos como os elementos de borda mais espessa, os *nós*, que possuem entradas e saídas, são os elementos que processam a funcionalidade das VIs, e os "*wires*" interligam os elementos do diagrama de blocos.

A.2 Estruturas de Controle de Fluxo do Programa

Além de funções em *Diagrama de Blocos*, o LabVIEW possui uma série de estruturas para o controle do fluxo do programa, as quais foram utilizadas neste trabalho, e estão detalhadas a seguir.

Estrutura *While Repeat*, equivalente a *Do Loop* e *Repeat Until* Esta estrutura executa um subdiagrama até a condição de controle ser atendida. Na figura A.3 existe um exemplo da estrutura *While*, em que o ciclo de comparações se mantém enquanto a condição de comparação da temperatura, oriunda da Sub VI *termometer.vi*, for falsa. Na parte inferior do diagrama o ícone circular à direita indica a condição de controle e o ícone "i" a esquerda fornece o número de interações já realizadas.

Estrutura *For* A estrutura *For*, executa um sub-diagrama, uma quantidade predeterminada de vezes. Na figura A.4 a cada ciclo, um valor decimal aleatório está sendo enviado a um gráfico no *Painel de Controle* pelo bloco semelhante a um par de dados. O terminal de interação "i" é um terminal de saída e indica o número

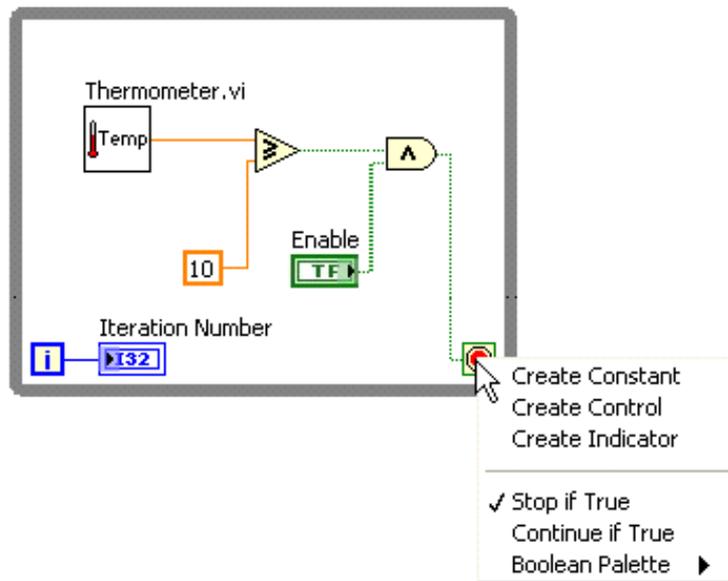


Figura A.3: *Exemplo da estrutura While Repeat do LabVIEW (extraído de [16])*

de interações já executadas iniciando na interação 0 (zero), até a interação “N”, que neste caso é 100 (cem).

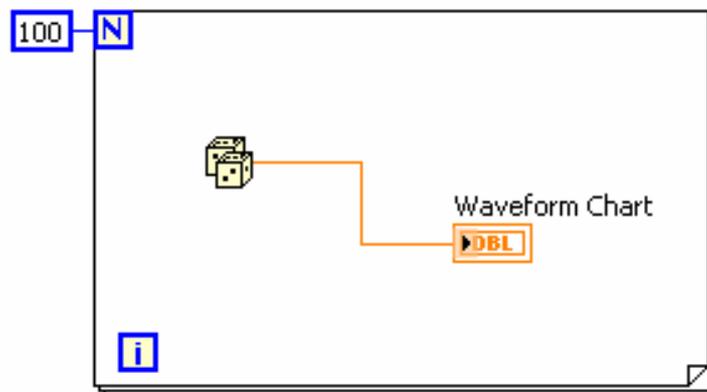


Figura A.4: *Exemplo da estrutura For do LabVIEW (extraído de [16])*

Estrutura Case A estrutura Case foi amplamente utilizada neste trabalho, ela equivale à um comando *Case* ou *If, Then, Else* nas linguagens de programação baseadas em texto. A estrutura *Case* possui dois ou mais subdiagramas ou “cases”, apenas um subdiagrama é visível por vez na VI e a estrutura executa apenas um “case” por vez. Na figura A.5 há um exemplo da estrutura case para o cálculo da

raiz quadrada de um número com a emissão de mensagem de erro no caso de uma entrada negativa. A estrutura case por padrão apresenta os modos *True* e *False*.

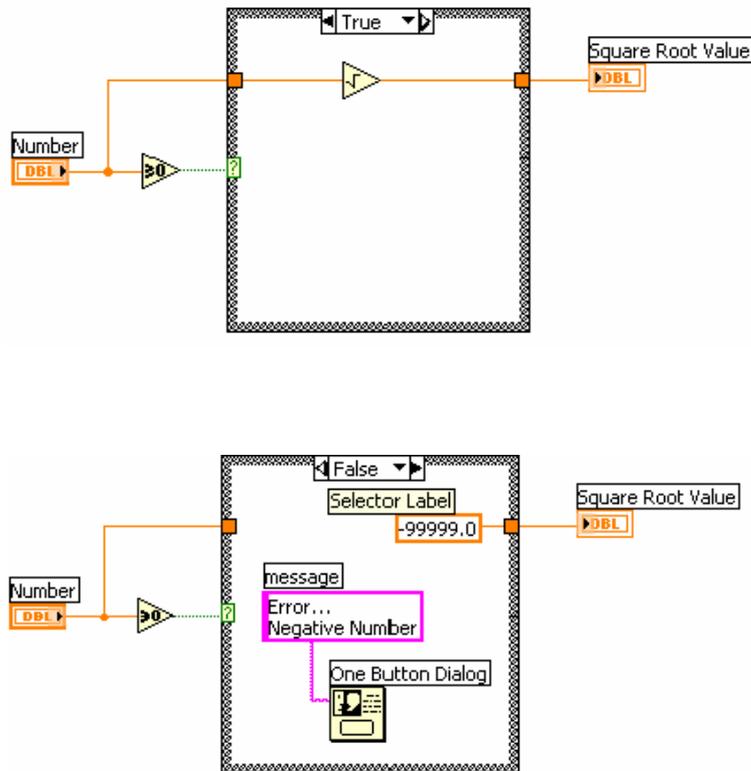


Figura A.5: Exemplo da estrutura Casado LabVIEW (extraído de [16])

Estrutura Sequence A estrutura de *Sequence* possui um ou mais subdiagramas chamados *Frames* ou *Quadros*, que executam os blocos internos aos *Frames* em ordem seqüencial. Na figura A.6 existe uma ilustração sobre a ordem de execução de uma estrutura de três *Frames* com a seta indicando a ordem de execução temporal do *Frame* 0(zero) ao *Frame* (2), um após o outro.

O programa LabVIEW suporta a utilização de outras VIs, sub VIs, dentro do *Diagrama de Blocos* principal de forma semelhante às subrotinas em linguagens textuais, o que permite a simplificação dos *Diagramas de Blocos*, do gerenciamento e da manutenção do código.

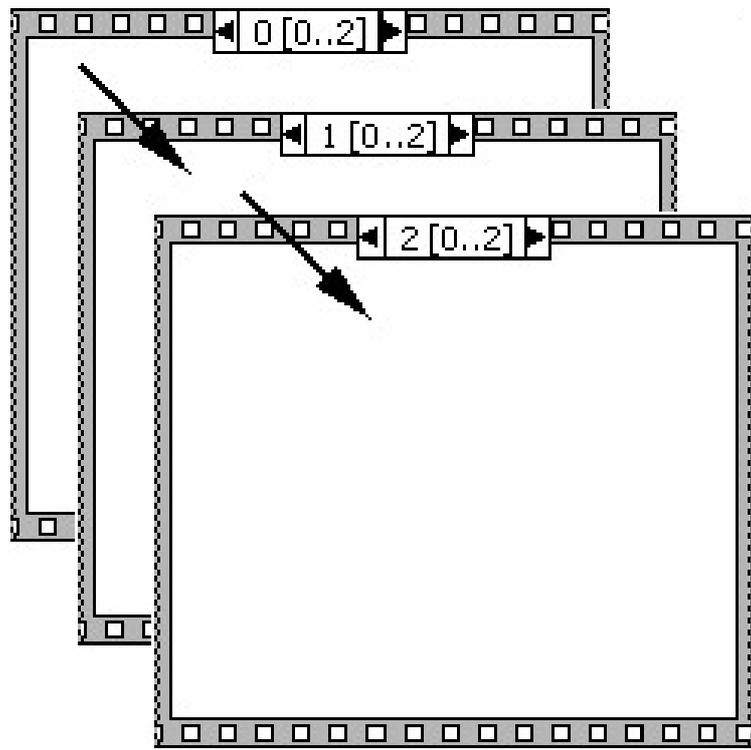


Figura A.6: *Exemplo da estrutura Sequencedo LabVIEW (extraído de [16])*