

Felipe Rinaldi de Campos

Gabriel Prata Bakiewicz

Tiago Cabral Padovani

Winston Quirino França

Weblab: aprendizado e controle à distância

Monografia apresentada à Escola Politécnica da Universidade de São Paulo (EPUSP) como parte dos requisitos para a obtenção do Título de Engenheiro Eletricista no Departamento de Engenharia de Computação e Sistemas Digitais (PCS).

Felipe Rinaldi de Campos

Gabriel Prata Bakiewicz

Tiago Cabral Padovani

Winston Quirino França

Weblab: aprendizado e controle à distância

Monografia apresentada à Escola Politécnica da Universidade de São Paulo (EPUSP) como parte dos requisitos para a obtenção do Título de Engenheiro Eletricista no Departamento de Engenharia de Computação e Sistemas Digitais (PCS).

Área de concentração: Educação à distância

Orientador:

Prof. Dr. Carlos Eduardo Cugnasca

Resumo

BAKIEWICZ, G. P.; CAMPOS, F. R.; FRANÇA, W. Q.; PADOVANI, T. C. **Weblab: aprendizado e controle à distância**. 2007. 106 f. Monografia - Escola Politécnica da Universidade de São Paulo, São Paulo, 2007.

Este projeto de formatura tem como objetivo a criação de um sistema web que permita a disponibilização de ambientes remotos diversos. A comunicação entre os pontos finais do sistema exige uma solução distribuída, com interoperabilidade entre módulos e uso de protocolos distintos. Uma métrica de desempenho crítica ao sistema é o tempo de resposta, já que deve haver suporte à aplicações que exijam compromisso com o tempo (sistema de tempo real). O sistema é composto por três módulos interconectados: aplicação web, aplicação interna (camada do meio) e hardware. A primeira é responsável por capturar os comandos do usuário, enviá-los à aplicação interna e apresentar os dados resultantes, quando houver. A camada do meio apenas encaminha os dados para o dispositivo indicado através de um protocolo definido para a aplicação. Os dados são então repassados ao hardware, que opera como interface entre o sistema e o ambiente de operação. Este nível se faz necessário apenas para acionar dispositivos como *jumpers* e relés, que demandam estímulos elétricos para alterarem seu estado.

Como aplicação, o sistema foi projetado para atender à execução de experiências do Laboratório de Processadores da EPUSP. Por meio dele o aluno será capaz de agendar experiências, guardar os resultados de suas atividades e executar comandos sobre o *kit* didático, baseado no microcontrolador 8051. O Weblab foca unicamente na parte operacional da experiência pois o grupo entende que todo o processo de avaliação do aluno pode ser realizado, com bons resultados, através do uso de sistemas como o COL 3.3.

Tem-se ainda o sistema Weblab como exemplo de uma arquitetura simples e flexível que faz uso de *frameworks* importantes do mercado como o Struts 2 e o Spring 2.

Palavras-chave: sistemas web, ensino a distância, arquitetura web, controle remoto.

Abstract

BAKIEWICZ, G. P.; CAMPOS, F. R.; FRANÇA, W. Q.; PADOVANI, T. C. **Weblab - Remote learning and control**. 2007. 106 f. Dissertation (Graduate Degree) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2007.

This graduate project aims to provide a web system that allows remote interaction with distinct environments. The communication between the end points of the system requires a distributed solution, with interoperability between modules and use of diverse communication protocols. A critical parameter of the system is the answer time on the execution of activities, since applications with time requirements should be supported (real time systems). WebLab system is composed by three interconnected modules: web application, internal application (middle layer) and hardware. The web application is responsible for capturing user commands and passing them to the internal application. The middle layer performs an analysis on the data received and responds as requested by the protocol. The hardware acts as an interface between the environment and the system controlled. It is only necessarily because components like jumpers and relays need electrical input to change state.

As application, the system was configured to fulfil the execution of experiences from Microprocessors Laboratory from EPUSP. By its ways, the student should be able to schedule experiences, save results from its activities and execute commands on the experimental board, based on the 8051 microcontroller. The focus is on the operational aspect only, as the group assumes that the evaluation process of the students can be accomplished, with satisfactory results, using systems like COL 3.3.

The system is also an example of a simple and flexible architecture reached with the use of important frameworks, like Struts 2 and Spring 2.

Lista de Figuras

1.1	Diagrama de Blocos Microcontrolador 8051	17
1.2	Diagrama fluxo de trabalho do Laboratório de Processadores sem Weblab .	20
1.3	Diagrama fluxo de trabalho do Laboratório de Processadores com Weblab .	21
3.1	Diagrama de blocos do sistema Weblab	29
3.2	Diagrama de blocos da Arquitetura Web	32
3.3	Frameworks e tecnologias usadas dispostas em camadas	33
3.4	Diagrama de Caso de Uso do Usuário (Aluno)	47
3.5	Diagrama de Caso de Uso do Administrador (Professor)	47
3.6	Classes da Package action - 1 de 3	48
3.7	Classes da Package action - 2 de 3	49
3.8	Classes da Package action - 3 de 3	50
3.9	Classes da Package actionespec	51
3.10	Classes da Package interceptor	51
3.11	Classes da Package kit	51
3.12	Classes da Package result	51
3.13	Classes da Package service	52
3.14	Classes da Package dao	53
3.15	Classes da Package util	54
3.16	Diagrama de Pacotes WebLab	55
3.17	Arquitetura Web Services	57
3.18	Fluxograma de trabalho do Programa Kit	61
3.19	Package DIP	63

3.20 Package PLCC	63
3.21 Package QFP	63
3.22 Package TQFP	64
3.23 Diagrama de blocos do Hardware	66
3.24 Layout completo do circuito impresso	67
3.25 Layout da face do circuito impresso	68
3.26 Layout do verso do circuito impresso	69
3.27 Face do circuito impresso finalizado	70
3.28 Verso do circuito impresso finalizado	71
A.1 Características do TIL111 da FAIRCHILD	76
A.2 Dimensões do TIL111 da FAIRCHILD	77
B.1 Tela de login do Weblab	82
B.2 Tela de boas vindas do Weblab	83
B.3 Navegação sobre os menus do Weblab	84
B.4 Tela de login do Weblab	85
B.5 Realização de consulta de um usuário	86
B.6 Exibição dos detalhes de um usuário	87
B.7 Usuário sendo editado	88
B.8 Confirmação da alteração de usuário	88
B.9 Erro na conexão com o banco de dados	89
B.10 Solicitação de confirmação para exclusão de usuário	89
B.11 Mensagem de usuário excluído com sucesso	90
B.12 Mensagem de usuário cadastrado com sucesso	91
B.13 Cadastro de disciplinas	92
B.14 Informações do Cadastro de Bancadas	92
B.15 Informações do Cadastro de Experiências	93

B.16 Edição do Cadastro de Protocolo	93
B.17 Mensagem de erro ao executar experiência sem agenda	94
B.18 Selecionando a experiência	95
B.19 Selecionando o horário	95
B.20 Mensagem de confirmação de agendamento	96
B.21 Tela de realização de experiências	97
B.22 Uso do terminal durante a experiência	97
B.23 Gerenciador de arquivos Weblab	98
B.24 Mensagem de sucesso da compilação do arquivo selecionado	99
B.25 Mensagem de erro na compilação	99
B.26 Visualização de detalhes da compilação	100
B.27 <i>Download</i> de arquivo	100
B.28 Painel de Controle Weblab	101
B.29 Informações Cadastro de Painéis	102
B.30 Informações da Imagem	102
B.31 Alteração da disposição dos elementos na tela	103
B.32 Cadastro de Permissões dos Usuários	104
B.33 Alteração das Informações das Permissões	104
B.34 Cadastro de parâmetros do sistema	105
B.35 Alterando a senha	106

Lista de Tabelas

2.1	Tecnologia de comunicação	23
2.2	Matriz de Riscos	23
2.3	Matriz de Responsabilidades	25

Lista de Abreviaturas

PR Professor

AL Aluno

USR Usuário

SUSR Usuário com poder administrativo

WL Weblab

LP Laboratório de Processadores

UCP Unidade Central de Processamento

API Application Programming Interface

IP Internet Protocol

TCP Transmission Control Protocol

UDP User Datagram Protocol

HTTP HyperText Transfer Protocol

XML eXtensible Markup Language

RAM Random Access Memory

ROM Read Only Memory

UART Universal Asynchronous Receiver/Transmitter

QFN Quad Flat package No leads

PLCC Plastic Leaded Chip Carrier

TQFP Thin Quad Flat Pack

DIP Dual In-line Package

LQFP Low-profile Quad Flat Package

SDCC Small Device C Compiler

Sumário

1	Introdução	14
1.1	Motivação e Justificativa	14
1.2	Laboratório de Processadores	15
1.2.1	Microprocessadores	15
1.2.2	Microcontroladores	16
1.2.3	A família 8051	16
1.2.4	A placa experimental do 8051	17
1.2.5	Programa monitor	18
1.2.6	Execução de experiências	19
2	Planejamento e Metodologia	22
2.1	Modelo de comunicação	22
2.2	Fluxo de trabalho	22
2.3	Tecnologias usadas na comunicação	23
2.4	Matriz de Riscos do Projeto	23
2.5	Matriz de Responsabilidades	24
3	Especificação Funcional	26
3.1	Requisitos do Sistema	26
3.1.1	Requisitos Funcionais	26
3.1.2	Requisitos Não Funcionais	27
3.2	Descrição geral do Sistema	28
3.3	Sistema Web	30

3.3.1	Descrição Arquitetura	30
3.3.2	Descrição dos Casos de Uso	33
3.3.3	Diagrama de Casos de uso	47
3.3.4	Diagrama de Classes	48
3.4	Programa Kit	56
3.4.1	Descrição	56
3.4.2	Possibilidades	56
3.4.3	Opção escolhida	58
3.4.4	Protocolo	59
3.4.5	Fluxograma Programa Kit	61
3.5	Hardware	61
3.5.1	Descrição	61
3.5.2	O processo de escolha	62
3.5.3	Solução para obter a quantidade de bits desejada	64
3.5.4	Projeto e especificação do Hardware	65
3.5.5	Diagrama de blocos da Interface Kit 8051	66
3.5.6	Layout circuito impresso completo	67
3.5.7	Layout face do circuito impresso completo	68
3.5.8	Layout verso do circuito impresso completo	69
3.5.9	Face do circuito impresso finalizado	70
3.5.10	Verso do circuito impresso finalizado	71
4	Evoluções Futuras	72
4.1	Avaliação automática das atividades	72
4.2	Conclusão	73
	Referências	74

Anexo A – TIL111	75
Apêndice A – Instalação do sistema Weblab	78
A.1 Manual de Instalação	78
A.1.1 Sistema Web	78
Apêndice B – Operação do sistema Weblab	82
B.1 Manual de Operação	82
B.1.1 Iniciando o Weblab	82
B.1.2 Estrutura do menu	83
B.1.3 Cadastros	84
B.1.4 Fazendo experiências	94
B.1.5 Administrando o Sistema	101
B.1.6 Alterando a senha	105

1 Introdução

Deseja-se produzir, com este projeto, um sistema que permita a realização e monitoramento de experiências em laboratórios remotos. O sistema provê aos usuários acesso e controle dos equipamentos presentes no laboratório, além de ferramentas de administração para o uso dos equipamentos e avaliação dos resultados. O objetivo da solução é oferecer uma plataforma de execução de atividades para a área de ensino à distância através da internet. Aplicações em áreas como monitoramento residencial ou testes em dispositivos digitais também devem ser possíveis.

O sistema está configurado para uso no Laboratório de Processadores da Escola Politécnica, com o desenvolvimento dos módulos específicos à aplicação. Ele permite aos alunos realizar experiências programadas através da web sob a supervisão dos professores. Os equipamentos do laboratório que estarão disponíveis para as experiências são os *kits* didáticos baseados no microcontrolador 8051. É possível interagir com ele através de comandos que permitem, entre outros: carregar e executar programas, verificar e alterar o conteúdo da memória. O acesso aos dispositivos periféricos é disponibilizado através de uma placa de interface, responsável por ativar eletricamente dispositivos como teclado, *jumpers*, câmera *IP*, que transmite imagem do *display* alfanumérico e som emitido pelo speaker. O *kit* didático e seus periféricos são apresentados mais abaixo, na seção 1.2.4.

O sistema é composto por 3 módulos: interface web com o usuário, núcleo de conexão em tempo real e interface com os equipamentos do laboratório. O núcleo é totalmente desacoplado do restante da aplicação, permitindo assim o seu reuso em diferentes laboratórios.

1.1 Motivação e Justificativa

Dada a estrutura tecnológica disponível, é cada vez mais freqüente que sistemas convirjam tecnologias diferentes, principalmente utilizando a Internet. Além de simplesmente integrar sistemas, é também importante a garantia de que eles funcionem em condições

adversas (erros de conexão, atrasos, má qualidade da comunicação) e ainda sim tenham tempos de respostas mínimos para a execução adequada dos processos.

O produto deste projeto vem ao encontro desta tendência, permitindo que alunos e professores executem suas respectivas atividades, no que diz respeito a laboratórios, remotamente. Há ainda a possibilidade de que outras instituições de ensino utilizem este sistema fazendo adaptações e melhorias aos laboratórios.

1.2 Laboratório de Processadores

A aplicação escolhida para o sistema Weblab foi a disciplina de Laboratório de Processadores, obrigatória para alunos da ênfase Computação do curso de Engenharia Elétrica da Escola Politécnica da USP. O objetivo da disciplina é apresentar técnicas e métodos de desenvolvimento baseados em microprocessadores, através de teoria e desenvolvimento de aplicações em componentes de famílias de sucesso comercial. A disciplina é oferecida em dois módulos complementares; ao final do último há a necessidade de se desenvolver um projeto dirigido. Duas famílias de componentes são apresentadas e utilizadas nas atividades do curso: a do 68000, da Motorola e a do 8051, da Intel.

1.2.1 Microprocessadores

O microprocessador é um dispositivo lógico programável que incorpora as funções de uma UCP¹, em um único *chip*. Sua origem data de 1971 (STEWART; CARVALHO, 1998), quando foi lançado ao mercado o microprocessador 4004. Ele operava com vias de endereço de 12 bits, possibilitando endereçar 4kbytes de memória e com via de dados de 4 bits. A partir dessa tecnologia foi possível desenvolver calculadoras mais modernas, sistemas embarcados e o computador pessoal.

A evolução dos microprocessadores vem seguindo o modelo de Moore (MOORE, 1965), que em sua famosa lei sugere que o número de transistores que podem ser dispostos em um circuito integrado, de maneira economicamente viável, cresce exponencialmente, dobrando em aproximadamente 2 anos.

¹A Unidade Central de Processamento - UCP (em inglês, Central Processing Unit - CPU) é a responsável pelo processamento e execução dos programas armazenados na memória principal. As funções da UCP são: executar as instruções e controlar as operações no computador.

1.2.2 Microcontroladores

Algum tempo após o lançamento dos microprocessadores, surgiram os microcontroladores, que possuem a *UCP*, recursos de memória (*ROM* e *RAM*) e de entradas e saídas (serial, *I2C*), *timers* e outros periféricos em um único componente. Eles apresentam desempenho inferior a um computador convencional, mas a um preço menor e com menor consumo de energia. São destinados à aplicações onde o tamanho, custo e gasto de energia são críticos, caso de sistemas embarcados.

O primeiro microcontrolador lançado foi o 8048, da Intel, sucedido pelo 8051, lançado em 1977 e que alcançou grande sucesso, sendo usado até hoje em modelos baseados em sua arquitetura e produzidos por muitos fornecedores distintos.

1.2.3 A família 8051

A família 8051 tem como características:

- Via de dados de 8 bits - pode acessar 8 bits de dados por operação
- Via de endereços de 16 bits - pode acessar até 64kB de memória *RAM* e *ROM*
- 128 bytes de *RAM* interna
- 4 portas de entrada e saída, bidirecionais, endereçadas por byte ou bit.
- Porta serial (*UART*)
- 2 *timers* de 16 bits
- 2 canais de interrupção com prioridade
- Modo de baixo consumo

Modelos baseados no 8051 incluem componentes como contadores e *timers* extras, conversores A/D e D/A, RTC, oscilador, interface *I2C*, USB e outros. A família apresenta muitos recursos de programação, através de um conjunto de instruções poderoso e material e ferramentas de apoio ao desenvolvimento de aplicações. Também é possível usar linguagens de alto nível para a programação de microcontroladores, como a Linguagem C, limitando o uso da Linguagem Assembly a rotinas onde o desempenho e o uso de memória são críticos.

1.2.4 A placa experimental do 8051

Para permitir ao aluno uma plataforma de desenvolvimento para o 8051, foi desenvolvida na EPUSP uma Placa Experimental com diversos tipos de dispositivos periféricos acoplados ao microcontrolador 8051 da Intel (MATSUNAGA; TSUGAWA, 1998), também chamada de kit didático, permitindo ainda a conexão com dispositivos externos diversos. Foi incorporada a ela o "Programa Monitor", que através da via serial, viabiliza a conexão da Placa Experimental com um computador ou terminal de vídeo. Esse recurso proporciona ao aluno capacidade de carregar, executar e depurar programas desenvolvidos por ele.

A placa contém diversos dispositivos periféricos como relógio de tempo real, interface para teclado, conversor A/D e interface para display (LCD); há espaço para expansão através de uma placa de montagem de protótipos de circuitos. Todos os componentes estão dispostos em uma única placa de circuito impresso, especialmente projetada para aplicações didáticas e desenvolvimento de protótipos. Na Figura 1.1 pode ser vista a arquitetura do sistema.

O modo de endereçamento do microcontrolador está configurado para a utilização de memória externa, usando as portas de entrada e saída P0 e P2 como vias de dados e endereçamento. A porta P1 é utilizada em sua função normal, para a conexão de um teclado matricial 4X4 (hexadecimal). Os periféricos estão mapeados em endereços da memória externa e são acessíveis por comandos de manipulação de memória. Uma área dessa memória está sobreposta à memória de programa (que permite apenas leitura), possibilitando carregar programas e executá-los em seguida.

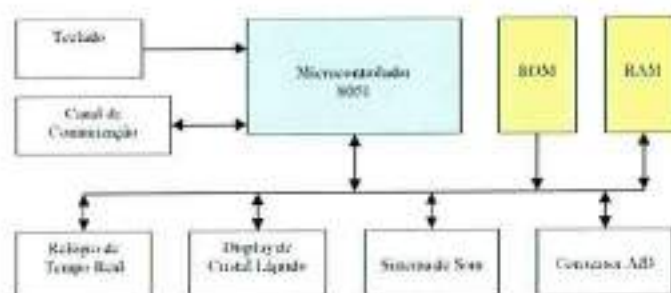


Figura 1.1: Diagrama de Blocos Microcontrolador 8051

1.2.5 Programa monitor

Em execução na Placa Experimental, o Programa Monitor oferece uma interface de acesso para usuários aos recursos da placa através da via serial. Foi definido um protocolo simples que permite a troca de dados através de comandos compostos por caracteres ASCII² (case-sensitive), listados abaixo:

DM - Display Memory

Descrição: mostra o conteúdo das posições de memória

Parâmetros: dois caracteres alfa-numéricos; definem área de memória.

Retorno: conteúdo das posições de XX00 a XXFF, formatado para exibição em terminal.

DR - Display Register

Descrição: mostra o conteúdo dos registradores

Parâmetros: não tem

Retorno: conteúdo da memória interna de 0x00 a 0x7F, formatado para exibição em terminal.

G - Executa

Descrição: execução de programas.

Parâmetros: quatro caracteres alfa-numéricos; definem endereço da posição inicial do programa

Retorno: não tem

L - Lê e carrega HEX

Descrição: carregamento de programas em memória

Parâmetros: arquivo ASCII de extensão .ihx³

Retorno: não tem

M - Modificação de memória

Descrição: modificação do conteúdo da memória.

Parâmetros: quatro caracteres alfa-numéricos; definem a posição de memória.

Retorno: posição de memória (dois dígitos menos significativos). Após o retorno, o usuário deve inserir o conteúdo para a posição mostrada. Um "enter" encerra o comando.

S - Substitui memória

Descrição: substituição do conteúdo de memória após sua verificação. Após o retorno, o usuário deve inserir o novo conteúdo para a posição de memória.

Um "enter" encerra o comando.

²American Standard Code for Information Interchange - é um conjunto de códigos para o computador representar números, letras, pontuação e outros caracteres. Surgido em 1961, um dos seus inventores foi Robert W. Bemer.

Parâmetros: quatro caracteres alfa-numéricos; definem posição de memória.

Retorno: conteúdo da posição de memória solicitada.

T - Testa RAM

Descrição: teste de memória RAM.

Parâmetros: não tem

Retorno: se o teste foi finalizado com sucesso ou não.

A - Testes gerais

Descrição: testa dispositivos do microcontrolador através de interação com display e teclado.

Parâmetros: não tem

Retorno: não tem

1.2.6 Execução de experiências

Hoje, a execução de experiências é feita da seguinte maneira:

1. O Professor informa ao aluno qual assunto estudar e solicita uma pré experiência.
2. O Aluno estuda a experiência e prepara os arquivos que serão carregados no microcontrolador.
3. O Aluno leva esses arquivos para a aula e testa os arquivos, validando os conceitos adquiridos.

Em muitos casos esse tempo de validação leva praticamente todo tempo da aula. A figura 1.2 mostra um diagrama do fluxo de trabalho que é realizado hoje em dia no Laboratório de Processadores.

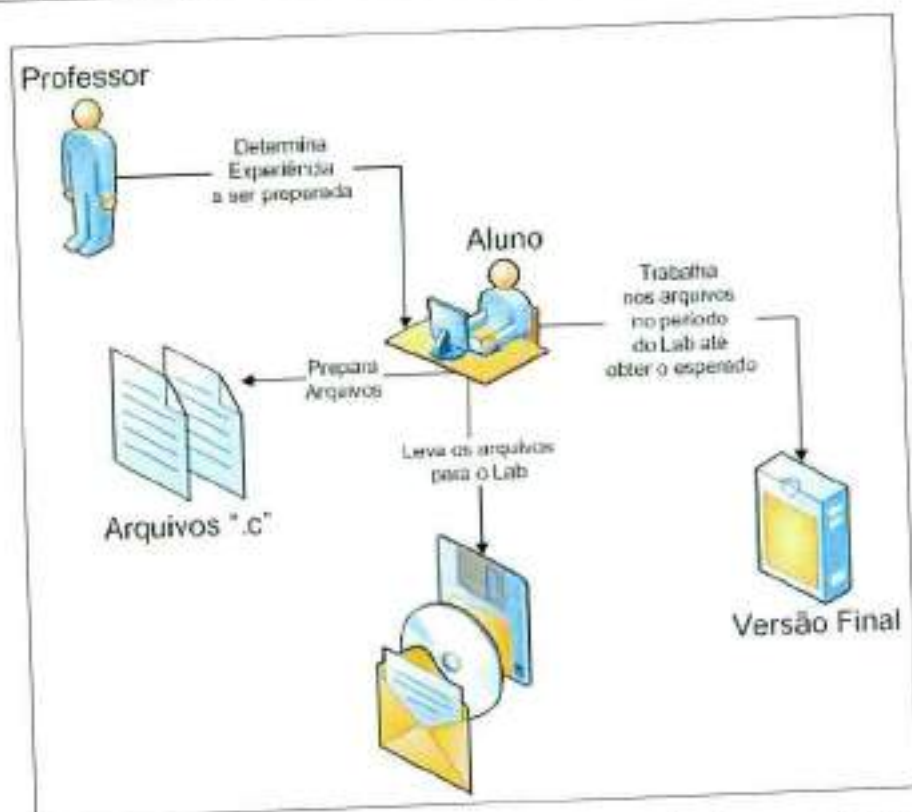


Figura 1.2: Diagrama fluxo de trabalho do Laboratório de Processadores sem Weblab

Com o Weblab haverá uma mudança no fluxo de trabalho do aluno e do professor. E mais importante ainda, o professor contará com a possibilidade de o Aluno estar em sala com suas atividades testadas, permitindo ao professor abordar outros conceitos, explorando melhor a criatividade de seus Alunos.

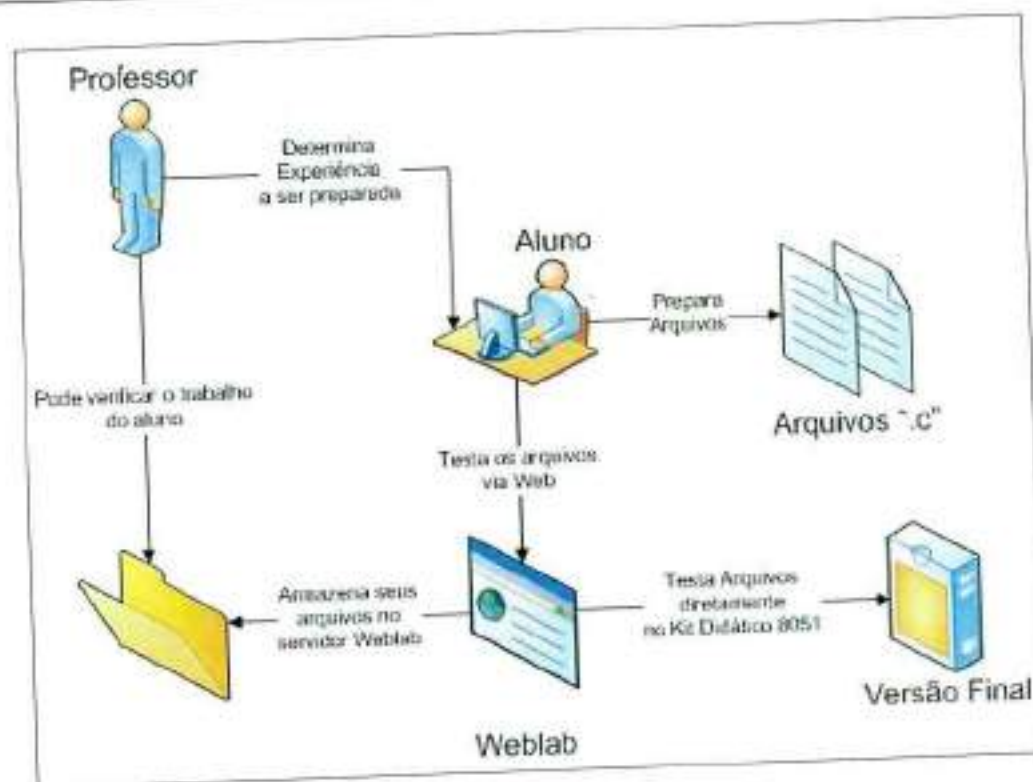


Figura 1.3: Diagrama fluxo de trabalho do Laboratório de Processadores com Weblab

2 Planejamento e Metodologia

2.1 Modelo de comunicação

Para o desenvolvimento de todas essas atividades serão utilizados alguns tópicos, relevantes para este projeto, da metodologia de desenvolvimento ágil *Scrum*¹. Tais tópicos são os seguintes:

- Encontros entre os alunos serão diários e rápidos (no máximo 15 minutos de conversa) para que cada um fale do progresso e dos obstáculos ao bom andamento das atividades;
- Todos devem saber de todas as atividades;
- Foco nos entregáveis;
- Encorajamento de uma equipe auto-organizável.

2.2 Fluxo de trabalho

Todas as atividades serão reportadas para o aval dos orientadores. Logo, os alunos direcionarão seus feitos para os professores, que reportarão se as decisões assumidas estão dentro do esperado. Dessa forma, os alunos são responsáveis por tomadas de decisões do projeto, mesmo antes de conversar com os orientadores.

Nesse contexto, tem-se a seguinte situação:

- Professores responsáveis por verificar se os reais objetivos estão sendo alcançados.
- Alunos responsáveis pelas decisões de projetos, conforme consenso entre eles.

¹Scrum é um processo incremental e iterativo para gerenciamento e controle de atividades, atualmente muito utilizado durante o desenvolvimento de software.

2.3 Tecnologias usadas na comunicação

Durante todas as fases do projeto **WebLab** foram usadas os canais para comunicação, abaixo listados, com os respectivos propósitos:

Tabela 2.1: Tecnologia de comunicação

Tecnologia	Objetivos
Email	Avisar aos alunos e professores sobre datas de encontros.
	Solicitar ajuda na resolução de problemas.
	Avisar sobre imprevistos.
Wiki	Gerar listas de discussões.
	Guardar diário de bordo dos alunos.
	Avisar professores do andamento do projeto.
	Colocar os principais tópicos das atas de reuniões.
	Armazenamento da bibliografia.
SubVersion	Armazenar artefatos gerados.
	Armazenar arquivos de atas.
	Armazenamento dos resultados de pesquisas.
	Armazenamento do código fonte gerado.
	Controle de versão de todos os arquivos.

2.4 Matriz de Riscos do Projeto

Os seguintes riscos foram levantados no início do projeto:

Tabela 2.2: Matriz de Riscos

Ameaças	Probabilidade	Impacto
Não conseguir compilar o kernel do Linux com o RTLlinux	Alta	Alto
RTLlinux não fornecer tempos de resposta desejados	Média	Baixo
Problemas com a resolução da câmera de vídeo	Baixa	Alto
Taxa de quadros/s da câmera de vídeo inferior à desejada	Alta	Alto
Não conseguir adaptar relés no lugar do teclado do kit	Média	Alto
Não conseguir controlar relés remotamente	Média	Alto
Dificuldades na implementação da camada de negócio	Baixa	Muito Alto
Dificuldades na implementação da interface web	Baixa	Baixo
Não conseguir um computador para ser usado como servidor	Baixa	Muito Alto
Erros na arquitetura do sistema	Média	Alto
Erros de codificação do software	Baixa	Baixo

Após o levantamento dessa matriz, o grupo decidiu focar nas atividades de maior risco, antecipando-as no cronograma. Isso forneceu ao grupo, tranquilidade e maturidade na

obtenção dos resultados. Todos os problemas eram compartilhados e discutidos entre os membros. Contudo, a resolução dos problemas ficava sempre focado no responsável. Caso ele não conseguisse resultados no prazo esperado o grupo elegia um auxiliar. Como dito no item anterior, as reuniões de acompanhamento eram usadas para detectar barreiras à obtenção de resultados.

2.5 Matriz de Responsabilidades

Para desenvolvimento das atividades criamos uma matriz de responsabilidades que ligou os membros ao trabalho a ser realizado. Apesar do projeto ser complexo, o grupo decidiu criar as responsabilidades em um único nível, definindo apenas que o responsável de cada função é, juntamente com o professor, a autoridade da tarefa ao qual foi incumbido. Os demais membros atuavam, quando solicitados, como consultores para ajudar em eventuais problemas.

A matriz 2.3 é uma estruturação de um gráfico conhecido como Gráfico de RACI².

²Chama-se RACI por indicar os nomes das funções: Responsável, Reportar-se, Consultoria e Informar (Responsible, Accountable, Consult, Inform, em inglês)

Tabela 2.3: Matriz de Responsabilidades

Atividades	Felipe	Gabriel	Tiago	Winston	Orientador
Elaboração do Documento de Abertura	C	C	C	R	I
Levantamento dos Requisitos de Software	C	R	C	C	I
Modelagem do Diagrama de Classes	C	C	R	C	I
Elaboração da Documentação do usuário	R	C	C	C	I
Modelagem do Diagrama de Arquitetura	C	R	C	C	I
Modelagem do Diagrama Entidade-Relacionamento	C	C	R	C	I
Estimativa de horas	C	C	C	R	I
Controle de horas	C	C	C	R	I
Identificação dos riscos	R	C	C	C	I
Controle dos riscos	R	C	C	C	I
Estimativa da duração das atividades	C	R	C	C	I
Levantamento do sequenciamento das atividades	C	R	C	C	I
Elaboração do cronograma das atividades	C	C	C	R	I
Elaboração do plano de testes dos cadastros	C	C	C	R	I
Elaboração do plano de testes da camada de negócios	C	C	R	C	I
Elaboração do plano de testes da camada de acesso ao BD	C	R	C	C	I
Elaboração do plano de testes de integração	R	C	C	C	I
Elaboração do plano de implantação	C	C	C	R	I
Implementação dos Cadastros	R	C	C	C	I
Implementação das Consultas	C	R	C	C	I
Implementação do controle do kit	C	C	R	C	I
Implementação da comunicação com o kit didático	C	C	R	C	I
Implementação do programa KIT	C	C	R	C	I
Implementação da camada banco de dados	R	C	C	C	I
Implementação do painel de controle do kit	C	R	C	C	I
Implementação da formatação dos dados de saída	C	C	C	R	I
Implementação das telas dos cadastros	R	C	C	C	I
Implementação das telas de relatórios	C	R	C	C	I
Implementação da tela de acesso às ferramentas	C	C	R	C	I
Implementação da tela de controle de arquivos	C	C	C	R	I
Script Tabelas	C	R	C	C	I
Script Relacionamentos	C	R	C	C	I
Scripts Queries	R	C	C	C	I
Script Stored Procedures	R	C	C	C	I
Testes dos cadastros	C	R	C	C	I
Testes da camada de negócios	C	C	C	R	I
Testes da camada de acesso ao BD	C	C	R	C	I
Testes de integração	R	C	C	C	I
Implementação dos testes de integração	R	C	C	C	I
Implementação dos testes de cobertura	R	C	C	C	I
Diário de Bordo	C	C	C	R	I
Elaboração do relatório de resultados	C	C	C	R	I
Elaboração do relatório da teoria aplicada	C	C	C	R	I

3 Especificação Funcional

3.1 Requisitos do Sistema

3.1.1 Requisitos Funcionais

3.1.1.1 Cadastros

O Professor ou o Monitor devem poder cadastrar, alterar e remover disciplinas, experiências e bancadas. Também devem poder cadastrar usuários para os alunos da disciplina.

3.1.1.2 Agendamento de Horários

Para poder realizar uma experiência, o aluno deverá agendar um horário disponível. Os blocos de agendamento são gerados com base nas datas definidas para a experiência e no horário de funcionamento do laboratório. O aluno poderá apenas realizar um agendamento por experiência; ele também deverá poder cancelá-lo e agendar um novo horário. A bancada na qual fará a experiência será alocada automaticamente pelo sistema.

3.1.1.3 Repositório de arquivos para documentos, códigos-fonte ou códigos-objeto

O aluno deve ter acesso a um repositório de arquivos, organizado com pastas distintas para cada experiência. Deve ser possível, a qualquer momento, navegar pelas pastas, fazer *upload* e *download* de arquivos e compilar os arquivos de código-fonte em linguagem C. Esses arquivos poderão ser acessados pelos administradores.

3.1.1.4 Controle de duração de experiência para cada Usuário

Os recursos do painel de realização da experiência estarão disponíveis para o aluno somente durante o tempo agendando. Será permitido ao aluno começar a experiência após o horário

inicial; o horário final, porém, não deverá ser excedido.

3.1.1.5 Execução de experiências

O aluno deve poder executar, através do painel de controle da experiência, os comandos definidos pelo Programa Monitor e ter acesso aos seguintes dispositivos periféricos: teclado hexadecimal, display alfanumérico, botão de *reset* e *jumper* de controle de interrupção.

3.1.1.6 Possibilidade de execução simultânea de experiência em bancadas diferentes

Desde que a questão física esteja resolvida (computadores, câmeras, pontos de rede e hardwares de interface), o sistema deve ser capaz de gerenciar mais de um Usuário navegando pelo site, executando experiências ou mesmo gerenciando suas atividades.

3.1.1.7 Protocolo de comunicação personalizável

O administrador do sistema poderá definir um protocolo de comunicação para adaptá-lo ao ambiente a ser disponibilizado remotamente. O Weblab também permite o envio de conteúdo de arquivos ou simplesmente o envio de bits quaisquer conforme configuração do Usuário.

3.1.1.8 Painel de execução configurável

O Painel de Controle é o ambiente do Weblab que permite a criação de telas por meio de inserção de imagens. Essas imagens podem ser cadastradas com suas características de trabalho e as ações de cada estado podem estar ligadas a eventos que acionam protocolos. Por exemplo, um botão que mude de forma ao ser pressionado e que peça a execução de um arquivo dentro do ambiente remoto.

3.1.2 Requisitos Não Funcionais

3.1.2.1 Usabilidade

O Sistema deve ser de fácil operação para todos os alunos de Universidades.

3.1.2.2 Disponibilidade

O Sistema deve estar disponível durante todo o período letivo, inclusive nos fins de semana e feriados.

3.1.2.3 Segurança

Não permitir acesso de usuários não autorizados e acesso do aluno ao painel de controle da experiência fora do horário agendado.

3.1.2.4 Portabilidade

O Sistema não deve estar fixo a um sistema operacional, podendo funcionar em sistemas baseados no Unix e no Windows.

3.1.2.5 Desempenho

O atraso entre uma requisição do usuário via web e a visualização do resultado não deve exceder 2 segundos.

3.2 Descrição geral do Sistema

O Weblab é um sistema de suporte a aplicações de monitoramento e controle de experiências e testes em dispositivos localizados remotamente, por meio da internet. As ferramentas oferecidas pelo sistema permitem o acesso aos dispositivos através de um painel virtual configurável pelo administrador do sistema, exibido em página web. Outras funcionalidades, voltadas à nossa aplicação, complementam uma solução para o oferecimento de disciplinas no âmbito acadêmico, oferecendo funções de auxílio para gerenciamento de usuários, agendamentos para uso dos recursos e notificações.

A arquitetura do sistema é definida em três níveis, distribuídos em módulos executados em ambientes distintos. No primeiro nível, tem-se o módulo de interação com o usuário, definido como a aplicação web, apresentada por páginas dinâmicas acessadas pelo usuário através de um *browser*. A aplicação web é composta por uma base de dados e uma aplicação escrita em linguagem JAVA.

No segundo, há o programa de comunicação com o *kit*, executado em um PC com acesso às vias de comunicação do dispositivo; este nível pode ser ignorado para dispositivos

com acesso por rede *IP*, como uma câmera. No último nível está a comunicação com os dispositivos, em protocolo suportado pelo mesmo. A Figura 3.1 mostra a disposição do sistema Weblab em blocos.

A parte web é composta por páginas dinâmicas, visualizadas pelo usuário através de um *browser*. Os dados presentes nesse nível são armazenados em um banco de dados. Esses dados são acessados pela aplicação que manipula as páginas, tanto na extração quanto na apresentação, mantendo as informações sobre os usuários, disciplinas e dispositivos.

O programa de comunicação com o *kit* é responsável pela interface de acesso aos dispositivos remotos. Ele é executado em um PC com acesso físico às portas de comunicação das vias utilizadas pelos dispositivos. Se apenas um deles fosse acessado por vez pelo sistema, poder-se-ia usar as portas do próprio servidor onde é executada a aplicação Web. Entretanto, o objetivo é permitir que as experiências possam ser conduzidas simultaneamente em bancadas distintas. Um protocolo de apoio é utilizado para indicar para qual via deve ser encaminhados os dados recebidos e se deve haver retorno de dados. Neste caso a escolha é feita entre serial e paralela.

A última camada é de conexão aos dispositivos que serão controlados ou monitorados. No caso do Laboratório de Processadores, precisou-se de comunicação com o *kit* didático, que deve ser feita com comandos definidos pelo programa monitor sobre a via suportada. Uma placa de interface foi desenvolvida para acionar dispositivos usando-se a porta paralela do PC. Na presente aplicação a placa é utilizada para acionar um teclado hexadecimal conectado ao *kit*, assim como o botão de *reset* e *jumpers*.

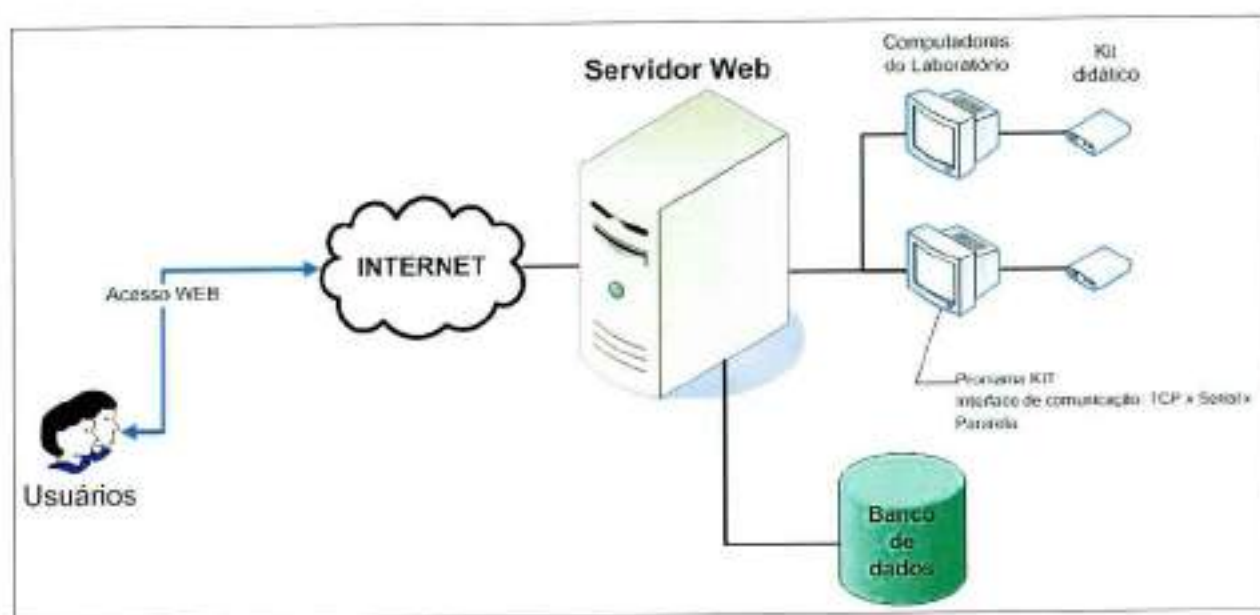


Figura 3.1: Diagrama de blocos do sistema Weblab

3.3 Sistema Web

A distância entre a melhor prática de engenharia de software e as práticas medianas é muito grande - talvez maior até do que em qualquer outra disciplina da engenharia.

Fred Brooks

3.3.1 Descrição Arquitetura

O sistema Weblab foi integralmente construído na linguagem de programação Java, que é uma linguagem de programação orientada a objetos. A notação utilizada para o desenvolvimento foi a definida pela Sun (MICROSYSTEMS, 2007). As mais básicas são:

- Nomes de classes são iniciados por letras maiúsculas.
- Nomes de métodos, atributos e variáveis são iniciados por letras minúsculas.
- Em nomes compostos, cada palavra do nome é iniciada por letra maiúscula - as palavras não são separadas por nenhum símbolo.

Mais informações sobre notações podem ser encontradas em Microsystems (2007).

Para tornar ágil o desenvolvimento, alguns *frameworks* Java, como o Struts 2 e o Spring 2, foram utilizados.

3.3.1.1 Struts

Um dos projetos Apache, o Struts é um *framework* para construção de aplicações web. Algumas das suas ferramentas disponíveis para o programador são:

- Um *framework* de arquitetura acoplável que permite ciclos de vida de requisição customizáveis.
- *Framework* para validação de regras de negócio.
- Facilidades para internacionalização (disponibilizar o sistema em várias línguas).

- *Injeção de dependência*¹ integrada que gerencia os ciclos de vida dos componentes e suas dependências.
- *Tags* de interfaces reusáveis que facilitam o desenvolvimento orientado a objetos.
- Temas em AJAX que simplificam a criação de páginas dinâmicas.
- Fácil integração com outros *frameworks* populares como o Hibernate, Spring e SiteMesh.

Mais informações sobre o Struts podem ser obtidas em Struts (2007).

3.3.1.2 Spring

Spring é um *framework* que auxilia a criação de aplicações J2EE. O Spring inclui:

- *Container* peso-leve com várias funcionalidades, fornecendo centralização, configuração automática e controle da ligação dos objetos da aplicação.
- Camada de abstração para gerenciamento de transações, permitindo o gerenciamento de transações acopláveis.
- Camada de abstração JDBC inclusa, que oferece um detalhamento da hierarquia de exceções do SQL.
- *Framework* para aplicações MVC, que é bem configurável e acomoda múltiplas tecnologias para visualização dos dados.

Mais informações sobre o Spring podem ser obtidas em Spring (2007).

¹É um padrão de desenvolvimento de programas de computadores utilizado quando é necessário manter baixo o nível de acoplamento entre diferentes módulos de um sistema. Nesta solução as dependências entre os módulos não são definidas programaticamente, mas sim pela configuração de uma infraestrutura de software (container) que é responsável por “injetar” em cada componente suas dependências declaradas. A Injeção de dependência se relaciona com o padrão Inversão de controle mas não pode ser considerada um sinônimo deste.

3.3.1.3 Diagrama da Arquitetura

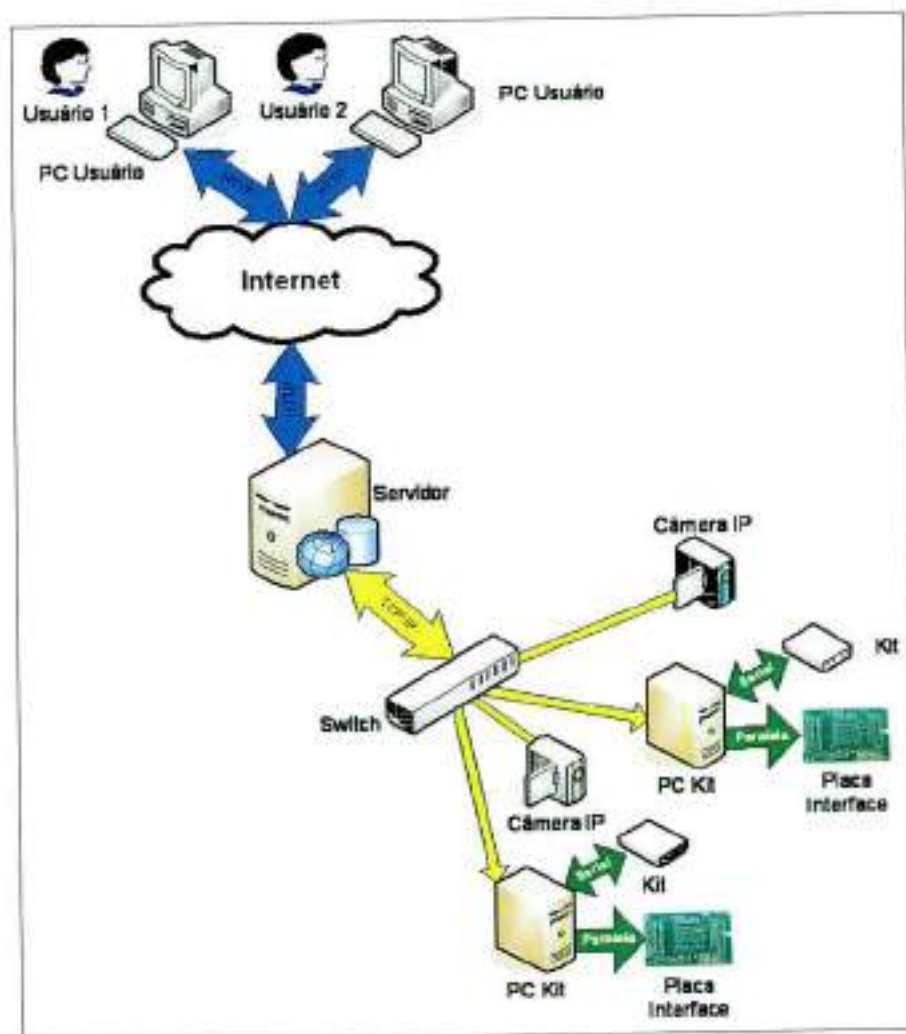


Figura 3.2: Diagrama de blocos da Arquitetura Web

Na figura 3.2 é possível visualizar a localização dos módulos do Weblab em seus vários pontos de uso. Já na figura 3.3 estão apontados os frameworks utilizados e em quais camadas estes foram utilizados.

3.3.1.4 Diagrama da Arquitetura

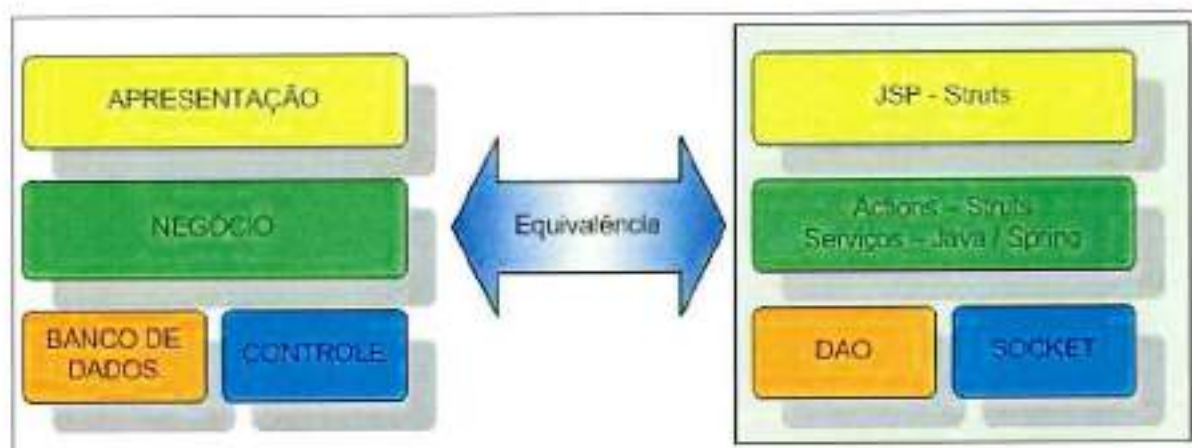


Figura 3.3: Frameworks e tecnologias usadas dispostas em camadas

3.3.2 Descrição dos Casos de Uso

3.3.2.1 Carregar arquivos no microcontrolador

Descrição: No painel da experiência, o Aluno determina o carregamento de arquivo com extensão .ihx para a memória do microcontrolador.

Evento Iniciador: seleção do comando de carregamento de arquivo no painel da experiência.

Atores: Aluno.

Pré-Condição: Aluno identificado pelo sistema ("logado"), com agendamento efetuado e autenticado para realizar experiência.

Seqüência de Eventos:

1. Sistema exibe, no painel de interface com o kit, lista de arquivos .ihx do Aluno;
2. Aluno seleciona o arquivo desejado;
3. Aluno aciona comando para envio do arquivo;
4. Sistema envia arquivo para o programa-monitor do 8051.

Pós-Condição: Arquivo .ihx do Aluno enviado ao kit.

Extensões: não há.

Inclusões: não há.

3.3.2.2 Compilar código-fonte

Descrição: Compilação de um código-fonte em linguagem C armazenado no repositório para código-objeto do 8051.

Evento Iniciador: seleção da opção "Compilar" em um arquivo específico na tela de controle de arquivos.

Atores: Aluno

Pré-Condição: Aluno identificado pelo sistema ("logado"), com agendamento efetuado e autenticado para realizar experiência.

Seqüência de Eventos:

1. Sistema executa o compilador Small Device C Compiler (SDCC) passando como parâmetro o arquivo escolhido pelo usuário;
2. Sistema realiza a compilação e armazena o código-objeto na mesma pasta do código-fonte;
3. Sistema exibe mensagens de retorno do processo ao usuário.

Pós-Condição: arquivo compilado e mensagens de retorno exibidas.

Extensões: não há.

Inclusões: não há.

3.3.2.3 Acionar jumpers e botões

Descrição: usuário determina o acionamento de um botão ou a mudança de estado de um botão comutador.

Evento Iniciador: clique na figura do botão no painel de interface do kit.

Atores: Aluno.

Pré-Condição: Aluno identificado pelo sistema ("logado"), com agendamento efetuado e autenticado para realizar experiência.

Seqüência de Eventos:

1. Sistema Web identifica o botão pressionado pelo Aluno;

2. Sistema Web busca no Protocolo a mensagem correspondente ao estado do botão;
3. Sistema Web envia mensagem ao programa kit;
4. Programa kit encaminha a mensagem à porta adequada (paralela ou serial);
5. Sinal chega à placa experimental e aciona o dispositivo adequado.

Pós-Condição: imagem do botão atualizada para o seu estado seguinte. Resposta (se houver) exibida no painel.

Extensões: 1. No passo 3, se não houver uma mensagem associada ao botão ele não terá ação alguma.

Inclusões: não há.

3.3.2.4 Enviar arquivos

Descrição: usuário faz o *upload* de arquivos de qualquer tipo (documento, código-fonte, código-objeto, etc) para armazenamento no Sistema Web.

Evento Iniciador: usuário seleciona opção “arquivos” no menu.

Atores: Aluno.

Pré-Condição: Aluno identificado pelo sistema (“logado”).

Seqüência de Eventos:

1. Usuário clica em “Arquivo”;
2. Usuário seleciona um arquivo para ser enviado;
3. Usuário clica em “Enviar”;
4. Sistema faz o *download* do arquivo para a pasta atualmente exibida ao usuário;
5. Sistema atualiza lista de arquivos da pasta.

Pós-Condição: Arquivo armazenado na pasta escolhida pelo usuário.

Extensões:

1. No passo 3, quando a experiência já tiver sido executada, a operação de envio de arquivos não será permitida para a pasta correspondente a esta experiência.

Inclusões: não há.

3.3.2.5 Obter arquivos

Descrição: usuário faz o *download* de arquivos armazenados no servidor, em sua pasta.

Evento Iniciador: usuário seleciona opção "Arquivo" no menu.

Atores: Aluno.

Pré-Condição: Aluno identificado pelo sistema ("logado").

Seqüência de Eventos:

1. Sistema exibe lista de arquivos na pasta atual;
2. Usuário clica sobre o nome de um arquivo;
3. O arquivo é enviado ao usuário.

Pós-Condição: não há.

Extensões: não há.

Inclusões: não há.

3.3.2.6 Agendar horário

Descrição: Aluno seleciona data e horário disponível para realizar uma experiência. O sistema confirma a reserva do horário escolhido pelo Aluno.

Evento Iniciador: seleção do item de menu "Agendar Experiência"

Atores: Aluno.

Pré-Condição: Aluno identificado pelo sistema ("logado"), existência de pelo menos uma experiência cadastrada e não encerrada e blocos de horários gerados para experiência.

Seqüência de Eventos:

1. Sistema verifica se Aluno pode realizar agendamentos;
2. Sistema exibe lista de experiências abertas;
3. Aluno seleciona experiência;
4. Sistema exibe calendário da semana atual, com os horários das experiências e o número de bancadas disponíveis em cada um;

5. Aluno seleciona bloco de horário;
6. Sistema pede confirmação do agendamento;
7. Sistema confirma o agendamento e envia e-mail ao Aluno.

Pós-Condição: bloco alocado ao Aluno. Log do agendamento criado pelo sistema.

Extensões:

1. No passo 1: Aluno não pode realizar agendamentos (bloqueado pelo Professor). Sistema informa ao Aluno ele está impossibilitado de realizar agendamentos;
2. No passo 3: Aluno já tem agendamento para a experiência selecionada. Sistema informa ao Aluno a data agendada para a experiência e exibe botão para cancelar o agendamento;
3. No passo 4: Aluno aciona botão de rolagem do calendário. Calendário é atualizado para mostrar nova semana e seus blocos;
4. No passo 6: Aluno não confirma agendamento. Volta para o passo 4.

Inclusões: não há.

3.3.2.7 Acessar painel virtual

Descrição: Aluno acessa tela com painel virtual para realizar experiência em horário agendado a ele.

Evento iniciador: seleção do item de menu "Realizar Experiência".

Atores: Aluno.

Pré-Condição: Aluno identificado pelo sistema ("logado").

Seqüência de Eventos:

1. Sistema verifica se horário atual está dentro do intervalo concedido ao Aluno;
2. Sistema carrega painel da experiência.

Pós-Condição: Aluno com acesso ao painel de experiência. Log do acesso criado pelo sistema

Extensões:

1. No passo 1: horário atual não está no intervalo concedido ao Aluno. Sistema informa que Aluno não pode acessar o painel naquele momento.

Inclusões: não há.

3.3.2.8 Enviar comandos por console

Descrição: Aluno envia comandos através de painel de experiência que emula console.

Evento iniciador: inserção de comando na tela de console.

Atores: Aluno.

Pré-Condição: Aluno logado e com acesso a painel da experiência.

Seqüência de Eventos:

1. Sistema envia comando do Aluno para a porta serial da bancada da experiência;
2. Sistema exibe dados de retorno na tela de console.

Pós-Condição: não há.

Extensões:

1. No passo 2: não há retorno de dados para comando. Sistema exibe tela para inserção do próximo comando.

Inclusões: não há.

3.3.2.9 Cadastrar usuário

Descrição: Professor ou administrador do sistema cadastra um usuário no sistema (por exemplo: Aluno ou Professor).

Evento Iniciador: seleção da opção de cadastro de usuários na tela de usuários.

Atores: Professor ou administrador.

Pré-Condição: usuário autenticado no sistema e com permissão de cadastrar usuários.

Seqüência de Eventos:

1. Sistema carrega tela de cadastro de usuários;

2. Professor/Administrador entra com os dados do novo usuário (código, nome, e-mail e nível de acesso);
3. Professor/Administrador pressiona botão para confirmar o cadastro;
4. Sistema cria usuários que podem acessar o sistema com login e senha padrões (iguais ao código do usuário).

Pós-Condição: usuários cadastrados no sistema, com login e senha de acesso. Log do cadastro criado pelo sistema.

Extensões:

1. No passo 3: Professor/Administrador cancela a operação. Usuário não será cadastrado no sistema.

Inclusões: não há.

3.3.2.10 Cadastrar disciplina

Descrição: Professor cadastra as disciplinas do semestre no sistema.

Evento Iniciador: seleção da opção de inserir disciplinas na tela de disciplinas.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema.

Seqüência de Eventos:

1. Sistema carrega tela de cadastro de disciplinas;
2. Professor entra com os dados da nova disciplina (sigla, nome, data de início e data de fim);
3. Professor pressiona botão para confirmar cadastro;
4. Sistema armazena nova disciplina.

Pós-Condição: disciplina armazenada. Log do cadastro criado pelo sistema.

Extensões:

1. No passo 3: Professor cancela a operação. Disciplina não é armazenada no sistema.

Inclusões: não há.

3.3.2.11 Alocar Alunos na disciplina

Descrição: Professor aloca Alunos em uma disciplina.

Evento Iniciador: seleção da opção alocar Alunos na tela de informações sobre a disciplina.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema. Disciplina cadastrada no sistema. Aluno cadastrado no sistema.

Seqüência de Eventos:

1. Sistema exibe lista de Alunos cadastrados no sistema;
2. Professor faz uma marca em cada Aluno que deseja adicionar à disciplina;
3. Professor pressiona botão para confirmar a alocação dos Alunos;
4. Sistema armazena o relacionamento entre a disciplina e os Alunos selecionados.

Pós-Condição: Alunos alocados. Log do evento criado pelo sistema.

Extensões:

1. No passo 3: Professor cancela a operação. Sistema não armazena alocações.

Inclusões: não há.

3.3.2.12 Cadastrar bancada

Descrição: Professor cadastra uma bancada, que possui um computador com IP fixo e opcionalmente câmera IP, que será controlada remotamente pelo servidor Web.

Evento Iniciador: seleção da opção de inserir bancada na tela de bancadas.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema.

Seqüência de Eventos:

1. Sistema exibe tela de cadastro de bancada;

2. Professor entra com os dados da nova bancada (número, equipamentos, condição, último teste, endereço IP da bancada, porta TCP da bancada, IP da câmera e porta TCP de acesso externo à câmera);
3. Professor pressiona botão para confirmar o cadastro;
4. Sistema armazena nova bancada, disponível para experiências.

Pós-Condição: bancada armazenada no sistema. Log do cadastro criado pelo sistema.

Extensões:

1. No passo 3: Professor cancela a operação. Bancada não é armazenada.

Inclusões: não há.

3.3.2.13 Cadastrar experiência

Descrição: Professor cadastra experiência no sistema, disponibilizando horários para agendamento pelos Alunos.

Evento Iniciador: seleção da opção de inserir experiência na tela de experiências.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema. Disciplina cadastrada no sistema.

Seqüência de Eventos:

1. Sistema exibe tela de cadastro de experiência.
2. Professor entra com os dados da nova experiência (título, descrição, data de início, data de fim, duração máxima, disciplina associada e rótulo para criação de pastas);
3. Professor pressiona botão para confirmar o cadastro;
4. Sistema armazena experiência;
5. Sistema cria automaticamente blocos de agendamento desta experiência de acordo com a duração e datas definidas pelo Professor.

Pós-Condição: experiência armazenada no sistema. Blocos de agendamento criados e disponíveis aos Alunos. Log do cadastro criado pelo sistema.

Extensões:

1. No passo 3: Professor cancela a operação. Sistema não armazena experiência nem cria blocos de agendamento.

Inclusões: não há.

3.3.2.14 Bloquear Aluno

Descrição: Professor bloqueia acesso de Aluno ao agendamento de experiências.

Evento Iniciador: seleção da opção bloquear usuário na tela de informações do usuário.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema. Aluno cadastrado no sistema.

Seqüência de Eventos:

1. Sistema pede confirmação do bloqueio;
2. Professor confirma bloqueio;
3. Sistema marca Aluno como bloqueado para fazer agendamentos.

Pós-Condição: Aluno bloqueado para agendamentos. Log do evento criado pelo sistema.

Extensões:

1. No passo 2: Professor cancela a operação. Aluno não é bloqueado.

Inclusões: não há.

3.3.2.15 Marcar bancada como indisponível

Descrição: Professor marca uma bancada como indisponível, por possuir algum defeito que impede a realização da experiência pelo Aluno, cancelando assim possíveis agendamentos nesta bancada.

Evento Iniciador: seleção da opção marcar como indisponível na tela de bancadas.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema. Alunos, experiências, bancadas e agendamentos cadastrados no sistema.

Seqüência de Eventos:

1. Sistema pede confirmação da operação;
2. Professor confirma a indisponibilidade da bancada;
3. Sistema marca bancada como indisponível;
4. Sistema cancela todos os agendamentos para aquela bancada, informando aos Alunos por e-mail.

Pós-Condição: bancada marcada como indisponível. Alunos avisados por e-mail. Log do evento criado pelo sistema.

Extensões:

1. No passo 2: Professor não confirma a operação. Sistema não altera condição da bancada;
2. No passo 4: não há agendamentos para a bancada. Sistema não emite aviso por e-mail.

Inclusões: não há.

3.3.2.16 Cadastrar comando

Descrição: Professor cadastra um comando do protocolo para comunicação com o programa da bancada, com definição das mensagens a serem associadas com um botão do painel de realização da experiência.

Evento Iniciador: seleção da opção de inserir comando na tela de protocolo.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema.

Seqüência de Eventos:

1. Sistema exibe tela de cadastro de comando;
2. Professor entra com os dados de comando (nome, descrição e mensagem);
3. Professor pressiona botão para confirmar cadastro;
4. Sistema armazena novo comando do protocolo de comunicação.

Pós-Condição: comando armazenado no sistema. Log do evento guardado no sistema.

Extensões:

1. No passo 3: Professor cancela a operação. Sistema não cadastra o comando.

Inclusões: não há.

3.3.2.17 Cadastrar elementos do painel

Descrição: Professor insere um elemento que aparecerá no painel de execução de experiências.

Evento Iniciador: seleção da opção inserir painel na tela do painel de controle.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema.

Seqüência de Eventos:

1. Sistema exibe tela de cadastro de painel;
2. Professor entra com os dados do painel (nome, classe, posição X, posição Y, comprimento e altura);
3. Professor pressiona botão para confirmar cadastro;
4. Sistema armazena novo elemento no painel de experiências.

Pós-Condição: elemento adicionado ao painel. Log do evento gerado.

Extensões:

1. No passo 3: Professor cancela a operação. Sistema não armazena o elemento.

Inclusões: não há.

3.3.2.18 Cadastrar imagens dos botões

Descrição: Insere uma imagem representando um dos estados do botão, que será associada a um comando do protocolo.

Evento Iniciador: seleção da opção inserir imagem na tela de informações sobre o painel.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema. Painel cadastrado e previamente selecionado.

Seqüência de Eventos:

1. Sistema exibe tela para cadastro de imagens;
2. Professor entra com os dados da imagem (arquivo, altura e comprimento);
3. Professor pressiona botão para confirmar o cadastro;
4. Sistema armazena arquivo da imagem e suas informações.

Pós-Condição: imagem adicionada ao botão. Log do evento gerado.

Extensões:

1. No passo 3: Professor cancela a operação. Sistema não armazena imagem.

Inclusões: não há.

3.3.2.19 Alterar disposição do painel

Descrição: altera a posição de cada elemento no painel de execução de experiências.

Evento Iniciador: seleção da opção alterar disposição na tela do painel de controle.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema. Elementos de painel cadastrados.

Seqüência de Eventos:

1. Sistema exibe tela com todos os elementos do painel localizados em suas devidas posições;
2. Professor move cada elemento no painel, clicando sobre o mesmo com o mouse e arrastando-o para a posição desejada;
3. Professor seleciona a opção "Salvar posições";
4. Sistema armazena a nova disposição do painel.

Pós-Condição: elementos reposicionados. Log da operação gerado.

Extensões: não há.

Inclusões: não há.

3.3.2.20 Fazer download de arquivos dos Alunos

Descrição: Faz o download de um arquivo armazenado pelo Aluno no repositório do sistema.

Evento Iniciador: seleção da opção arquivos do menu pelo Professor.

Atores: Professor.

Pré-Condição: Professor autenticado no sistema. Aluno cadastrado no sistema e existência de algum arquivo no repositório.

Seqüência de Eventos:

1. Sistema exibe uma lista das disciplinas que o Professor ministra;
2. Professor escolhe uma disciplina;
3. Sistema exibe lista de experiências da disciplina escolhida;
4. Professor escolhe uma experiência;
5. Sistema exibe uma lista com os códigos e nomes dos Alunos;
6. Professor escolhe um Aluno;
7. Sistema exibe arquivos do Aluno;
8. Professor escolhe um arquivo;
9. Sistema envia o arquivo.

Pós-Condição: não há.

Extensões:

1. No passo 6, se o Professor selecionar a opção de baixar todos os arquivos, receberá um arquivo compactado com todos os arquivos de todos os Alunos para a experiência escolhida.

Inclusões: não há.

3.3.3 Diagrama de Casos de uso

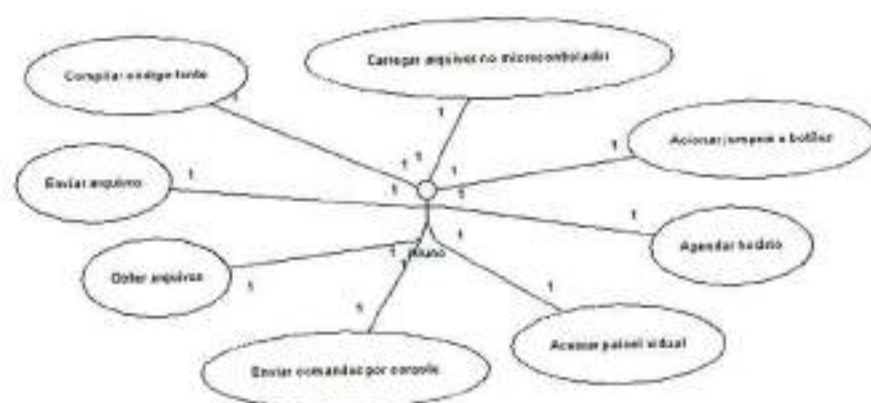


Figura 3.4: Diagrama de Caso de Uso do Usuário (Aluno)



Figura 3.5: Diagrama de Caso de Uso do Administrador (Professor)

3.3.4 Diagrama de Classes

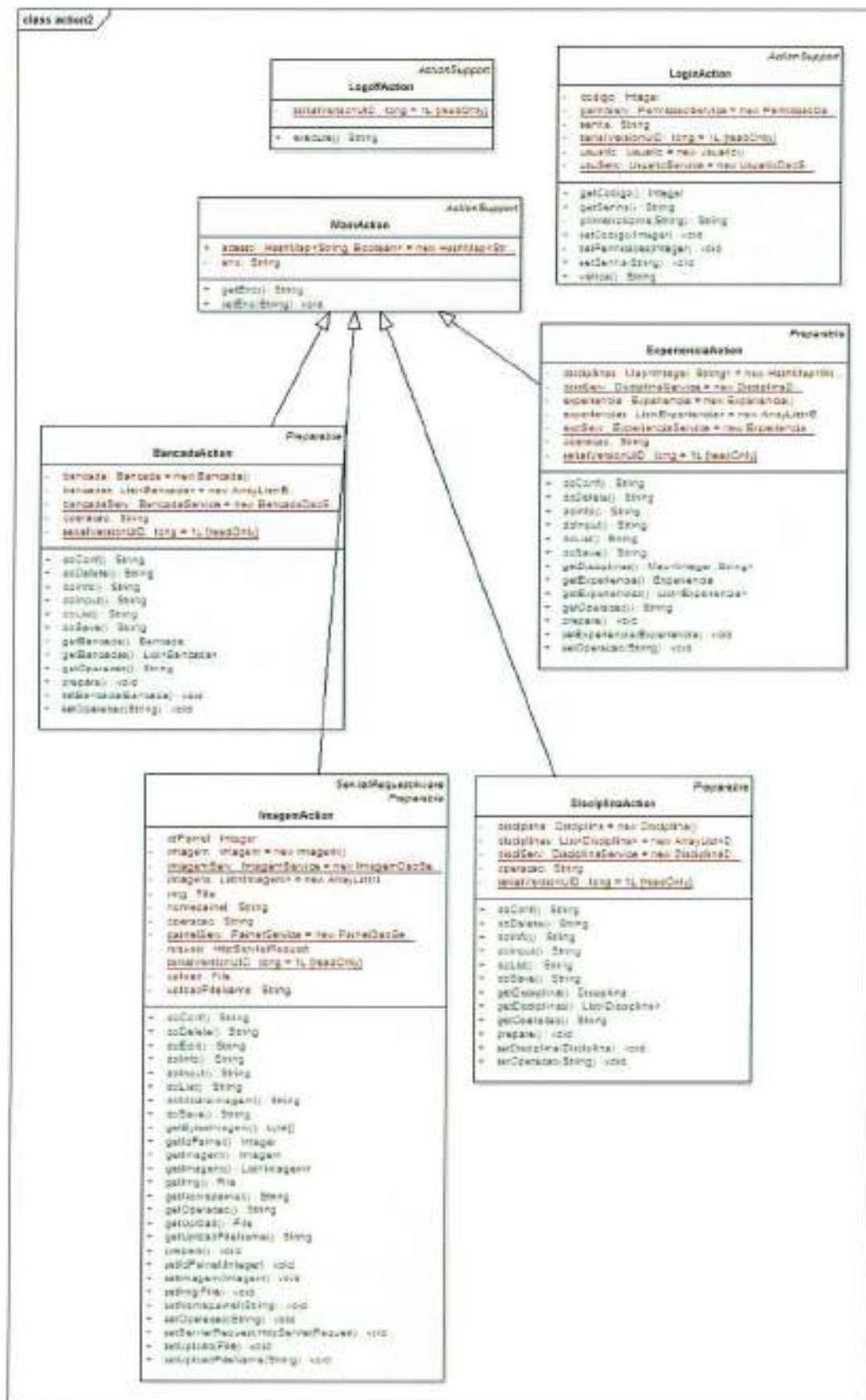


Figura 3.6: Classes da Package action - 1 de 3

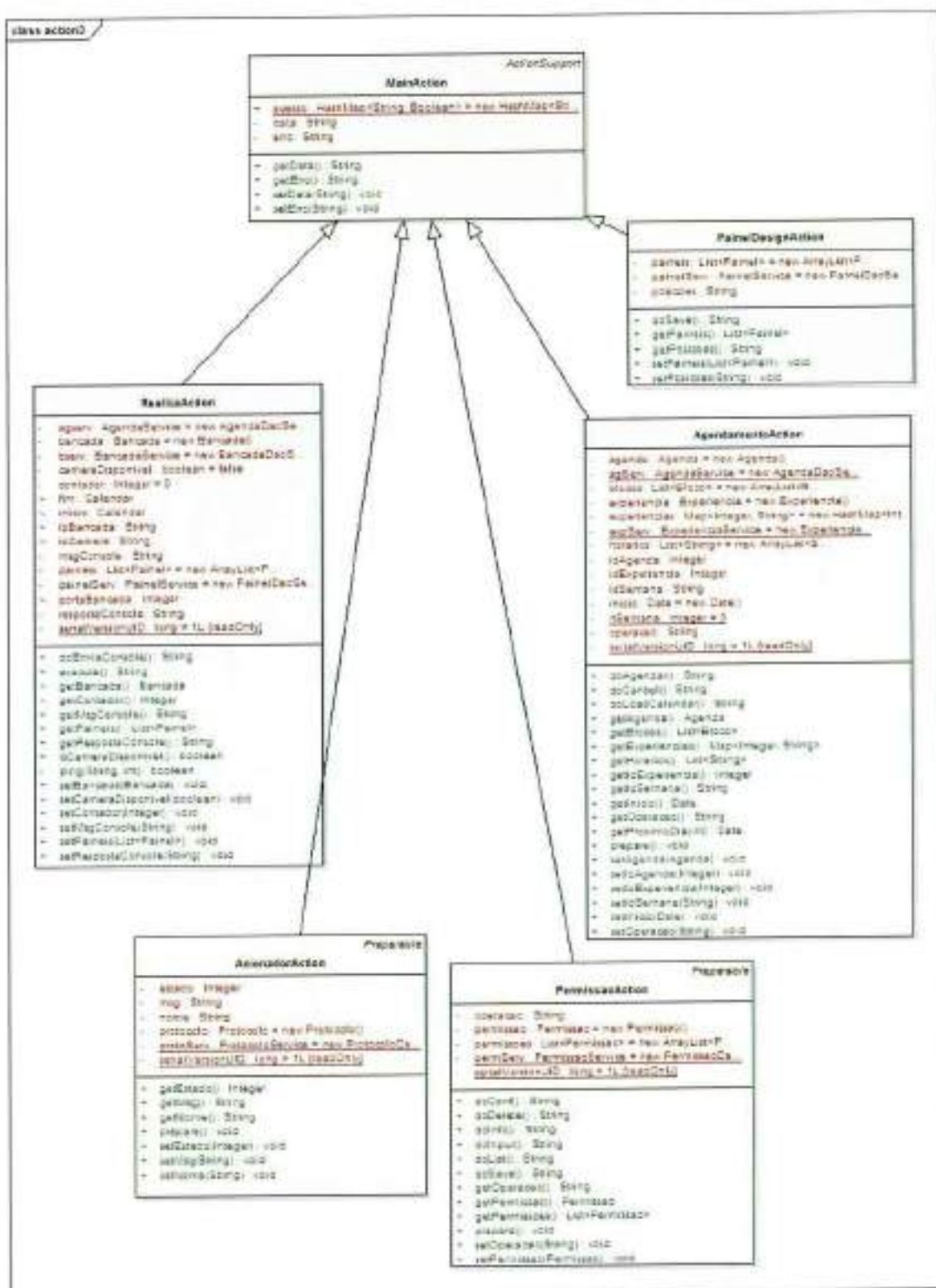


Figura 3.7: Classes da Package action - 2 de 3

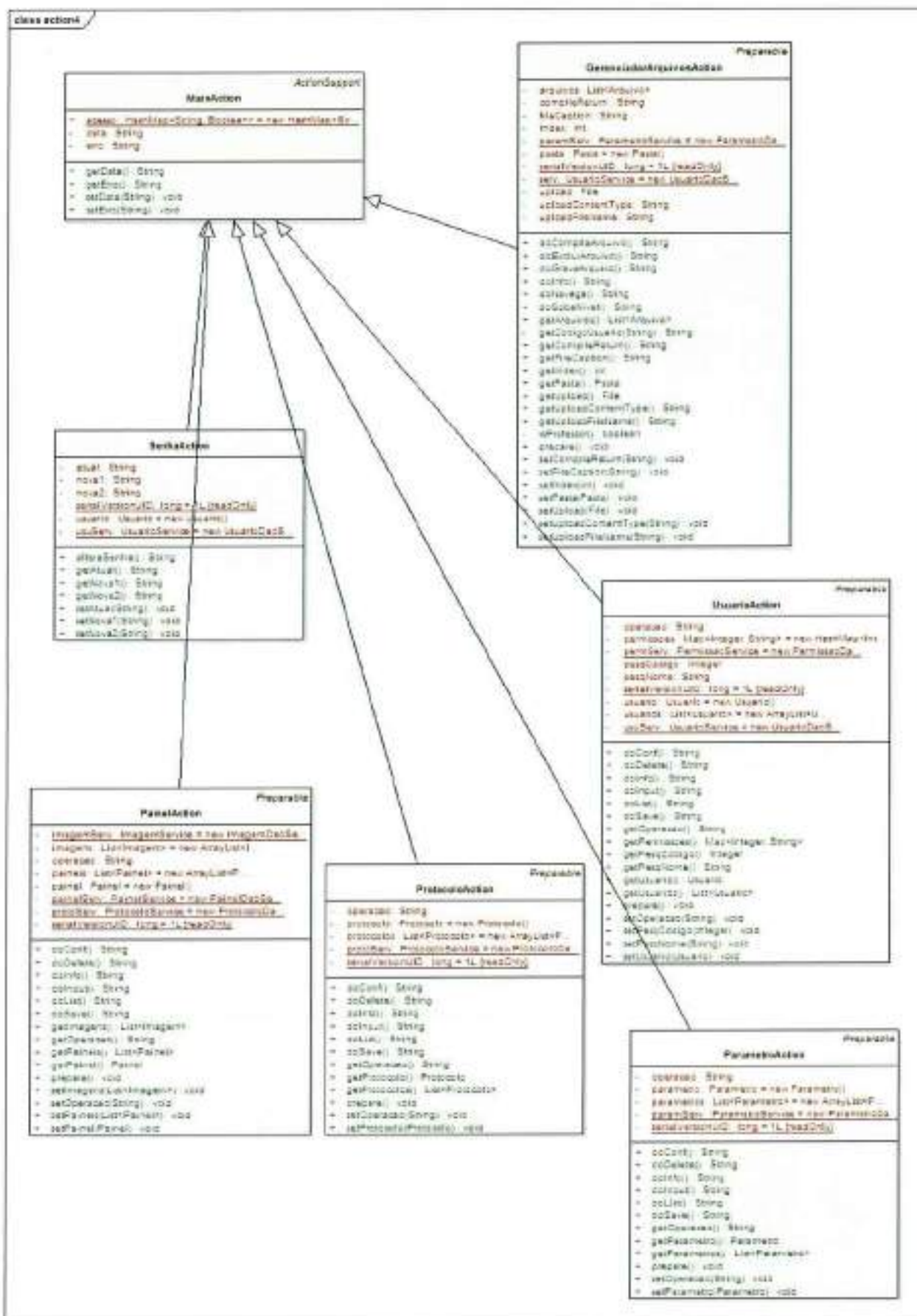


Figura 3.8: Classes da Package action - 3 de 3



Figura 3.9: Classes da Package actionespec

A Figura 3.9 mostra a *package* onde ficarão actions específicas do ambiente a ser disponibilizado remotamente. Para o caso do Laboratório de Processadores, os arquivos .ihx são uma particularidade do microcontrolador 8051.



Figura 3.10: Classes da Package interceptor

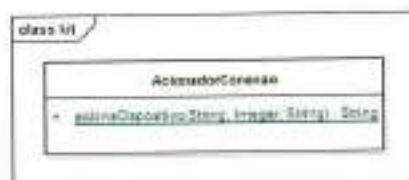


Figura 3.11: Classes da Package kit



Figura 3.12: Classes da Package result

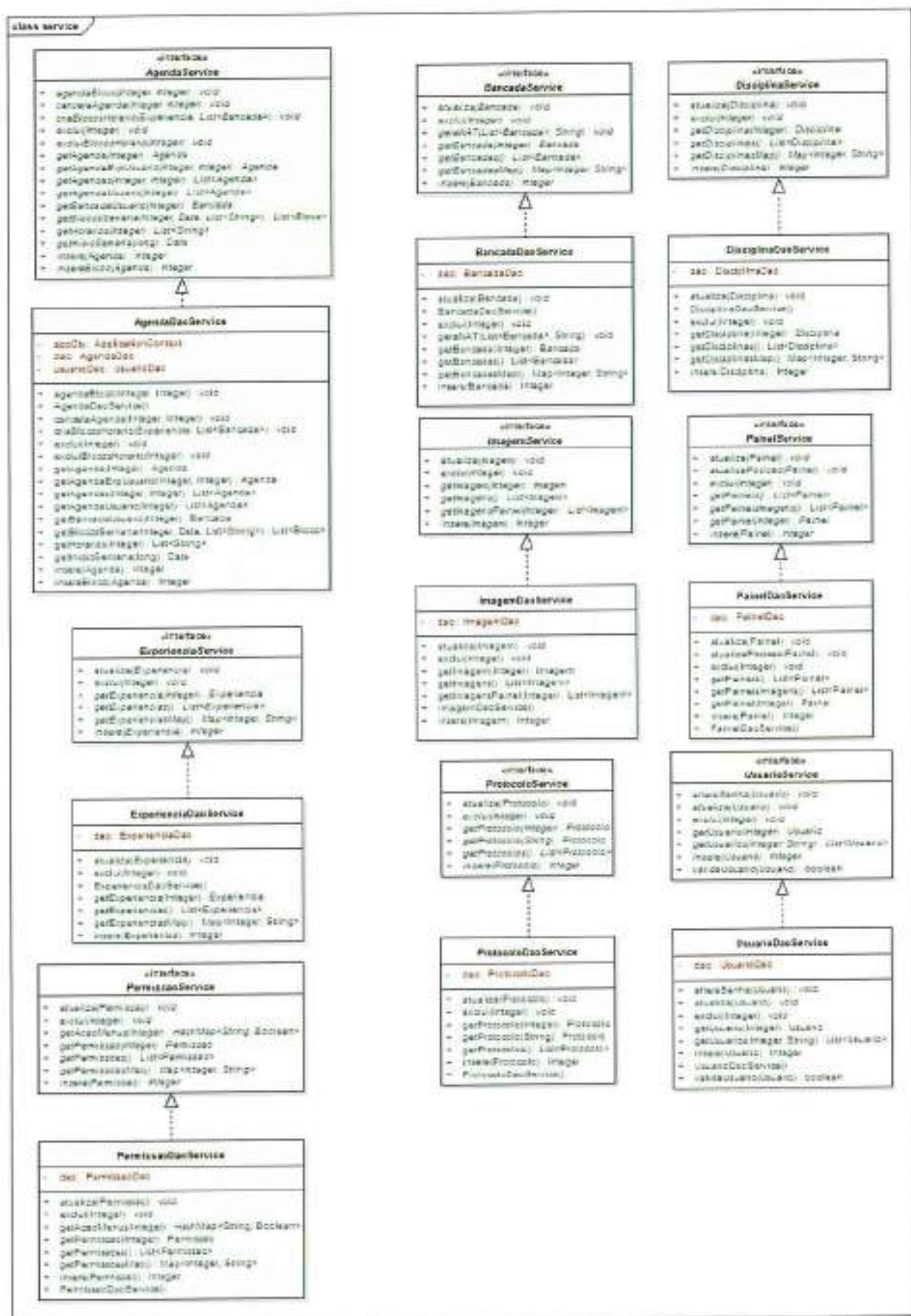


Figura 3.13: Classes da Package service

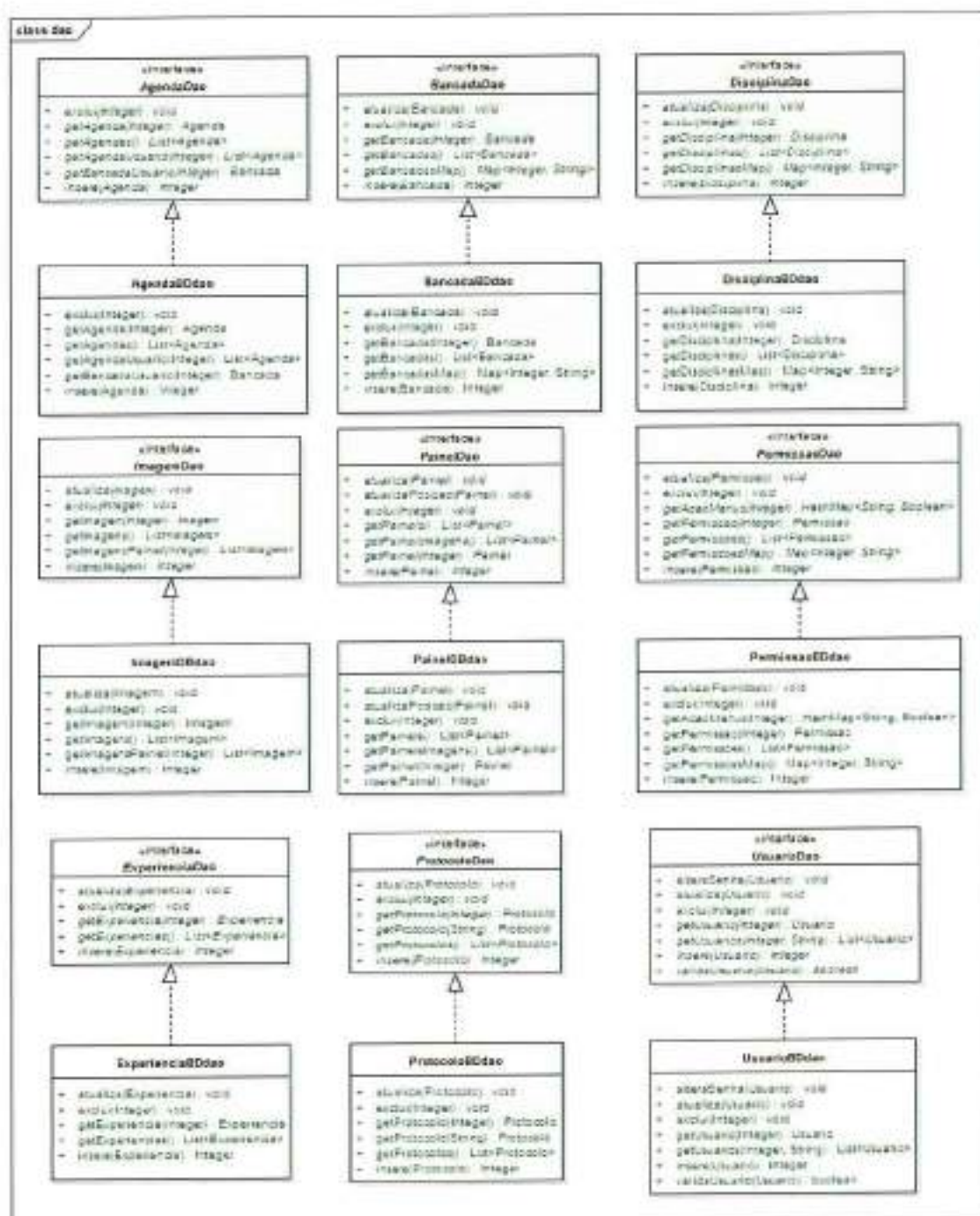


Figura 3.14: Classes da Package dao

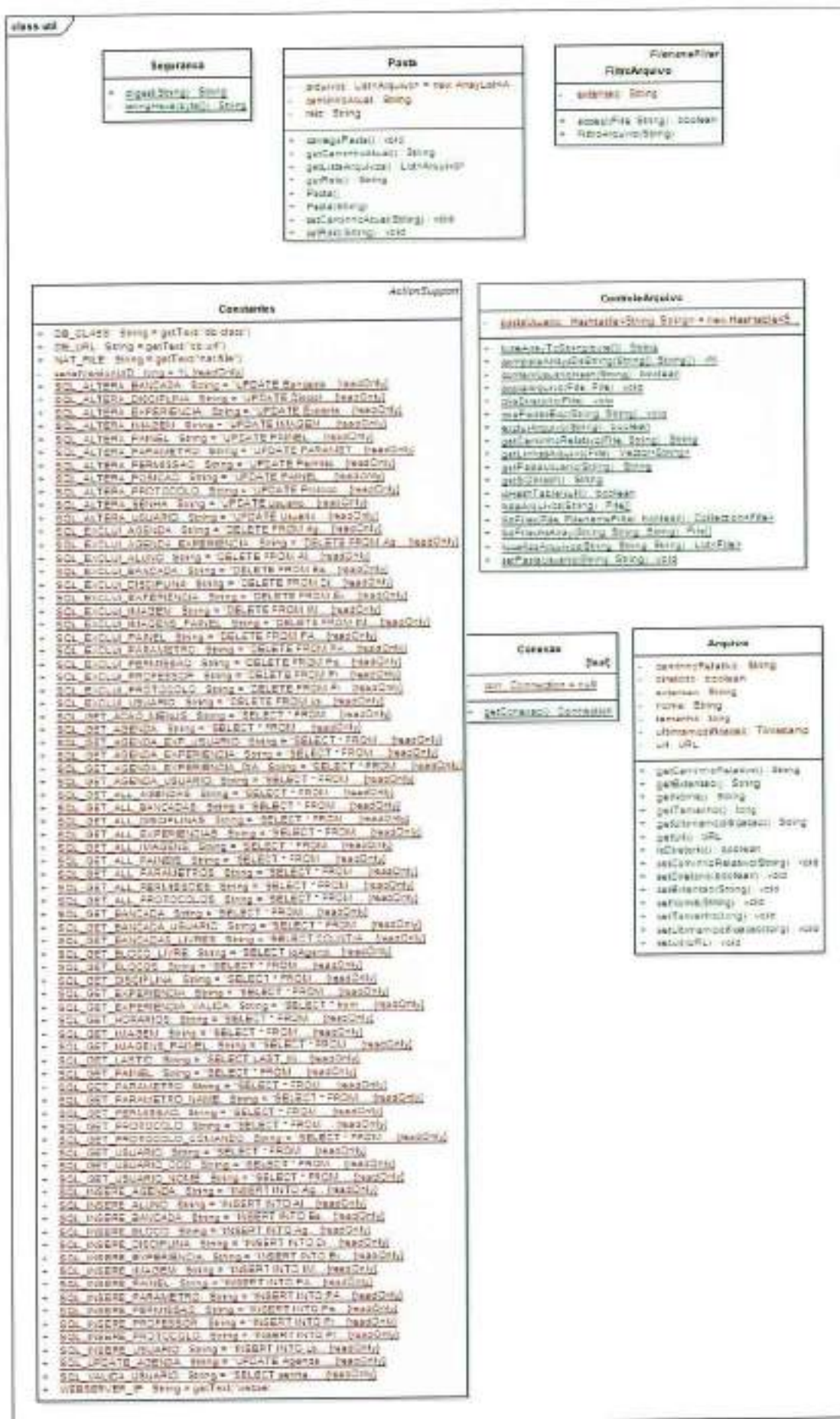


Figura 3.15: Classes da Package util

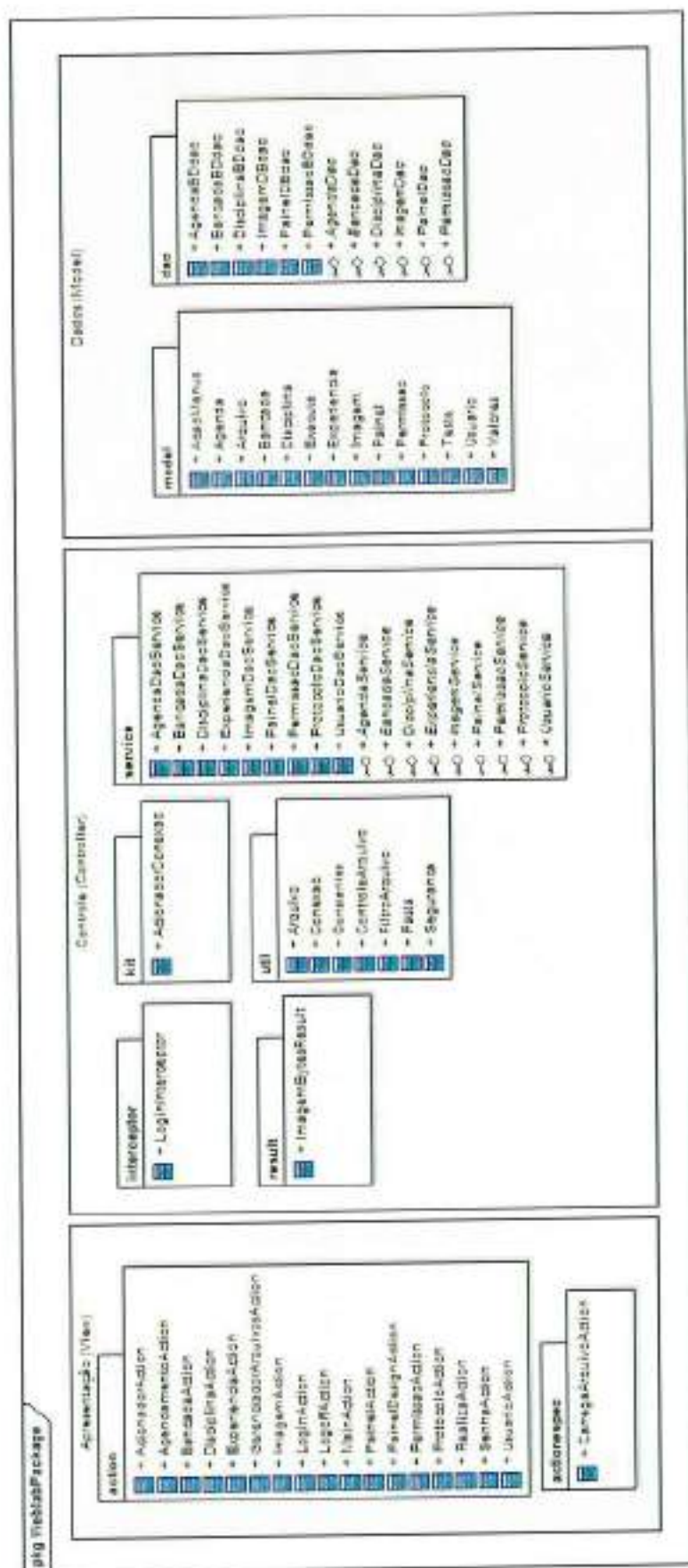


Figura 3.16: Diagrama de Pacotes WebLab

3.4 Programa Kit

3.4.1 Descrição

O Programa Kit tem como função disponibilizar acesso às portas de comunicação que estão presentes no ambiente em que é executado e às quais se deseja encaminhar dados da aplicação web. Ele roda sobre um computador com acesso direto, através da porta adequada, aos dispositivos que serão controlados. Na presente aplicação, para o Laboratório de Processadores, estes dispositivos são a placa experimental e a placa de interface para controle, acessíveis pela porta serial e paralela respectivamente. A câmera *IP*, usada para visualizar o *display* e receber som, não necessita de interface do programa *kit* pois possui interface *IP* e pode ser acessada diretamente.

Para a comunicação entre o servidor de aplicação web e os diversos computadores ligados aos dispositivos, foi preciso usar uma tecnologia voltada a sistemas distribuídos. Algumas delas foram avaliadas e suas características são apresentadas abaixo.

3.4.2 Possibilidades

3.4.2.1 O problema

A matriz de riscos apontou a possibilidade de o sistema não ter um tempo de resposta adequado para a execução das atividades. No caso, experiências do Laboratório de Processadores. Logo, a intercomunicação entre as partes do Weblab deveria ser rápida, confiável e de implementação ágil para que os testes fossem executados o quanto antes. Era preciso interligar um sistema web com um sistema numa rede local e com um hardware. Também foi preciso levar em consideração os protocolos *HyperText Transfer Protocol (HTTP)*, *Ethernet*, a comunicação serial e a paralela.

3.4.2.2 Possibilidade 1 - XML-RPC

XML-RPC é uma especificação e um conjunto de implementações que permite que o software rode em variados sistemas operacionais e em diferentes ambientes para fazer chamadas de procedimentos sobre a internet. É uma chamada remota de procedimentos usando o protocolo *HTTP* como transportador e *eXtensible Markup Language (XML)* como encodificador.

Listing 3.1: Exemplo requisição XML-RPC

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: betty.userland.com
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>41</i4></value>
    </param>
  </params>
</methodCall>
```

3.4.2.3 Possibilidade 2 - SOA

SOA, Service Oriented Architecture ou Arquitetura Orientada a Serviços é um modelo de arquitetura para criação de sistemas que usem seu modelo de negócios como serviços, disponibilizando assim uma infraestrutura para troca de dados entre diferentes participantes, independentemente do sistema operacional ou das linguagens de programação envolvidas.

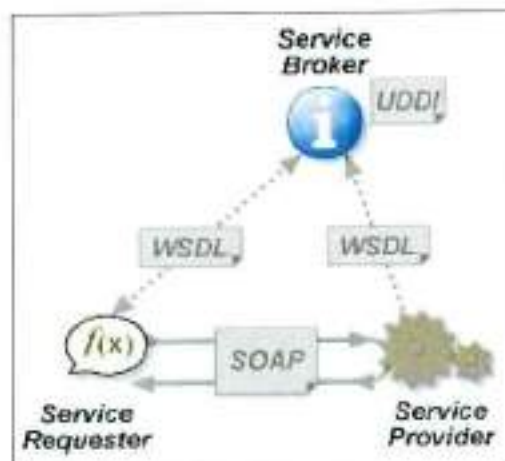


Figura 3.17: Arquitetura Web Services

3.4.2.4 Possibilidade 3 - JMS

Java Message Service - JMS é uma especificação que agrega ao J2EE a capacidade de comunicar-se por meio do envio e recebimento de mensagens. Diferente do e-mail convencional, a JMS atua de forma a permitir a comunicação entre sistemas (aplicações ou

módulos) sem a necessidade de conhecer as interfaces. Basta que tenham um destinatário (local para onde enviar as mensagens e de onde serão consultadas) em comum e que compartilhem uma mesma mensagem.

3.4.2.5 Possibilidade 4 - Socket

Socket é uma *API* que oferece recursos para comunicação inter-processos localmente ou através de uma rede *IP*. Seus recursos são associados ao sistema operacional, que executa um processo para receber e enviar os dados, através do protocolo de transporte escolhido. Os dados que o definem são: endereço *IP* local, porta local, endereço *IP* remoto, porta remota. Por meio dele podemos usar uma arquitetura cliente-servidor.

3.4.3 Opção escolhida

Em Olson e Ogbuji (2007) são comparadas tecnologias de troca de mensagem para Python. Embora a linguagem usada seja diferente da adotada no projeto, os resultados podem ser estendidos para outras implementações. Outros fatores além do desempenho foram levados em conta, como tempo de aprendizado para o uso da tecnologia e disponibilidade de ferramentas livres. Por sua simplicidade de implementação e baixo *overhead* associado às mensagens, optou-se pelo uso de Sockets para a comunicação entre a aplicação web e os Programas kit. O protocolo de transporte escolhido foi *TCP*, por proporcionar entrega confiável de dados, evitando que se tenha problemas associados ao tráfego na rede, como mensagens perdidas, recebidas fora de ordem, duplicadas ou corrompidas. Foi criado um protocolo de aplicação para codificar as mensagens, descrito na seção 3.4.4. O papel de servidor foi atribuído ao Programa Kit, executado em *background* nos computadores do laboratório conectados aos kits didáticos. O cliente é o módulo da aplicação web responsável pela comunicação com os dispositivos.

No Laboratório de Processadores é preciso interagir com o *kit* de desenvolvimento para microcontroladores da família 8051. Entre as ações que o usuário deve ter acesso estão leitura e modificação do conteúdo da memória, carga e execução de programas feitos pelo usuário e acesso aos dispositivos periféricos, como teclado e *display*. Pode-se separá-las em dois grupos: o que acessa recursos internos e externos. O primeiro grupo deve interagir com o programa residente no *kit*, chamado Programa Monitor, via RS-232². O segundo precisa de um hardware de apoio para realizar suas funções através da porta paralela.

²padrão de troca de dados serial

Nas mensagens destinadas à porta paralela, é necessário guardar o estado dos bits da porta, já que não pode haver interferência entre as ações que os compartilham. Como exemplo, podemos citar um caso em que um *jumper* estivesse fixo numa posição e o usuário desejasse acionar uma tecla do teclado. Se mandássemos apenas os bits equivalentes ao acionamento da tecla, esquecendo-se do estado em que estivessem, a posição do *jumper* poderia ser alterada de forma indesejada. Os dados são codificados de modo que cada byte represente um bit da porta. Bits que não devem ser alterados são representados pelo caractere ASCII "x". Foi previsto o uso de 12 bits da porta paralela: 8 de dados e 4 de controle; efetivamente, fazemos uso de 9 bits. Não há retorno de dados da paralela.

Por imposição dos computadores presentes no Laboratório de Processadores, o Programa Kit foi projetado para operar no sistema operacional Windows 95; a implementação abrange também as versões 98 e XP. A linguagem escolhida para implementação foi C++, pela facilidade que oferece no acesso às vias de comunicação (através de bibliotecas) e por permitir modelagem orientada a objetos.

Foram usados três recursos externos de apoio ao desenvolvimento: módulos de acesso a socket, porta paralela e serial. O acesso ao Socket foi feito com a *API* de Sockets, na linguagem C/C++, para a plataforma Windows: o Winsock. Para a porta serial foi usada a *API* do Windows e para a paralela uma biblioteca dinâmica.

É permitida a alteração de parâmetros da porta serial (nome da porta, taxa de transmissão, data e stop bits, paridade) e Socket (porta *TCP*) através de arquivo de configuração. Novos módulos podem ser desenvolvidos para outras vias de comunicação e integrados ao protocolo.

3.4.4 Protocolo

Foi criado um protocolo para a troca de mensagens entre os programas, usando o protocolo *TCP/IP*, que define campos para parâmetros de controle e dados codificado em ASCII. Esse protocolo contém basicamente 3 campos: o primeiro campo informa ao Programa Kit em qual via devem ser enviados os dados; o segundo, usado apenas para vias que retornam dados, informa se a mensagem espera retorno de dados; por último, são dispostos os dados, no formato adequado aos dispositivos.

Considerou-se o desenvolvimento de um protocolo mais complexo, que definisse comandos para os tipos de mensagem e evitasse repetição de dados. Um exemplo: a mensagem para enviar dados pela porta paralela referente a apertar uma tecla do teclado,

composta por 8 caracteres ASCII, poderia ser substituída por 2 caracteres, que definiriam o tipo da mensagem (controle de teclado) e o valor (tecla). Contudo, adotou-se o primeiro modelo, já que os dados a mais não acarretavam problema de desempenho e a solução é mais simples.

Na implementação, foi usado um algoritmo simples que permite ao Programa Kit identificar a via e encaminhar os dados. Como comentado anteriormente, usou-se as vias RS-232 (serial) e paralela como destino dos dados. Na serial foi usado um parâmetro a mais, para definir se há retorno ou não de dados. Em caso afirmativo, o programa deve realizar operação de leitura do *buffer* da porta serial e enviar os dados resultantes ao cliente, sem nenhum tipo de tratamento. A figura 3.18 mostra o fluxo de trabalho do programa Kit.

O programa monitor apresenta uma dificuldade adicional no envio de alguns comandos, como DM, S e M. Nesses casos, há mais de uma interação com o programa, já que, após a entrada do comando em um terminal, o programa monitor envia uma mensagem de eco. Isto impossibilita o envio imediato do comando e seus parâmetros. A solução adotada foi dividir as mensagens na aplicação Web, de modo que, nesses casos, os dados sejam enviados de forma fragmentada, em mensagens distintas. Isto também vale para o carregamento de programas no *kit*, quando são transmitidos arquivos cujo tamanho pode exceder o *buffer* de recepção. Não há problemas em enviar os dados fragmentados para o Programa Monitor.

3.4.5 Fluxograma Programa Kit

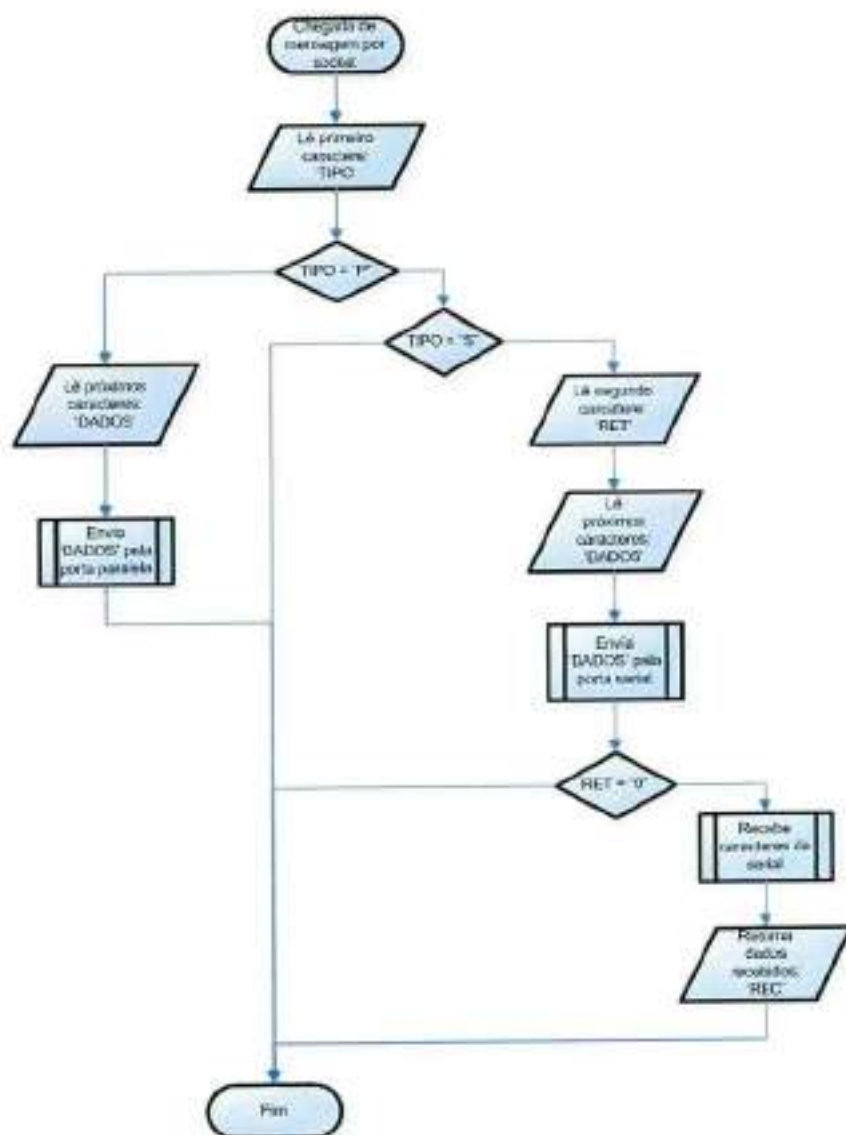


Figura 3.18: Fluxograma de trabalho do Programa Kit

3.5 Hardware

3.5.1 Descrição

O último nível do sistema é composto pelos dispositivos que se deseja controlar ou monitorar. Estes dispositivos podem ser acessíveis por software, através de protocolo específico sobre uma via de comunicação, ou hardware, quando não há interface de comunicação. No Laboratório de Processadores, o acesso à placa experimental é feito por software, através de comandos definidos pelo Programa Monitor. Para acesso aos periféricos da placa, porém, é necessário um hardware de adaptação, que permita a ativação através de

contato mecânico ou elétrico.

Uma placa de circuito impresso foi projetada para atender às necessidades da aplicação. Não foi possível adaptar todos os dispositivos usados em experiências da disciplina de Laboratório de Processadores. Os escolhidos, que permitem a realização da maior parte das experiências, foram:

- Teclado hexadecimal
- Relógio de tempo real
- Reset

Além deles, o display alfanumérico e o som emitido pelo speaker também são acessíveis, através da câmera *IP*.

3.5.2 O processo de escolha

Inicialmente o grupo cogitou centralizar todo fluxo de dados em uma única porta do computador: a serial. Isso porque a quantidade de bits necessária era grande e excedia a capacidade da via de dados da porta paralela (era preciso no mínimo 12 bits e só se tem 8 bits de dados pela paralela). Foi feita uma pesquisa sobre as possibilidades de transformar os sinais paralelos em seriais; achou-se conversores serial-paralelos, da empresa Texas Instruments, de protocolo fechado com 2 *start* bits, 3 bits de endereço, 4 bits de dados e 1 stop bit. Na mesma linha de pesquisa foi possível encontrar soluções mais modernas para conversão serial-paralela. Constatou-se que essas soluções são bastante empregadas em dispositivos móveis pois a transmissão de longa distância de dados é “serial” e o processamento interno paralelo. Esses dispositivos são *transceivers*, referidos como Universal Asynchronous Receiver/Transmitter, e possuem várias aplicações interessantes:

- Controle de fluxo automático ou por hardware ou por software
- *Buffer* tanto para transmissão como para recepção (poucos bytes)
- Geração de paridade de dados
- Controle automático do *start* bit e do *stop* bit (controle feito durante a programação do protocolo)

Os modelos disponíveis pela Texas Instrument são:

- TL16C2550 - package disponível : 32 Quad Flat package No leads, 44 Plastic Leaded Chip Carrier, 48 Thin Quad Flat Pack
- TL16C450 - package disponível: 40 Dual In-line Package, 44 Plastic Leaded Chip Carrier
- TL16C554A - package disponível: 68 Plastic Leaded Chip Carrier, 80 Low-profile Quad Flat Package

Estão listadas abaixo as imagens exemplificando os tipos de packages:



Figura 3.19: Package DIP



Figura 3.20: Package PLCC



Figura 3.21: Package QFP



Figura 3.22: Package TQFP

3.5.3 Solução para obter a quantidade de bits desejada

O teclado usado com a placa experimental é composto por 16 teclas, de 0 a F. Não seria possível usar um bit para o acionamento de cada tecla, já que tem-se apenas 8 bits de dados na paralela. Para minimizar o número de bits utilizados, adotou-se uma representação em forma de matriz, com o uso de decodificadores. Foram usados 2 bits para representar linha e 2 bits para coluna, chegando a uma matriz 4X4. Um bit adicional foi usado para representar o estado do teclado: ativado ou desativado. Esse controle foi feito através da entrada *enable* dos decodificadores. O total de bits usados pela aplicação é 9; um bit da via de controle da paralela foi usado para complementar o total.

Precisou-se ainda de:

- 1 bit para controle do Real-Time Controller (RTC)
- 1 bit para controle do botão reset
- 2 bits para controle de outros componentes do *kit* didático.

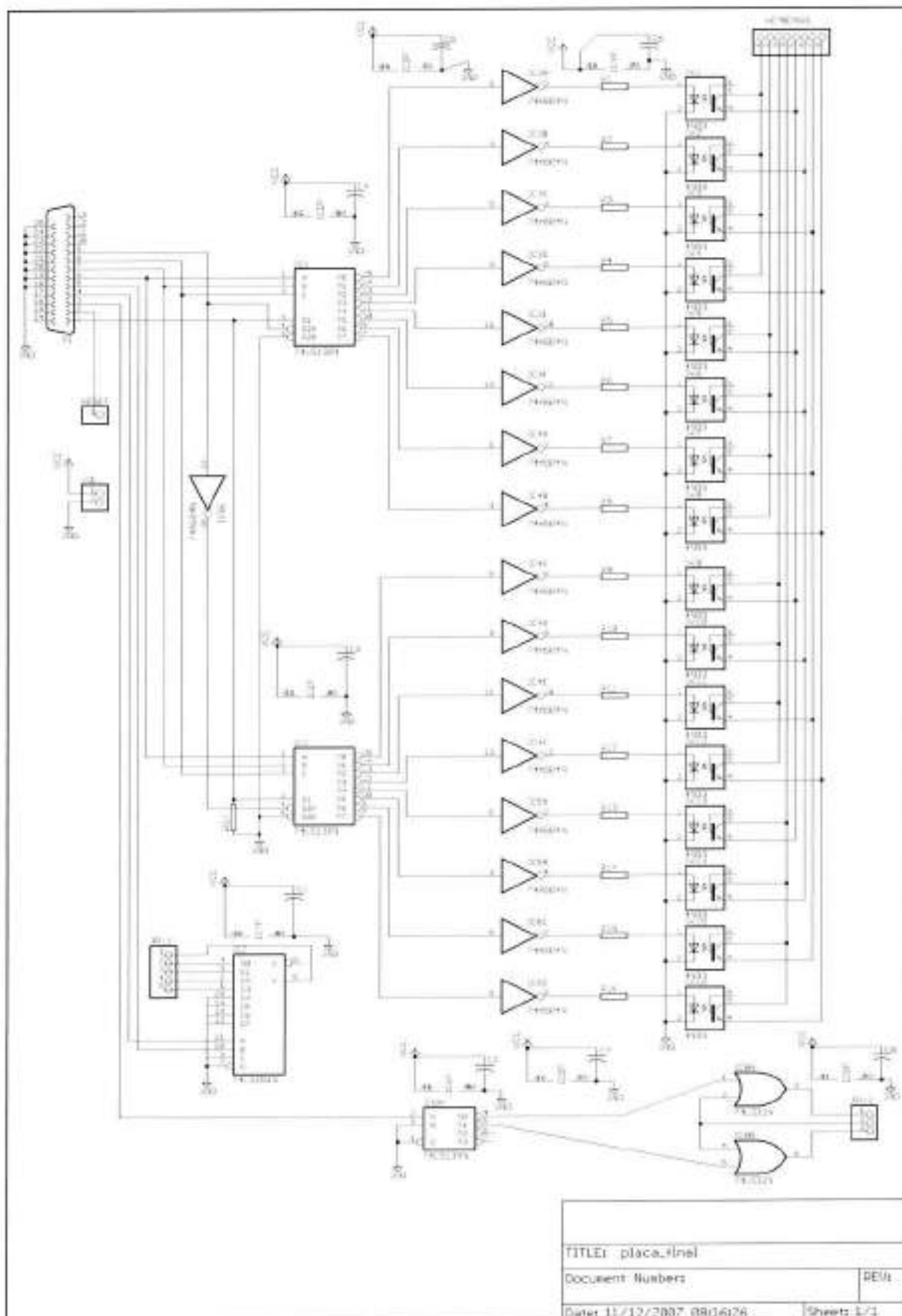
Na tentativa de fazer o fechamento dos circuitos de acionamento das teclas, primeiramente foi usado um relé 5V de bobina e 220V de tensão máxima nos contatos, mas não foi obtida corrente suficiente da porta paralela do computador; seria necessário o uso de circuitos auxiliares para fornecer a potência adequada. Como alternativa para o acionamento das teclas, foi escolhido um acoplador ótico (TIL111). O anexo A contém mais informações sobre o TIL111. A potência para o acionamento do botão *reset* e dos *jumpers* é atendida pela porta paralela.

3.5.4 Projeto e especificação do Hardware

Os seguintes componentes foram utilizados para adaptação do ambiente remoto do Laboratório de Processadores:

- 8 capacitadores poliéster 100nF
- 3 circuitos 7404 (portas NOT)
- 1 circuito 7432 (portas OR)
- 2 circuitos 74138 (decodificador 3x8)
- 1 circuito 74139 (decodificador 2x4)
- 1 circuito 74151 (multiplexador 8 entradas)
- 17 resistores 10k Ω

3.5.5 Diagrama de blocos da Interace Kit 8051



3.5.6 Layout circuito impresso completo

A placa projetada para adaptar o Laboratório de Processadores possui as seguintes características:

- Dupla face - no verso se concentram a maior quantidade de trilhas
- Dimensão: 19,3cm x 17,5cm
- Trilhas de 28mm (largas)
- Baixa densidade (grande área e baixo número de componentes)

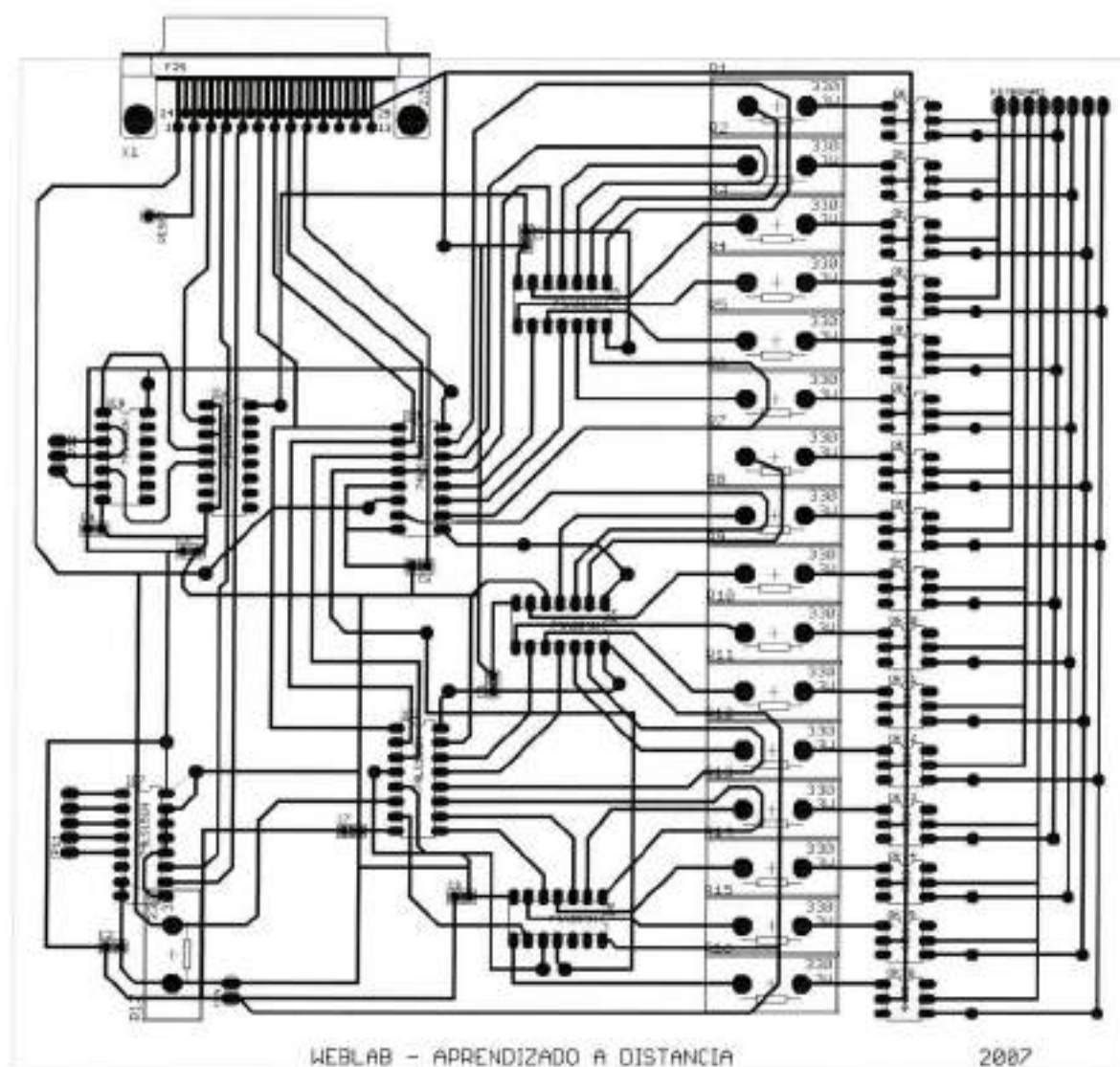


Figura 3.24: Layout completo do circuito impresso

3.5.7 Layout face do circuito impresso completo

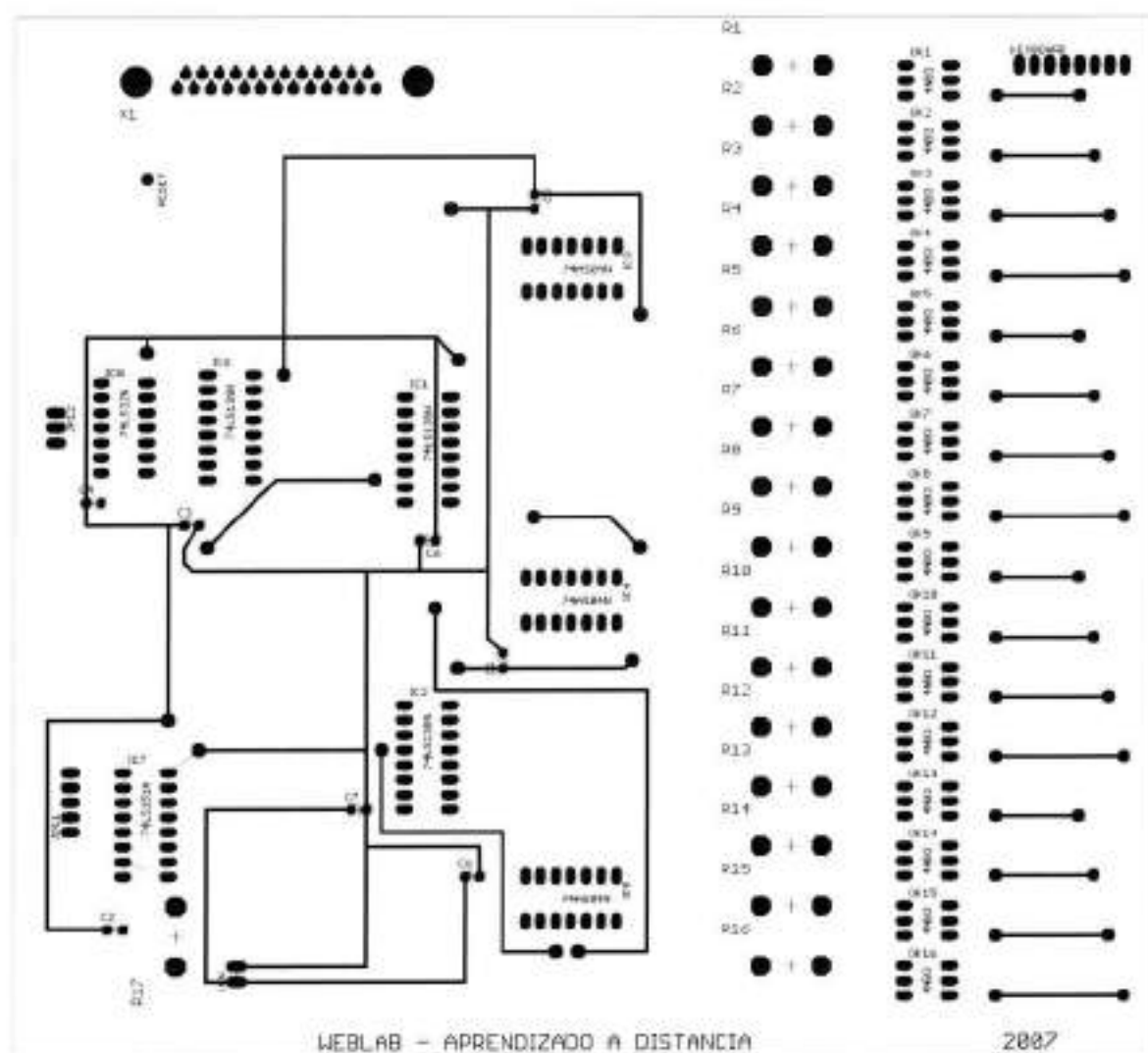


Figura 3.25: Layout da face do circuito impresso

3.5.8 Layout verso do circuito impresso completo

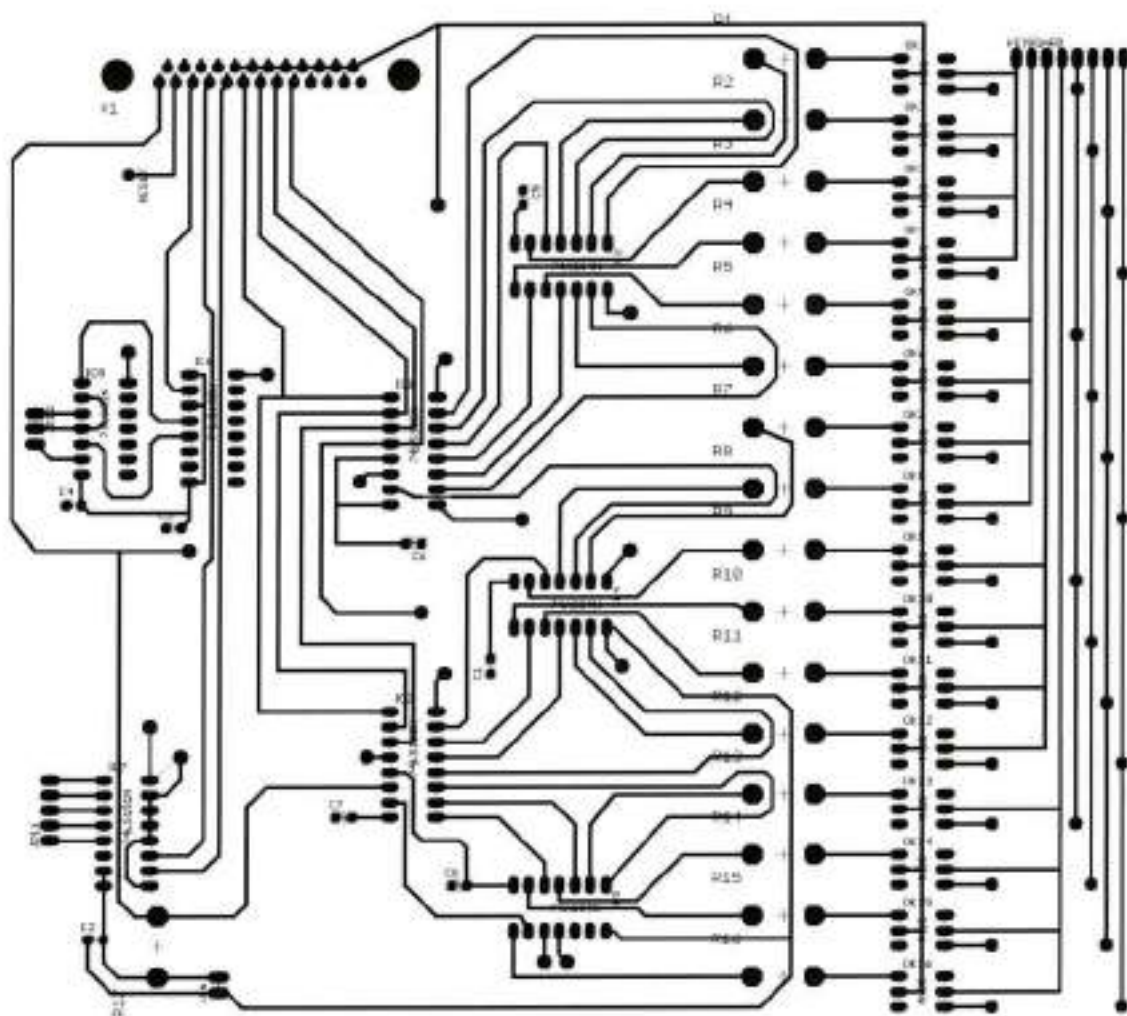


Figura 3.26: Layout do verso do circuito impresso

3.5.9 Face do circuito impresso finalizado

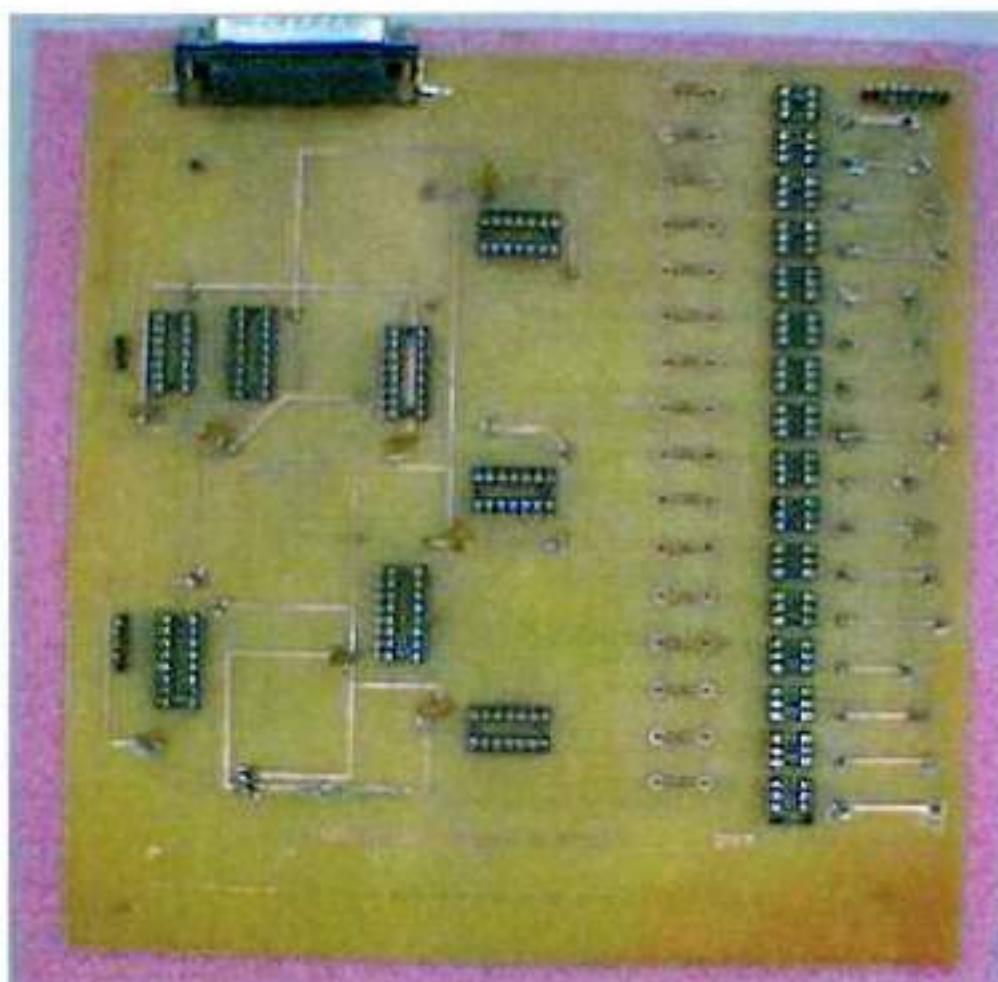


Figura 3.27: Face do circuito impresso finalizado

3.5.10 Verso do circuito impresso finalizado

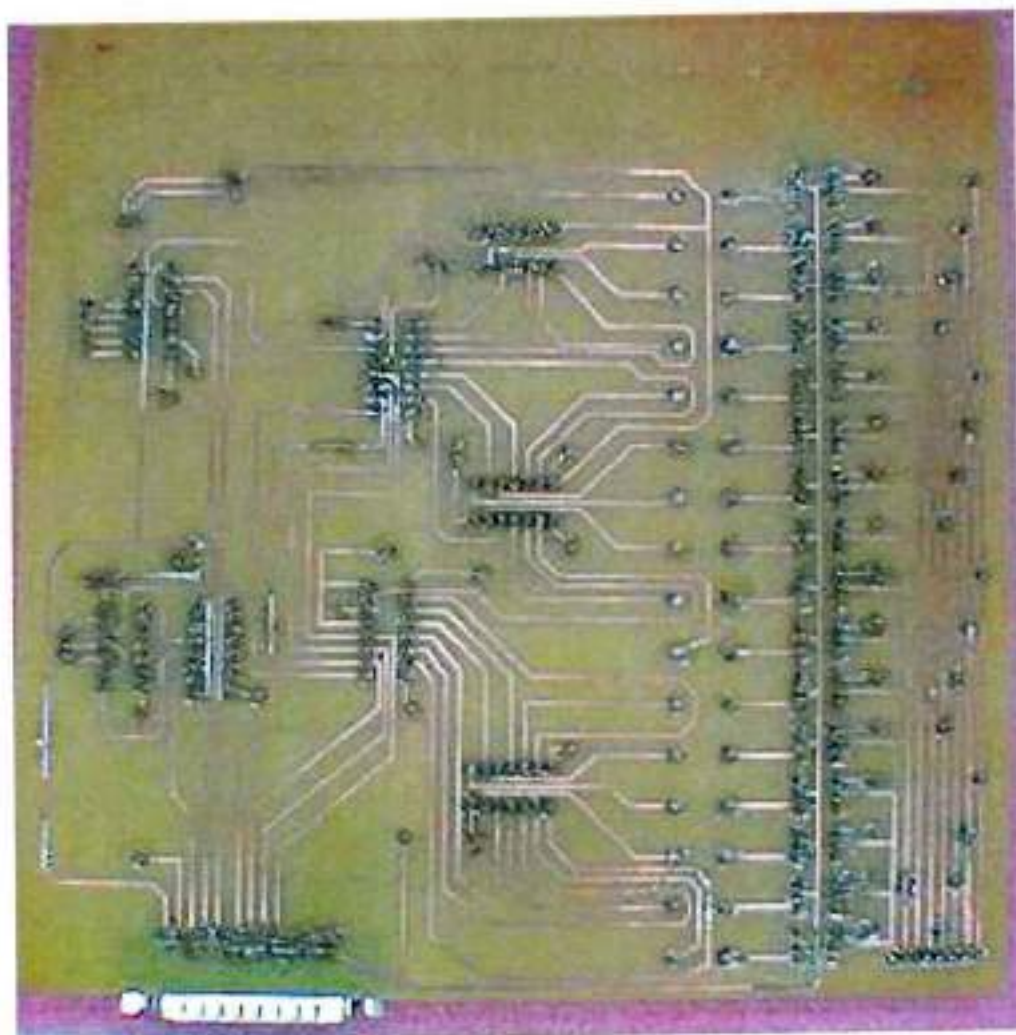


Figura 3.28: Verso do circuito impresso finalizado

4 Evoluções Futuras

Conforme as necessidades de cada ambiente, podem ser adicionadas ao sistema Weblab novas funcionalidades. Entretanto, algumas funcionalidades são consideradas mais importantes para a próxima versão do Weblab:

- Inclusão de novas portas de comunicação, por exemplo USB.
- Integração do sistema com o cadastros de usuários do Sistema COL.
- Criar um nível de controle entre o professor e a disciplina ministrada por ele.
- Criar módulo de correção automático de atividades.

4.1 Avaliação automática das atividades

Dos itens informados como evolução futura, merece destaque a possibilidade de o sistema oferecer ao avaliador automatização do processo de validação das atividades. Essa avaliação teria a seguinte sequência operacional:

1. O avaliador (Professor) prepara um arquivo de testes informando ao sistema o nome do arquivo e os métodos a serem testados. Além dessas informações, é passada uma lista de valores de entrada e de respostas esperadas.
2. O avaliador disponibiliza a experiência e informa ao aluno qual o nome da função que ele precisa desenvolver e também o nome do arquivo que será testado.
3. O Usuário (Aluno) prepara sua atividade, realiza os testes e disponibiliza o arquivo na sua pasta pessoal.
4. Após o término da experiência, o Weblab envia um aviso (e-mail) para o Professor.

5. O Professor se conecta ao sistema e solicita a correção dos arquivos da lista de alunos selecionados, obtendo as respostas das funções de cada aluno. Lembrando que os valores das entradas foram passados pelo professor ao sistema no momento da configuração dos testes. É feita uma comparação entre os valores obtidos e esperados e um resumo indicando sucesso ou falha é apresentado ao Professor.

4.2 Conclusão

O desenvolvimento do sistema Weblab proporcionou o contato com diversos conceitos da área de Engenharia de Computação. Foi abordado o processo de desenvolvimento de software, por meio do uso de métodos e técnicas adequados, bem como o de hardware, através do projeto de uma placa de circuito impresso, envolvendo descrição lógica, seleção de componentes e produção. Na parte de software, foram usadas tecnologias próprias para o desenvolvimento de aplicações web e de inter-conexão entre módulos distintos do sistema. Foi dada muita atenção à arquitetura adotada, que refletiu grande estudo para simplificação e atribuição de responsabilidades de maneira eficiente. Por ser extensível a outras aplicações, houve preocupação adicional do grupo em manter uma arquitetura com módulos fracamente acoplados.

A solução implementada constitui uma ferramenta complementar na área de ensino. Por meio dela o professor pode oferecer contato antecipado ao aluno com os dispositivos que serão abordados na disciplina, permitindo a familiarização do aluno e abre a possibilidade de execução de atividades mais complexas no decorrer da disciplina. Além do mais, o aluno pode ter mais tempo de contato com os temas abordados, já que pode acessá-los de onde estiver, bastando uma conexão com a internet.

Outras aplicações que podem fazer uso do sistema, mediante algumas modificações, incluem a área de automação industrial e experimentação de novos dispositivos, como processadores e micro-controladores.

Referências

- BORGES, A. P. *Instrumentação virtual aplicada a um laboratório com acesso pela Internet*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2002.
- MATSUNAGA, A. M.; TSUGAWA, M. O. *MC8051 Placa Experimental*. Disponível em: <<http://www.pcs.usp.br/pcs2497/>>. Acesso em: 01 de Novembro de 1998.
- MICROSOFT. *Serial Communications in Win32*. Disponível em: <<http://msdn2.microsoft.com/en-us/library/ms810467.aspx>>. Acesso em: 01 de Setembro de 2007.
- _____. *Winsock Reference*. Disponível em: <<http://msdn2.microsoft.com/en-us/library/ms741416.aspx>>. Acesso em: 01 de Setembro de 2007.
- MICROSYSTEMS, S. *Code Conventions for the Java Programming Language*. Disponível em: <<http://java.sun.com/docs/codeconv/>>. Acesso em: 01 de Agosto de 2007.
- MOORE, G. E. *Computer History Museum*. Disponível em: <<http://www.computerhistory.org>>. Acesso em: 01 de Novembro de 1965.
- OLSON, M.; OGBUJI, U. *The Python Web services developer: Messaging technologies compared*. Disponível em: <<http://www.ibm.com/developerworks/library/ws-pyth9/author>>. Acesso em: 10 de Agosto de 2007.
- SPRING. *Spring Official Site*. Disponível em: <<http://www.springframework.org/>>. Acesso em: 09 de Outubro de 2007.
- STEWART, J.; CARVALHO, K. X. M. *The 8051 Microcontroller : Hardware, Software and Interfacing*. [S.l.]: Prentice Hall, 1998. ISBN 0-135-31948-X.
- STRUTS. *Struts Official Site*. Disponível em: <<http://struts.apache.org/>>. Acesso em: 09 de Outubro de 2007.
- VARELLA, W. A. *Laboratório remoto em ambiente aberto com protocolo IPv6*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2003.

Anexo A – TIL111

O TIL111 é um fototransistor com um isolador optico acoplado. Um raio infravermelho emitido por um diodo produzido especialmente para isso. Sua aplicação varia desde controles industriais até entradas para microcontroladores. As principais características do TIL11 são mostradas nas figuras A.1 e A.2.

ELECTRICAL CHARACTERISTICS (At 25°C Free-Air Temperature)						
INDIVIDUAL COMPONENT CHARACTERISTICS						
PARAMETER	SYMBOL	TIL111			UNIT	TEST CONDITIONS
		MIN.	TYP.	MAX.		
INPUT DIODE						
Input diode static reverse current	I_r			10	μA	$V_a = -3 V$
Input diode static forward voltage	V_f		1.2	1.4	V	$I_a = 16 mA$
OUTPUT TRANSISTOR						
Collector-base breakdown voltage	V_{CBR}	70			V	$I_c = 10 \mu A, I_e = 0, I_b = 0$
Collector-emitter breakdown voltage	V_{CEB}	30			V	$I_c = 1 mA, I_b = 0, I_e = 0$
Emitter-base breakdown voltage	V_{EBR}	7			V	$I_c = 10 \mu A, I_e = 0, I_b = 0$
Transistor static forward current transfer ratio	h_{FE}	100	300			$V_{CE} = 5 V, I_c = 10 mA, I_b = 0$
TRANSFER CHARACTERISTICS						
PARAMETER	SYMBOL	TIL111			UNIT	TEST CONDITIONS
		MIN.	TYP.	MAX.		
On-state collector current	Phototransistor operation	$I_{C(on)}$	2	7	mA	$V_{CE} = 0.4 V, I_b = 16 mA, I_e = 0$
	Photodiode operation	$I_{C(on)}$	7	20	μA	$V_{CE} = 0.4 V, I_b = 16 mA, I_e = 0$
Off-state collector current	Phototransistor operation	$I_{C(off)}$	1	50	nA	$V_{CE} = 10 V, I_b = 0, I_e = 0$
	Photodiode operation	$I_{C(off)}$	0.1	20	nA	$V_{CE} = 10 V, I_b = 0, I_e = 0$
Collector-emitter saturation voltage	$V_{CE(sat)}$		0.25	0.4	V	$I_c = 2 mA, I_b = 16 mA, I_e = 0$
SWITCHING CHARACTERISTICS (At 25°C Free-Air Temperature)						
PARAMETER	SYMBOL	TIL111			UNIT	TEST CONDITIONS
		MIN.	TYP.	MAX.		
Rise time	Phototransistor operation	t_r	5	10	μs	$V_{CE} = 10 V, I_{C(on)} = 2 mA, R_L = 100 \Omega$
Fall time	Phototransistor operation	t_f				
Rise time	Photodiode operation	t_r	1		μs	$V_{CE} = 10 V, I_{C(on)} = 20 \mu A, R_L = 1 k\Omega$
Fall time	Photodiode operation	t_f				
ISOLATION CHARACTERISTICS						
PARAMETER	SYMBOL	TIL111			UNIT	TEST CONDITIONS
		MIN.	TYP.	MAX.		
Input-to-output internal resistance	r_{io}	10^{11}			Ω	$V_{CC} = \pm 1.5 kV$
Input-to-output capacitance	C_{io}		1	1.3	pF	$V_{CC} = 0, f = 1 MHz, \text{ See Note 6}$
Isolation voltage	V_{iso}	7500			VAC-PEAK	$I_o \leq 1 \mu A, 1 \text{ minute}$
		5300			VAC-RMS	$I_o \leq 1 \mu A, 1 \text{ minute}$

Figura A.1: Características do TIL111 da FAIRCHILD

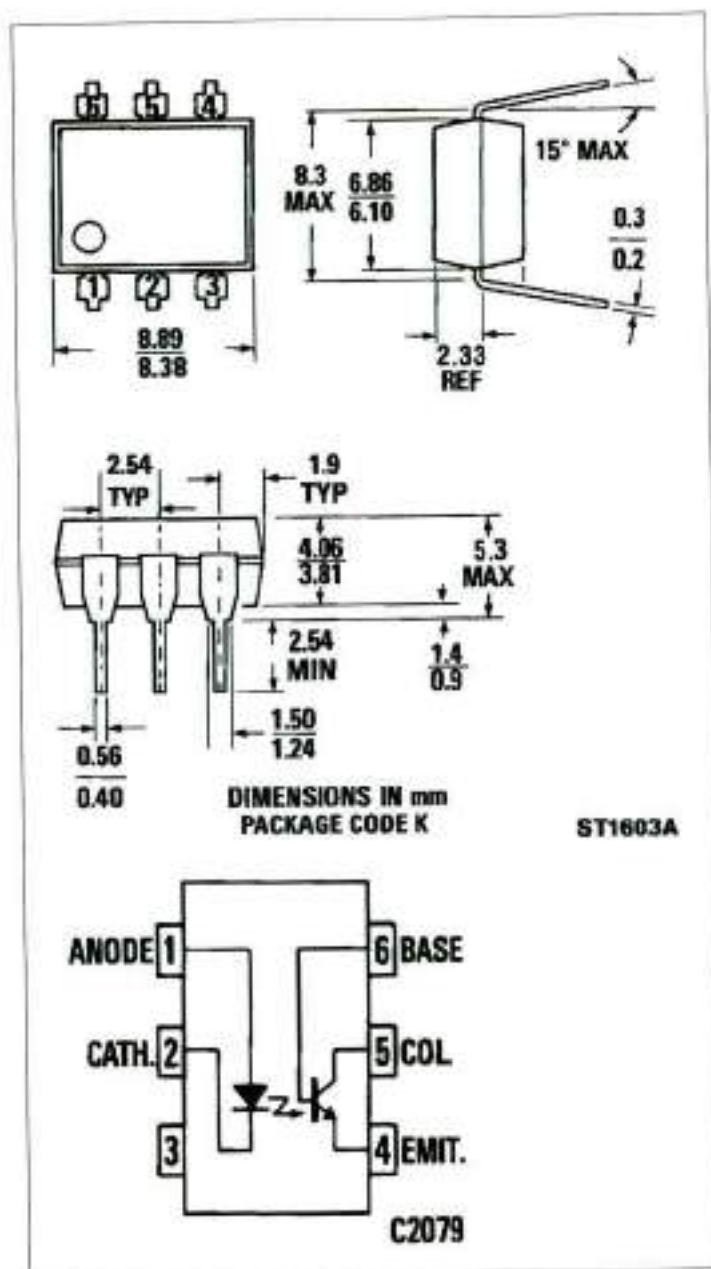


Figura A.2: Dimensões do TIL111 da FAIRCHILD

Apêndice A – Instalação do sistema Weblab

A.1 Manual de Instalação

A.1.1 Sistema Web

Pré-requisitos para a instalação do Weblab:

- Sistema Gerenciador de Banco de Dados.
- Servidor Apache Tomcat.

A.1.1.1 Sistema Gerenciador de Banco de Dados

Como em qualquer sistema de informação, o Weblab necessita de um gerenciador de banco de dados para o seu funcionamento. O sistema permite a utilização de qualquer gerenciador de banco de dados relacional. Após seguir as instruções específicas de instalação e inicialização do servidor escolhido, basta configurar os parâmetros “db.class” e “db.url”, que serão detalhados posteriormente, e inserir o *driver* correspondente na pasta WEB-INF/lib da aplicação Weblab instalada no Tomcat. Caso queira utilizar o MySQL (que foi o escolhido para o desenvolvimento do projeto) basta configurar as variáveis, pois o *driver* já está presente no pacote de bibliotecas.

Antes de iniciar o uso da aplicação Web ainda é necessário criar as tabelas no banco de dados. Para isso, execute o script “weblab.sql” de acordo com instruções específicas para o gerenciador escolhido.

A.1.1.2 Apache Tomcat

O WebLab é uma aplicação J2EE, portanto necessita de um servidor de aplicações que execute as especificações de Servlets e JavaServer Pages (JSP) da Sun Microsystems.

Existem vários servidores para esta finalidade, desenvolvidos por diferentes fabricantes, tais como o IBM WebSphere, BEA WebLogic, Sun iPlanet, Oracle Application Server, JBoss, mas este manual tratará especificamente do software de código aberto Apache Tomcat, que é referência oficial na implementação das tecnologias de Servlets e JSP.

Apesar de funcionar em diferentes sistemas operacionais, neste manual será abordada apenas a instalação do Tomcat na plataforma Linux, a qual foi utilizada no desenvolvimento deste projeto de formatura. Presume-se que o Java Development Kit (JDK), requerido pelo Tomcat, já está instalado no computador. A seguir serão descritos os passos necessários para a sua instalação utilizando o ambiente textual, ou seja, o interpretador de comandos do Linux:

1. Obtenha a versão mais recente do Tomcat já compilado (binários) no formato tar.gz. Na edição deste documento, a última versão disponível era a 6.0.14. O software pode ser obtido gratuitamente da página oficial do projeto (<http://tomcat.apache.org>).
2. Estando o arquivo na pasta desejada para a instalação, entre nesta pasta e descompacte-o. Obs.: dependendo da pasta escolhida, será preciso privilégios de superusuário para a instalação. Exemplo:

```
cd /opt
tar -xzf apache-tomcat-6.0.14.tar.gz
```

Uma nova pasta deve ter sido criada contendo os arquivos extraídos.

3. Informe ao Tomcat a localização do Java Development Kit (JDK) instalado na sua máquina. Isto é feito através da definição de uma variável de ambiente. Exemplo:

```
export JAVA_HOME=/opt/jdk1.6.0_01
```

4. Inicie o servidor simplesmente executando o script fornecido. Exemplo:

```
apache-tomcat-6.0.14/bin/startup.sh
```

5. Pronto! O servidor já está no ar. Confirme acessando através de um navegador web o seguinte endereço:

```
http://localhost:8080/
```

A.1.1.3 Aplicação Weblab

A próxima etapa é instalar a aplicação no servidor. Para isso, contamos com a ferramenta “Tomcat Manager”, que pode ser acessada a partir do *link* de mesmo nome na página inicial (<http://localhost:8080/>) ou diretamente no endereço <http://localhost:8080/manager/html>. Porém, antes precisamos definir um usuário que tenha permissões de acesso a esta ferramenta. Para isso, acrescente no arquivo “tomcat-users.xml”, localizado na pasta “conf”, as seguintes linhas entre as *tags* tomcat-users:

```
<tomcat-users>
<role rolename="manager"/>
<user username="tomcat" password="tomcat" roles="manager"/>
</tomcat-users>
```

Assim, definimos o usuário “tomcat” com senha “tomcat” que possui privilégios de gerente do Tomcat. Reinicie o servidor para que as alterações tenham efeito:

```
apache-tomcat-6.0.14/bin/shutdown.sh
apache-tomcat-6.0.14/bin/startup.sh
```

Acesse agora o “Tomcat Manager” com as credenciais definidas acima. É exibida uma lista com as aplicações já instaladas no servidor e informações sobre seu estado. Para instalar uma nova aplicação, localize a área de “Deploy”, item “WAR file to deploy”, campo “Select WAR file to upload”. Clique em “Arquivo...” e selecione o arquivo Weblab.war, que é o pacote que contém toda a aplicação Weblab juntamente com as descrições utilizadas pelo servidor para a instalação. Finalmente clique em “Deploy” e a aplicação será automaticamente instalada no servidor e já estará em funcionamento no endereço <http://localhost:8080/Weblab>.

A.1.1.4 Ajuste de Parâmetros

Para que a aplicação funcione corretamente no ambiente específico onde foi instalada, alguns parâmetros devem ser ajustados para se adequar ao banco de dados utilizado e a configurações desejadas. Altere o arquivo “weblab.properties”, na pasta “webapps/Weblab/WEB-INF/classes” (a partir da pasta de instalação do Tomcat) para se adequar à sua instalação em particular. Os seguintes parâmetros podem ser alterados:

- db.class → classe que implementa o driver de comunicação com o banco de dados;

- db.url → string contendo a localização, o nome do banco de dados, login e senha de acesso;
- webserver.ip → endereço do próprio servidor Web, para acesso externo;
- nat.file → nome do script que será gerado para fazer o redirecionamento de portas para acesso às câmeras IP;
- gerenciadorarquivo.root → local onde os arquivos dos usuários serão armazenados;
- imagens.root → local onde as imagens dos botões do painel serão armazenadas.

A.1.1.5 Inicialização

A última etapa de configuração é fazer o servidor Apache Tomcat iniciar junto com a inicialização do Linux, assim como o script gerado para o redirecionamento de portas TCP. Para isso, basta acrescentar as linhas seguintes ao arquivo que é executado na inicialização do sistema, alterando os locais de acordo com o que foi instalado. Na maioria das distribuições Linux, este arquivo é `"/etc/rc.local"`. Exemplo:

```
export JAVA_HOME=/opt/jdk1.6.0_01
/opt/apache-tomcat-6.0.14/bin/startup.sh
```

Assim, o servidor estará pronto para receber requisições logo que o sistema operacional for iniciado.

Apêndice B – Operação do sistema Weblab

B.1 Manual de Operação

B.1.1 Iniciando o Weblab

O Usuário deve acessar o Weblab primeiramente através de seu endereço eletrônico HTTP disponibilizado para o ambiente. A página será carregada e o Usuário poderá então digitar seus dados de acesso. A figura B.1 mostra a tela acesso ao sistema.



Figura B.1: Tela de login do Weblab

O Usuário que tiver acesso ao sistema, após sua confirmação receberá a tela de boas vindas e terá acesso ao menu com funções, conforme perfil do usuário (figura B.2).

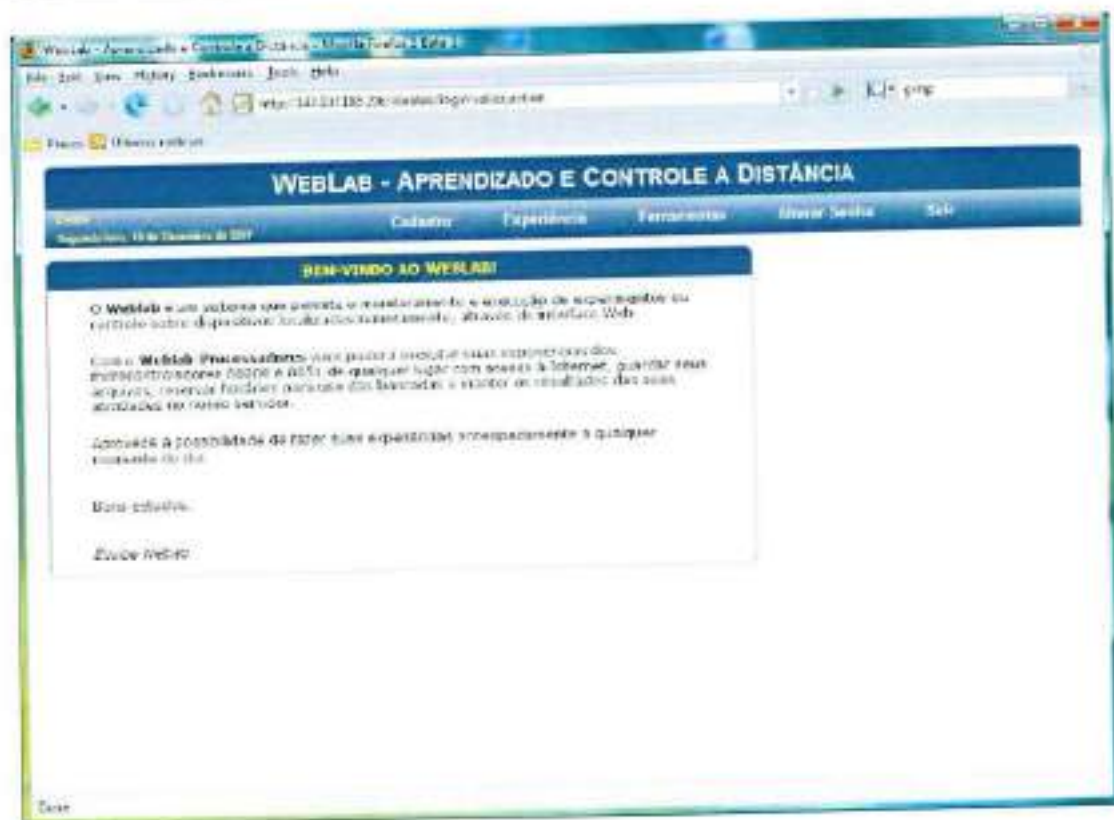


Figura B.2: Tela de boas vindas do Weblab

B.1.2 Estrutura do menu

O Weblab possui uma 5 opções no menu principal:

- Cadastro - Contém os links para as telas de cadastros
 1. Cadastro de usuários do sistema
 2. Cadastro de disciplinas
 3. Cadastro de experiências
 4. Cadastro de bancadas
 5. Cadastro de protocolos
- Experiência - Contém os links para as telas de realização das experiências
 1. Agendar experiências
 2. Fazer experiências
 3. Arquivos
- Ferramentas - Contém os links para as ferramentas administrativas

1. Painel de controle

2. Permissões

3. Parâmetros

- Alterar Senha - Opção para o usuário alterar a própria senha
- Sair - Opção para o usuário sair do sistema

Para navegar no Weblab basta passar o mouse sobre o menu (figura B.3). Dependendo de como o administrador do sistema configurou o perfil do usuário algumas opções podem não estar disponíveis.

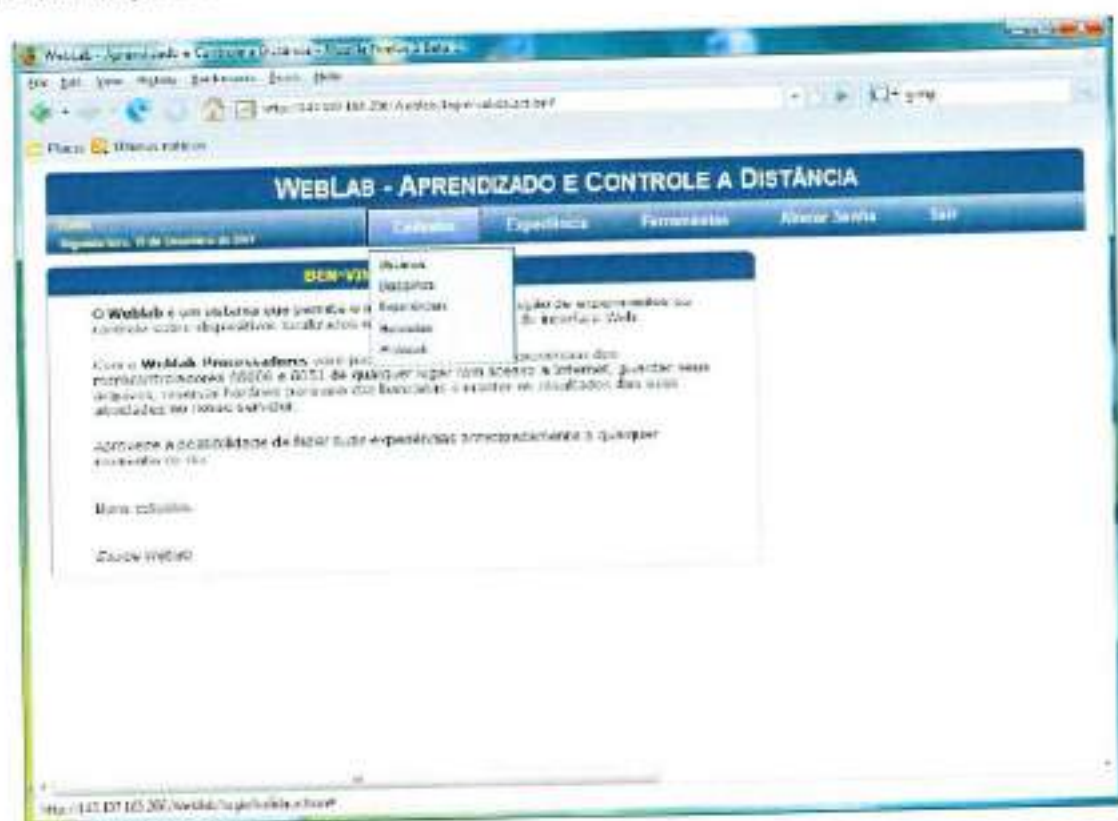


Figura B.3: Navegação sobre os menus do Weblab

B.1.3 Cadastros

Os cadastros possuem a mesma aparência e comportamento em todos os pontos. Execução feita ao cadastro de usuários que apresenta na primeira tela uma opção para realização de consulta. A figura B.4 mostra a primeira página do cadastro de usuários.

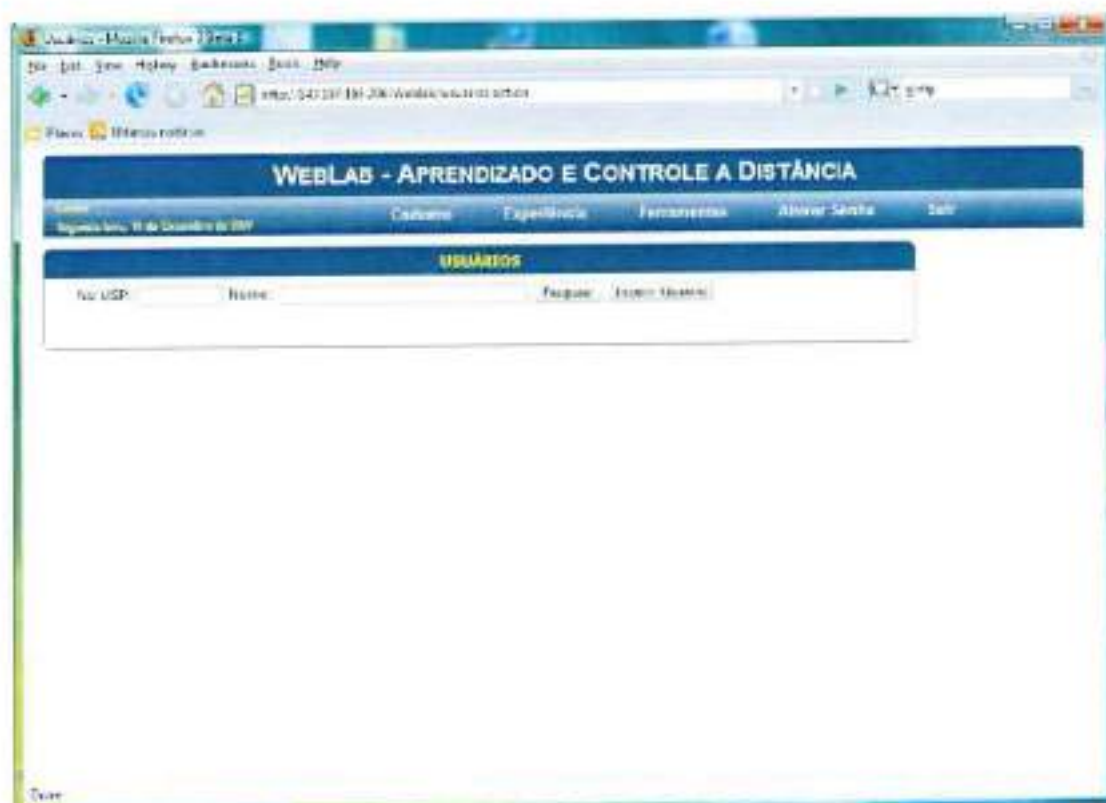


Figura B.4: Tela de login do Weblab

Para pesquisar um aluno basta digitar o número USP do mesmo, ou o seu nome (figura B.5). Se o Professor desejar visualizar detalhes sobre o usuário, ele pode clicar sobre o link disponibilizado (em azul).

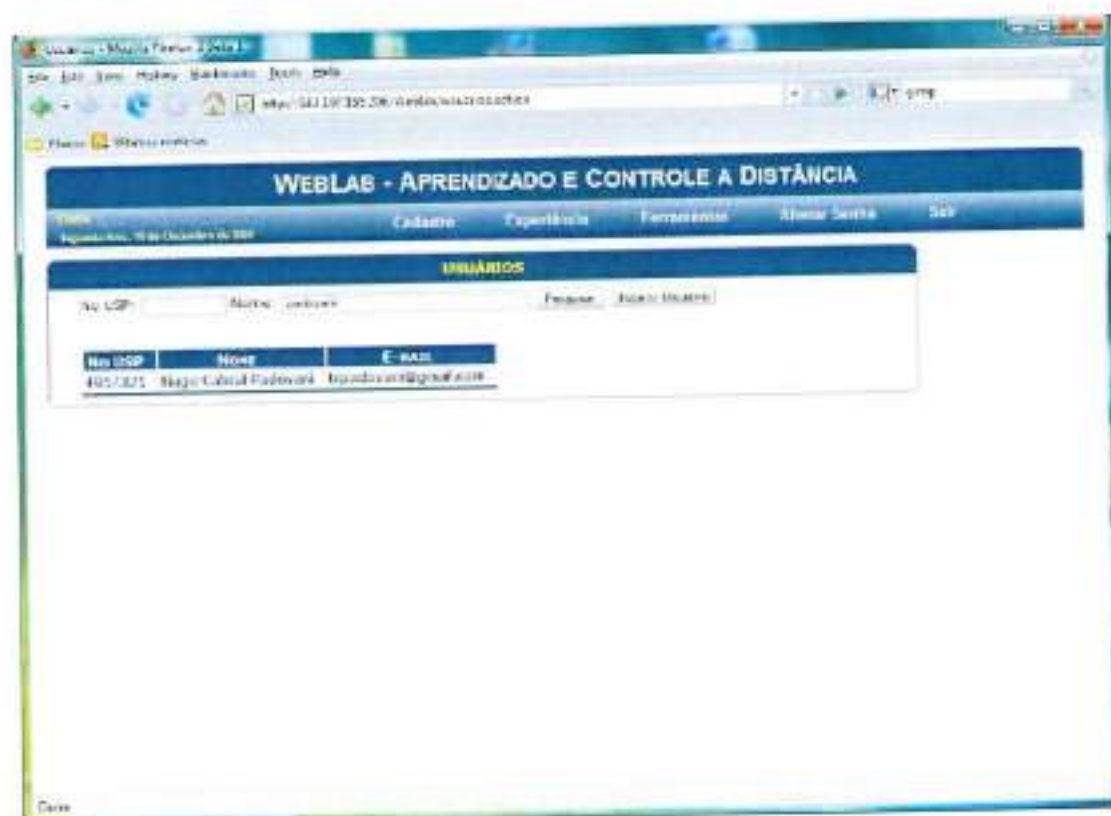


Figura B.5: Realização de consulta de um usuário

A tela de informações do usuário (figura B.6) permite que o Professor altere as informações do Aluno ou exclua o Aluno do sistema.

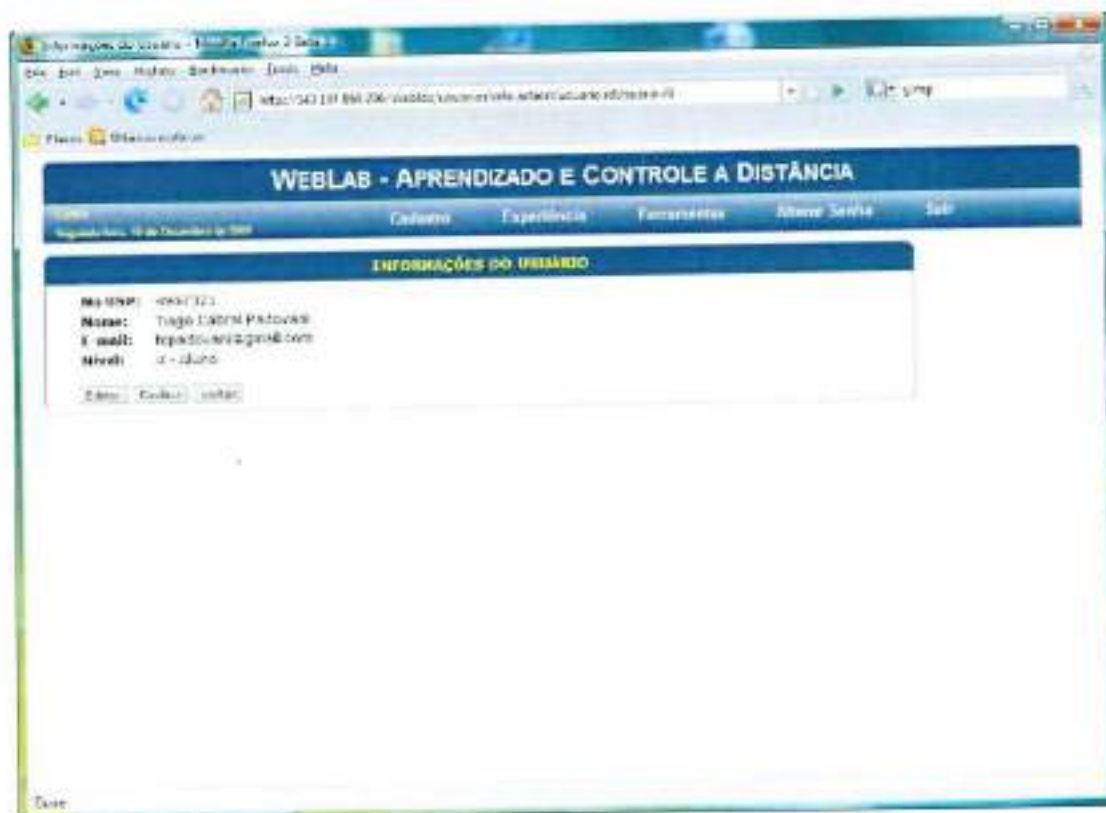


Figura B.6: Exibição dos detalhes de um usuário

Para editar as informações de um Aluno, basta clicar no botão editar e fazer as alterações (figura B.7). Ao fim delas, pode-se decidir entre confirmar as alterações ou cancelar o evento. Se as alterações forem confirmadas, o sistema emitirá tela confirmando as alterações (figura B.8). Caso aconteça um erro o sistema avisará que a transação não foi realizada (figura B.9).

Para excluir um Aluno basta clicar no botão excluir. O sistema emitirá uma mensagem de confirmação da operação (figura B.10). Se a opção sim for escolhida, o sistema emitirá uma mensagem confirmando que o usuário foi excluído com sucesso (figura B.11).



Figura B.7: Usuário sendo editado

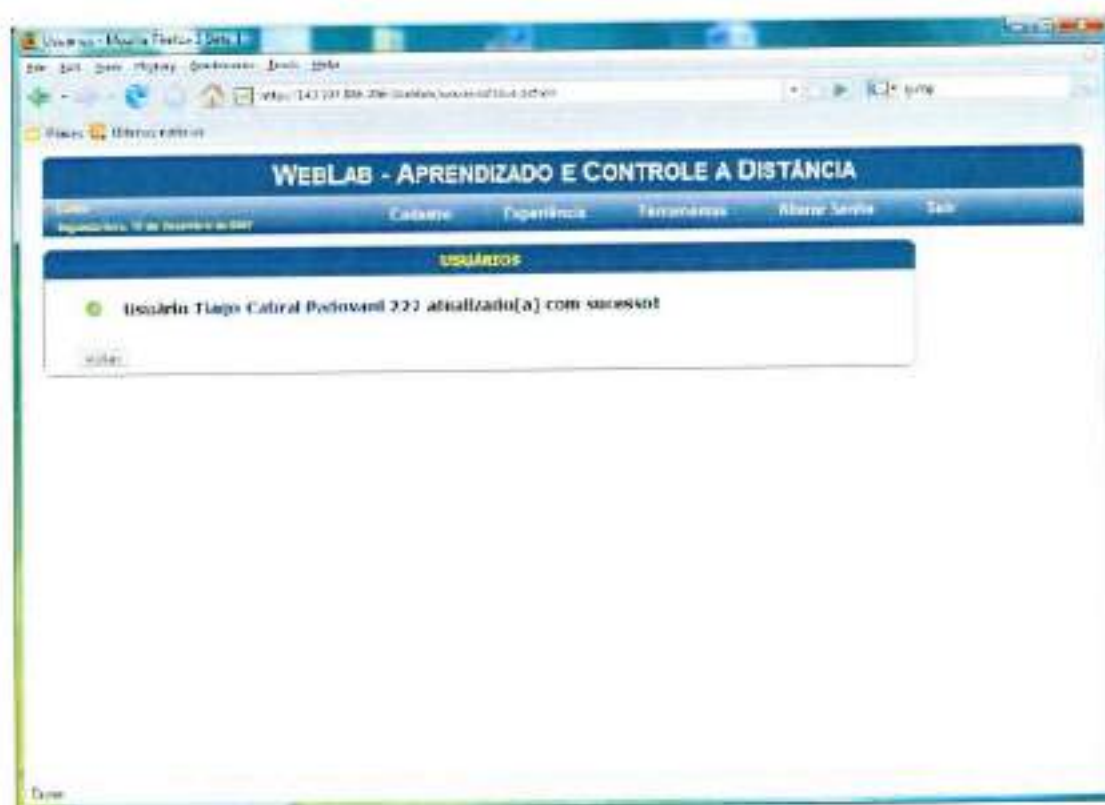


Figura B.8: Confirmação da alteração de usuário



Figura B.9: Erro na conexão com o banco de dados

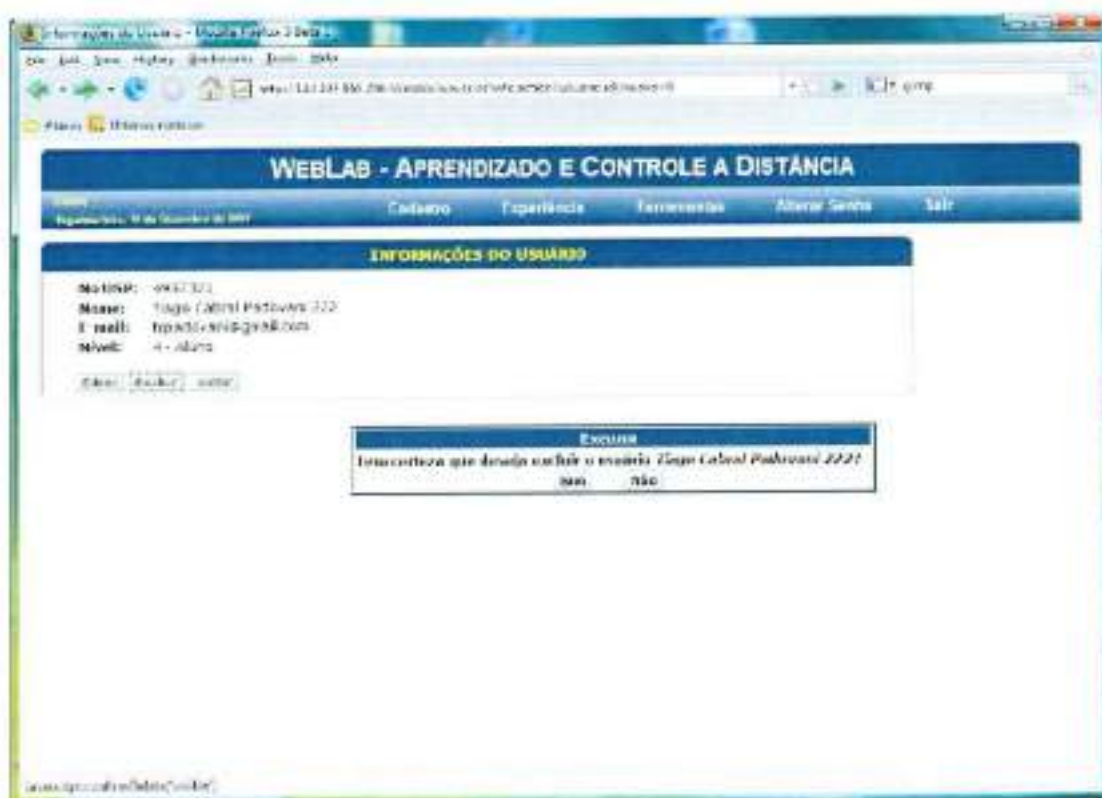


Figura B.10: Solicitação de confirmação para exclusão de usuário

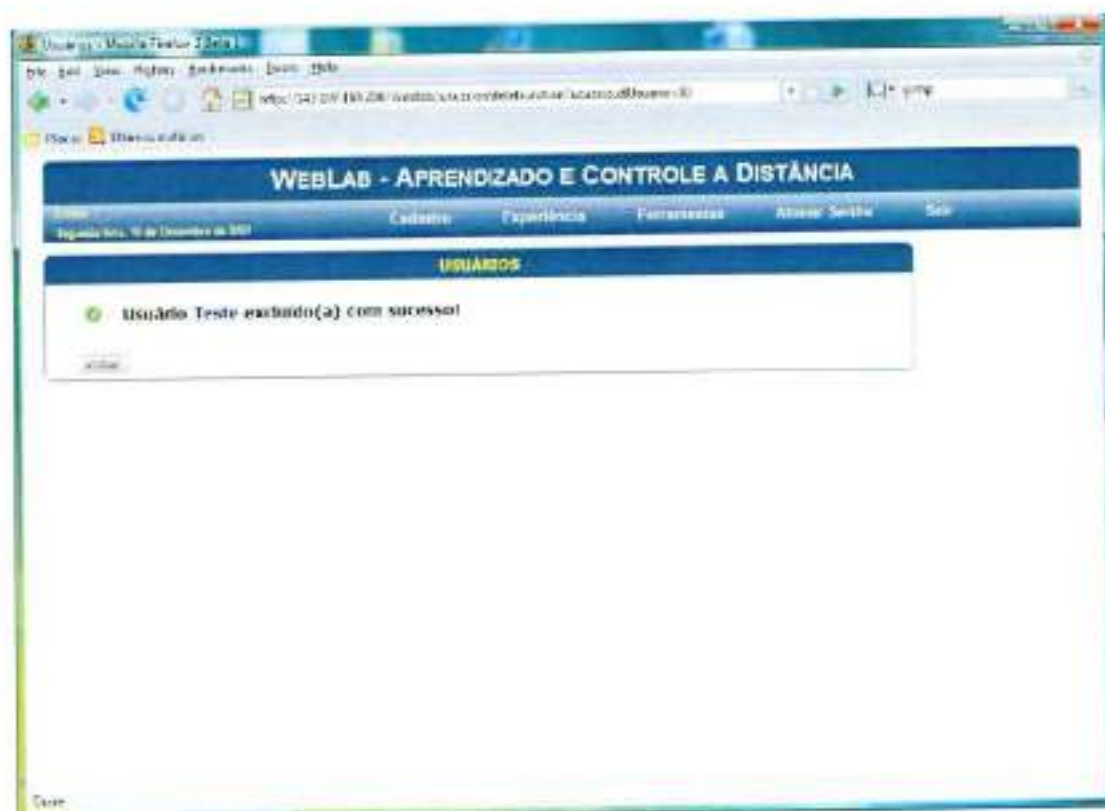


Figura B.11: Mensagem de usuário excluído com sucesso

Ao incluir um usuário basta clicar no botão incluir mostrado na figura B.4. Ao confirmar a inclusão o sistema exibirá uma mensagem de confirmação caso o preenchimento esteja correto (figura B.12).

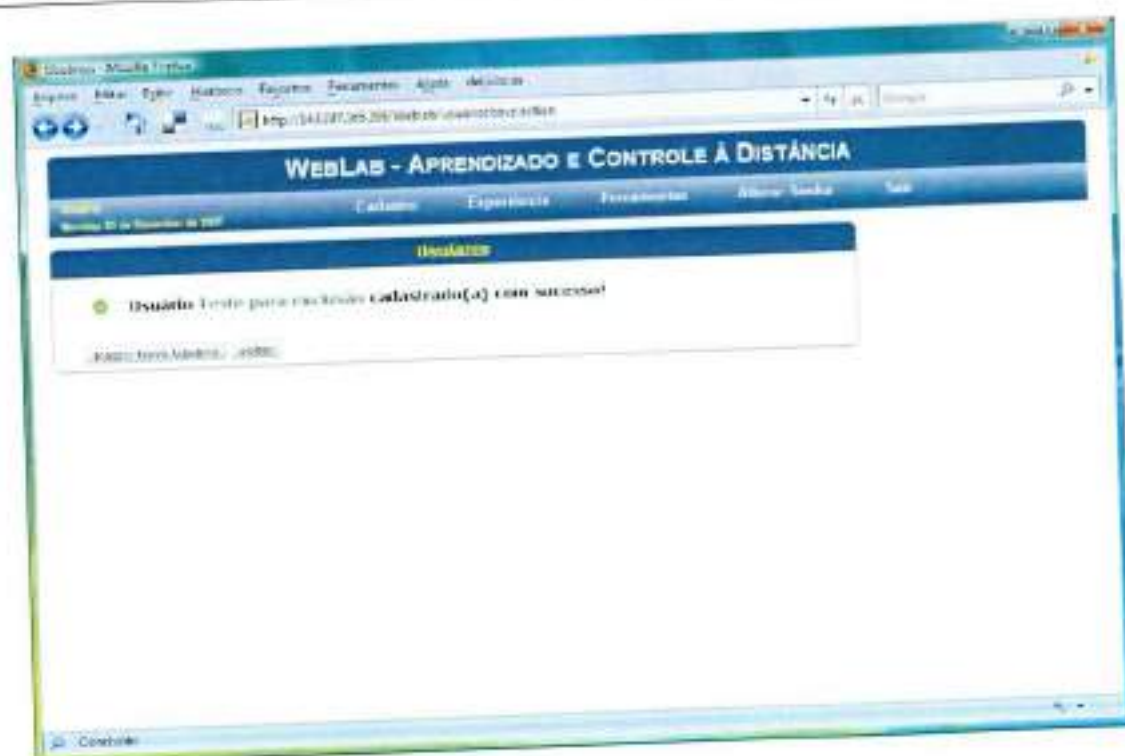


Figura B.12: Mensagem de usuário cadastrado com sucesso

B.1.3.1 Outros Cadastros

Os demais cadastros seguem um padrão bastante similar ao que acabamos de ver. As telas a seguir mostram, respectivamente, a tela de cadastro de disciplinas (figura B.13), cadastro de bancadas (figura B.14), cadastro de experiências (figura B.15) e edição de protocolos (figura B.16).

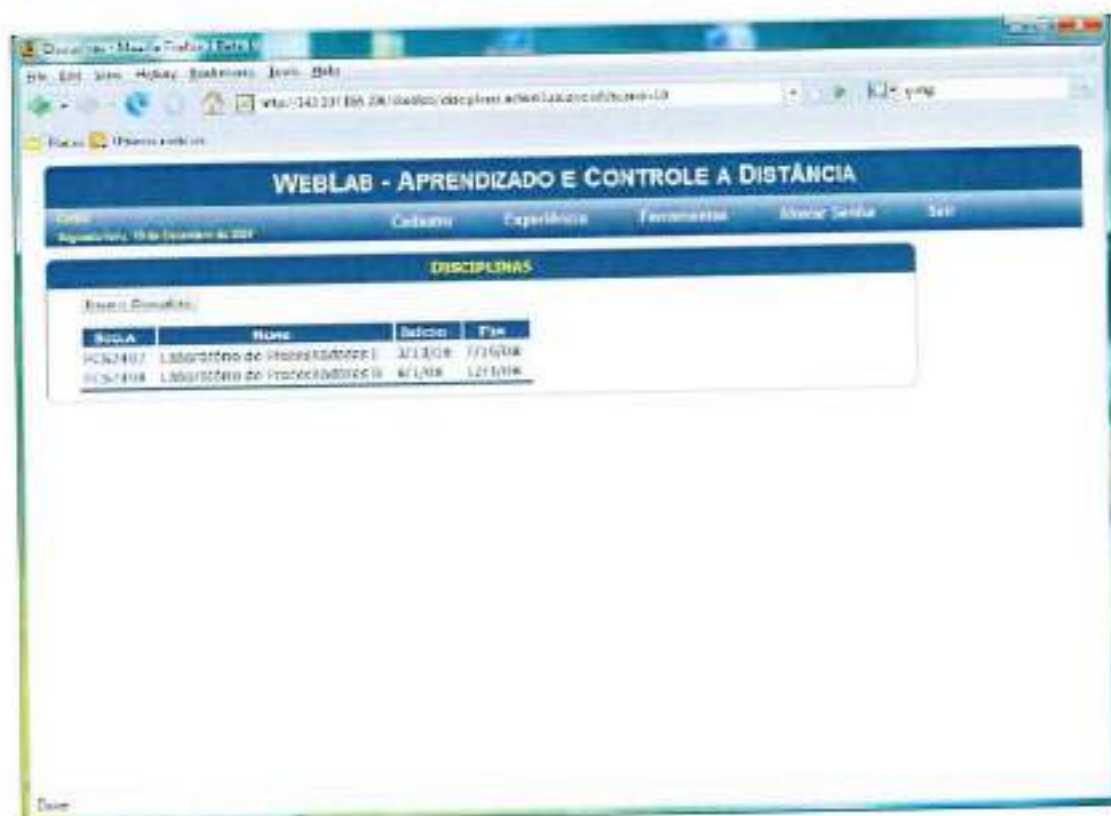


Figura B.13: Cadastro de disciplinas

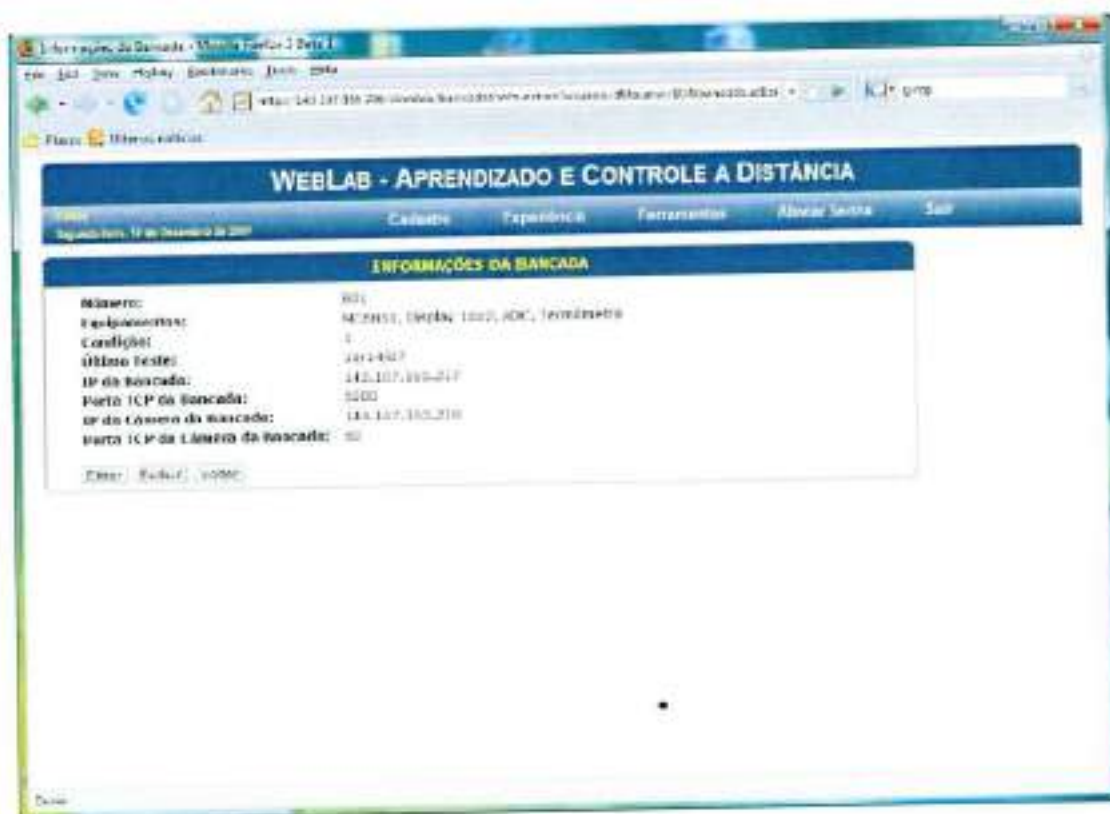


Figura B.14: Informações do Cadastro de Bancadas

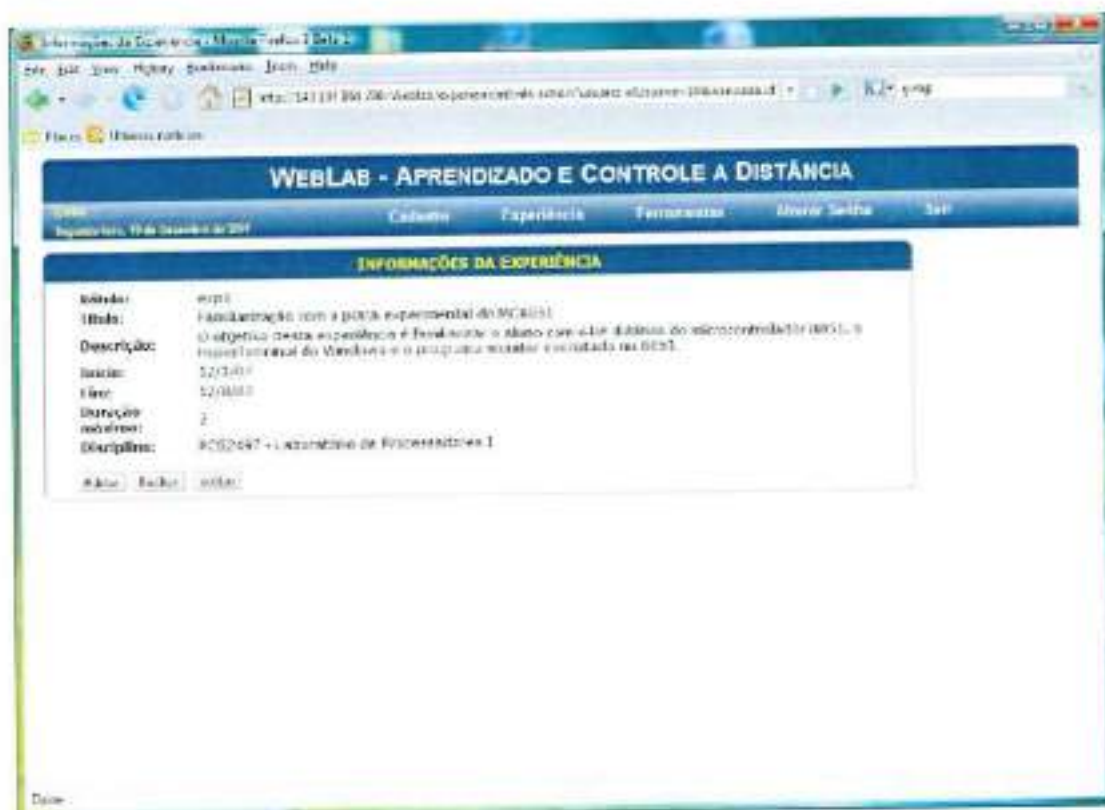


Figura B.15: Informações do Cadastro de Experiências

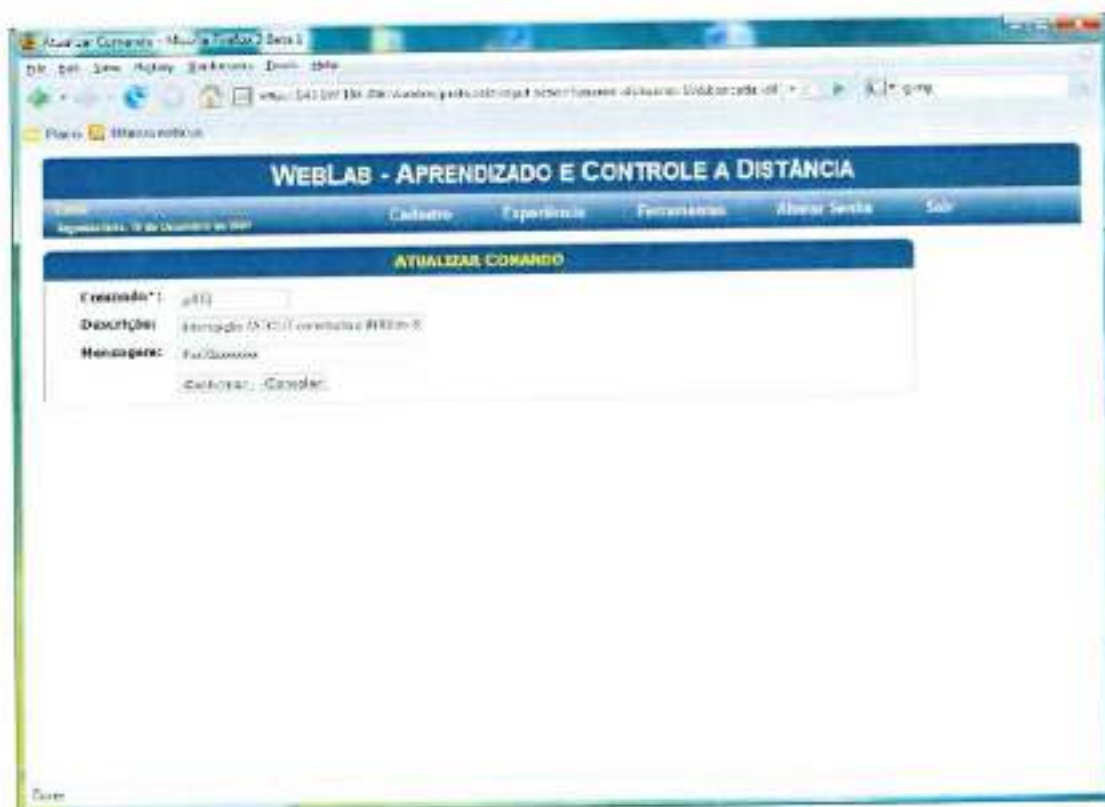


Figura B.16: Edição do Cadastro de Protocolo

B.1.4 Fazendo experiências

Para realizar uma experiência o usuário precisa ter agendado um horário, reservando uma bancada. Caso ele não tenha feito isso ao tentar executar uma experiência receberá a mensagem da figura B.17.

Para agendar um horário é preciso inicialmente escolher para qual experiência o horário será agendado, como é mostrado na figura B.18. Em seguida, o Usuário deve escolher um horário disponível e clicar em Agendar (figura B.19). Ele então receberá a mensagem da figura B.20.

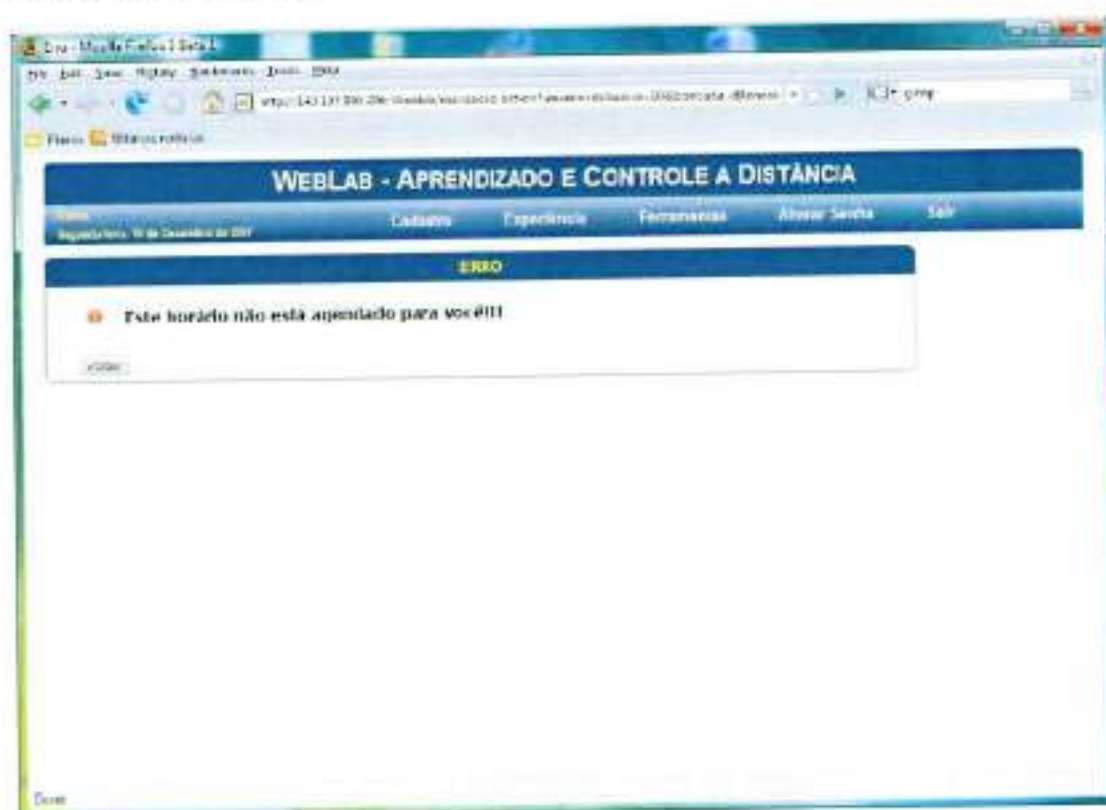


Figura B.17: Mensagem de erro no executar experiência sem agenda

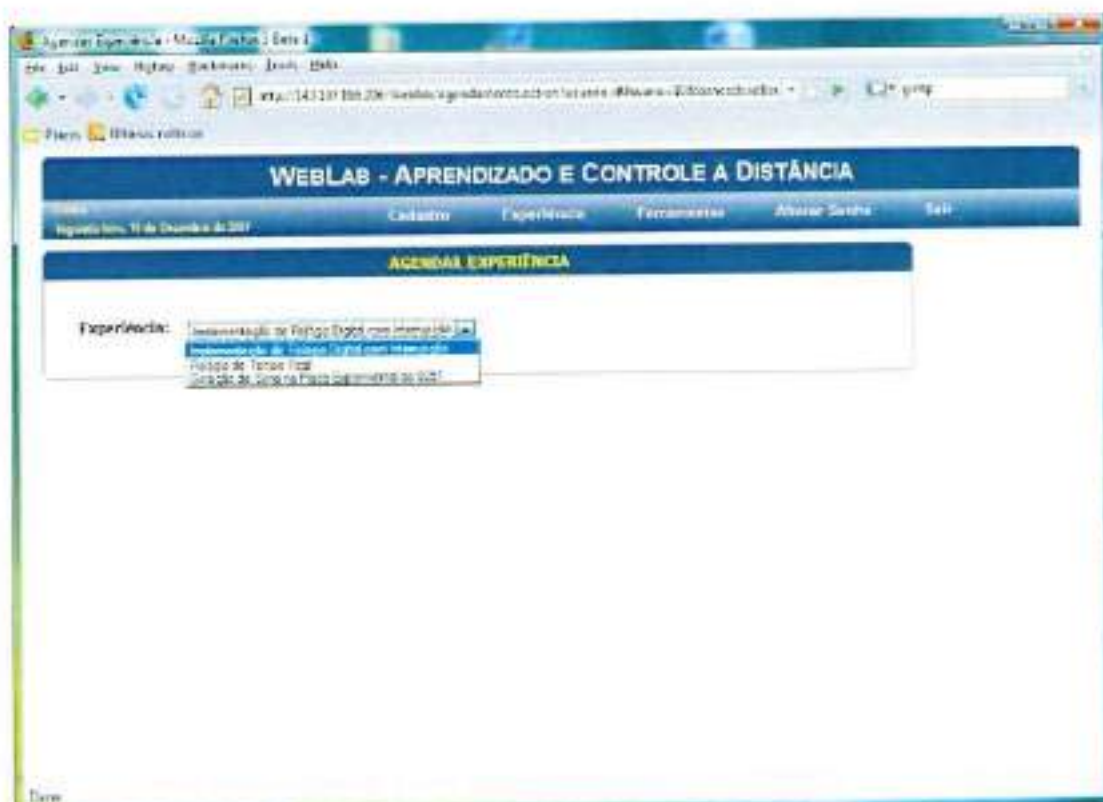


Figura B.18: Selecionando a experiência



Figura B.19: Selecionando o horário

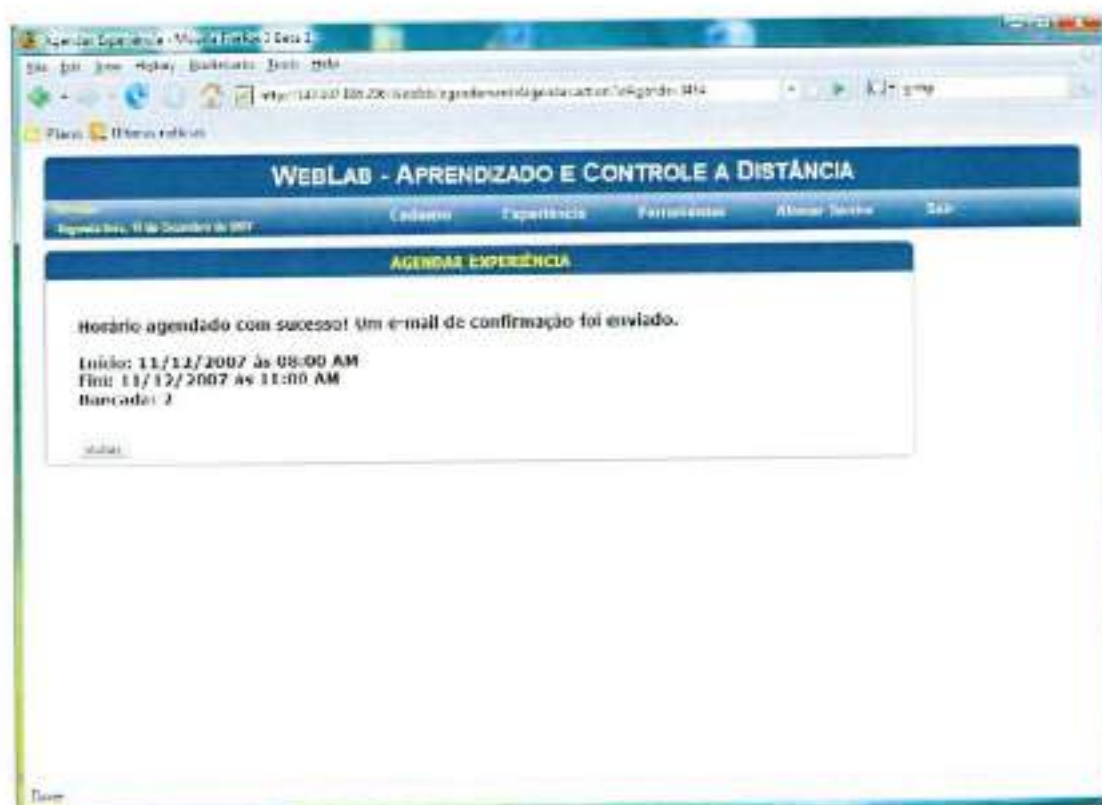


Figura B.20: Mensagem de confirmação de agendamento

Durante a experiência o usuário terá a disposição as ações que o administrador do sistema dispôs. No exemplo da figura B.21 a execução da experiência está configurada a câmera, alguns botões, uma tela de envio de comandos para o terminal (e recepção da resposta do mesmo) e o envio de arquivos .ihx para o microcontrolador. Um exemplo do uso do terminal está mostrado na figura B.22.



Figura B.21: Tela de realização de experiências



Figura B.22: Uso do terminal durante a experiência

O Usuário poderá manter os seus arquivos no servidor do Weblab. Cada usuário do sistema terá uma pasta própria para armazenamento de arquivos e só poderá acessar o próprio nível de conteúdo, ou seja, as próprias pastas. As sub-pastas serão criadas automaticamente pelo sistema a medida que as experiências são criadas. O Usuário poderá incluir arquivos, excluir arquivos e fazer download deles.

Além disso, poderá compilar arquivos remotamente, clicando no botão correspondente que está disponível para arquivos com a extensão .c. O resultado é demonstrado na própria tela e, em caso de erro, o Usuário terá acesso ao erro retornado clicando no botão Detalhes.

A figura B.23 mostra a tela de gerenciamento de arquivos do Weblab. Os casos de compilação estão mostrados nas figuras B.24 (em caso de sucesso da compilação), B.25 (em caso de erro) e B.26 (tela de detalhamento do erro). Caso o Usuário clique no arquivo, o *browser* tenta exibir o arquivo, como na figura B.27, ou simplesmente baixa o arquivo.



Figura B.23: Gerenciador de arquivos Weblab

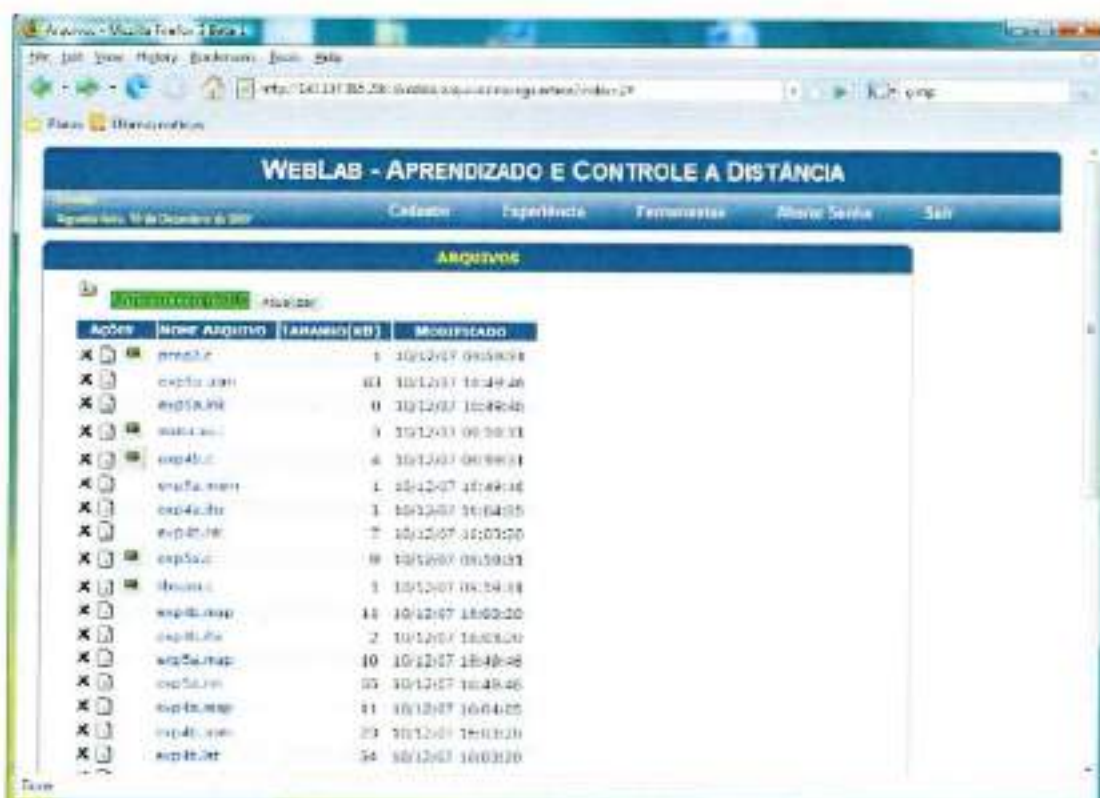


Figura B.24: Mensagem de sucesso da compilação do arquivo selecionado

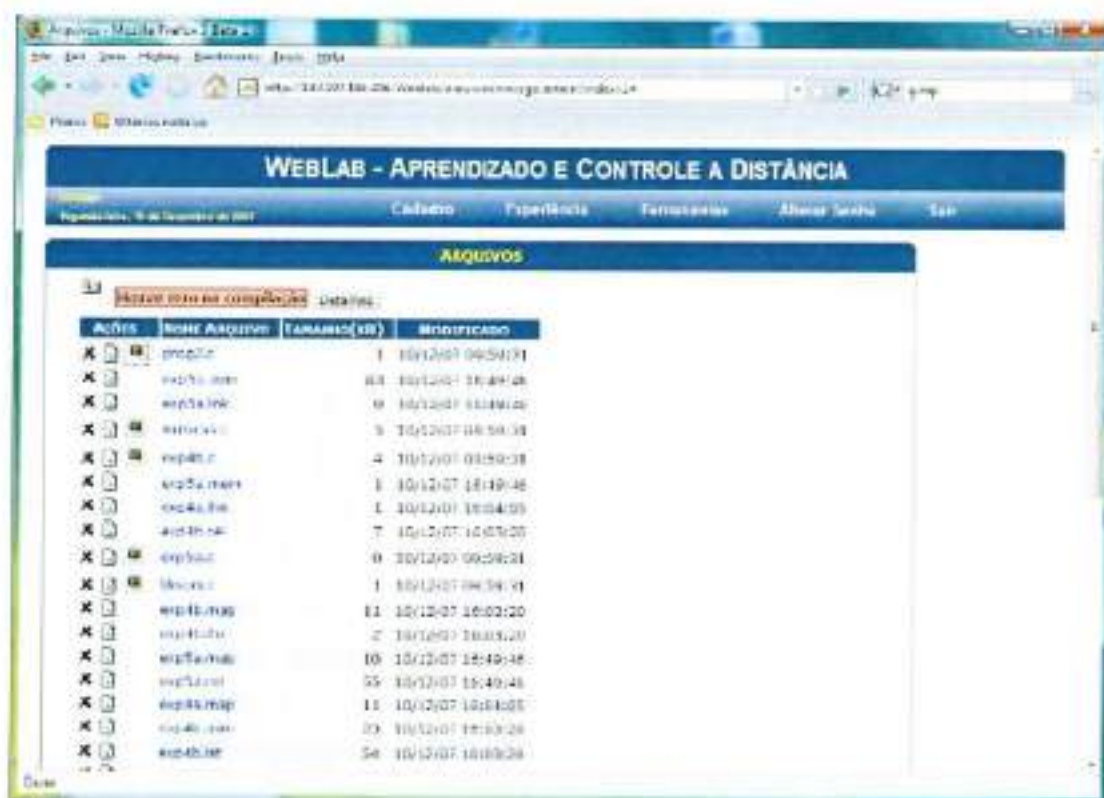


Figura B.25: Mensagem de erro na compilação



Figura B.26: Visualização de detalhes da compilação

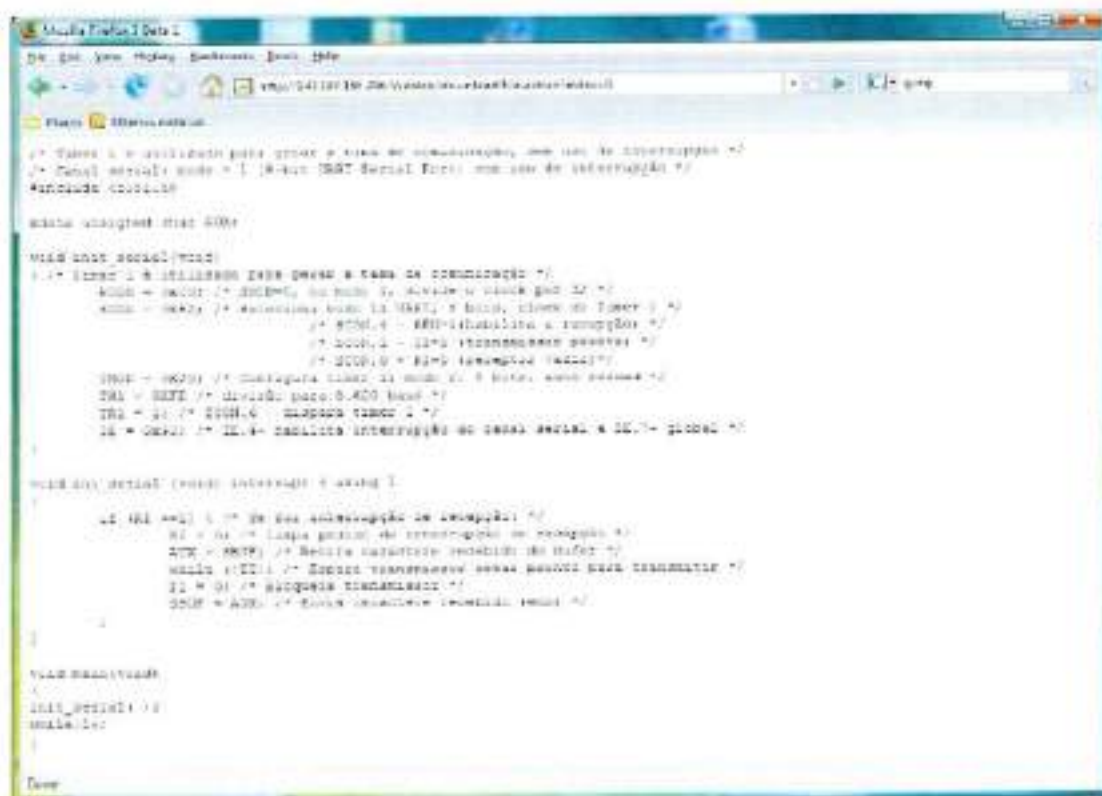


Figura B.27: Download de arquivo

B.1.5 Administrando o Sistema

B.1.5.1 Painel de Controle

Para preparar uma experiência deve-se produzir um ambiente parecido ao que o usuário encontraria no real. O Painel de controle é composto por um conjunto de painéis. Os painéis são compostos por um conjunto de imagens que podem ter várias ações: apenas clique, duplo estado (por exemplo um botão que fique pressionado) ou mesmo vários estados. Há ainda objetos pré-configurados pelo sistema como a câmera e o envio de arquivos. A figura B.28 mostra a tela de cadastro dos painéis do painel de controle.

Ao cadastrar um painel deve-se informar o tamanho desejado do painel e sua posição inicial na tela. Na tela de informações do painel também serão exibidas as imagens associadas a ele (figura B.29). Selecionando uma imagem é possível ter acesso as suas características (figura B.30).

NOME	CLASSE	POS X	POS Y	COMPRIMENTO	ALTURA
CONTROL	1	61	61	320	100
PIR	1	303	276	208	68
JP11	1	41	264	216	110
JP12	1	267	316	180	60
SUPORTE1	2	109	496	80	30
SUPORTE2	2	314	479	80	30
MESA1	0	464	525	50	50
MESA2	0	404	67	426	200
TOCADA	4	374	299	526	110

Figura B.28: Painel de Controle Weblab



Figura B.29: Informações Cadastro de Painéis

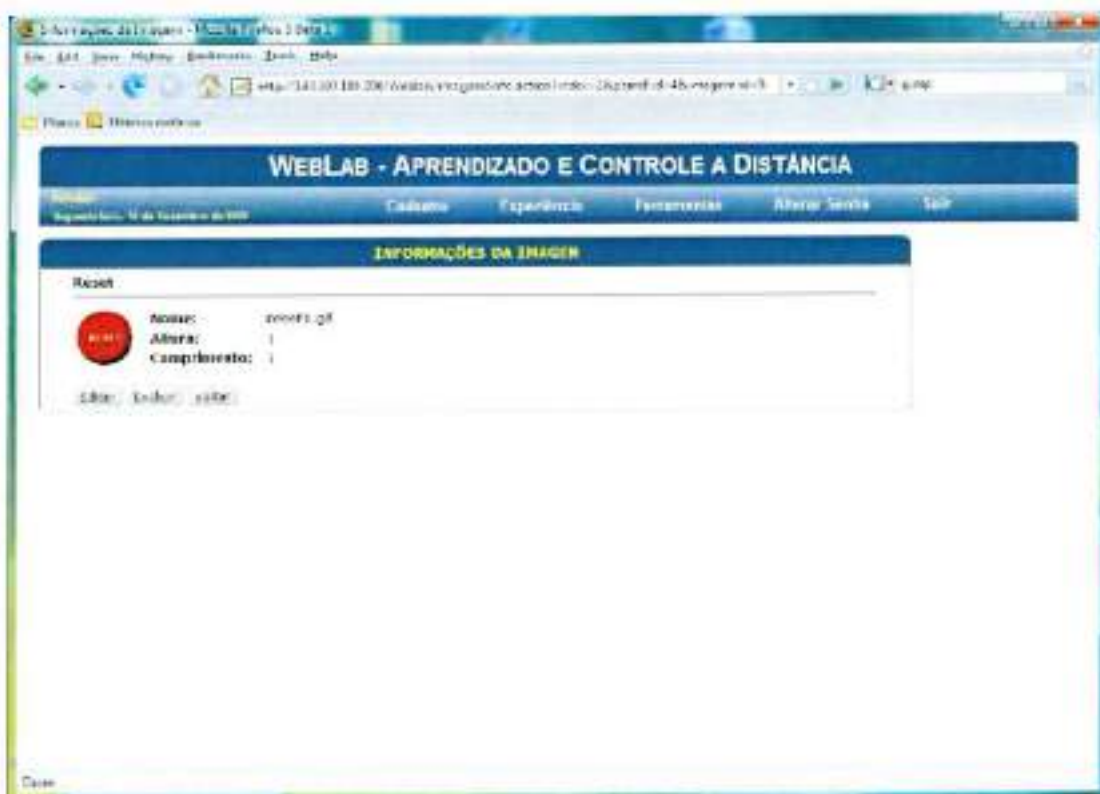


Figura B.30: Informações da Imagem

B.1.5.2 Alterar disposição

Visando melhorar a usabilidade do sistema foi criada a ferramenta de alteração de disposição dos itens da bancada. Os itens estão dispostos de acordo com os parâmetros de posição na tela definidos na seção anterior. O Usuário poderá alterar essa disposição simplesmente arrastando os itens pela tela com o mouse para a posição que achar mais conveniente, clicando em seguida no botão Salvar posições. Essa tela é mostrada na figura B.31.



Figura B.31: Alteração da disposição dos elementos na tela

B.1.5.3 Permissões

Cada usuário cadastrado tem um nível de acesso ao sistema. Esses níveis vão da visualização das páginas e até o que cada usuário pode fazer com uma informação (alterar ou excluir). A figura B.32 mostra os níveis incluídos no Weblab para o Laboratório de Processadores. A figura B.33 mostra como são gerenciadas as permissões de cada perfil.

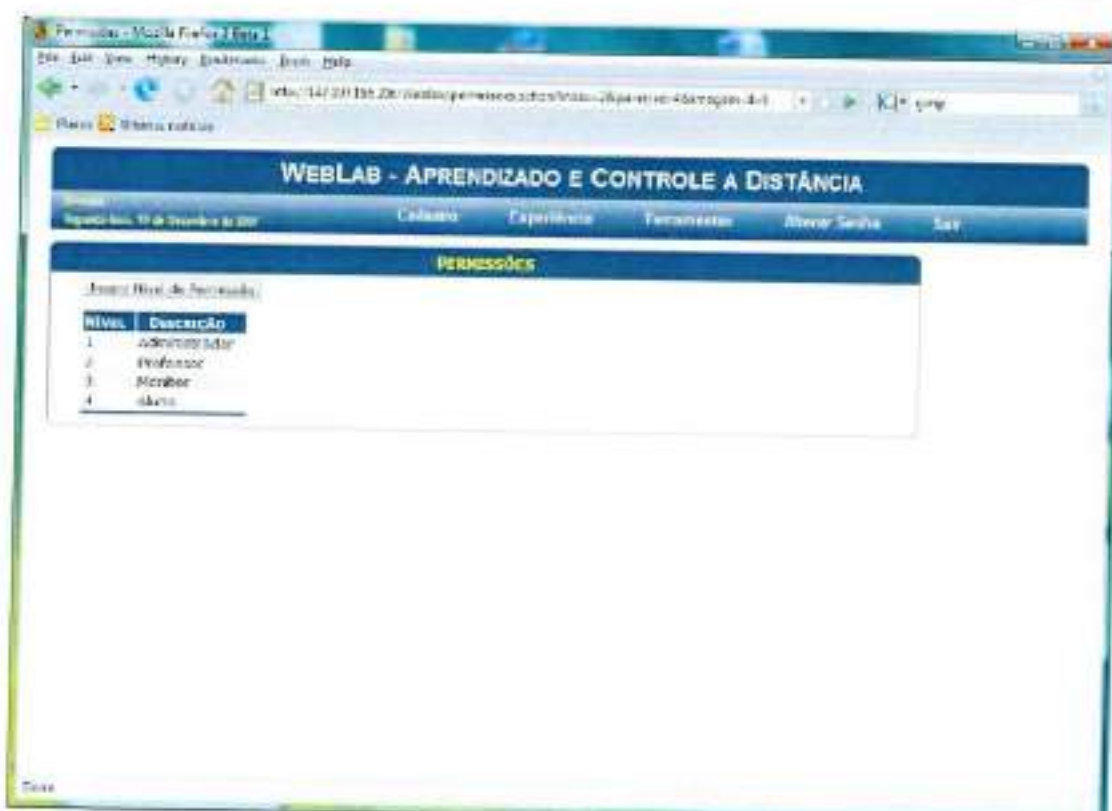


Figura B.32: Cadastro de Permissões dos Usuários

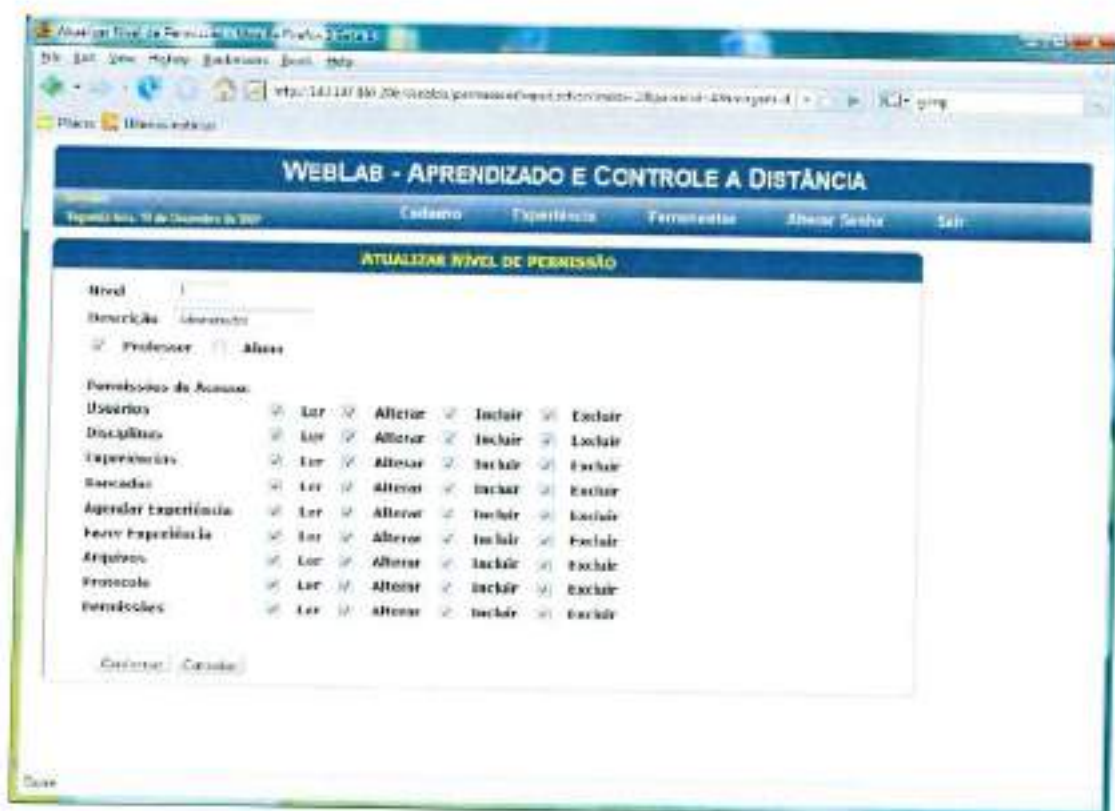


Figura B.33: Alteração das Informações das Permissões

B.1.5.4 Parâmetros

Caso o Usuário deseje, o sistema permite que o mesmo manipule algumas variáveis do sistema através dessa tela (figura B.34). Os parâmetros do nosso sistema são, por exemplo, linhas de comando do Small Device C Compiler e horário de início e fim dos agendamentos.

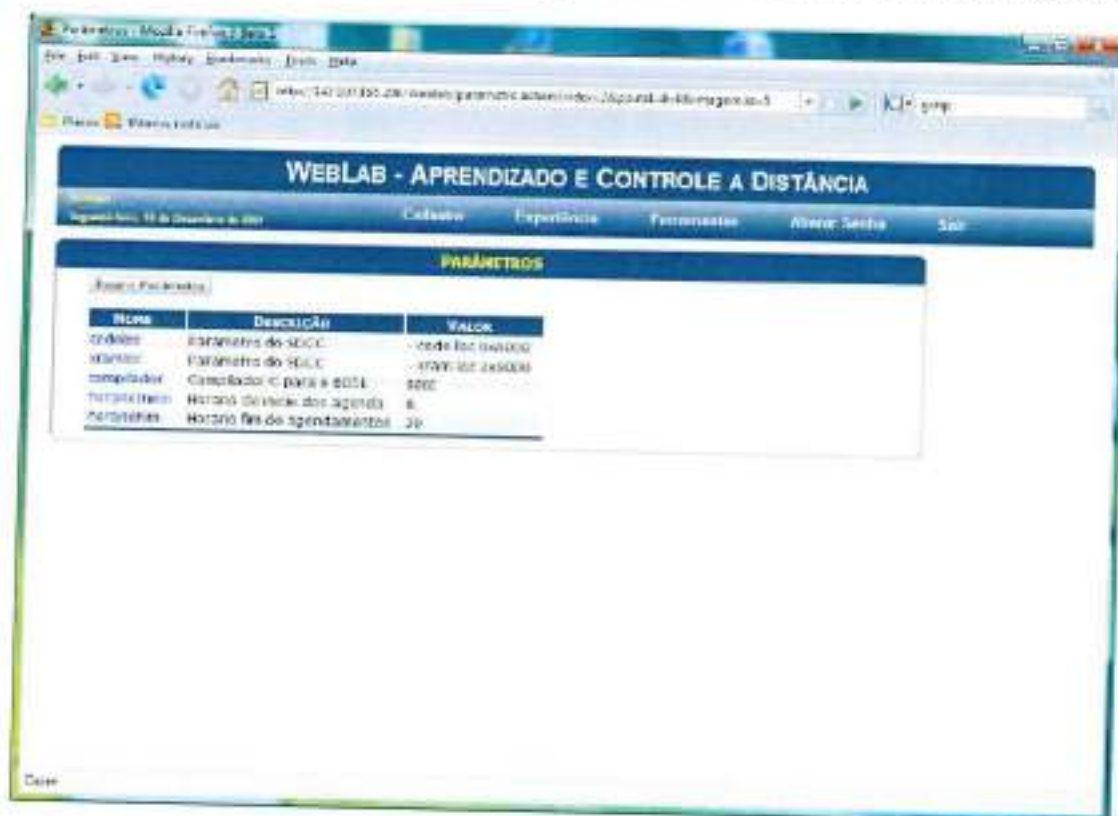


Figura B.34: Cadastro de parâmetros do sistema

B.1.6 Alterando a senha

Para alterar a própria senha basta clicar no link Alterar Senha. A figura B.35 mostra a tela de alteração de senha.

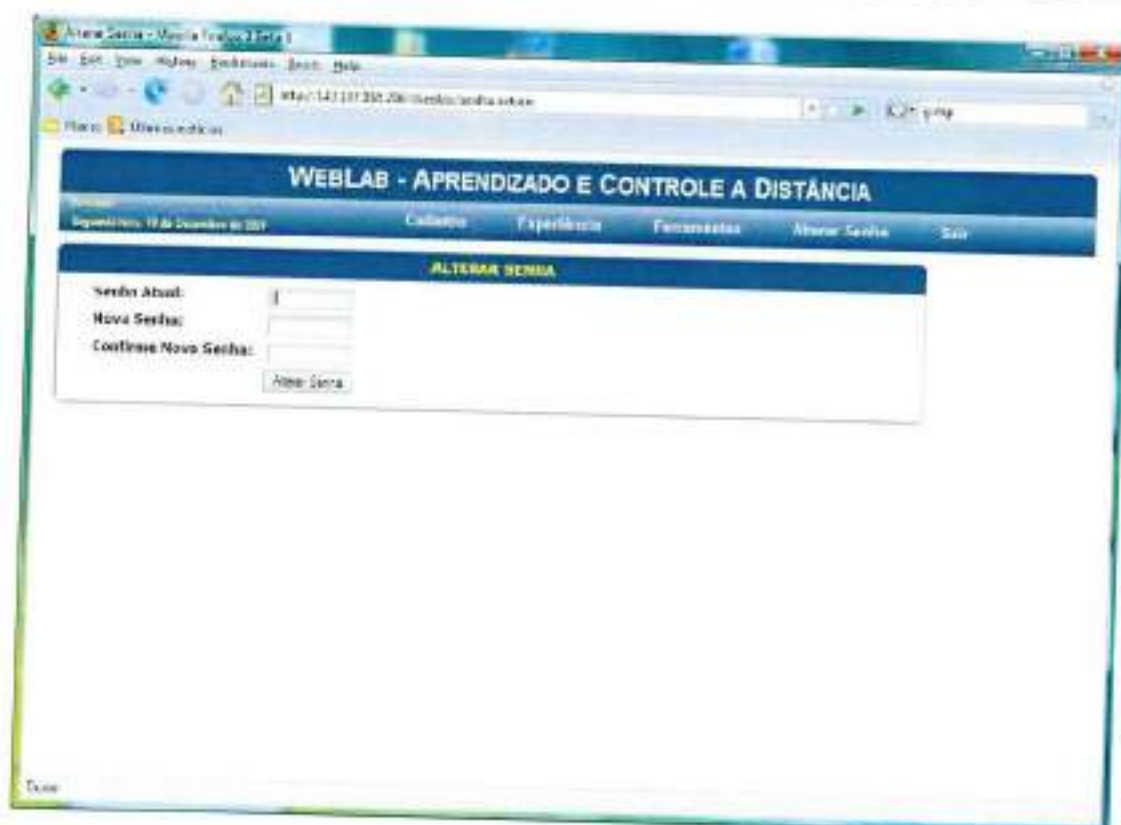


Figura B.35: Alterando a senha