

**UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**LABVAE: “UMA SOLUÇÃO PARA EXPERIMENTAÇÃO  
DE APRENDIZAGEM ELETRÔNICA”**

**MARCOS PEREIRA DOS SANTOS**

**São Cristóvão/SE**

**2014**

**UNIVERSIDADE FEDERAL DE SERGIPE**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA**  
**COMPUTAÇÃO**

**MARCOS PEREIRA DOS SANTOS**

**LABVAE: “UMA SOLUÇÃO PARA EXPERIMENTAÇÃO**  
**DE APRENDIZAGEM ELETRÔNICA”**

**Dissertação** apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal do Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

**Orientador:** Prof. Dr. Marco Túlio Chella

**São Cristóvão/SE**

**2014**

MARCOS PEREIRA DOS SANTOS

**LABVAE: “UMA SOLUÇÃO PARA EXPERIMENTAÇÃO  
DE APRENDIZAGEM ELETRÔNICA”**

**Dissertação** apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal do Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

**BANCA EXAMINADORA**

Prof. Dr. Marco Túlio Chella, Presidente  
Universidade Federal de Sergipe (UFS)

Prof. Dr. Tarcísio Rocha  
Universidade Federal de Sergipe (UFS)

Prof. Dr. Rafael Thyago Antonello  
Instituto Federal de Alagoas (IFAL)

# **LABVAE: “Uma Solução para Experimentação de Aprendizagem Eletrônica”**

Este exemplar corresponde à redação parcial da dissertação de mestrado, sendo o exame de qualificação do mestrando **MARCOS PEREIRA DOS SANTOS** para ser aprovado pela Banca Examinadora.

São Cristóvão/SE, 3 de fevereiro de 2015

---

Prof. Dr. Marco Túlio Chella  
Orientador

---

Prof. Dr. Tarcisio Rocha  
Membro

---

Prof. Dr. Prof. Rafael Thyago Antonello  
Membro

## **Agradecimentos**

Agradeço primeiramente a Deus por tudo que conquistei, que estou conquistando e tudo que conquistarei.

A minha família, por sempre me dar apoio durante esta jornada.

Ao meu professor orientador Marco Túlio Chella, por me dar a oportunidade de participar deste programa de Mestrado.

Aos professores Ricardo Salgueiro, Rogério Nascimento e Edward Davi, por seus apoios e conselhos.

A Vice-reitora da Universidade Tiradentes, Amália Maria, pelas oportunidades concedidas para prosseguimento deste desafio.

Ao meu amigo de trabalho e estagiário deste projeto de pesquisa, Flávio Henrique, por me ajudar durante uma fase importante desta dissertação.

Aos meus estimados amigos Ângela, Glauco Luiz, Luiz Otávio, Luana Barreto, Vana Hilma, Exson Machado, Fábio Gomes Rocha, Tiago, Rafael, Marcelo Maynard e Marlon Prata por toda a ajuda prestada.

E a todos que de maneira direta ou indireta contribuíram para este trabalho.

Deixo meu sincero muito obrigado.

Diante da importância dos laboratórios para as aulas práticas dos estudantes das áreas de engenharia e computação e do fato de que há um custo considerável e um processo burocrático para manter esses laboratórios em funcionamento, esta pesquisa propõe a criação de um laboratório de acesso remoto, que possibilitará a realização dos experimentos de forma remota, com segurança e redução de custos de investimentos. Esta dissertação apresenta a concepção, desenvolvimento e avaliação de uma plataforma em *hardware* e *software* denominada Laboratório Virtual de Aprendizagem Eletrônica (LABVAE), visando mostrar a aplicabilidade do laboratório remoto, designada ao ensino e ao trabalho colaborativo nos cursos de ciência da computação, sistemas de informação, redes de computadores, mecatrônica e às engenharias. Testes foram realizados em nível de *hardware* e *software* para a validação da plataforma, bem como apresentados os resultados de uso dos recursos de rede que possibilitaram o levantamento necessário para estimar as necessidades e características mínimas para sua funcionalidade. No que diz respeito aos requisitos de eficiência dos resultados e de avaliação do desempenho, as tecnologias apresentadas foram satisfatórias, demonstrando que o LABVAE faz um bom uso da rede, a exemplo de um aproveitamento de 97,57% do total de pacotes transmitidos; uma média de 23,64 pacotes por segundo; uma média de tamanho de 875 bytes por segundo e uma largura de banda de 24,28Mbps. Considerando que o desenvolvimento da arquitetura proposta é aplicável dos pontos de vista de usabilidade, disponibilidade e funcionalidade, pode-se confirmar a hipótese deste estudo, segundo a qual a plataforma do laboratório virtual, projetada para viabilizar o conceito prático de laboratório remoto, promove a colaboração na pesquisa prática e no ensino de experimentos remotos.

**Palavras-chave:** Laboratório Remoto, Experimentos, Rede, *Arduino*, Aprendizagem, Controle, LABVAE.

## ABSTRACT

---

Due to the importance of laboratories for practical classes for students of engineering and computing, and because there is a considerable cost and bureaucratic process to keep these labs in operation, this research proposes the establishment of a laboratory for remote access, which will enable the experiments remotely, securely and cost reduction investments. Presents the design, development and evaluation of a hardware and software platform named Virtual Laboratory for E-Learning (LABVAE), to show the applicability of the remote laboratory, designated to education and collaborative courses in computer science, information systems, computer networks, to engineering and mechatronics. Tests were conducted in hardware and software for the validation of the platform and presented the results of use of network resources that enabled the survey necessary to estimate the needs and minimum features for their functionality. Regarding efficiency requirements and the results of performance evaluation, the technologies presented were satisfactory, for the LABVAE makes good use of the network, such a utilization of 97.57% of the total packets transmitted; 23.64 an average of packets per second; an average size of 875 bytes per second and a bandwidth of 24.28Mbps. Whereas the development of the proposed architecture is applicable from the points of view of usability, availability and functionality, you can confirm the hypothesis of this study, according to which the platform of virtual lab, designed to facilitate the practical concept of remote lab, promotes collaboration in research practice and in the teaching of remote experiments

**Keywords:** Laboratory Remote, Experiments, Network, Arduino, Learning, Control, LABVAE.

## LISTA DE FIGURAS

---

Figura 1:	Troca de Mensagem (adaptado de <i>WebSocket-Sample</i> ).....	8
Figura 2:	Componentes do Frame <i>WebSocket</i> .....	9
Figura 3:	<i>Bitmap x Outline</i> .....	11
Figura 4:	Proposta da arquitetura em camadas do LABVAE.....	20
Figura 5:	Diagrama de classe.....	21
Figura 6:	Diagrama de classes: laboratórios, usuários e reservas.....	22
Figura 7:	Diagrama de Sequência – Usuário.....	24
Figura 8:	Sequência do protocolo <i>WebSocket</i> .....	26
Figura 9:	Troca de mensagem HTTP <i>x WebSocket</i> .....	27
Figura 10:	Componentes do frame <i>WebSocket</i> .....	27
Figura 11:	Placa de micro controlador PIC modelo 18F2550.....	28
Figura 12:	Placa <i>arduino</i> UNO.....	29
Figura 13:	Potenciômetro acoplado ao servo motor.....	29
Figura 14:	Diagrama de Implantação.....	30
Figura 15:	Diagrama de Pacotes.....	31
Figura 16:	Diagrama de Sequência – Administrador.....	32
Figura 17:	Caso de Uso do LABVAE (Administrador).....	34
Figura 18:	Tela de administração do site.....	35
Figura 19:	Caso de Uso do LABVAE (Usuário).....	35
Figura 20:	Fluxo de dados camada de <i>middleware</i> .....	36
Figura 21:	Fluxo de dados páginas Web.....	37
Figura 22:	Interface IDE <i>arduino</i> .....	38
Figura 23:	Tela inicial dos experimentos.....	39
Figura 24:	Tela de transferência do arquivo programado pelo estudante.....	40
Figura 25:	Imagem da WebCam exibida na tela inicial da aplicação.....	41
Figura 26:	Simulação de Funcionamento de leds usando SVG.....	41
Figura 27:	Fluxo de dados WebCam.....	42
Figura 28:	Fluxo de dados Placa de Controle.....	44
Figura 29:	Fluxo de dados Placa de <i>Arduino</i> .....	45
Figura 30:	Modelo de testes para a RFC 2544.....	47
Figura 31:	Diagrama do Ambiente de Medição.....	49
Figura 32:	Vazão (LABVAE).....	50

---

Figura 33:	Utilização de ferramenta similar ao LABVAE.....	51
Figura 34	Avaliação do domínio sobre o <i>arduino</i> .....	52
Figura 35:	Avaliação do conhecimento sobre linguagem de programação C.....	52
Figura 36:	Avaliação do LABVAE por comparação com os itens descritos.....	53
Figura 37:	Satisfação relacionada aos experimentos disponíveis no LABVAE.....	53

## LISTA DE SIGLAS

---

API	<i>Application Programming Interface</i>
DAQ	<i>Data Acquisition</i>
DNS	<i>Domain Name System</i>
FPGA	<i>Field-Programmable Gate Array</i>
HTML	<i>HyperText Markup Language</i>
IDL	<i>Interface Definition Language</i>
IETF	<i>Internet Engineering Task Force</i>
LABVAE	Laboratório Virtual de Aprendizagem Eletrônica
SLD	<i>Distance Laboratory System</i>
LAR	Laboratório de Acesso Remoto
OpenCV	<i>Open Source Computer Vision Library</i>
PBL	<i>Problem Based Learning</i>
RA	<i>Remote Access Laboratory</i>
REDLART	<i>ReconFigurable Digital Laboratory for Advanced Research and Teaching</i>
RFID	<i>Radio-Frequency IDentification</i>
SLD	Sistema de Laboratório a Distância
SVG	<i>Scalable Vector Graphics</i>
TIC	Tecnologia da Informação e Comunicação
USB	<i>Universal Serial Buss</i>

## SUMÁRIO

---

1	Introdução .....	1
1.1	Definição do Problema .....	3
1.2	Hipótese .....	3
1.3	Objetivos.....	4
1.3.1	Objetivo Geral.....	4
1.3.2	Objetivos Específicos.....	4
1.4	Justificativa .....	4
1.5	Metodologia.....	4
1.6	Organização da Dissertação.....	5
2	Fundamentação Teórica.....	7
2.1	Cliente Web .....	7
2.2	Protocolo HTTP.....	8
2.3	<i>JavaScript</i> .....	8
2.4	SVG .....	9
2.5	<i>Python</i> .....	10
2.6	<i>WebSocket</i> .....	11
2.7	Camada de <i>Middleware</i> e Tecnologias.....	13
2.8	Camada de Arquitetura de <i>Hardware</i> e Rede.....	15
2.9	Trabalhos Relacionados.....	17
2.9.1	Trabalhos correlatos quanto ao desenvolvimento de solução de laboratório de acesso remoto.....	17
2.9.2	Trabalhos correlatos quanto à motivação relacionada ao uso de laboratório de acesso remoto.....	20
3	A Solução LABVAE .....	23
3.1	Arquitetura.....	23
3.1.1	Camada Cliente .....	26
3.2	Modelagem .....	29
3.3	Funcionamento .....	33
3.2.1	Páginas Web.....	37
3.2.2	Programar <i>Arduino</i> .....	38
3.2.3	Aquisição <i>WebCam</i> .....	42

3.2.4 Aquisição Placa de Controle .....	43
3.2.5 Aquisição <i>Arduino</i> .....	44
3.3 Coleta e Análise de Tráfego .....	45
3.3.1 Perfil de Tráfego .....	46
3.3.1.1 Análise de aproveitamento de pacotes .....	46
4 Avaliação do LABVAE.....	48
4.1 Avaliação Quantitativa .....	48
4.2 Avaliação Qualitativa .....	50
5 Análise dos Resultados dos Questionários .....	51
6 Conclusões.....	54
6.1 Contribuições.....	55
6.2 Trabalhos Futuros .....	55
Referências .....	57
Apêndice.....	61

---

## 1 Introdução

Este capítulo apresenta uma visão geral, os objetivos geral e específicos, bem como uma contextualização que remete aos trabalhos relacionados, condizentes ao desenvolvimento de um estudo de caso para o ensino e pesquisa em laboratórios virtuais de aprendizagem. Também são apresentadas justificativa, metodologia e a organização do presente trabalho.

Tanto a formação teórica baseada em aulas tradicionais, quanto à experiência de fazer na prática proporcionada pelos laboratórios, são imprescindíveis na formação do profissional de engenharia (HUA; GANZ, 2003).

A grande dificuldade, no entanto, é disponibilizar e organizar configurações de laboratório disponíveis em períodos extensivos de tempo, o que acarreta um investimento financeiro elevado para a manutenção desses laboratórios. Nesse caso, o uso de um laboratório remoto possibilita a melhoria na utilização dos recursos, uma vez que estará disponível num tempo de 24 por 7 (24 horas durante os 7 dias da semana), podendo ser utilizado por qualquer outra instituição de ensino e abranger um maior número de alunos, que podem realizar seus experimentos simultaneamente em locais e horários distintos.

Analisando pelo viés econômico, observa-se que, por não ser necessária a presença do aluno dentro do laboratório, esse espaço pode ter sua área física reduzida, o que, conseqüentemente, impactará na diminuição dos custos de manutenção. Outro ponto a ser considerado é a disponibilização do uso de equipamentos e de tecnologias complexas para os alunos, provendo a realização de seus experimentos na prática, para que vivenciem a essência da teoria estudada, agregando, dessa forma, valor à sua formação profissional. Quando se disponibiliza o acesso para a realização de experimentos que requerem equipamentos de alto custo, sensíveis e frágeis, a faixa de utilização desses equipamentos pode ser expandida, levando, de determinada forma, a uma economia de escala (SCHAFER; SEIGNEUR; DONELLY, 2002).

Fazendo uma análise pelo viés pedagógico, acredita-se que, da mesma forma que um conteúdo que pode ser aprendido dentro de uma sala de aula, também pode ser disponibilizado usando as Tecnologias da Informação e Comunicação (TICs) em ambiente virtual de aprendizagem, dentro de uma abordagem divergente daquela praticada na sala de aula tradicional, pois, de acordo com Silva (2014, p. 01): “O professor propõe o conhecimento. Não o transmite. Não o oferece à distância para a recepção audiovisual ou

“bancária” (sedentária, passiva), como criticava o educador Paulo Freire”. O autor prossegue enfatizando que: “A maneira do design de *software* interativo, ele [o professor] constrói um conjunto de territórios a explorar, não uma rota”, de modo que, qualquer que seja o ambiente de ensino-aprendizagem, “a sala de aula socializa liberdade, diversidade, diálogo, cooperação e co-criação quando tem sua ‘materialidade da ação’ baseada nestes mesmos princípios” (SILVA, 2014, p. 01).

A experiência com a prática de laboratório deve integrar o aluno com o uso de equipamentos, condutas laboratoriais e possibilitar o trabalho colaborativo para explorar conceitos teóricos. Esses elementos são consideráveis para viabilizar a boa formação de profissionais capacitados para o mercado de trabalho.

Pode-se perceber, também, que outro desafio apresentado na linha da pesquisa de laboratório colaborativo é o compartilhamento de recursos e equipamentos entre instituições de ensino e pesquisa através de acesso remoto, com a viabilidade de acréscimo da faixa de utilização desses equipamentos, buscando otimizar o uso dos recursos e ampliando o número de alunos e pesquisadores envolvidos no contexto.

Embora haja recurso tecnológico, ainda é viável construir uma solução de laboratório de acesso remoto para auxiliar uma infraestrutura de aula presencial já existente. Para atender essa realidade, alguns requisitos devem ser atendidos, tais como, segurança e eficiência do acesso remoto ao aparato experimental e a obtenção dos resultados. É aceitável que umas das maiores dificuldades estejam no fato de que cada laboratório exija uma solução específica, que demande soluções únicas que requer tempo de implementação.

Os laboratórios remotos surgiram diante da necessidade de disponibilizar laboratórios reais a qualquer momento, sem contenção de tempo e de espaço. Com este conceito, é possível que seja adotada uma nova técnica de ensino, permitindo a troca de informações, interação entre alunos, professores e tutores, assim como uma flexibilização de local e de tempo de estudo. Além disso, tem como objetivo minimizar os problemas de logística, facilitando o compartilhamento de recursos e possibilitando o trabalho colaborativo. Espera-se que o laboratório acesso remoto seja um ambiente de aprendizado no qual o aluno e professor possam ter o envolvimento de forma remota em suas atividades laboratoriais, com a mesma eficiência que teriam se estivessem presentes, fisicamente, no laboratório real.

Em modos gerais, o uso de laboratório de acesso remoto consiste em uso de ferramentas que são utilizadas para fazer a aquisição, o controle e a monitoração de dados do experimento. A aquisição e a monitoração são realizadas a partir de equipamentos de *hardware* integrados ao modo de aquisição específico baseado em *software*. Da mesma maneira, o controle do

experimento é realizado a partir de uma página web, projetado com controles específicos para o experimento.

Conforme Santos (2003), o ambiente de aprendizagem mediado pela Internet surge como um dos meios promissores de apoio ao processo educacional. Com base nessa afirmação, este trabalho propõe o desenvolvimento de uma plataforma baseada em *hardware* e *software* para o ensino, pesquisa e experimentação a distância denominada Laboratório Virtual de Aprendizagem Eletrônica (LABVAE). Essa proposta permite que alunos e pesquisadores programem remotamente os seus próprios experimentos e os integrem ao LABVAE, aproveitando, deste modo, os benefícios de controle, monitoração e aquisição das informações dos experimentos que a arquitetura do laboratório oferece. Este laboratório virtual foi projetado para viabilizar o conceito da prática de laboratório remoto, possibilitando a colaboração no fortalecimento do conhecimento teórico, através de uma prática de laboratório remoto.

### **1.1 Definição do Problema**

As formações em ciência da computação, sistemas de informação, redes de computadores, mecatrônica e as engenharias, possuem algumas disciplinas que necessitam da execução de tarefas em laboratórios, como a disciplina de circuitos digitais, em que há necessidade de manipular equipamentos eletrônicos, instrumentos de medição, dispositivos e componentes para montagem e análise de circuitos para realização de experimentos em aulas práticas.

Manter um laboratório funcionando exige investimento e custos elevados com equipamentos e corpo técnico responsável pelo espaço físico. Outros aspectos identificam-se na concorrência entre alunos pelo uso do laboratório e nos riscos de danos causados aos dispositivos por mau uso.

### **1.2 Hipótese**

Parte-se da hipótese segundo a qual a plataforma do laboratório virtual, projetada para viabilizar o conceito prático de laboratório remoto, promove a colaboração na pesquisa prática e no ensino de experimentos remotos.

## **1.3 Objetivos**

### **1.3.1 Objetivo Geral**

Propor um sistema de *software* e de *hardware* cuja principal função é realizar experimentos em sistema de *hardware*, eletronicamente interconectado, atualizando sistemas embarcados através de acesso remoto em tempo real.

### **1.3.2 Objetivos Específicos**

- Analisar propostas semelhantes em publicações de temas correlatos;
- Identificar as tecnologias que apresentem respostas em tempo real;
- Implementar um sistema baseado em *software* e *hardware* para concretização de experimentos a serem realizados remotamente;
- Validar as tecnologias selecionadas para o desenvolvimento da solução;
- Sistematizar o processo de criação de um laboratório remoto, disponibilizando o produto final para a pesquisa de campo e homologar da solução.

## **1.4 Justificativa**

A opção pelo tema decorre de análises sobre a problemática descrita neste estudo, de forma que se tornou motivador propor a criação de um laboratório de acesso remoto (LAR), com o intuito de possibilitar a realização dos experimentos de forma remota, com segurança e redução de custos. Considera-se um tema de relevância por disponibilizar ao aluno um laboratório provisionado com tais equipamentos e para profissionais da Ciência da Computação e correlatas.

## **1.5 Metodologia**

O presente estudo foi elaborado a partir de uma pesquisa bibliográfica, que é definida como aquela que se realiza a partir do registro disponível, decorrente de pesquisas anteriores, em documentos impressos, como livros, artigos, dissertações e teses (SEVERINO, 2007).

Quanto aos fins, foi realizada uma pesquisa exploratória, experimental e descritiva, visando proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses (SEVERINO, 2007).

Também foi desenvolvido um estudo de caso para medição, modelagem e simulação, em ambiente real, de um mecanismo de mensuração de resultado para consumo de recursos de rede, a fim de medir a eficiência da solução. Em seguida, foi apresentado seu perfil de tráfego, utilizando as seguintes métricas: vazão, volume de tráfego, tamanho do pacote, dentre outras.

Com o intuito de reforçar os resultados esperados, foi realizada uma pesquisa de campo, utilizando questionário como instrumento aplicado a 54 alunos, no sentido de validar a solução LABVAE, de modo a reforçar a hipótese do estudo. O universo da pesquisa concentrou-se na Universidade Tiradentes (UNIT), onde os alunos dos cursos de Ciências da Computação, Engenharia Mecatrônica, Sistema de Informação e Redes de Computadores foram avaliados. A amostra foi do tipo aleatória.

Os dados foram tratados estatisticamente e analisados qualitativamente. O método de abordagem caracteriza-se como indutivo, uma vez que se partiu de uma experiência particular para a proposta de universalização. Os métodos de procedimentos foram o estatístico e o comparativo.

A coleta de dados ocorreu em três etapas: na primeira foi apresentada a proposta de pesquisa aos sujeitos, disponibilizando a solução em horários predefinidos; na segunda foi aplicado um questionário (Apêndice) aos alunos para validação da solução; e na terceira foram realizadas as medições para avaliar o desempenho da ferramenta na rede de computadores da Universidade Tiradentes.

## **1.6 Organização da Dissertação**

Neste capítulo, encontra-se uma abordagem geral do contexto no qual pode ser aplicado este trabalho, destacando-se os seguintes itens: Problemática, hipótese, objetivos, justificativa e metodologia.

O segundo capítulo consiste na fundamentação teórica, onde estão descritos conceitos e noções de cliente Web, protocolo HTTP, *JavaScript*, *Scalable Vectorial Graphics* (SVG), *Python* e *WebSocket*. O último item deste capítulo apresenta catorze trabalhos relacionados.

No terceiro capítulo, o LABVAE é descrito, enquanto solução, abordando-se os seguintes aspectos a ele correspondentes: Arquitetura; as camadas cliente, de *middleware* e

tecnologias, de arquitetura de *hardware* e rede; modelagem; funcionamento; páginas web; programar *arduino*; aquisição webcam; aquisição placa de controle e aquisição *arduino*.

No quarto capítulo encontra-se a avaliação do LABVAE, dividida em duas seções: Avaliação quantitativa e qualitativa da pesquisa.

No quinto capítulo, encontram-se descritos os resultados obtidos através da aplicação de um questionário a alunos, a fim de testar a funcionalidade do LABVAE.

No último capítulo, além das conclusões, estão algumas contribuições e sugestões para a elaboração e/ou aprofundamento de pesquisas futuras.

---

## 2 Fundamentação Teórica

Laboratórios remotos para a educação em assuntos de controle constituem um método comum usado pelas universidades. Esse método oferece um serviço flexível de horários, de modo a obter maiores e melhores resultados dos recursos disponíveis. Os laboratórios remotos têm sido utilizados para controlar remotamente os dispositivos físicos. Além disso, os laboratórios remotos têm sido usados para a transferência de identificação do equipamento real (SANTANA *et al.*, 2012).

Os laboratórios remotos permitem acomodar muitos estudantes para realizar experimentos simultaneamente e quase da mesma maneira que os laboratórios reais, assim como ocorre nas plataformas FPGA, as quais aceitam os serviços desenvolvidos nas linguagens de programação Verilog e HDL e as FPGA setup, que executam os serviços web. Quanto às tarefas, elas podem ser executadas em PCs de estudantes, a fim de descarregar obras das plataformas. Assim, um compartilhamento eficiente de plataformas pode ser alcançado. Isso pode também melhorar o desempenho interativo especialmente em tempo de design de entrada/simulação. Desse modo, a finalidade é tornar possível aos alunos realizarem investigação(ões) remotamente quanto aos resultados que foram obtidos na fase de execução anterior. O analisador de lógica moderna baseada em PC permite que tais medidas sejam armazenadas em nuvem para os estudantes possam repetir seus experimentos nos próprios PCs. Os arquivos são transferidos através de armazenamento em nuvem, tornando-se fácil migrar entre diferentes modos de laboratório em laboratório remoto (MAAROUF *et al.*, 2012).

A esse respeito, Yanaze (2009) faz referência à tecno-pedagogia, que abrange os aspectos lúdicos e pedagógicos da comunicação digital, está revolucionando o modo de formação tradicional, trazendo a experiência do laboratório em sala de aula, em casa e em vários locais.

### 2.1 Cliente Web

A utilização de um cliente web se faz da forma que o usuário possa fazer solicitações ao servidor, que, por sua vez, exhibe para ele o resultado do que foi pedido. O usuário acessa esse cliente web via navegador (browser) e o browser se comunica com o servidor, interpreta o

código HTML e exibe a página para o usuário. O processo se inicia com a conexão entre o computador onde está instalado o servidor web e o computador do cliente. A partir daí, é processado o pedido do cliente e, conforme as restrições de segurança e a existência da informação solicitada, o servidor devolve os dados. Quando o servidor responde a uma solicitação, geralmente envia instruções para o browser escritas em HTML, que informa ao browser como exibir o conteúdo ao usuário. O HTML possui um conjunto de *tags* e atributos de *tags*.

O uso amigável do aplicativo da web é devido à integração transparente de *software* (MATLAB e *Simulink*), tipicamente usada em laboratório com o Sistema de Laboratório a Distância (SLD) (SANTANA *et al.*, 2012).

## **2.2 Protocolo HTTP**

O HTTP é outro protocolo de rede que tem características específicas para web, mas depende do TCP/IP para obter a solicitação e a resposta completas de um lugar para o outro. O TCP/IP é responsável por garantir que um arquivo enviado de um nó da rede para o outro chegue íntegro ao seu destino, mesmo que o arquivo esteja dividido em blocos no momento de envio. O servidor usa o HTTP para enviar o HTML para o cliente. O objetivo do HTTP é permitir a transferência de arquivos (essencialmente no formato HTML). Isso é possível pela URL e por um servidor web.

O protocolo HTTP possui vários métodos, porém os que são usados com mais frequência são o GET e o POST. Eles informam que tipo de solicitação está sendo realizada e o formato da mensagem.

O GET é um método simples do HTTP e seu principal trabalho é pedir ao servidor que consiga um recurso e o envie de volta, ou seja, o objetivo é pedir alguma coisa ao servidor. Pode-se usar o GET para enviar dados também, mas ele tem o tamanho de caracteres limitado e os dados enviados ficam aparecendo na barra de endereços.

O POST está associado ao envio de dados de um formulário para o servidor. Pode enviar um conjunto maior de dados e de diferentes tipos.

## **2.3 JavaScript**

Conhecida por derivar da linguagem *JAVA*, a *JavaScript* se destina ao desenvolvimento web, de páginas ou de servidores (*client-side*, *server-side*). Ela permite que o programador

tenha acesso aos elementos de uma página qualquer, como imagens, elementos de um formulário e links, que, por sua vez, podem ser alterados via programação, pois esta linguagem permite a captura de eventos, desde um clique no mouse até uma tecla pressionada. Portanto a linguagem permite criar ações baseadas nas ações do usuário.

O primeiro browser a introduzir a *JavaScript* foi o Netscape, ao contrário da Microsoft, que começou a incorporar alguns recursos apenas a partir do Internet Explorer 3.0. Com o passar dos anos, outros browsers foram incorporando o *JavaScript*, fazendo com que esta linguagem crescesse e estendesse o seu suporte. A Microsoft, para não ficar atrás, criou a linguagem chamada JScript, que é a mesma linguagem da *JavaScript* com alguns adicionais, gerando problema de incompatibilidade com os browsers.

## 2.4 SVG

O SVG é um arquivo XML que contém comandos específicos e simples capazes de gerar uma imagem vetorizada e de alta qualidade. Mesmo se a resolução da imagem for alterada, não perderá a qualidade, pois se trata de uma imagem vetorizada. Em imagens fixas e em animações, o SVG pode ser utilizado junto com o *JavaScript*, adicionando eventos a essas imagens. O SVG permite três tipos de objetos gráficos: imagens, textos e formas geométricas. Quando se trabalha com imagens, elas necessitam de um alto nível de detalhes, ou seja, elas requerem uma resolução muito grande e ocuparão bastante espaço. Do contrário, haverá perda de qualidade na imagem e seu redimensionamento resultará em distorção.



**Figura 1:** *Bitmap x Outline*

**Fonte:** SANTOS, M.P. Autor do Projeto (2014).

No SVG, porém, as imagens são escaláveis, ou seja, elas podem ser redimensionadas sem haver distorção, conforme mostrado na Figura 1. Além disso, o seu armazenamento

ocupa bem menos espaço que o de uma imagem convencional *.jpeg* ou *.png*, pois as imagens do SVG são arquivos XML.

## 2.5 Python

A programação para web ganha cada vez mais espaço no mercado virtual. As linguagens de programação estão mais poderosas e ao mesmo tempo produtivas e com excelente desempenho. Uma delas é o *Python*, que pode rodar com o servidor web Apache na geração de páginas dinâmicas. Como exemplo, teremos um cadastro de produtos de supermercado que povoará uma tabela do banco de dados *Mysql*.

A linguagem *Python* foi criada em 1991, por Guido van Rossum, com muitas características da linguagem ABC. Ela é orientada a objetos, interpretada e interativa, tendo como atributo marcante o controle de bloco por indentação, sem a necessidade de símbolos delimitadores, e com códigos legíveis, sendo extensível para C/C++ em caso de acesso a bibliotecas construídas nessas linguagens.

Após o primeiro contato, é possível seguir com um exemplo de montagem de servidor HTTP, conforme as bibliotecas disponíveis pela própria linguagem (*BaseHTTPServer*, *BaseHTTPRequestHandler*, *HTTPServer*). Parte-se do princípio da criação de uma classe para capturar os *Posts* e *Gets* que são utilizados para armazenar e retornar valores, fator significativo no desenvolvimento HTTP.

Os nomes das definições *do\_GET* e *do\_POST* são padrões do *BaseHTTPRequestHandler*. O *do\_GET* é usado para renderização de qualquer arquivo. Por exemplo: tem-se uma *index.html* na pasta *htdocs*, que trabalhará obtendo o arquivo *index.html* (*/home/usuarioteste/python/httpd/htdocs/index.html*) e renderizando o *html* dele.

Na var *NameVirtualHost* passa-se IP e PORTA para o servidor HTTP. Exemplos: liberar só para acesso local: *localhost:8000* e liberar para qualquer IP que estiver configurado na máquina: *\*:8000*.

Na biblioteca *BaseHTTPServer*, há uma função em que o mesmo server, para rodar o servidor, recebe dois parâmetros: o primeiro é IP/PORTA e o segundo é a classe onde estão os métodos (GET, POST etc.), que, no caso em questão, é a classe *html*. Com as informações apresentadas e a definição criada anteriormente, já se pode dar início ao processo de *start* do HTTP, obtendo:

- *DocumentRoot*: Pega a pasta local que estamos e concatena com *"/htdocs/"*;

- PORT: Porta-padrão do servidor;
- HOST: Host-padrão do servidor;
- Rodando o servidor: `usuarioteste@program-8:~/python/httpd$ python httpd.py *:8000 Start server HTTP IN *:8000`
- Finalizando o servidor e exibindo os logs de relação `usuarioteste@program-8:~/python/httpd$ python httpd.py localhost:8000 Start server >HTTP IN localhost:8000 localhost - - [14/Sep/2010 09:42:05] "GET /index.html >HTTP/1.1" 200 - localhost - - [14/Sep/2010 09:42:54] "GET /index.html >HTTP/1.1" 200 - ^CShutting down server >HTTP marcos@program-8:~/python/httpd$`

## 2.6 *WebSocket*

*WebSocket* é uma API de desenvolvimento que permite às páginas da web usar os recursos do protocolo *WebSocket*, que foi definido pelo IETF, a fim de estabelecer a comunicação de duas vias com um host remoto. Essa tecnologia teve seu documento produzido em 5 de fevereiro de 2004, sob a política de patentes da W3C (HICKSON, 2009).

Uma vez criada, a conexão permanece aberta, possibilitando comunicação permanente entre o cliente e o servidor, sendo fechada, tão-somente, por determinação explícita do desenvolvedor, o que garante uma conexão *full-duplex*, ou seja, simultânea nas duas direções: cliente servidor e servidor cliente, com pacotes extremamente pequenos em relação às conexões dos sistemas atuais que não usam o *WebSocket*.

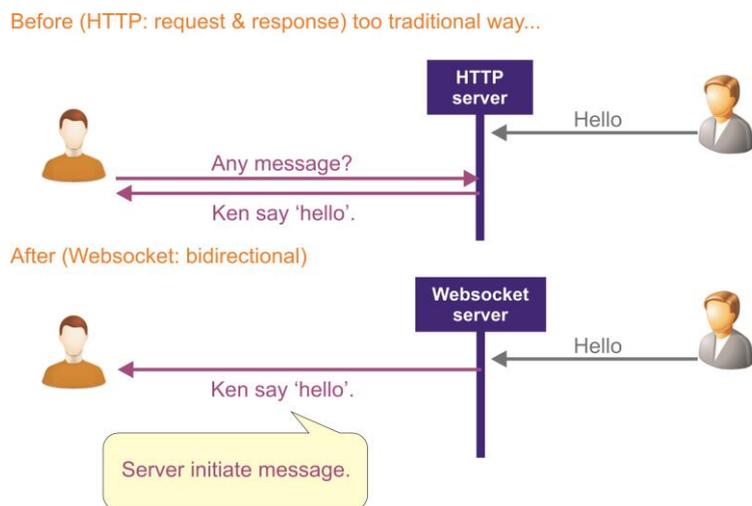
O construtor do *WebSocket* é definido como *WebSocket* (url, protocolos), em que a url faz referência à página que vai carregar e o “protocolo”, quando existe, pode ser uma *string* ou um *array* de *strings*.

Pode-se considerar que é um dos melhores recursos para a comunicação bidirecional de HTTP em alguns aspectos específicos, como a latência. Por ter que estabelecer novas conexões TCP para cada mensagem HTTP e por não existir um *WebSocket*, um servidor web padrão é sobrecarregado com cabeçalhos HTTP a cada pedido de conexão. Com o novo recurso *WebSocket*, reduz-se “o tráfego de cabeçalhos HTTP em 500:1 e, dependendo do tamanho dos cabeçalhos HTTP, a redução pode chegar a 1000:1. A redução de latência é d3:1” (HICKSON, 2009, p. 01). Desse modo, os cabeçalhos em *kilobytes* de sobrecarga em HTTP são reduzidos para dois bytes de *WebSocket* e a dificuldade do servidor de associar as

conexões de saída às conexões de entrada é inexistente, por haver apenas uma conexão TCP em *WebSocket*.

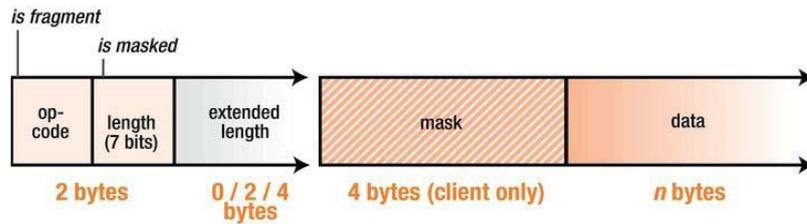
Após o aperto de mão e com o estabelecimento da conexão, o cliente e o servidor podem enviar mensagens a qualquer momento. Cada conexão é representada por uma instância *WebSocketConnection*. A mensagem é escrita de acordo com o protocolo *WebSocket*. Os bytes que precedem os dados marcam o início, o comprimento e o tipo do quadro. Os quadros de texto são codificados em UTF-8. No servidor, cada conexão *WebSocket* é um *asyncore.dispatcher\_with\_send*, que é um soquete assíncrono com suporte para enviar buffer.

Ambos os quadros de texto e de binário podem ser enviados em qualquer direção simultaneamente. Os dados são minimamente enquadrados com apenas dois bytes. No caso de quadros de texto, cada quadro começa com um byte 0x00, termina com um byte 0xFF e contém dados UTF-8 no meio. Quadros de texto *WebSocket* usam um terminador, enquanto quadros binários usam um prefixo de comprimento, conforme apresentado na Figura 2.



**Figura 2:** Troca de Mensagem (adaptado de *WebSocket-Sample*)  
**Fonte:** SANTOS, M.P. Autor do Projeto (2014), pelo *WebSocket-Sample* (2010).

Os dados enviados a partir do navegador para o servidor são mascarados. Mascaramento é uma característica comum do protocolo *WebSocket*. Cada byte de dados de carga útil é XORed com uma máscara randômica para garantir que o tráfego *WebSocket* não tenha necessidade de observar outros protocolos (Figura 3). O *hash Sec-WebSocket-Key* está destinado a garantir uma forma misteriosa de contra-ataque não compatível com a infraestrutura de rede.



**Figura 3:** Componentes do Frame *WebSocket*  
**Fonte:** Lubbers, Brian e Salim (2011).

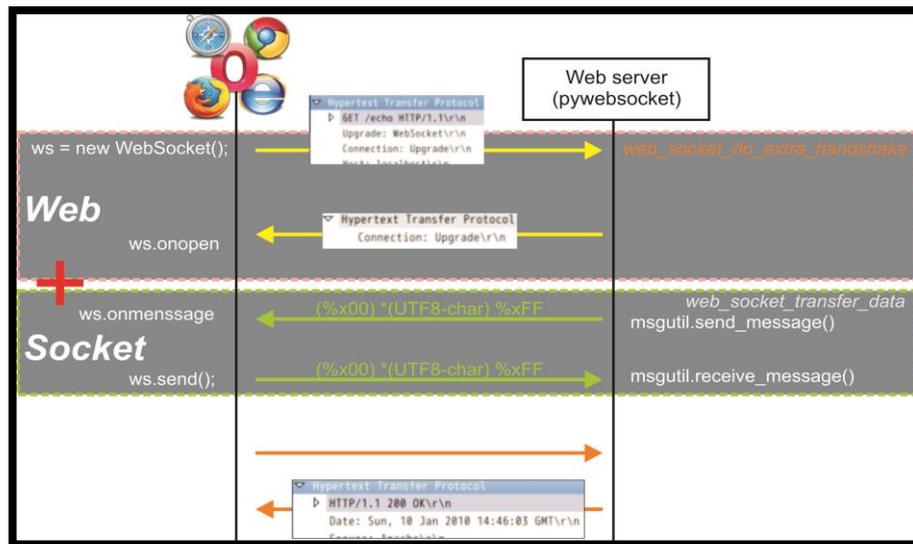
O *WebSocket* possui dependências sobre os conceitos fundamentais do HTML e sobre o uso da semântica dos blocos de *Interface Definition Language* (WebIDL). A IDL permite fazer a descrição das interfaces independentemente da linguagem.

## 2.7 Camada de *Middleware* e Tecnologias

A camada de *middleware* é responsável pela comunicação para entrega das requisições feitas pela camada de aplicação e, pela camada de *hardware*, através da rede local e ou interfaces seriais (USB) que estão conectadas às placas *arduino* e controle.

O *pyWebSocket* é uma API de desenvolvimento que permite às páginas da web usar os recursos do protocolo *WebSocket*, que foi definido pelo *Internet Engineering Task Force* (IETF), a fim de estabelecer a comunicação de duas vias com um host remoto. Essa tecnologia teve seu documento produzido em 05 de fevereiro de 2004, sob a política de patentes da W3C (HICKSON, 2009).

Uma vez criada, a conexão permanece aberta, possibilitando comunicação permanente entre o cliente e o servidor, sendo fechada, tão-somente, por determinação explícita do desenvolvedor, o que garante uma conexão *full-duplex*, ou seja, simultânea nas duas direções: cliente→servidor e servidor→cliente, semelhante às conexões dos sistemas atuais que não usam o *WebSocket*, conforme demonstrado na Figura 4.



**Figura 4:** Sequência do protocolo *WebSocket*

Fonte: SANTOS, M. P. Autor do Projeto (2014), pelo *WebSocket-Sample* (2012).

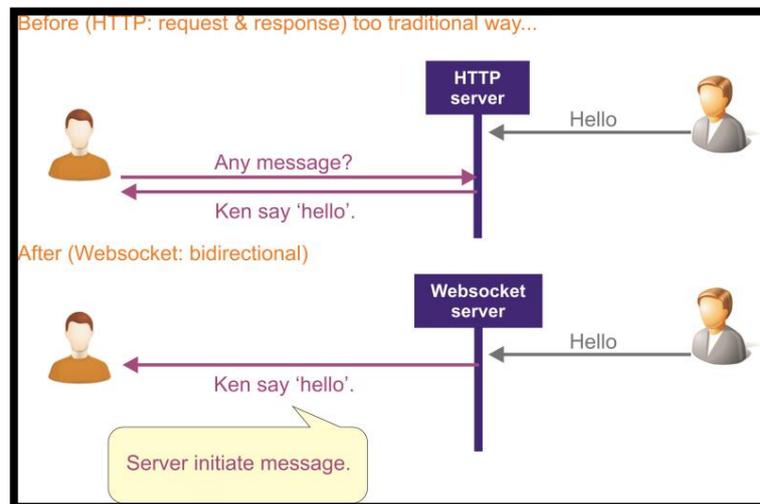
Após a *handshake* e com o estabelecimento da conexão, o cliente e o servidor podem enviar mensagens a qualquer momento de forma bidirecional. Cada conexão é representada por uma instância *WebSocketConnection*. A mensagem é escrita de acordo com o protocolo *WebSocket*. Os *bytes* que precedem os dados marcam o início, o comprimento e o tipo do quadro. Os quadros de texto são codificados em UTF-8. No servidor, cada conexão *WebSocket* é um *asyncore.dispatcher\_with\_send*, que é um soquete assíncrono com suporte para enviar buffer.

Quando a página da aplicação é aberta no *browser*, o navegador começa com a URL e extrai a seção do nome do host usando o *Domain Name System* (DNS) – Servidor de Domínio de Nomes – para fazer o mapeamento a um endereço IP equivalente e usa o endereço IP para formar a conexão TCP com o servidor. A partir do momento que esta conexão está pronta, o navegador e o servidor web usam o HTTP para se comunicarem; desta forma, o navegador envia uma requisição para carregar uma página específica, e o servidor responde enviando uma cópia da página (COMER, 2006).

O navegador envia um comando GET do HTTP para iniciar a conexão com aplicação e em seguida a conexão *WebSocket* é iniciada.

Os dados podem ser enviados entre o cliente e o servidor em modo *full-duplex*. Ambos os quadros de texto e de binário podem ser enviados em qualquer direção, simultaneamente. Os dados são minimamente enquadrados com apenas dois *bytes*. No caso de quadros de texto, cada quadro começa com um *byte* 0x00, termina com um *byte* 0xFF e contém dados UTF-8

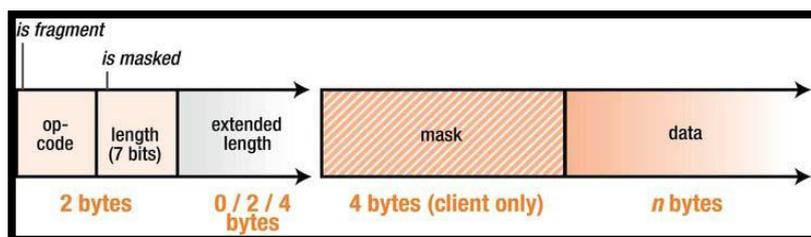
no meio. Quadros de texto *WebSocket* usa um terminador, enquanto quadros binários usam um prefixo de comprimento, conforme apresentado na Figura 5.



**Figura 5:** Troca de mensagem HTTP x *WebSocket*

Fonte: SANTOS, M. P. Autor do Projeto (2014), pelo *WebSocket-Sample* (2012).

Os dados enviados a partir do navegador para o servidor são mascarados. Mascaramento é uma característica comum do protocolo *WebSocket*. A Figura 6 representa a estrutura do quadro. Cada byte de dados de carga útil usa uma máscara randômica para garantir que o tráfego *WebSocket* não tenha necessidade de observar outros protocolos.

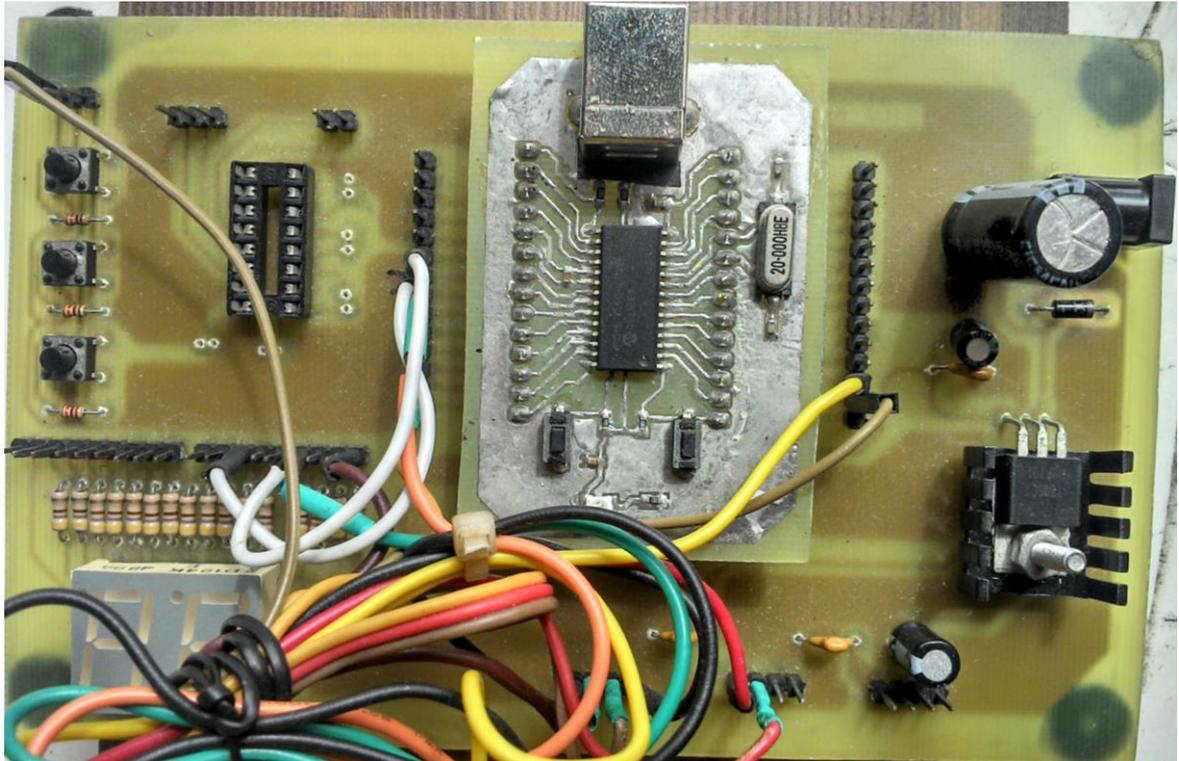


**Figura 6:** Componentes do frame *WebSocket*

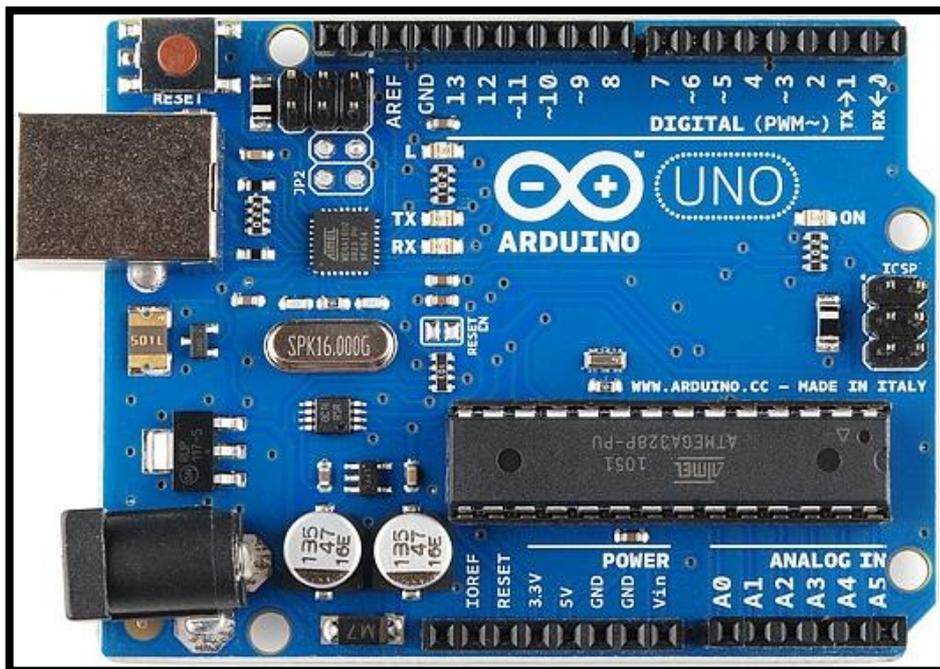
Fonte: Lubbers, Brian e Salim (2012).

## 2.8 Camada de Arquitetura de *Hardware* e Rede

A camada de arquitetura de *hardware* e rede é responsável por estabelecer a troca de informações com a camada de *middleware* do LABVAE. Esta camada representa a infraestrutura de *hardware* e rede que utiliza recurso como a placa de controle com circuito integrado de micro controlador *PIC* modelo *18F2550* e *arduino UNO*, conforme as Figuras 7 e 8, que estão interconectadas a um servidor através de cabo *Universal Serial Buss* (USB).



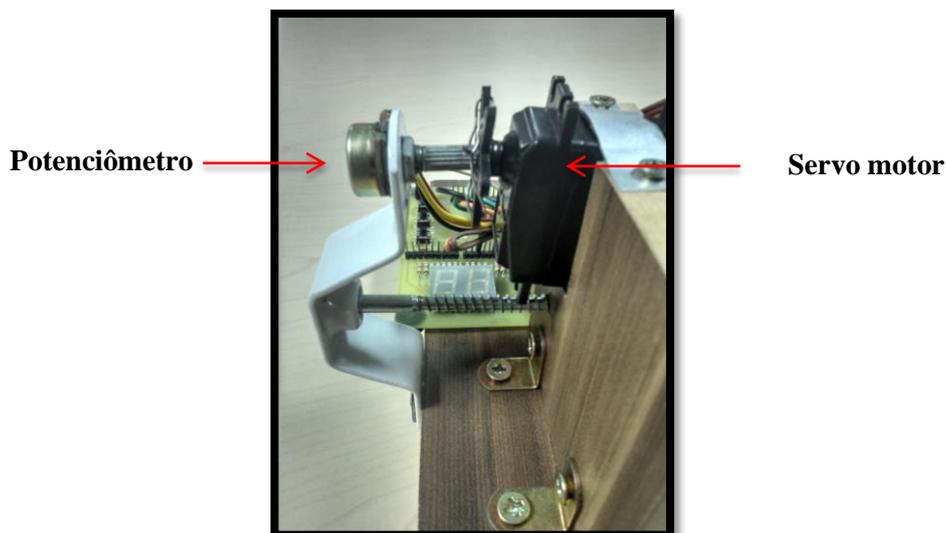
**Figura 7:** Placa de micro controlador PIC modelo 18F2550  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).



**Figura 8:** Placa arduino UNO  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

Ainda faz parte do projeto o servo motor e o potenciômetro ilustrados na Figura 9. Atualmente, o laboratório pode ser utilizado na rede mundial de computadores (Internet), e

nas redes locais de computadores dedicadas ao ensino e pesquisa. O importante é que, independente da rede de computadores utilizada, a comunicação acontece fazendo uso dos protocolos de rede bem definidos a exemplo do TCP/IP descrito na RFC-1180 (*Network Working Group*).



**Figura 9:** Potenciômetro acoplado ao servo motor  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

## 2.9 Trabalhos Relacionados

### 2.9.1 Trabalhos correlatos quanto ao desenvolvimento de solução de laboratório de acesso remoto

No estudo desenvolvido por Moreira, Cardoso e Arantes (2008), a plataforma REDLART é uma solução adicional para auxiliar no processo de ensino-aprendizagem de práticas de laboratório de engenharia. A intenção não é substituir as aulas práticas de laboratório convencionais, mas fornecer uma solução que possibilite estender o conceito de laboratório real para que as experiências possam ser realizadas à distância. A plataforma usa dispositivos FPGA para desenvolver circuitos digitais com aplicações em processamento digital de sinais, sistemas de comunicações digitais, controle, dentre outras. A plataforma foi projetada e desenvolvida para a prototipagem. Este trabalho objetivou a viabilização de trabalho colaborativo em laboratórios distribuído em rede de pacotes de alta velocidade.

No artigo de Žovínová e Ožvoldová (2011) é descrita a criação de uma solução chamada de *World in Motion*, que foi desenvolvida utilizando o *Macromedia Flash 8*, permitindo criar aplicações interativas dinâmicas para os sites mais comuns. Nesse trabalho,

foram apresentadas as seguintes dificuldades: ter um computador capaz de provocar o movimento dos objetos em fração de segundos e construir um equipamento experimental que permita mover o equipamento para sua posição inicial. Percebe-se que nessa solução foram agregados recursos tecnológicos que podem despertar o interesse dos alunos pela disciplina de física, porém, nas discussões técnicas, os autores do artigo não enfatizam a resposta em tempo real, o que fortalece a hipótese do presente projeto de pesquisa. Os autores utilizaram recursos como arquivos em *FLASH* e executável, conhecido como *exe*, a fim de disponibilizar aos alunos uma solução específica de visualização dos experimentos realizados.

Lerro *et al.* (2012) criaram um projeto com o objetivo de aplicar experiências de laboratórios remotos, algumas das quais foram realizadas para demonstrar a eficácia da utilização de laboratórios remotos para uma análise completa de um sistema térmico. Considerando-se o acesso remoto ao sistema térmico, foi utilizado o LMS (Learning Management System). O estudo incluiu a implantação de uma interface que permite aos usuários de uma plataforma *e-learning* terem acesso ao futuro desenvolvimento de laboratórios remotos. Para isso, os autores descrevem a implementação técnica do projeto, vislumbrando o fornecimento de critérios educacionais para integrar o novo desenvolvimento ao currículo de engenharia eletrônica.

Conforme esses autores, os LMSs e-educativa integram serviços de informação, notícias, mensagens, chats, fóruns de discussão, *software* repositório, *wikis*, pesquisas, videoconferências, avaliações, marcas, professores atualizados, dados dos alunos, calendário de eventos, dentre outros. De um ponto de vista técnico, o e-educativa LMS é desenvolvido em línguas PERL e HTML e utiliza sistemas de banco de dados MySQL. Suas principais características são:

- intranet / Internet multiplataforma;
- multissistemas: Windows, Linux, Unix, Solaris;
- multilíngua: inglês, espanhol, italiano, português;
- gestão e administração do sistema de alunos, tutores, cursos e avaliações de linha;
- compatibilidade com SCORM<sup>1</sup> (LERRO *et al.*, 2012).

---

<sup>1</sup> Conjunto de normas técnicas para produtos de *software* e-learning.

Santana *et al.* (2013) realizaram uma experiência de laboratórios remotos em aplicações de pesquisa, compartilhando complexos equipamentos entre diferentes pesquisadores, pela aplicação de experimentações remotas no controle de um robô 3-DOF paralelo, usando SLD. O trabalho demonstrou a eficácia do uso remoto laboratórios em experimentos de pesquisa relacionados ao sistema robótico. O desempenho do sistema é avaliado pelas possibilidades e funcionalidades da plataforma de laboratório remoto. Esses autores afirmam que o uso de laboratórios remotos para a educação está sendo cada vez recorrente nas universidades, a fim de oferecer um serviço flexível em horários com operação maior e melhor dos recursos disponíveis. No entanto, para atividades de pesquisa, laboratórios remotos não têm sido amplamente utilizados.

Grout, Murphy e Da Silva (2012) desenvolveram um projeto com base no entendimento de que o experimento de laboratório de acesso remoto pode ser feito através da utilização de tecnologia de radiofrequência – *Radio-Frequency Identification* (RFID). Cartões inteligentes sem contato são amplamente utilizados em muitas aplicações de cartões de viagem, bem como em construção de controle de acesso e controle de estoque. Esses cartões foram utilizados no projeto desses autores para acesso à experimentação de engenharia eletrônica em um ambiente de laboratório remoto, fornecendo a capacidade de experiências de interface, através deste meio, sem contato (sem fios) de conexão. Para isto, implementaram um projeto de estudo de caso com a finalidade de demonstrar tal meio, incorporando dados do experimento em um cartão inteligente sem contato e acesso a este, por meio de um leitor de cartão e disposição de servidor web.

Garcia-Valls e Basanta-Val (2013) descrevem as configurações práticas realizadas, no caso de uma bem-sucedida experiência de laboratório, que permite aos alunos simular a monitoração on-line e o controle de sistemas remotos, isto é, o tráfego de comboios em um sistema de metrô simulado. A ideia consistiu em reunir o *feedback* dos alunos em relação à operação da tarefa, mostrando que há uma motivação à aprendizagem dos alunos. As tecnologias de comunicação de *middleware* (camadas de *software*) são poderosos instrumentos para desenvolver e implantar laboratórios remotos para os alunos praticarem o controle de casos de aplicações reais. O *middleware* permite o rápido desenvolvimento e implementação de configurações reais distribuídas, uma vez que abstrai os detalhes das diferentes plataformas de *hardware* e meios de comunicação envolvidos no acesso dos alunos ao material de laboratório.

No trabalho de Samuelsen e Graven (2012), o laboratório é utilizado para a verificação de algoritmos de controle para helicópteros através de experiências práticas. A estratégia de

controle de danos deve não limitar ou impedir o movimento natural do equipamento, contanto que o comportamento seja mantido dentro de limites predefinidos. A natureza do sistema a ser controlado exclui o uso de restrições físicas que, normalmente, são utilizadas para danos de controle, uma vez que estes, de algum modo, vão impedir ou limitar o movimento do helicóptero.

Por fim, o trabalho de Choudhary *et al.* (2012) trazem a ideia de laboratórios remotos que fornecem a facilidade de experimentação on-line em processos reais. Esta arquitetura serve para demonstrar o processo de aprendizagem remota, usando a ferramenta gráfica de programação LabVIEW<sup>TM</sup>10. Os princípios da implementação de sessão de laboratório remoto são a entrega da imagem do painel de controle de instrumentos virtuais para o cliente remoto e a ligação do *hardware* experimental ao servidor, pela aquisição de interface de cartão de dados (DAQ). A prova de conceito foi demonstrada através do controle da velocidade de motor DC, utilizando o método *Ward-Leonard* e de gravação das medições usando instrumentos virtuais.

### **2.9.2 Trabalhos correlatos quanto à motivação relacionada ao uso de laboratório de acesso remoto**

Santana *et al.* (2012) realizaram um projeto com o objetivo de contribuir ao aplicar a experiência de controle de laboratório remoto para o campo da pesquisa. Mostram experiências de utilização remota de laboratórios para análise completa de um sistema térmico, o qual permite aos alunos ligar equipamentos via laboratório remoto de forma semelhante, quando estão fisicamente alocados no laboratório. Isso tem permitido aos controladores compreender o design dos equipamentos existentes no laboratório e colocá-los em funcionamento em tempo real, bem como uma análise completa do sistema térmico. Os estudantes avaliam positivamente a disponibilidade de todo o equipamento, por ele estar sempre pronto para ser utilizado, a fim de verificar e identificar diferentes controladores de sistemas, e também por ele lhes permitir obter dados e avaliar o sistema de desempenho em um pequeno período de tempo.

O principal benefício obtido até o momento é o reforço na aprendizagem dos alunos ao utilizarem as práticas remotas. Isso se deve ao fato de que as horas reais lhes permitem familiarizar-se e manipular o equipamento físico diretamente, enquanto o trabalho remoto permite-lhes testar diferentes algoritmos e técnicas de controle sem as restrições do cronograma. Essa metodologia acresce as atividades práticas, bem como oferece aos

estudantes mais tempo para dedicar ao assunto durante o semestre, de modo a serem recompensados na nota final (SANTANA *et al.*, 2012).

Para Maarouf *et al.* (2012), com a evolução da Internet e sua acessibilidade em todos os lugares e a qualquer momento, laboratórios remotos ou laboratórios a distância (LADs) estão atraindo um abrangente e crescente número pessoas, tornando-se soluções viáveis para o ensino a distância e para o compartilhamento de equipamentos sofisticados. Neste contexto, as TICs são a base para LADs e podem, também, proporcionar enriquecimento para o laboratório convencional. Essas novas possibilidades tecnológicas coincidem com o surgimento de novas abordagens de aprendizagem e levantam questões sobre o papel potencial do trabalho de laboratório na formação de alunos de engenharia.

Isso também permite ao aluno estar mais em contato com a realidade tecnológica do laboratório e até mesmo com o espaço industrial, através de visitas virtuais. Além disso, ao usar uma abordagem *Problem Based Learning* (PBL) – Aprendizagem Baseada em Problemas (ABP) –, as novas tecnologias permitem considerar um problema inicial na realidade industrial e em um *hardware* laboratório de configuração (MAAROUF *et al.* 2012).

Com o surgimento de abordagens de ensino ativo e o desenvolvimento do ensino a distância, o trabalho colaborativo é necessário como uma atividade eficaz para apoiar a aprendizagem. O trabalho de laboratório já está baseado no trabalho em equipe. Partindo de tais pressupostos, Santana *et al.* (2012) apresentam dois projetos típicos relacionados ao laboratório de distância, ou *Distance Laboratory System* (SLD). O primeiro SLD usa o controle do equipamento distante durante a experimentação entre diferentes usuários. O segundo SLD baseia-se em um kit eletrônico miniaturizado acoplado a um *software* adequado para executar diferentes experimentos eletrônicos. O resultado desses dois projetos é apresentado com as suas vantagens e desvantagens.

No trabalho de Maiti, Mahata e Maiti (2012), combinam-se laboratórios reais e laboratórios remotos para alcançar a partilha eficiente de FPGA design/avaliação de plataformas e uso eficiente de PCs por estudantes, de modo a reduzir o tempo de espera para plataformas e melhorar a performance interativa. O armazenamento em nuvem permite uma transferência suave e uniforme de arquivos e resultados de experiência. Nesse caso, o PC do aluno funciona como um controle remoto, podendo o mesmo desfrutar de um ambiente com designs com qualidade de usabilidade.

Cubillo *et al.* (2012) citam que o fato de adicionar informação virtual para o resultados observados pelos alunos aumenta a quantidade de informações, fornecendo-nos uma grande absorção dos conteúdos e uma expansão dos estudos de caso. É por isso que é possível

adicionar a resposta para perguntas como: “Por que isso acontece?” ou “O que aconteceria se as condições do ensaio forem diferentes?”. A realidade aumentada é uma maneira fácil de controlar um laboratório remoto, criando uma interface realista, simples e prática a partir de qualquer lugar do mundo.

A publicação de Lowe e Orou (2012) refere que o agendamento otimizado pode melhorar significativamente o nível de utilização de LAR, da expressão original em inglês *Remote Access Laboratory* (RAL), e diminuir o tempo de espera. Essa pesquisa analisa o algoritmo de agendamento utilizado pelo sistema RAL Sahara, a fim de investigar quaisquer limitações que afetam a utilização dele. Foi identificado um número de questões atuais a fim de propor modificações específicas para responder a eles. As modificações propostas devem conduzir a um aumento da utilização e uma melhora da experiência do estudante com o RAL. Ao longo da última década, os laboratórios remotos têm sido valiosos recursos educacionais, proporcionando o potencial para a melhoria dos resultados educacionais do aluno, flexibilidade, experiências mais ricas de laboratório e compartilhamento de recursos.

Na perspectiva de Lowe *et al.* (2012), laboratórios remotos podem ser objeto de desenvolvimento técnico e análise pedagógica. Grande parte da literatura associada usa argumentos sobre a relação de valor desses laboratórios para justificar o trabalho nesta área. Embora muitos desses argumentos estejam focados em oportunidades pedagógicas ou flexibilidade logística, muitas vezes eles também defendem os benefícios financeiros que se obtêm a partir da habilidade de compartilhar recursos laboratoriais. Os autores consideram a prevalência destes argumentos e a extensão em que eles são (ou não são) suportados por uma investigação baseada em provas.

---

## 3 A Solução LABVAE

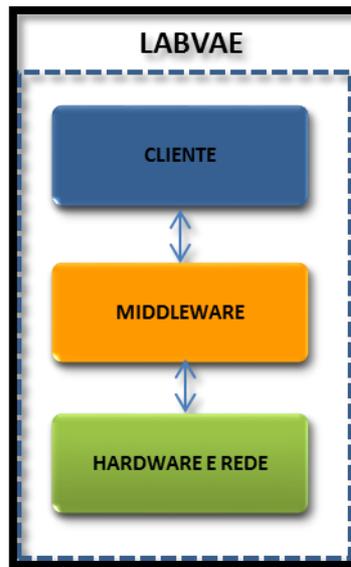
Neste capítulo é apresentada a proposta de arquitetura LABVAE, que tem por objetivo proporcionar um ambiente flexível por estar disponível na web sem a necessidade de *hardware* e *software* específico para funcionamento, de fácil usabilidade por possuir uma interface intuitiva através dos botões de controle e de uma arquitetura orientada a serviços e eventos. Essa arquitetura possibilita a interação com a plataforma em *hardware* e *software* reconfigurável para a execução de experimento real, com resultados em tempo real.

### 3.1 Arquitetura

Trata-se da arquitetura de um sistema para realizar experimentos em sistemas de *hardware*, eletronicamente interconectados e atualizando sistemas embarcados através de acesso remoto em tempo real, buscando economia e viabilidade pedagógica.

O sistema tem a finalidade de permitir acesso remoto aos laboratórios em tempo real para professores, alunos e pesquisadores que têm por objetivo fazer experimentos laboratoriais de sistemas digitais utilizando a plataforma LABVAE.

A modularização do sistema em camadas é apresentada a seguir na Figura 5, sendo que cada módulo possui interface com implementação independente, com o objetivo diminuir a complexidade do sistema, facilitando o desenvolvimento de cada módulo que faz parte da aplicação. A modularização da arquitetura possibilitou a escolha de requisitos técnicos para a implementação do sistema. Desse modo, a Figura 10 representa a proposta de arquitetura em camadas da aplicação.

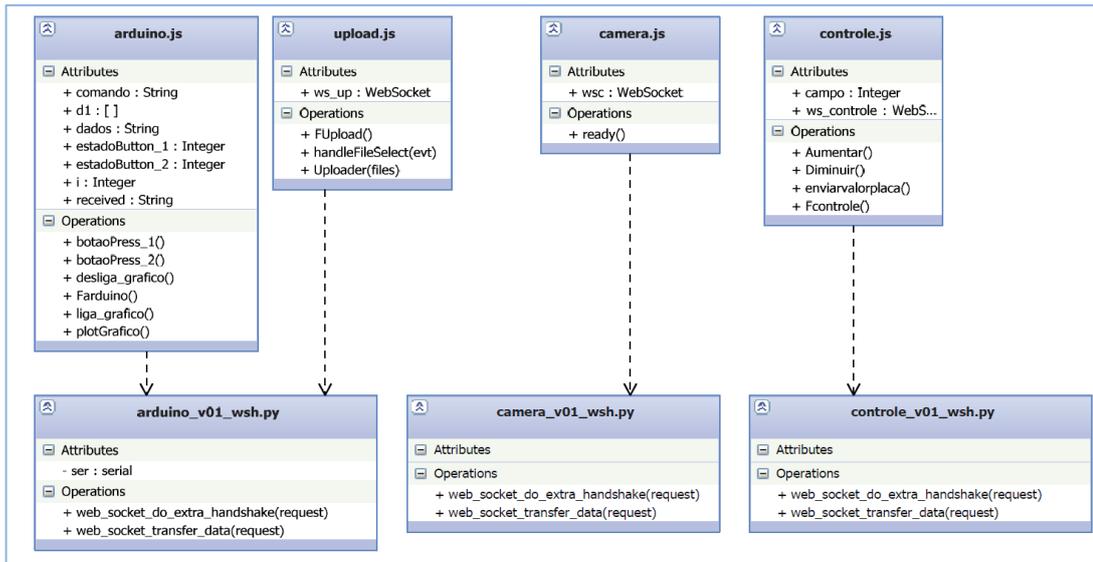


**Figura 10:** Proposta da arquitetura em camadas do LABVAE  
**Fonte:** SANTOS, M.P. Autor do Projeto (2014).

Dentro do contexto da proposta apresentada, esta arquitetura foi composta por três camadas, cada uma das quais com sua especificidade, sendo: cliente; *middleware*; *hardware* e rede. Cada uma dessas camadas tem sua especificidade, de modo que suas respectivas funções são executadas independentemente. Tal independência está na interface de programação de aplicação, da expressão em inglês *Application Programming Interface* (API), que define a forma de como é realizada a comunicação entre essas camadas.

O modelo descrito representa os dados da aplicação e as regras de negócio, que relacionam o acesso e a atualização desses dados requisitados. O modelo está implementado na camada do *middleware* que é responsável por controlar os dados da aplicação, responder às consultas da camada de apresentação a respeito do estado e atualizar o seu estado, quando solicitado pela camada de apresentação.

O diagrama de classes do LABVAE é representado na Figura 11.

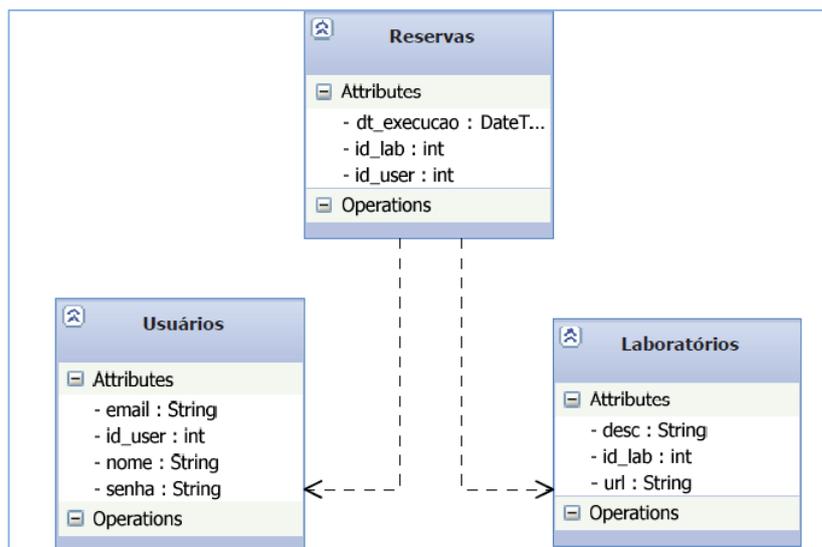


**Figura 11:** Diagrama de classe

**Fonte:** SANTOS, M. P. Autor do Projeto, pelo programa Visual Studio (2014).

As classes *arduino.js*, *upload.js*, *camera.js* e *controle.js* fazem parte da camada de aplicação que intermedia a troca de informações entre o *hardware* e as classes *arduino\_v01.wsh.py*, *câmera\_v01\_wsh.py* e *controle\_v01\_wsh.py* que são responsáveis por abrir um socket de conexão, com o objetivo de gerenciar os dados oriundos ou destinados ao funcionamento do *hardware* e da aplicação. A classe *arduino.js* tem sido utilizada para controle de envio e recebimento de comandos da placa *arduino*, enquanto a classe *upload.js* está encarregada de acionar comando para permitir o envio do arquivo compilado através do uso da IDE do *arduino*, para ser gravado no firmware. Quanto à classe *camera.js*, ela é responsável por permitir o envio do vídeo para a interface web, e o *controle.js*, por sua vez, intermedia o envio de comandos para a placa de controle para acionar o potenciômetro que faz a variação de tensão na entrada analógica do *arduino*.

As classes “usuários”, “laboratórios” e “reservas” fazem parte da camada de *software* para administração do laboratório (Figura 12). A classe “usuários” é responsável por manipular as informações de cadastro dos usuários na aplicação, enquanto a classe “laboratórios” gerencia as informações dos laboratórios cadastrados e a classe “reservas” permite o agendamento das realizações do laboratório a ser executado.



**Figura 12:** Diagrama de classes: laboratórios, usuários e reservas  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa Visual Studio (2014).

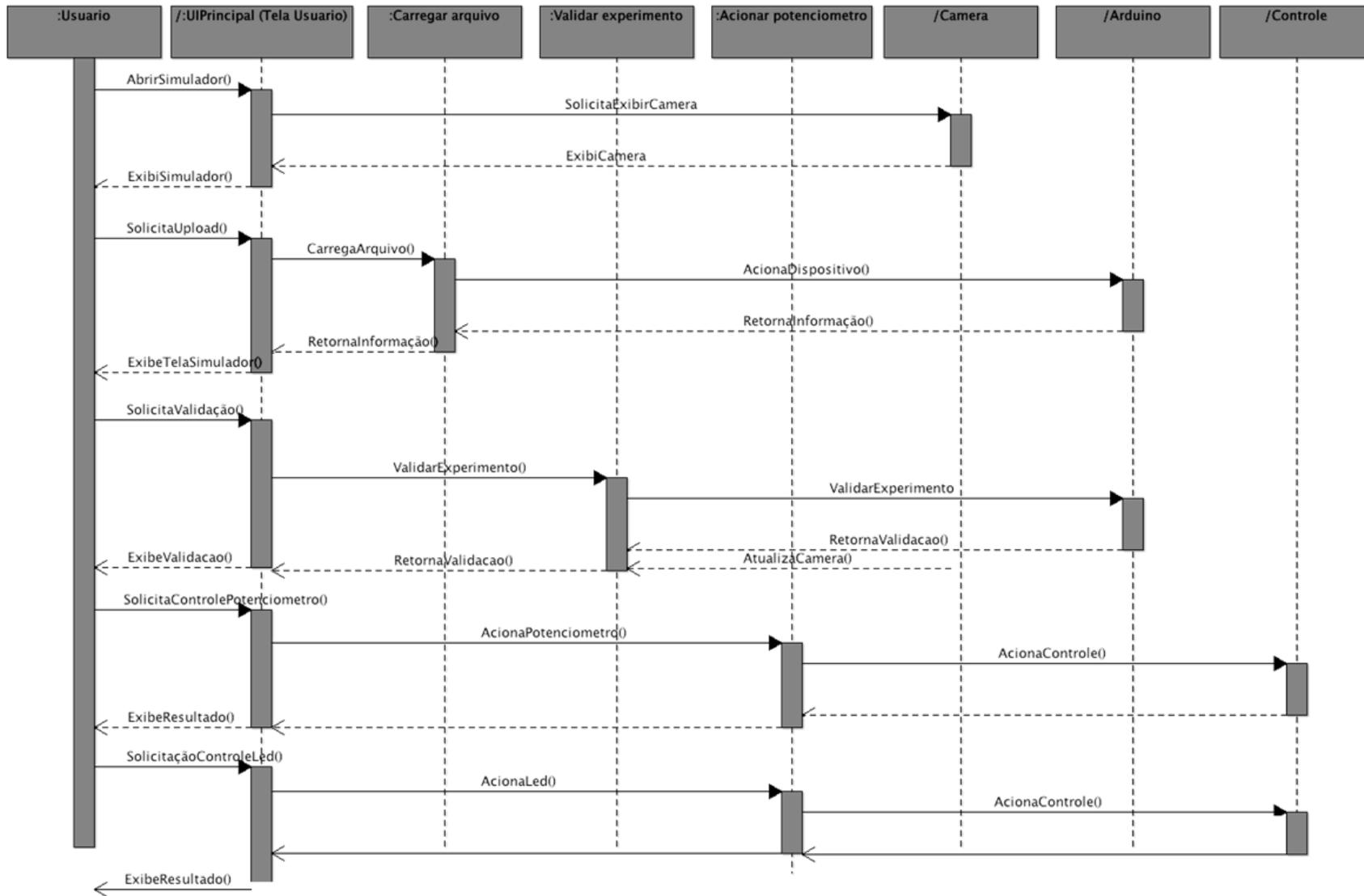
### 3.1.1 Camada Cliente

Nesta camada, foi desenvolvido um ambiente web para aquisição de dados de forma remota. Incluindo uma interface com o usuário para poder fazer solicitações ao servidor, que, conseqüentemente, apresenta o resultado do que foi solicitado. Desta forma, o usuário faz o acesso à página web via *browser*, sendo que este se comunica com a camada de *middleware* que interpreta as requisições do browser e retorna a página HTML para o usuário. O processo se inicia com a conexão entre o computador onde está instalado o servidor web e o computador do cliente.

Ainda na camada cliente LABVAE é exibido o experimento do laboratório que será executado pelo usuário e conseqüentemente são carregados os arquivos *upload.js*, *câmera.js*, *arduino.js* e *controle.js* que faram a interação com a camada de *middleware* para o controle do *hardware*. A seguir a modelagem mais concreta para cada fluxo de dados será apresentado nas secções a seguir.

Os fluxos de dados apresentados nesta camada estão representados na Figura 13, através da representação de um usuário acessando o ambiente do LABVAE. Quando o usuário tenta acessar o ambiente, o servidor web verifica o estado do usuário. Esta verificação consiste na utilização do *software Django* validando as credenciais do usuário. Se o usuário ainda possuir estado válido será apresentado ao mesmo a página inicial do ambiente. Se não, será solicitado que este usuário realize autenticação mediante inserção de login e senha. Esta medida foi adotada por questões de segurança e para realizar o controle acesso dos usuários,

principalmente quando houver necessidade de limitar este acesso a um usuário específico. Esta verificação é realizada através do banco de dados (*SQLite*) para autenticação de usuários no servidor de aplicações e as regras de controle de acesso estão definidas no arquivo *sql\_v01.py*. Se o login e senha forem válidos, será apresentada ao usuário a página inicial do ambiente. Se não, será apresentada uma página de erro e o usuário deverá realizar autenticação novamente.



**Figura 13:** Diagrama de Sequência - Usuário  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa ArgoUML (2014).

Ainda na camada cliente LABVAE é exibido o experimento do laboratório que será executado pelo usuário e conseqüentemente são carregados os arquivos *upload.js*, *controle.js*, *camera.js* e *arduino.js* que fazem a interação com a camada de *middleware* para o controle do *hardware*.

No arquivo *upload.js* existe a função *FUpload(i)* que instancia objeto *WebSocket* que executa o handler *pyWebSocket* usando o arquivo *quisicao\_arduino\_v01\_wsh.py* e a função *Uploader(files)*, que inicia o processo de gravação do arquivo enviando *byte a byte* o arquivo programado pelo usuário.

Já o arquivo *controle.js* possui funções como que garantem a funcionalidade dos controles para o potenciômetro (*function Aumentar()* e *function Diminuir()*) que servem para aumentar e diminuir os valores das aquisições a serem enviadas para o potenciômetro. A função *function enviarvalorplaca()*, serve para enviar um valor específico para a placa de controle bem com a função *Fcontrole()* que inicializa o objeto *WebSocket* que estabelece uma conexão para a placa de controle.

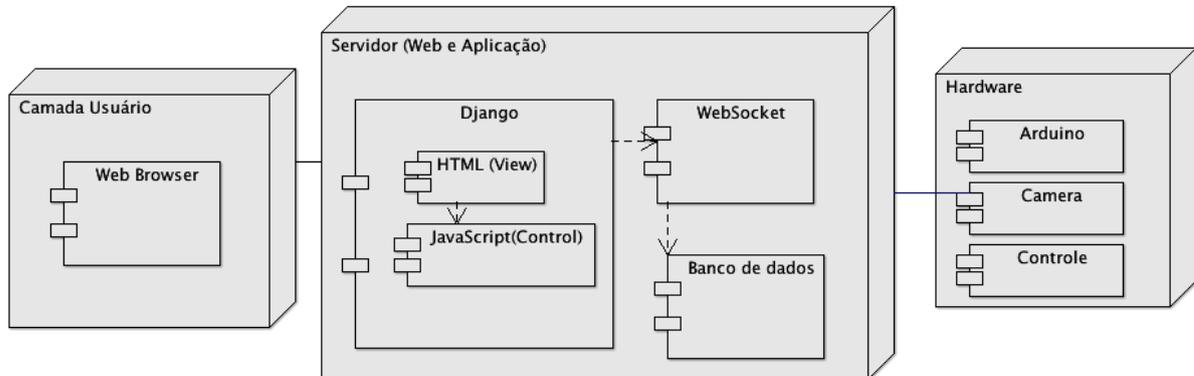
Para auxiliar o funcionamento da imagem na página inicial da aplicação, o arquivo *camera.js* possui a função *ready()*, que é usada para capturar imagens da webcam através do *WebSocket*, de modo que, após a conexão estabelecida pelo mesmo, as funções e controle das aquisições para a página do cliente através do *arduino*, existem as funções *liga\_grafico()* e *desliga\_grafico()* que serve para ligar e desligar a exibição da leitura das aquisições do potenciômetro, nas funções *botaoPress\_1()*, *botaoPress\_2()*, fazem o controle da troca da imagem da representação do funcionamento dos *leds* na página inicial da aplicação. A função *plotGrafico()* é usada para configurar e plotar o gráfico e a função *Farduino()* *WebSocket* que executa o handler *pyWebSocket* *quisicao\_arduino\_v01\_wsh.py*.

### 3.2 Modelagem

O diagrama de implantação, de acordo com Melo (2010, p. 215), “mostra a estrutura de nós nos quais os componentes e artefatos são implantados”.

No diagrama mostrado na Figura 14, a implantação foi dividida utilizando-se o modelo 3-tier, ou seja, três camadas físicas, sendo uma para o usuário, uma para os servidores e uma camada de recursos de *hardware*. Este diagrama foi utilizado por favorecer a condição de representar como é realizada a distribuição do sistema através do uso de nós de *hardware*, componentes e dependências de *software* bem com as devidas relações de comunicação. Além

disso, este tipo de diagrama modela o inter-relacionamento entre recursos de infraestrutura, de rede ou dispositivos de sistemas.

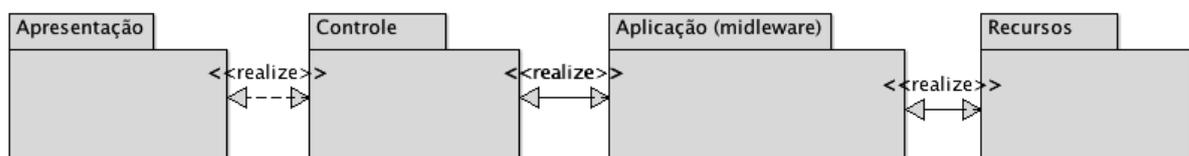


**Figura 14:** Diagrama de Implantação

Fonte: SANTOS, M. P. Autor do Projeto (2014), pelo programa ArgoUML (2014).

Este tipo de modelagem foi utilizado por ter a característica de modelar a arquitetura física de um sistema. Através do uso desse diagrama, é possível mostrar os relacionamentos entre os componentes de *software* e *hardware* no sistema e a distribuição física do processamento. Esses diagramas foram preparados durante a fase de desenvolvimento da implementação dos sistemas. Os nós representam dispositivos de *hardware* como, por exemplo, servidores, placas, motor e leds, bem como outros dispositivos que suportam o ambiente de tempo de execução de um sistema. Caminhos de comunicação e relacionamentos de implementação modelam as conexões do sistema.

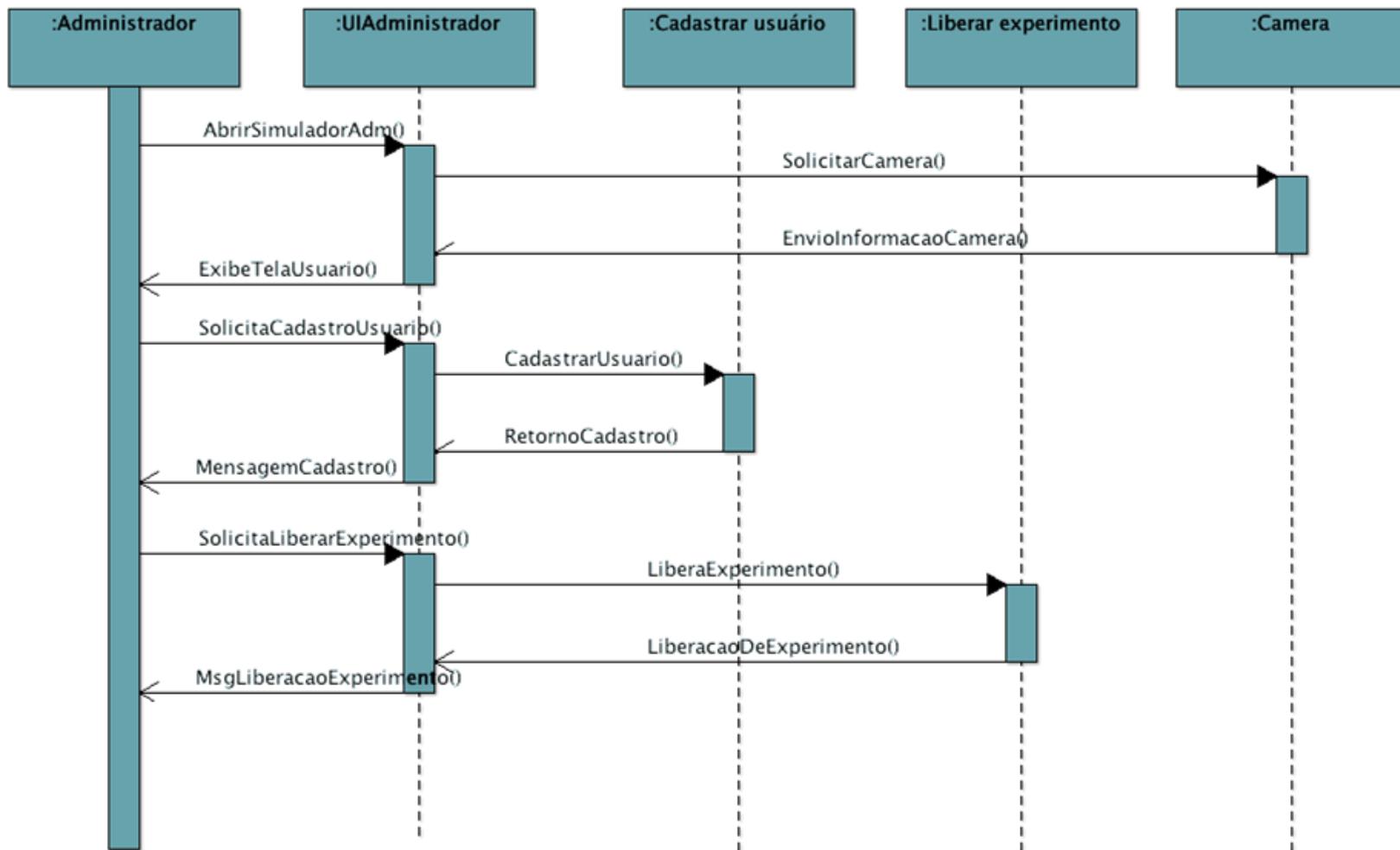
O diagrama de pacotes (oficializado a partir da UML 2.0) apresenta a divisão dos pacotes da aplicação, neste caso, a divisão lógica do sistema, sendo utilizado quatro pacotes, um responsável pela aplicação, um controlador que é responsável pela comunicação com o *middleware* (desenvolvido em *JavaScript*), o *middleware* responsável por acessar os recursos, totalmente desenvolvido em *python*. Para obter esse acesso, é necessário acessar a camada *middleware* que, por sua vez, finalmente, acessará a camada de *hardware* e rede, conforme ilustrado na Figura 15. Para entender os recursos utilizados na pela aplicação LABVAE.



**Figura 15:** Diagrama de Pacotes

Fonte: SANTOS, M. P. Autor do Projeto (2014), pelo programa ArgoUML (2014).

Segundo Blaha e Rumbaugh (2006, p. 142), “Um diagrama de sequência mostra os participantes nas interações e sequencia de mensagem entre eles. Um diagrama de sequência mostra as interações de um sistema com seus atores” (Figura 16).



**Figura 16:** Diagrama de Sequência - Administrador  
 Fonte: SANTOS, M. P. Autor do Projeto (2014), pelo programa ArgoUML (2014).

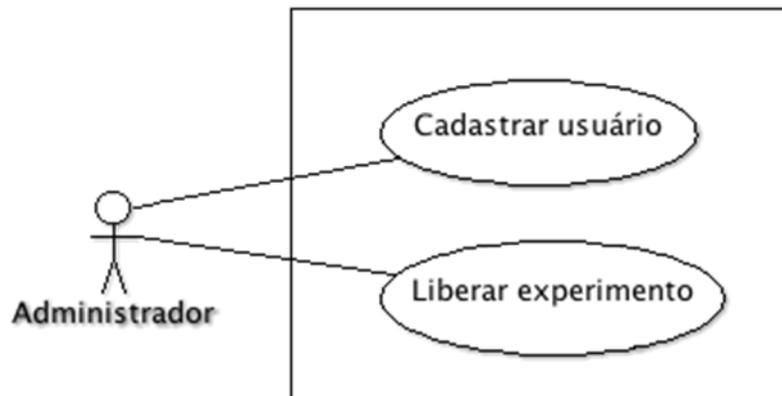
### 3.3 Funcionamento

A construção do LABVAE foi fundamentada no uso de *softwares* como o HTML5, JavaScript, Open Source Computer Vision Library (OpenCV), Django, Scalable Vector Graphics SVG, SQLite e PyWebSocket. Este último, segundo Pimentel e Nickerson (2012), é importante tratar de uma API de desenvolvimento, que diminui a latência na rede e permite às páginas a aplicação usar os recursos do protocolo WebSocket, definido pelo (Internet Engineering Task Force) IETF, possibilitando uma comunicação de duas vias com um host remoto. Uma vez estabelecida, a conexão permanecerá aberta, diferenciando-se na conexão HTTP que faz vários Handshake, um para cada objeto carregado na página, que no caso do WebSocket este processo ocorre uma única vez. Para o funcionamento do laboratório utilizou-se dos recursos de *hardware* e *software* especificados abaixo:

- Computador Intel Core i5, 4Gb de memória RAM, Windows 7 e 1Tb de disco;
- Comunicação do Arduino UNO e da placa de controle com o computador dar-se por meio das portas USB (Universal Serial Buss) do servidor;
- A placa arduino será o objeto de obtenção das informações e leituras pretendidas para cada experimento;
- Leds, servo motor e potenciômetro;
- Uso do OpenCV com a finalidade de dar eficiência computacional focando numa resposta em tempo real para a aplicação;
- O estudante deve fazer a programação para realizar seu experimento através do uso de uma IDE de desenvolvimento e transferir o firmware remotamente para a placa a fim de testar seus experimentos;
- A placa de controle serve como mecanismo de controle para girar o potenciômetro quando necessário;
- Toda e qualquer informação enviada remotamente da aplicação para o hardware e do hardware para a aplicação será feita através do protocolo WebSocket baseado em Python;
- Django software utilizada para a administração do site;
- O dados são armazenados no banco de dados SQLite nativo do Python compatível como o SQL;

- Para comunicar a página HTML da aplicação e o *WebSocket*, uma camada de *software* em *JavaScript* e outra em *Python* para intermediar a comunicação;
- Os resultados são apresentados em tempo real na página web e via webcam na interface do estudante e através de códigos HTML5 para exibição de gráficos.

Na Figura 17, é possível observar o diagrama de caso de uso para atender os requisitos implantação do LABVAE, bem como o administrador e os casos de uso necessário para implantar o sistema (Ligar Computador, Instalar *Django*, Instalar *Python* e suas bibliotecas, instalar OpenCV, instalar placa *arduino*, Instalar placa de controle, Instalar motor, leds e potenciômetro, iniciar servidor web, iniciar servidor *Python*, cadastrar usuário e disponibilizar experimentos). As tecnologias analisadas foram selecionadas de acordo com requisitos que permitissem o desenvolvimento de um ambiente onde os usuários, além de interagir com experimentos, pudessem visualizar o resultado de seus experimentos em tempo real. O administrador pode ser o professor ou o tutor da turma.



**Figura 17:** Caso de Uso do LABVAE (Administrador)  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa ArgoUML (2014).

O sistema possui características que permitem que os usuários cadastrados no sistema agendem seus experimentos previamente; a comunicação do *arduino* e da placa de controle com o computador dá-se por meio das portas USB, interligadas a um servidor; a placa *arduino* é o objeto de obtenção das informações e leituras pretendidas para cada experimento; o uso do OpenCV tem a finalidade de integrar o *hardware* da webcam à aplicação; o estudante deve fazer a programação para realizar seu experimento através do uso da IDE de desenvolvimento do *arduino* descrito na seção 3.3 desta dissertação. Para o melhor entendimento do sistema, encontra-se detalhado seu funcionamento baseando-se na Figura 16

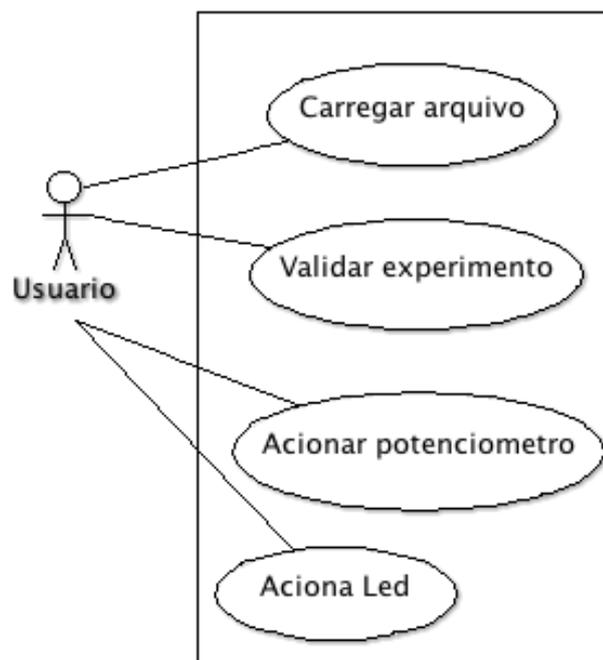
(Diagrama de Sequência – Administrador), através da representação lógica do diagrama de sequência. Os usuários e os laboratórios da aplicação são cadastrados pelo administrador, conforme tela de administração mostrada na Figura 18.



**Figura 18:** Tela de administração do site

**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa *Django* (2013).

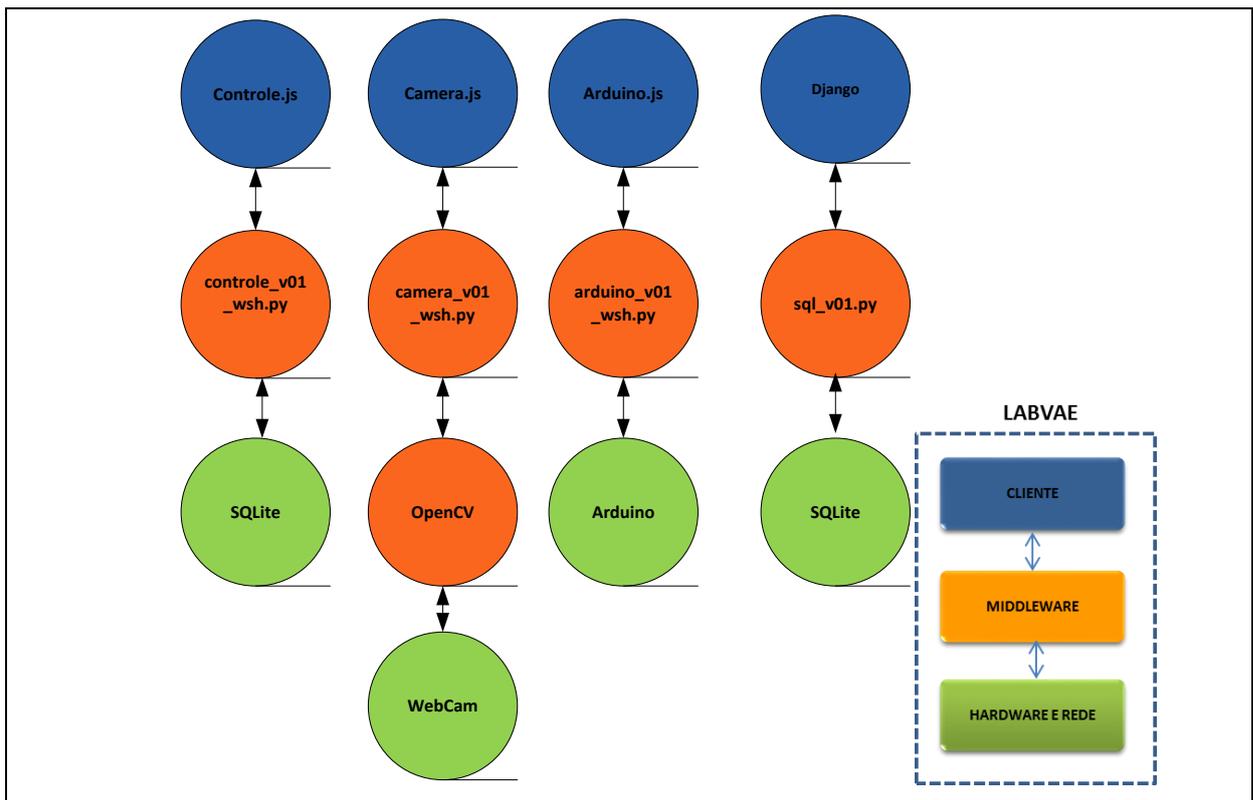
O caso de uso representado na Figura 19 detalha o acesso ao ambiente LABVAE, quando o usuário realiza os requisitos necessários para fazer seus experimentos (Programar *arduino*, compilar programa, acessar aplicação, carregar *javascripts*, iniciar a webcam, instanciar o *OpenVC* e o arquivo *camera\_v01\_wsh.py* e carregar o arquivo *.HEX*, validar experimento). Essas são medidas necessárias para que o usuário consiga fazer uso dos experimentos disponíveis na aplicação LABVAE.



**Figura 19:** Caso de Uso do LABVAE (Usuário)

**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa ArgoUML (2014).

Os arquivos *arduino.js*, *camera.js*, *controle.js*, *arduino\_v01\_wsh.py*, *camera\_v01\_wsh.py*, *controle\_v01\_wsh.py* estão armazenados localmente no servidor e são requisitados através dos arquivos *Controle.js*, *Camera.js* e *Arduino.js*. A existência de dados específicos refere-se àqueles pertinentes a obtenção de resultados do experimento que são obtidos a partir da camada de arquitetura de *hardware* e rede que, após o processamento desses dados, são visualizados na interface web da camada de aplicação, conforme Figura 20.



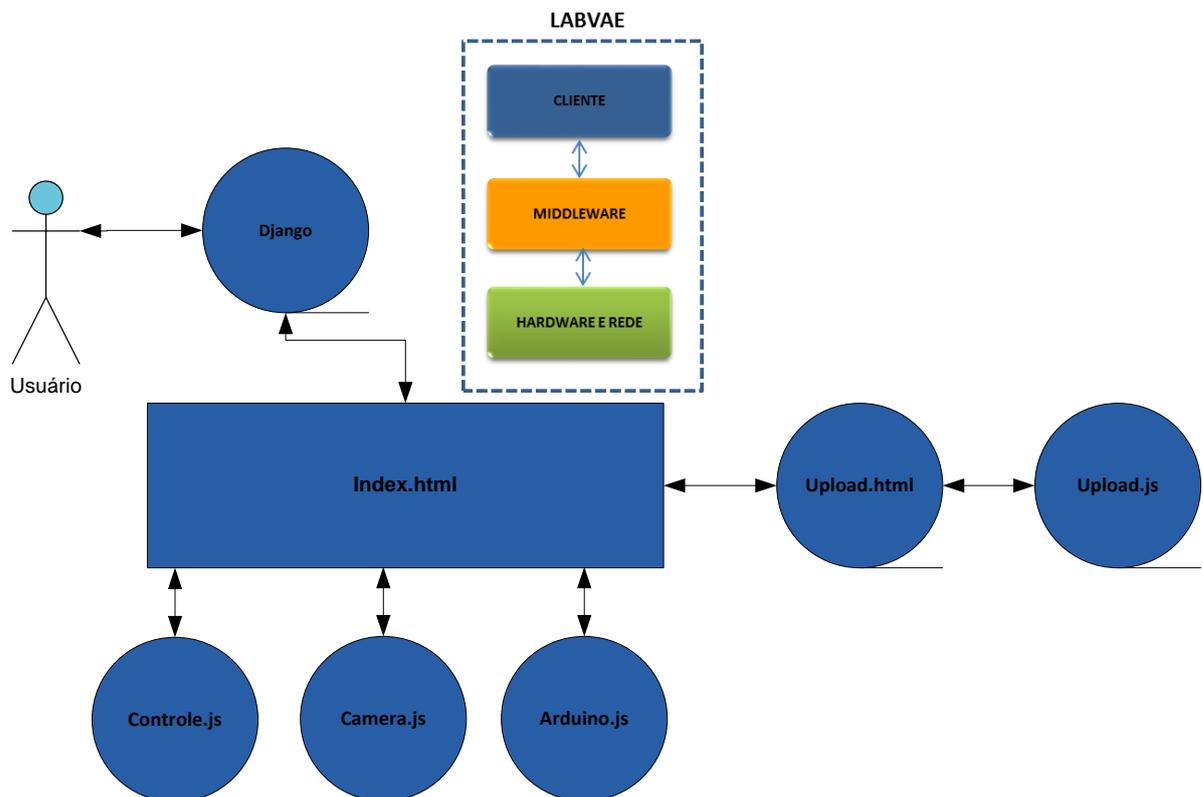
**Figura 20:** Fluxo de dados camada de *middleware*  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa Office Visio (2014).

Nos arquivos *arduino.js*, *camera.js* e *controle.js*, os *WebSockets* são inicializados através das linhas de códigos existentes nos respectivos arquivos, que colocam em funcionamento a camada de *middleware* através do uso dos arquivos *controle\_v01\_wsh.py*, *camera\_v01\_wsh.py*, *arduino\_v01\_wsh.py* que são instanciados na chamada de cada *WebSocket* iniciado.

- ✓ `ws_controle = new WebSocket("ws://10.121.7.4:9998/controle_v01");`
- ✓ `wsc = new WebSocket("ws://10.121.7.4:9998/camera_v01");`
- ✓ `ws = new WebSocket("ws://10.121.7.4:9998/arduino_v01");`

### 3.2.1 Páginas Web

As páginas Web representadas na Figura 21, são hospedadas em um servidor implementado em *python* nativo do *Django*, este serviço é iniciado na fazendo uso da porta 8000. No código HTML, são feitas as chamadas para os arquivos *javascript* (*arduino.js*, *camera.js* *controle.js*) para interagir com a *arduino*, câmera e placa de controle respectivamente. Através desta página o usuário terá botões de controle para poder analisar os seus experimentos com resultados apresentados graficamente em tempo real, tais como, simulação de funcionamento dos leds, acionamento do servo motor, multímetro para obter a valor de tensão adquirido através da placa do *arduino*.



**Figura 21:** Fluxo de dados páginas Web

**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa Office Visio (2014).

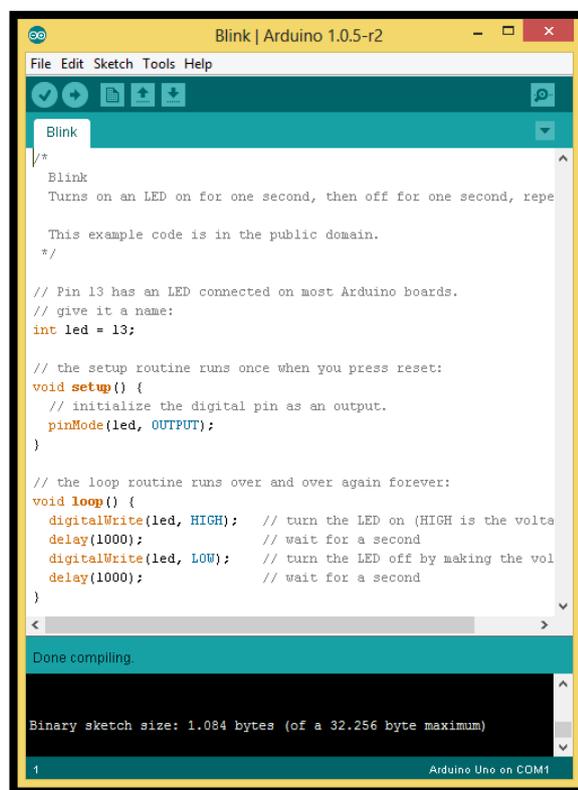
As linhas de códigos a seguir, representam o carregamento dos arquivos *javascript* existentes no servidor para a máquina do cliente.

```
<script type="text/javascript" src="{% static 'laboratorios/js/controle.js' %}"></script>
<script type="text/javascript" src="{% static 'laboratorios/js/camera.js' %}"></script>
<script type="text/javascript" src="{% static 'laboratorios/js/arduino.js' %}"></script>
```

A página *Upload.html* é carregada quando o usuário clica no botão “Gravar Arduino” que, conseqüentemente, carrega para a máquina do usuário o arquivo *upload.js*.

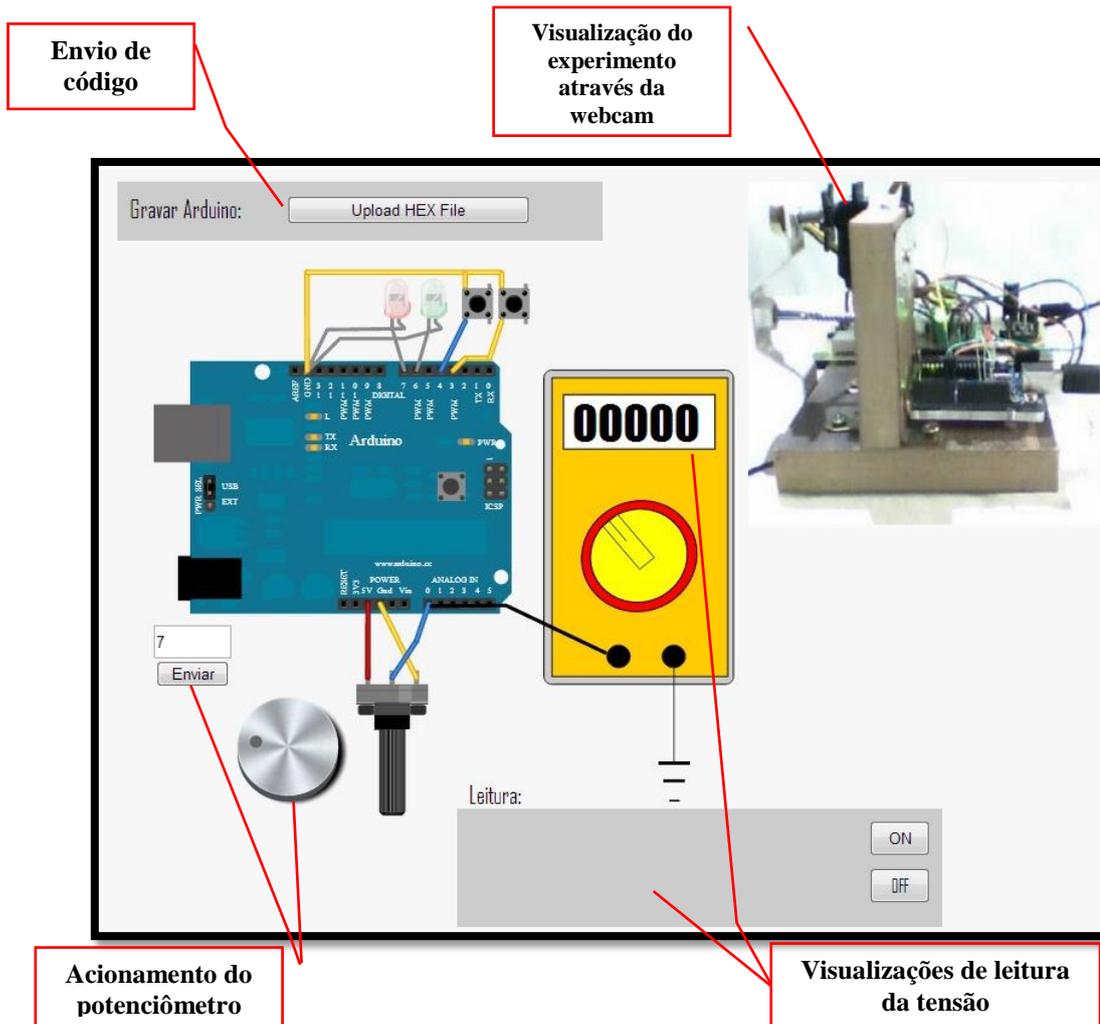
### 3.2.2 Programar Arduino

O usuário escreve seu código em linguagem de programação específica na IDE de desenvolvimento do *arduino* UNO (Figura 22) e após o processo de compilação gera um arquivo *hex* que é armazenado localmente na máquina do usuário.



**Figura 22:** Interface IDE *arduino*  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

Para transferir o código desenvolvido pelo usuário para ser gravado no *arduino*, esta ação é feita remotamente através da página principal da aplicação através do botão “*Upload HEX file*” (Figura 23).



**Figura 23:** Tela inicial dos experimentos  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

O usuário solicita ao servidor web, através de um navegador web padrão, uma página onde poderá enviar arquivos para o servidor. O servidor, através do navegador, disponibiliza a página e o carregamento desta acontece após o usuário clicar no botão “*Upload HEX file*”, ilustrado na Figura 23, acionando o método `<onclick="javascript:window.location.href='/laboratorios/{{ lab.id }}/upload"/>`. O servidor, através do navegador, disponibiliza a página para atualização gravação do arquivo no *firmware* do *arduino*. Em seguida, o usuário poderá selecionar o arquivo que desejar enviar. O servidor de aplicações processará a solicitação da transferência do arquivo quando o usuário fizer uso do botão “Escolher Arquivo” (Figura 24), para que este seja transferido para a placa do *arduino*.



**Figura 24:** Tela de transferência do arquivo programado pelo estudante  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

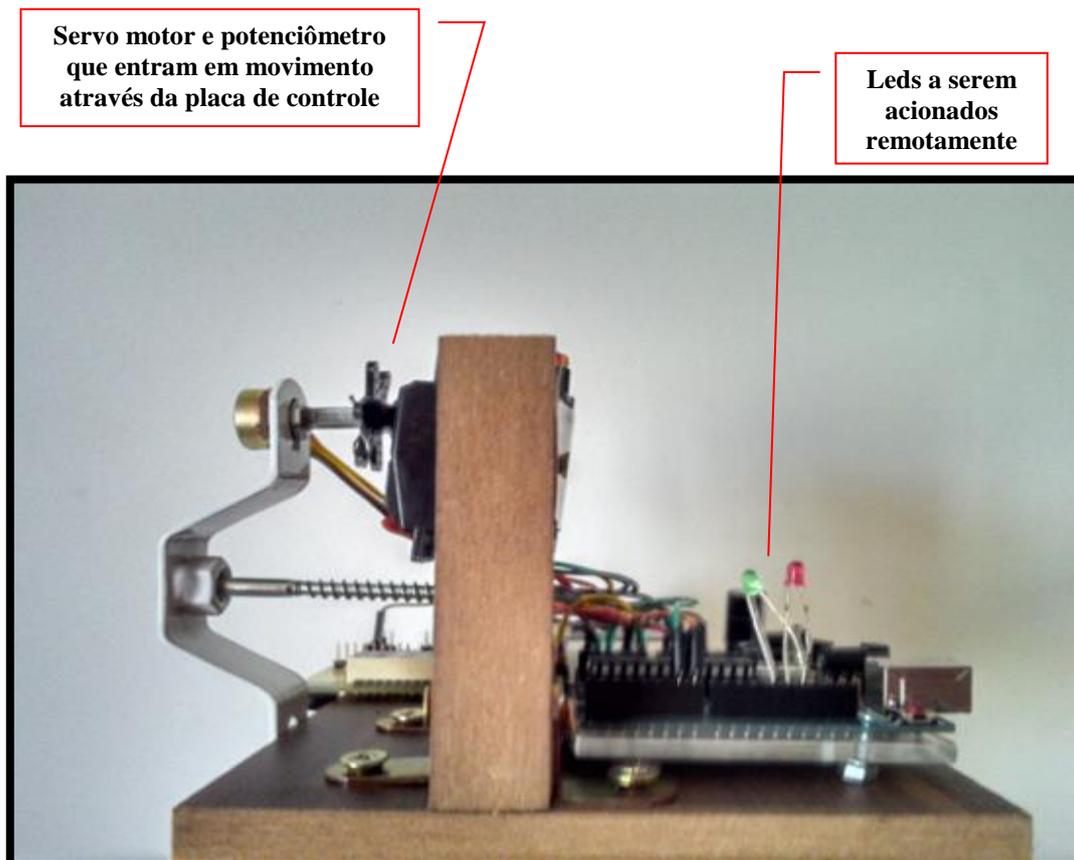
O LABVAE é um sistema que faz a aquisição de dados para a realização de experimento e testes em instrumentos eletrônicos para acionamento e medições. Esse sistema é baseado em computador e, do ponto de vista de *hardware*, é constituído de placas para controle, cabos e interfaces para conexão e uso de um protocolo de rede que viabiliza uma resposta em tempo real.

Quando um usuário está executando um experimento de medição de tensão na saída analógica do *arduino*, este faz a aquisição de comandos com valores que são necessários para rotacionar o servo motor que está acoplado a um potenciômetro (resistência variável).

A comunicação com o *hardware* das placas *arduino* e de controle pode ser estabelecida através da API *PyWebSocket*. A programação dos experimentos é enviada diretamente para a placa do *arduino*, a depender do tipo do laboratório que o estudante pretende realizar, possibilitando a visualização de informações em tempo real.

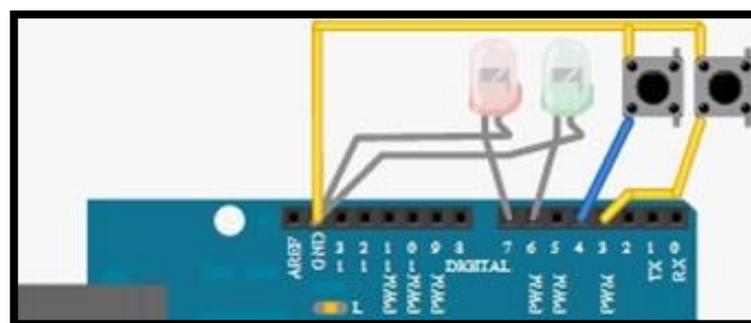
O LABVAE permite que os estudantes controlem o *login* e o acesso, para realizarem experimento para acionamentos de *leds* e medição de tensão. Outros laboratórios remotos podem ser construídos conforme necessidade de cada curso – Ciência da Computação, Engenharia Mecatrônica, Sistema de Informação e Redes de Computadores – possibilitando o compartilhamento de equipamentos de custo elevado e facilitando o gerenciamento dos laboratórios, o que é um desafio dentro das universidades.

A visualização do resultado do experimento na página web será feita pelo estudante por meio de gráficos e multímetro virtual, conforme apresentado na Figura 25. Desse modo, ele visualiza os efeitos do seu código desenvolvido por meio da imagem, que é transmitida pela Internet.



**Figura 25:** Imagem da WebCam exibida na tela inicial da aplicação  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

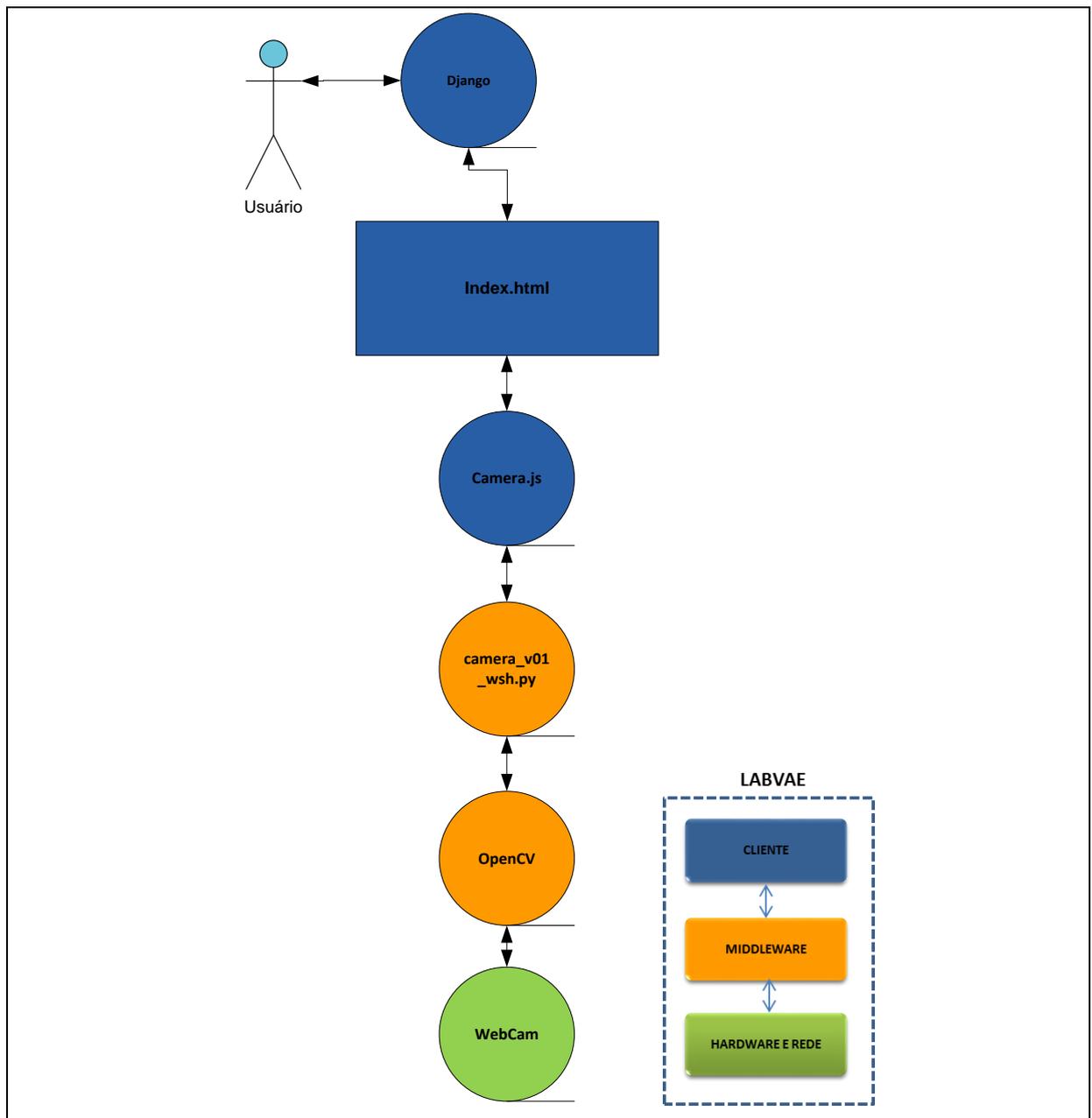
Para representar a visualização real exibida na figura acima, faz-se uso do SVG, conforme Figura 26. Trata-se de imagens escaláveis, ou seja, podem ser redimensionadas sem haver distorção, bem como o seu armazenamento ocupa menos espaço que o de uma imagem convencional *.jpeg* ou *.png*, pois as imagens do SVG são arquivos XML. Na proposta da presente pesquisa, as imagens que representam o acender e apagar os leds representados na Figura 26 foram construídas no formato SVG, aproveitando os benefícios deste tipo de imagem já citados anteriormente.



**Figura 26:** Simulação de Funcionamento de leds usando SVG  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa Fritzing (2013).

### 3.2.3 Aquisição WebCam

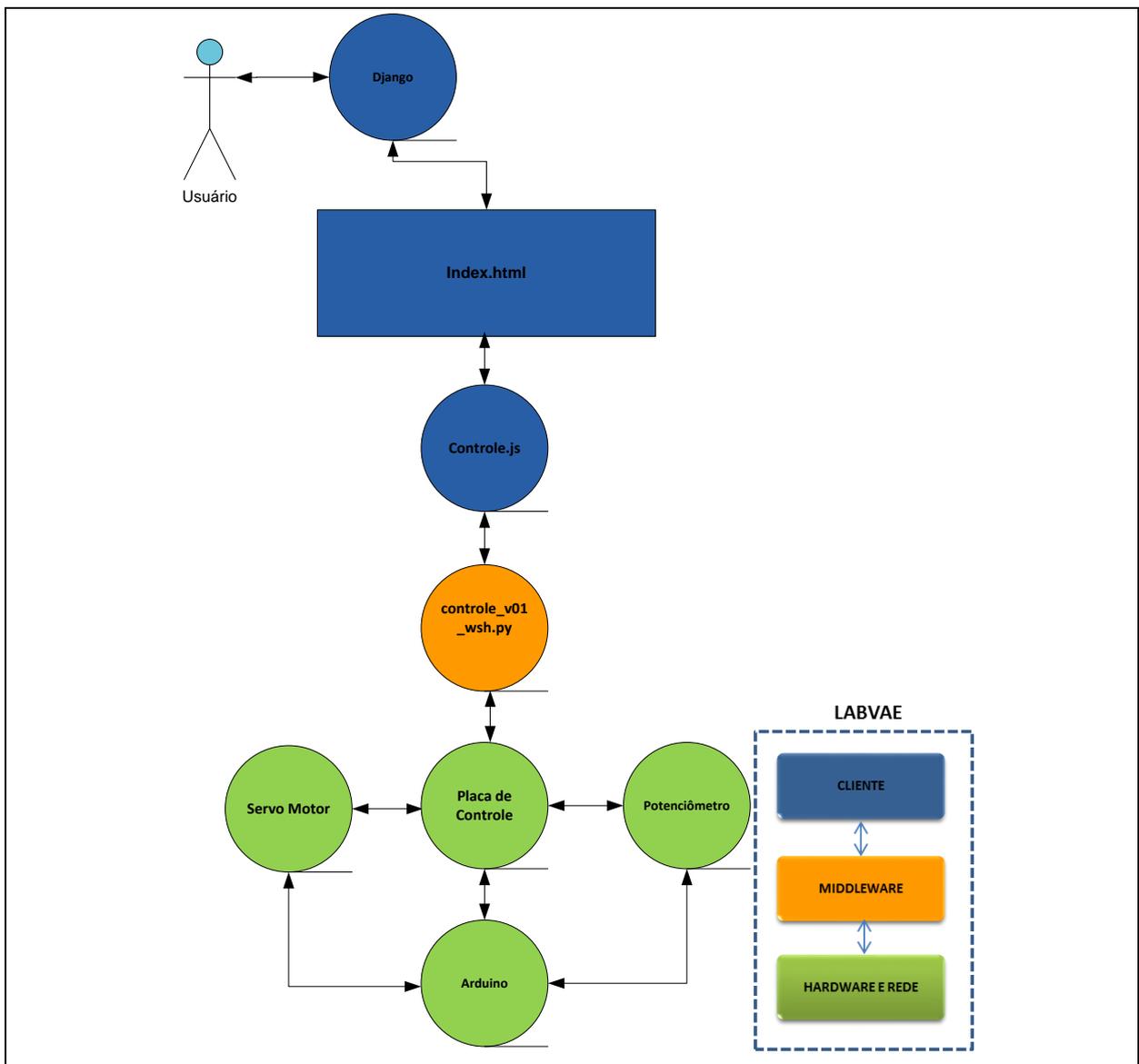
O diagrama de fluxo da Figura 27 representa os requisitos necessários para que a imagem seja exibida na página da aplicação do usuário. Para isso, cria-se o *WebSocket*, que inicializa a câmera e mantém atualizada a imagem *frame a frame* na página do usuário, pela biblioteca do *OpenVC* com o arquivo “*câmera\_v01\_wsh.py*”.



**Figura 27:** Fluxo de dados WebCam  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

### 3.2.4 Aquisição Placa de Controle

O fluxo de dados para a placa de controle, representado na Figura 28, é iniciado através da chamada do arquivo “*controle.js*” através do carregamento página web do usuário. Através deste arquivo é criada uma *WebSocket* na porta 9998 ((*ws\_controle = new WebSocket("ws://10.121.7.4:9998/controle\_v01")*)), invocando o arquivo “*controle\_v01.py*”, definindo a porta serial de comunicação *COM* para interagir com a placa de controle e fazer as aquisições, para aumentar e diminuir o valor que posicionará o servo motor através do botão de controle existente na página da aplicação, bem como o envio de valores específicos para o potenciômetro.

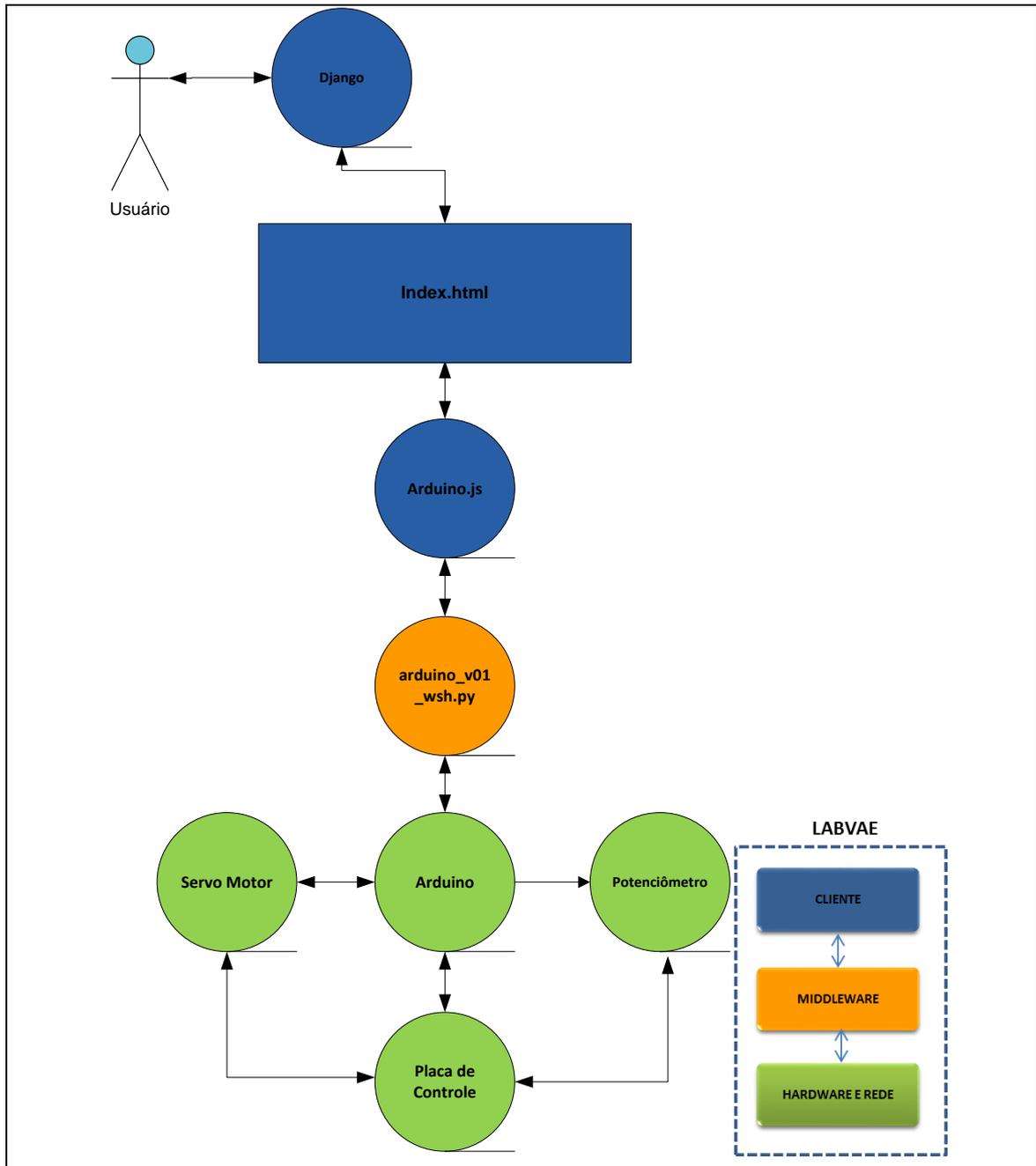


**Figura 28:** Fluxo de dados Placa de Controle

**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa Office Visio (2014).

### 3.2.5 Aquisição *Arduino*

O fluxo de dados para a placa de controle representado na Figura 29 é iniciado por meio da chamada do arquivo “*arduino.js*”, através do carregamento página web do usuário.



**Figura 29:** Fluxo de dados Placa de *Arduino*

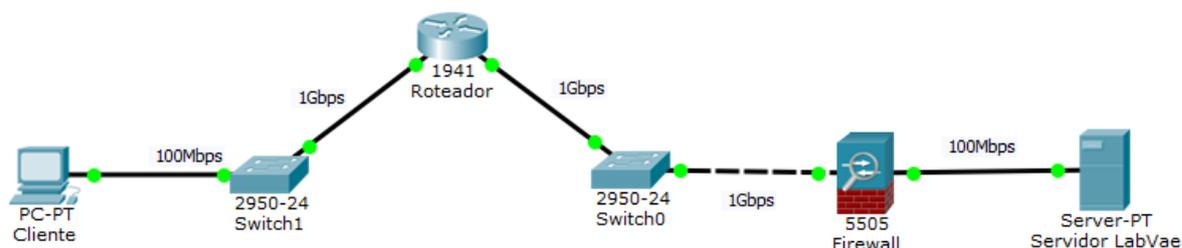
**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo programa Office Visio (2014).

Através deste arquivo, na porta 9998 (*ws = new WebSocket("ws://10.121.7.4:9998/arduino\_v01")*), invoca-se o arquivo “*arduino\_v01.py*”,

definindo a porta serial de comunicação *COM* para interagir com a placa *arduino* e atender as aquisições que vão simular a realismo dos controles para ascendimento dos leds, habilitar e desabilitar as aquisições oriundas do potenciômetro.

### 3.3 Coleta e Análise de Tráfego

Para a obtenção de amostras de tráfego do LABVAE vários experimentos foram realizados, em horas e dias diferentes, e com a realização de dois experimentos sendo um para medir tensão. A variedade de coletas tem como objetivo o aumento da representatividade dos resultados aqui apresentados. As coletas foram realizadas com o monitor de rede do Windows no cliente do LABVAE. Todas as medições foram feitas em um enlace com trechos diferenciados de velocidade de rede variando de 100Mbps a 1Gbps dentro do Campus Farolândia da Universidade Tiradentes. O diagrama do ambiente de medição montado pode ser visto na Figura 30. Será também feita uma análise de usabilidade da ferramenta, conforme especificado no capítulo 3 desta dissertação através da avaliação usabilidade dos alunos dos cursos de informática da Universidade Tiradentes.



**Figura 30:** Diagrama do Ambiente de Medição  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo Packet Tracer

A metodologia dos experimentos e as configurações utilizadas são explicadas nos próximos parágrafos.

Ambientes: Após uma avaliação preliminar do cenário de implantação do LABVAE, foi observada uma boa qualidade de desempenho da rede nos momentos e horários em que os testes foram realizados e, por consequência, foram significantes para o perfil de tráfego gerado. Desta forma, os experimentos foram realizados em horário normal de aula dentro do campus em função do maior tráfego de dados na rede.

Ações e fontes de tráfego externas: Cada experimento foi repetido em diferentes horários. Para melhor análise, foi levada em consideração a fonte de tráfego de vídeo da ferramenta LABVAE.

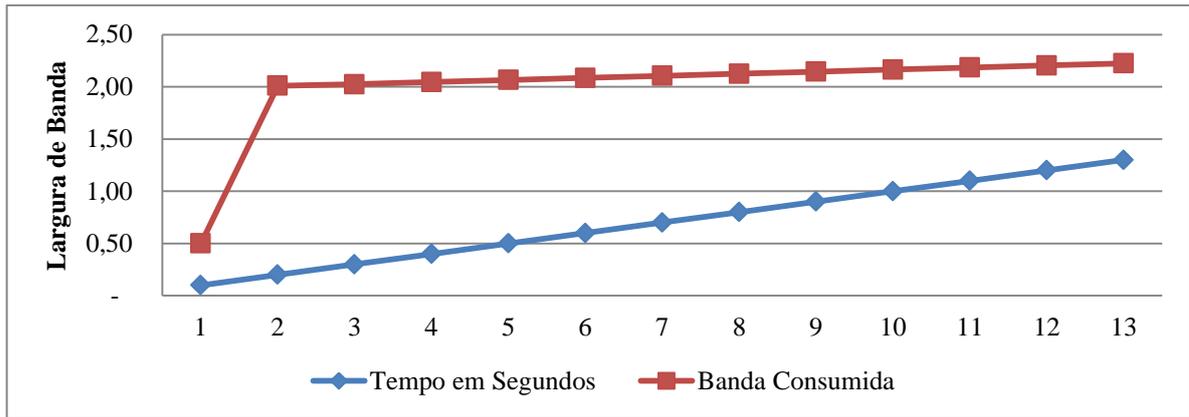
Métricas: As métricas avaliadas neste trabalho foram: Vazão, volume de tráfego (UDP/TCP), tamanho de pacote (incluindo os cabeçalhos TCP, UDP e IP), e tempo entre pacotes. Para capturar o tráfego gerado pelo cliente do LABVAE, foi utilizado o *sniffer wireshark*. Foi feita uma filtragem permitindo somente a captura do tráfego gerado pela comunicação entre cliente e servidor, evitando, assim, que o tráfego de outras aplicações interfira na qualidade dos resultados deste trabalho. Os resultados obtidos foram gravados em arquivos *.pcapng* e exportados para arquivos txt e posterior geração de gráficos utilizando o Excel. Os experimentos tiveram duração de 10 minutos e todas as métricas foram computadas continuamente, na medida em que os experimentos eram realizados. O tempo médio de cada experimento foi de 1,31min, em ambos os testes.

### **3.3.1 Perfil de Tráfego**

A seguir será realizada a análise do perfil de tráfego encontrado para a aplicação LABVAE. Métricas como vazão, volume de tráfego, tempo entre pacotes, entre outras, serão apresentadas e avaliadas. A latência na rede ficou variando entre 1 e 4 ms para ambos casos analisados.

#### **3.3.1.1 Análise de aproveitamento de pacotes**

A Figura 31 apresenta as séries temporais para a vazão da solução LABVAE proposta para um ambiente onde as aplicações eram testadas e avaliadas. Observando a vazão dos cenários coletados, nota-se que a velocidade da conexão de rede não afeta de maneira significativa a quantidade de tráfego gerado pela aplicação. O laboratório foi iniciado e, posteriormente, foram realizados testes para medição de tensão na placa de controle e acionamento de leds. Observou-se que a duração de tais experimentos foi de 1,30min, com o consumo de banda de 2,26 Mbps, para cada experimento. Tendo em vista que o total de experimentos realizados por 54 alunos, constatou-se um total consumo de banda de 122Mbps, num tempo de aproximadamente 1h10min.



**Figura 31:** Vazão (LABVAE)

**Fonte:** SANTOS, M. P. Autor do Projeto (2014), pelo Excel.

---

## 4 Avaliação do LABVAE

### 4.1 Avaliação Quantitativa

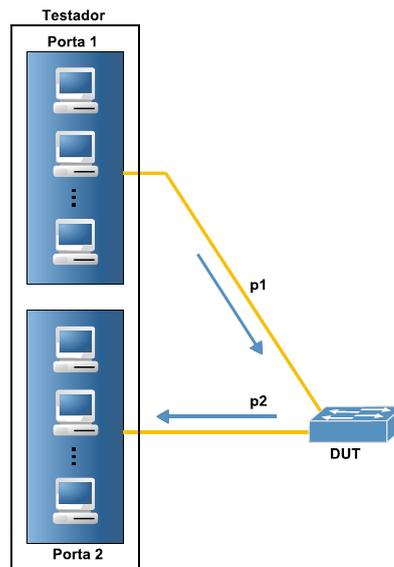
Partindo das premissas da RFC 2544, Taxa de transferência (Vazão) e Latência foi possível analisar a melhor forma de avaliar o desempenho da ferramenta desenvolvida (LABVAE), de modo a garantir que os serviços de tráfego proporcionem o nível de qualidade exigido e necessário para seu funcionamento.

A RFC 2544 descreve testes que podem ser usados para detalhar as características de desempenho de dispositivos de rede e o modo como os resultados devem ser apresentados. Assim, a RFC 1242 embasa o respectivo documento, introduzindo as terminologias para interconexão de dispositivos de rede (CECHIN, 2005).

Segundo Burgess (2004), devido à variedade de quadros – 64, 128, 256, 512, 1024, 1280 e 1518 bytes – os quais poderão ser usados na rede, sendo necessário conferir os resultados de cada um deles, o RFC 2544 define que sejam utilizados para teste quadros, que sejam enviados em um intervalo de tempo determinado e definido por um número estabelecido de vezes.

Conforme Bradner e McQuid (1999), perda de quadros, vazão, análise fim-a-fim (*back-to-back*) e latência são determinados pelos testes referidos no RFC 2544. O teste tráfego de quadros entre equipamentos envia um Burst (rajada de tráfego de curta duração) ao DUT (termo designativo do componente, equipamento ou sistema que deve ser testado), com espaços mínimos entre quadros e contar o número de quadros que forem encaminhados pelo DUT. O Burst terá seu comprimento aumentado e o teste será reexecutado, quando o número de quadros encaminhados de quadros transmitidos for igual. Caso o número de quadros encaminhados seja inferior ao número transmitido, o comprimento do Burst será reduzido e o teste será reexecutado. O número de quadros do Burst mais longo que o DUT consegue tratar sem haver perda de qualquer quadro é o valor fim-a-fim.

Um característico modelo de testes arquitetado para atender aos critérios de análise do RFC 2544 é mostrado na Figura 32:



**Figura 32:** Modelo de testes para a RFC 2544  
**Fonte:** Costa (2008, p. 16).

Os resultados de todos os testes, por recomendação do teste embasado no RFC 2544, devem ser expressos em texto e gráfico.

No que diz respeito à taxa de transferência, pelo fato de a vazão de dados expressar a quantidade máxima de dados, que pode ser conduzida de uma origem até o seu respectivo destino, o conceito de desempenho da rede, geralmente, é relacionado à sua velocidade.

Contudo, a complexidade em definir e medir a vazão é atribuída à necessidade de definir um nível aceitável de qualidade. Um exemplo é considerar aceitável 10% de quadros com erros ou perdidos, o que leva a vazão a ser medida a uma taxa de erros de 10%. A banda passante máxima absoluta, seja qual for o sistema Ethernet, corresponderá à taxa de dados, a exemplo de 10 Mbit/s, 100 Mbit/s ou 1000 Mbit/s. Em termos reais, esses números não podem ser alcançados, devido aos campos adicionais para que os quadros possam ser transportados e pelo espaçamento entre quadros, necessário para o funcionamento da rede (COSTA, 2008).

Conforme Burgess (2004), quanto aos pacotes menores, a vazão efetiva é menor em relação aos pacotes maiores, considerando que os bytes de preâmbulo e do espaço entre pacotes incluídos não contam como dados.

Quanto à Latência é definida como o tempo total consumido por um quadro, desde sua origem até o seu destino. Ou seja: É um tempo absoluto, que resulta da soma dos atrasos do processamento nos elementos da rede e o atraso de propagação ao longo do meio de transmissão (BURGESS, 2004).

A medida da latência é efetuada através da transmissão, pela rede, de um quadro de teste contendo uma marca de tempo (*timestamp*), a qual é analisada quando o quadro é recebido. Este processo exige que o quadro de teste retorne ao testador original através de um laço de retorno, o que caracteriza o atraso de ida e volta. Todavia, uma maior latência não significa que resultará degradação de voz, mas é possível ocorrer uma perda de sincronização.

Um exemplo é que uma boa qualidade em conexões telefônicas exige que a latência tenha um valor inferior ao patamar de 150 ms, o que, por sua vez, requer que sejam tomadas medidas de forma que determinem as alterações que deverão resultar na diminuição do tempo de empacotamento, transmissão e transporte dos dados.

## **4.2 Avaliação Qualitativa**

Um dos objetivos da avaliação qualitativa é validar a viabilidade do modelo do LABVAE com a finalidade que seja fornecida abstrações significativas aos usuários e programadores de aplicação, facilitando a configuração e programação de experimentos eletrônicos.

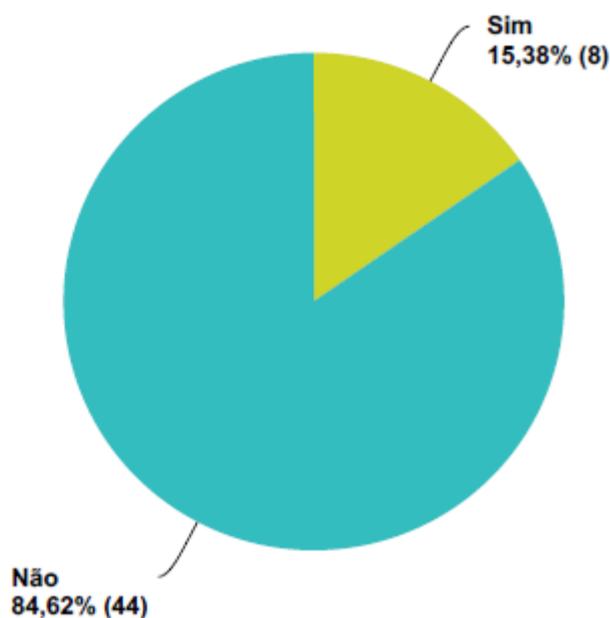
Através do LABVAE, o docente pode disponibilizar alguns circuitos eletrônicos previamente diagramados e elaborados, e conseqüentemente requisitar ao aluno que realize programação dos laboratórios para que seja possível realizar relacionado ao comportamento de um determinado dispositivo. Fazendo uso dos instrumentos de controles adequado existente na interface web do laboratório, os alunos podem conferir as respostas encontradas teoricamente com aquelas obtidas de maneira prática, através de um experimento realizado remotamente. Para que seja possível realizar esta atividade, é necessário que os usuários cadastrem-se no site que oferece os experimentos disponibilizados pelo LABVAE. Nela o usuário informa o tempo em que pretende ficar utilizando o instrumento eletrônico, a data e o horário de realização do experimento, e, por fim, lhe é disponibilizada uma listagem dos experimentos de modo que ele selecione aquele que será utilizado. Caso o horário escolhido para posse do dispositivo já tenha sido reservado, uma mensagem será exibida e será requisitado que o usuário informe um novo horário para reserva do laboratório.

---

## 5 Análise dos Resultados dos Questionários

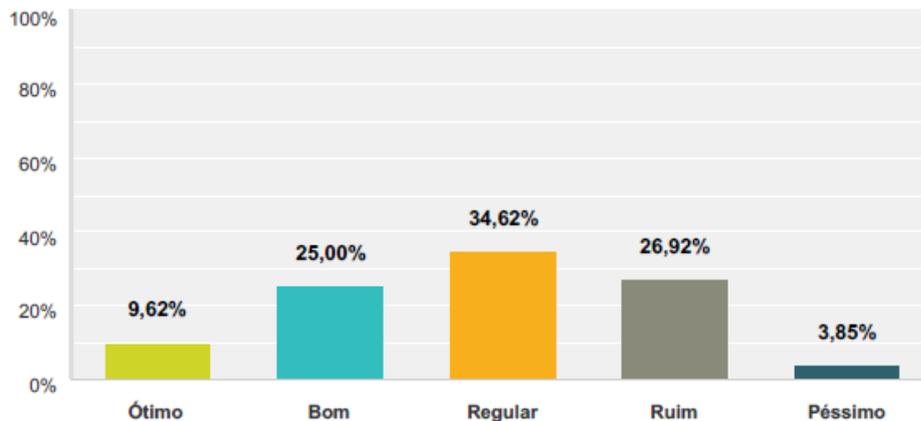
Dos 54 estudantes que participaram da pesquisa, respondendo o questionário, 88,7% correspondem aos indivíduos do sexo masculino. Quanto à idade, que variou de 17 a mais de 50 anos, houve prevalência (44,44%) da faixa etária de 21 a 24 anos de idade, seguida de percentual também significativo (33,33%) daqueles com idade de 17 a 20 anos.

Quanto à utilização de alguma ferramenta similar ao LABVAE, conforme ilustrado na Figura 33, nota-se que significativa parte dos estudantes (84,62%) não havia utilizado, cabendo destacar que dois dos participantes não responderam a referida questão.



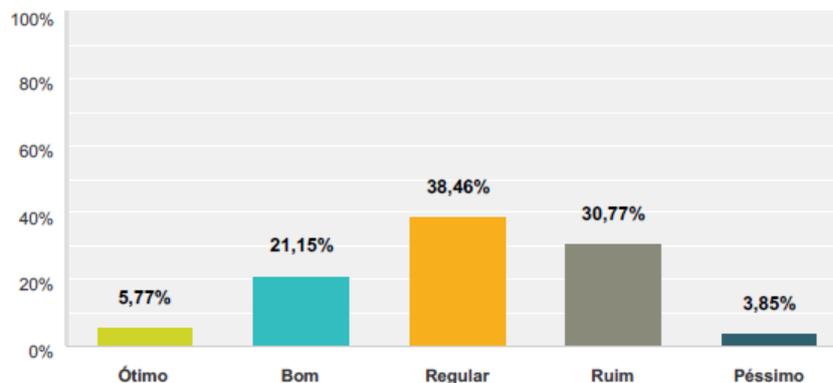
**Figura 33:** Utilização de ferramenta similar ao LABVAE  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

Acerca do domínio do estudante sobre o *arduino*, é possível observar, pela Figura 34, a prevalência da opção “regular” (34,62%), enquanto apenas 9,62% avaliaram o próprio domínio sobre o *arduino* como “ótimo”. Assim como ocorreu na questão anterior, dois estudantes não responderam.



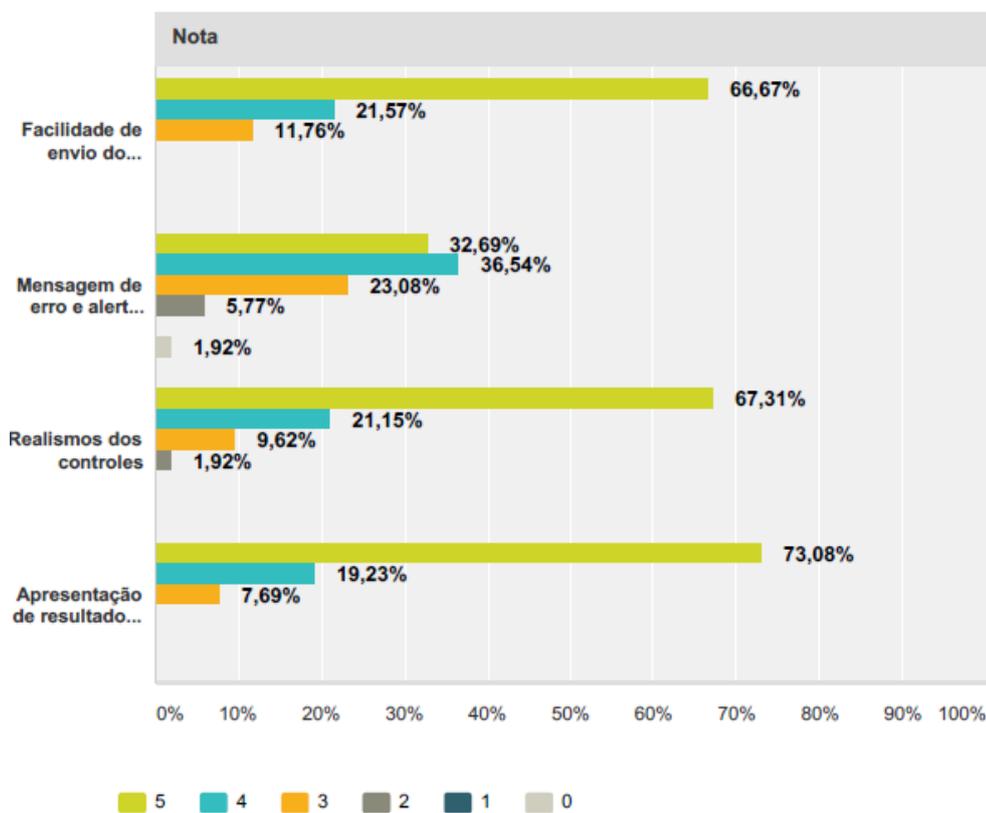
**Figura 34:** Avaliação do domínio sobre o *arduino*  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

A respeito do conhecimento sobre linguagem de programação C, excetuando dois dos participantes, os quais não responderam a referida questão, “regular” e “ruim” apresentam percentuais aproximados, respectivamente, 38,46% e 30,77%, à medida que 3,85% assinalaram que têm “péssimo” conhecimento, como mostrado na Figura 35.



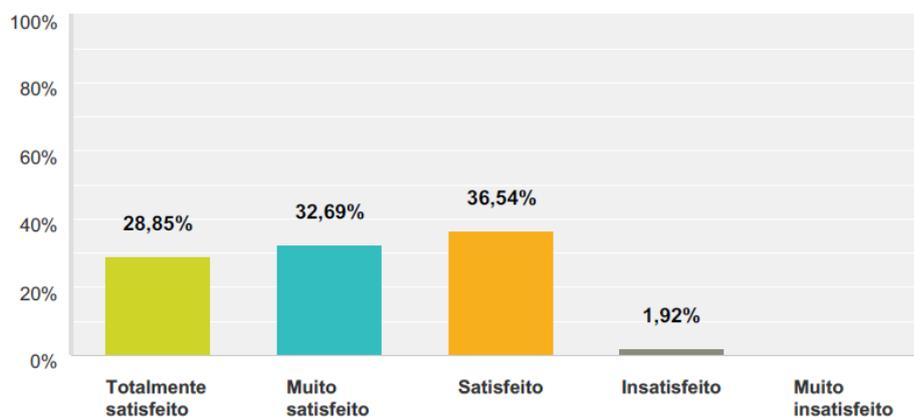
**Figura 35:** Avaliação do conhecimento sobre linguagem de programação C  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

Em outra questão, os participantes foram solicitados a avaliar, em uma escala de 0 a 5 (respectivamente de muito ruim a muito bom) a ferramenta LABVAE, em relação aos seguintes itens: Facilidade de envio do experimento; Mensagem de erro e alertas do sistema; Realismos dos controles e Apresentação de resultados dos experimentos. Conforme ilustrado na Figura 36, prevalece a opção “muito bom” em três dos itens e “bom” para o item Mensagem de erro e alerta do sistema. Este mesmo item foi o único avaliado como “muito ruim” (1,92%).



**Figura 36:** Avaliação do LABVAE por comparação com os itens descritos  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

Ao serem solicitados a avaliar a satisfação em relação aos experimentos disponibilizados pela ferramenta LABVAE, conforme ilustrado na Figura 37, o grupo de variáveis que expressa satisfação em alguma intensidade (98,08%) é significativamente superior ao grupo oposto, o qual representa apenas 1,92% dos participantes.



**Figura 37:** Satisfação relacionada aos experimentos disponíveis no LABVAE  
**Fonte:** SANTOS, M. P. Autor do Projeto (2014).

---

## 6 Conclusões

A partir da proposta deste trabalho, que consistiu em realizar um estudo de caso para medição, modelagem e simulação, fazendo de *software* e de *hardware* para a prática de laboratório remoto para fixação de conteúdo para os cursos de engenharia e computação, foi possível constatar que:

As tecnologias apresentadas foram satisfatórias quanto ao atendimento dos requisitos de eficiência dos resultados e de avaliação do desempenho, mostrando que o LABVAE faz uso intensivo dos recursos de rede, tais como: um aproveitamento de 97,57% do total de pacotes transmitidos; uma média de 23,64 pacotes por segundo; uma média de tamanho de 875 bytes por segundo e uma largura de banda de 24,28Mbps.

A proposta de uma arquitetura de sistema para desenvolver um laboratório de acesso remoto, visando à possibilidade de realizar experimentos remotos, facilitou a interação do usuário com o LABVAE, em virtude da permissão do acesso remoto ao experimento, através de uma interface disponível na Internet.

A estrutura, para medição, análise e modelagem de tráfego possibilita o levantamento de requisitos mínimos necessários para especificação da rede a ser utilizada.

O laboratório de acesso remoto viabiliza o uso de um número maior de pessoas, facilitando a aprendizagem de conteúdo com um custo menor investido em infraestrutura e recursos humanos. Além disso, incentiva o aluno, promovendo uma participação mais ativa no processo de ensino e aprendizagem.

Dentro do contexto de arquitetura de sistema que suporte a prática de laboratório remoto e a colaboração em conteúdos ministrados em sala de aula, um sistema integrado a uma plataforma (servidor e middleware da aplicação), proposto neste trabalho, possibilita o desenvolvimento de novos experimentos no escopo de laboratórios de acesso remoto.

O protocolo TCP foi utilizado significativamente nos cenários com o streaming de vídeo e para envio dos comandos de troca de mensagens. Assim, conclui-se que o cliente LABVAE utiliza o protocolo *WebSocket* para o transporte das mensagens de controle, bem como o protocolo TCP para streaming de vídeo.

Desse modo, a solução desenvolvida neste trabalho, denominada LABVAE, possibilitou a criação de experimentos remotos a partir de um conjunto de funcionalidades de *software*, que isolam completamente a complexidade do *hardware*, o qual pode ser utilizado de forma

genérica. Assim, ao apresentar a modelagem da plataforma LABVAE e sua forma de desenvolvimento, a arquitetura da solução proposta disponibilizou a execução de experimentos remotos através da Internet ou de uma rede local.

Portanto, com base nas inferências até aqui expostas e na avaliação a qual foram submetidos 54 alunos, é possível corroborar a hipótese deste estudo, tendo em vista que o desenvolvimento da arquitetura proposta é aplicável dos pontos de vista de usabilidade, disponibilidade e funcionalidade.

## **6.1 Contribuições**

As contribuições deste trabalho são listadas a seguir:

- O conhecimento das tecnologias que fazem parte do LABVAE, com detalhes sobre os servidores utilizados, protocolos e suas respectivas funções;
- Uma análise do perfil de tráfego gerado pelos clientes e servidores do LABVAE, mostrando resultados relevantes sobre vazão, volume de tráfego (TCP), dentre outros;
- Desenvolvimento de uma metodologia para modelagem de um laboratório virtual de aprendizagem tomando como base tecnologias, a exemplo do HTML 5;
- Criação de uma estrutura para medição, análise e modelagem de tráfego, compreendendo ferramentas como *Sniffer*, analisadores de tráfego;
- Aplicação de questionário para testar as funcionalidades da ferramenta LABVAE;
- Os mecanismos de medições e simulações criadas neste trabalho avaliam o modelo do LABVAE;
- Publicação de artigo em conferência Nacional;
- Publicação de artigo em conferência Regional;
- Submissão de Artigo a revista IEEE da América Latina.

## **6.2 Trabalhos Futuros**

Apesar serem frequentes pesquisas nessa linha de trabalho, outras questões podem ser tratadas em trabalhos futuros, dentre elas:

- Fazer uso da estrutura do trabalho proposto para avaliar outros laboratórios virtuais de aprendizagem realizando comparações de análise de tráfego.
- Estudar e usar novas tecnologias para aprimorar e dar continuidade a melhorias ao trabalho proposto nesta dissertação.
- Por se tratar de uma arquitetura aberta de *hardware* e *software*, estudos futuros podem aprimorar e avaliar o trabalho proposto para outras funcionalidades dentro da mesma linha de pesquisa.

---

## Referências

BLAHA, M.; RUMBAUGH, J. **Modelagem e projetos baseados em objetos com UML2**. Tradução por Daniel Vieira 2. ed. Rio de Janeiro: Elsevier, 2006;

BRADNER, S.; MCQUID, J. **Benchmarking Methodology for Network Interconnect Devices**: RFC 2544. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999;

BURGESS, N. **Testing of Ethernet Services in Telecom Networks**: RFC 2544. [S.l.]: Agilent Technologies, 2004;

CECHIN, S.L. **Avaliação de Desempenho em Redes de Computadores**. 2005. Tese (Doutorado em Ciência da Computação) – Instituto de Informática UFRGS. Porto Alegre, 2005;

CHOUDHARY, A.; SINGH, S. A.; MALIK, M. F. *et al.* Virtual lab: Remote access and speed control of DC motor using Ward-Leonard system. Technology Enhanced Education (ICTEE), **IEEE International Conference**, 2012;

COMER, D. E. **Interligação de redes com TCP/IP**: princípios, protocolos e arquitetura. Rio de Janeiro: Campus, 2006;

COSTA, Giovani Hoff da. **Métricas para avaliação de desempenho em redes QoS sobre IP**. 2008. 42f. Monografia (Especialização em Tecnologias, Gerência e Segurança de Redes de Computadores) – Universidade Federal do Rio Grande do Sul – Instituto de Informática. Porto Alegre/RS, 2008;

CUBILLO, J.; MARTIN, S.; CASTRO, M.; MEIER, R., Control of a remote laboratory by augmented reality. Teaching, Assessment and Learning for Engineering (Tale), **IEEE International Conference**, 2012;

GARCIA-VALLS, M.; BASANTA-VAL, P. Usage of DDS data-centric *middleware* for remote monitoring and control laboratories. **Industrial Informatics**, IEEE Transactions, 2013;

GROUT, I.; MURPHY, C.; DA SILVA, A. C. R. Remote laboratory experiment access via an RFID interface. Remote Engineering and Virtual Instrumentation (REV). In: **IX International Conference**, 2012;

HICKSON, I. **The Web Sockets API**. W3C Working Draft 29 October 2009. Disponível em: <<http://www.w3.org/TR/2009/WD-WebSockets-20091029/>>. Acesso em: 01 mar. 2014;

HUA, J.; GANZ, A. A new model for remote laboratory education based on next generation interactive technologies. **Microsoft Research Conference XP**. Disponível em: <<http://researchers.conferencexp.net/Lists/Research%20Papers2/Attachments/5/aseeivlab.pdf>>, abril, 2003. Acesso em: 28 nov. 2012;

LERRO, F.; MARCHISIO, S.; MARTINI, S. *et al.* Integration of an e-learning platform and a remote laboratory for the experimental training at distance in engineering education. Remote engineering and virtual instrumentation (REV). In: **IX International Conference in Bibao**, 2012;

LOWE, D.; DE LA VILLEFROMOY, M.; JONA, K.; YEOH, L. Remote laboratories: Uncovering the true costs. Remote Engineering and Virtual Instrumentation (REV). In: **IX International Conference**, 2012;

LOWE, D.; OROU, N. Interdependence of booking and queuing in remote laboratory scheduling. Remote Engineering and Virtual Instrumentation (REV). In: **IX International Conference**, 2012;

LUBBERS, P.; BRIAN, A.; SALIM, F. **Pro HTML5 programming**. 2. ed. USA: Editora Apress, 2011;

MAAROUF, S.; RADHI, M.; VAHE, N. *et al.* Collaborative activities in the remote laboratory work. Interactive Collaborative Learning (ICL). In: **XV International Conference**, 2012;

MAITI, A.; MAHATA, S.; MAITI, C.K. Common interface platform for development of remote laboratories. Remote Engineering and Virtual Instrumentation (REV). In: **IX International Conference**, 2012;

MOREIRA, V.R.; CARDOSO, F.A.C.M.; ARANTES, D.S. Plataforma em *hardware* reconfigurável para o ensino e pesquisa em laboratório de sistemas digitais a distância. In: **XIX Simpósio Brasileiro de Informática na Educação**. Campinas-SP, 2008. p. 542-551;

NETWORK WORKING GROUP. **A TCP/IP Tutorial**. Disponível em: <<https://tools.ietf.org/html/rfc1180>>. Acesso em: 28 nov. 2012;

PIMENTEL, V.; NICKERSON, B. G. Communicating and Displaying Real-Time Data with *WebSocket*. In: **IEEE transactions on industrial informatics**, v. 4, n. 16, p. 45-53, jul./ago., 2012;

SAMUELSEN, D. A. H.; GRAVEN, O. H. Virtual fences as protection against damage on physical equipment used in remote laboratories. **Remote Engineering and Virtual Instrumentation (REV)**. 9. Ed. International Conference, 2012;

SANTANA, I.; FERRE, M.; IZAGUIRRE, E. *et al.* **Laboratories for education and research purposes in automatic control systems industrial informatics**. Vol. 9. IEEE Transactions, 2013;

SANTANA, I.; HERNANDEZ, L.; FERRE, M.; *et al.* Analysis of a Thermal system through remote laboratories. **Global Engineering Education Conference (Educon)**, IEEE, 2012;

SANTOS, N. Agentes de *Software* em Ambientes Educacionais Mediados por Computador. **Revista Brasileira de Informática na Educação**, v. 11, n. 1, 2003;

SCHAFER, T.; SEIGNEUR, J. M.; DONELLY, A. **PEARL**: A Generic Architecture for Live Experiments in a Remote Lab. 2002. Disponível em:  
<<http://iet.open.ac.uk/pearl/publications/icsee03.pdf>>. Acesso em: 28 nov. 2012;

SEVERINO, A. J. **Metodologia do trabalho científico**. 23. ed. rev. e atual. São Paulo: Cortez, 2007;

SILVA, M. **A era da interatividade**. Disponível em:  
<[http://www.saladeaulainterativa.pro.br/leia\\_educacao.htm](http://www.saladeaulainterativa.pro.br/leia_educacao.htm)>. Acesso em: 02 abr. 2014;

SQLite. Disponível em: <<https://SQLite.org/>>. Acesso em: 28 nov. 2012;

**WEBSOCKET-SAMPLE**. **PyWebSocket works like as follows** (protocol sequence and indicating when api works). 20 jan. 2010. Disponível em:  
<<http://code.google.com/p/WebSocket-sample/wiki/Tips>>. Acesso em: 28 nov. 2012;

YANAZE, L.H.K. **Tecno-pedagogia**: os aspectos lúdicos e pedagógicos da comunicação digital. 2009. 235f. Dissertação (Mestrado em Ciências da Comunicação) – Universidade de São Paulo – Escola de Comunicação e Artes. São Paulo, 2009;

ŽOVÍNOVÁ, Michaela; OŽVOLDOVÁ, Miroslava. New multimedia teaching tool using remote physics experiments. In: **14th International Conference on Interactive Collaborative Learning**, 21-23 set., pp. 395-399, 2011;

## Apêndice

### Questionário

#### Utilização do Laboratório Virtual de Aprendizagem Eletrônica (LABVAE)

**1) Faixa etária**

- 17 a 20 anos     21 a 24 anos     25 a 28 anos  
 29 a 32 anos     33 a 40 anos     Mais de 50 anos

**2) Sexo**

- Masculino  Feminino

**3) Em qual curso você está matriculado?**

- Ciência da Computação     Sistema de Informação  
 Mecatrônica     Redes de Computadores

**4) Você já utilizou na prática alguma solução similar ao LABVAE?**

- Sim  Não

**5) Como você avalia o seu domínio sobre o *Arduino*?**

- Ótimo     Bom     Regular     Ruim     Péssimo

**6) Como você avalia o seu conhecimento sobre linguagem de programação C?**

- Ótimo     Bom     Regular     Ruim     Péssimo

**7) Avalie em uma escala de 0 a 5, onde 0 é muito ruim e 5 muito bom, a solução LABVAE em relação aos itens descritos abaixo:**

- \_\_\_ Facilidade de envio do experimento  
\_\_\_ Mensagem de erro e alertas do sistema  
\_\_\_ Realismos dos controles  
\_\_\_ Apresentação de resultados dos experimentos

**8) Como você avalia a sua satisfação em relação aos experimentos disponíveis na solução LABVAE?**

- Totalmente satisfeito     Muito satisfeito     Satisfeito  
 Insatisfeito     Muito insatisfeito

**9) Você acha interessante que a complementação dos conteúdos das aulas fosse aplicados desta forma?**

- Sim  Não

**10) Você gostaria de sugerir melhorias para o LABVAE?**