

**PPGI** PROGRAMA  
DE PÓS-GRADUAÇÃO  
EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

PAULO ROBERTO DE AZEVEDO SOUZA

DISSERTAÇÃO DE MESTRADO

**LABVAD: DESENHO E IMPLEMENTAÇÃO DO  
LABORATÓRIO VIRTUAL DE ATIVIDADES DIDÁTICAS  
COM ROBÓTICA**

RIO DE JANEIRO  
2015

  
Instituto de Matemática

  
Instituto Tércio Pacitti de Aplicações  
e Pesquisas Computacionais



**PPGI** PROGRAMA  
DE PÓS-GRADUAÇÃO  
EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

## **LABVAD: DESENHO E IMPLEMENTAÇÃO DO LABORATÓRIO VIRTUAL DE ATIVIDADES DIDÁTICAS COM ROBÓTICA**

Paulo Roberto de Azevedo Souza  
prasouza@ufrj.br

Orientadores: Fábio Ferrentini Sampaio, Ph.D.  
Marcos da Fonseca Elia, Ph.D.

Área de Pesquisa:  
Informática, Educação e Sociedade

**RIO DE JANEIRO  
DEZEMBRO DE 2015**

**PAULO ROBERTO DE AZEVEDO SOUZA**

**LABVAD: DESENHO E IMPLEMENTAÇÃO DO  
LABORATÓRIO VIRTUAL DE ATIVIDADES  
DIDÁTICAS COM ROBÓTICA**

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Informática do Instituto de Matemática e do Instituto Tércio Pacitti da Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Aprovada em 02 de dezembro de 2015.

---

Prof. Fábio Ferrentini Sampaio, Ph.D., NCE e PPGI/UFRJ (Presidente)

---

Prof. Marcos da Fonseca Elia, Ph.D., NCE/UFRJ (Coorientador)

---

Prof. Josefino Cabral Melo Lima, Docteur, PPGI/UFRJ

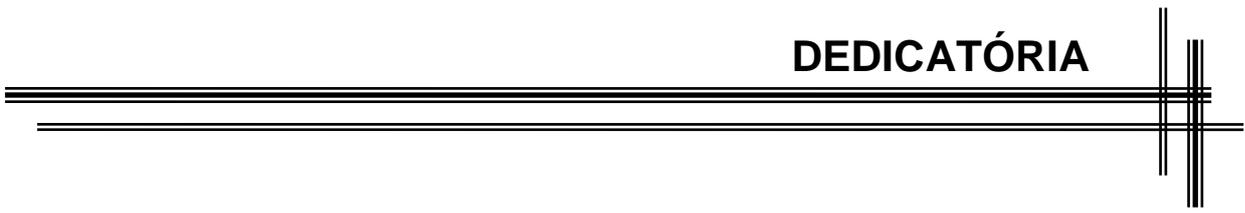
---

Prof. Adriano Joaquim de Oliveira Cruz, Ph.D., UFRJ

---

Prof. Leonardo Cunha de Miranda, D.Sc., UFRN

## DEDICATÓRIA



Para meus pais, minha filha Isabel e, especialmente,  
Viviane, que me fez encontrar o meu caminho de volta.

## AGRADECIMENTOS



“Que darei eu ao Senhor, por todos os benefícios que me tem feito?”.

Salmos 116:12.

Nesta intensa jornada do mestrado, creio que consegui realizar o sonho de construir algo que julgo útil para informática educativa, norteado pelas fantásticas orientações de Fábio Ferrentini e Marcos Elia.

Pela primeira vez, em minha vida, percebo que vale a pena olhar para trás e saber o que eu fui e como amadureci. Entrei no PPGI com 2 artigos e hoje possui 7 artigos publicados, sendo um deles, a realização de profissional de escrever com o Ferrentini e Elia.

Em função da imensa quantidade de artigos lidos, consegui me corresponder com pessoas como: Dan Cross, Dr. Vassilis, Sharona Levy, pesquisadores da Northwestern e até do MIT.

Agradeço a oportunidade de aprender e escrever com Marcos Borges, Paulo Pires, José Orlando, além dos seminários de acompanhamento com meus orientadores, Cláudia Mota, Priscila e Carlo Emmanoel, ou seja, a ELITE.

Não posso me esquecer de que sempre pude contar com o prof. Elia, Fábio, Angélica, Lucimeri, César, Serafim e o André, que além de colega, se tornou amigo e padrinho de casamento.

Agradeço ao apoio dos amigos: Rafael Souza, Léo Procópio, Lucimeri Ricas, Charlis, Verônica, Risomar, Viviane (minha esposa), minha família e, em especial, ao Roberto (meu pai), que enfrentou comigo cada “viagem” de 240 km para estudar no NCE.

*“Programar é um trabalho criativo, gigantesco e recompensador. Você pode criar complexidades fantásticas, bem como rotinas inteligentes, devastadoramente rápidas e enganosamente simples para superar todos os obstáculos”.*

(Eben Upton)

Cocriador do Raspberry Pi

SOUZA, Paulo Roberto de Azevedo. **Labvad: Desenho e Implementação do Laboratório Virtual De Atividades Didáticas Com Robótica**. 2015. 131 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

O presente trabalho possui como objetivo a proposição, o desenvolvimento e a validação conceitual de um laboratório virtual de acesso remoto para atividades didáticas com Robótica. Trata-se de um ambiente que foi construído sobre a plataforma eletrônica de baixo custo do projeto Arduino, conectada a alguns dispositivos eletrônicos, tais como displays, motores, sensores, LEDs e câmeras.

O design da plataforma é completamente baseado em softwares livres, utilizando tecnologias como: PHP, HTML, CSS, JavaScript e MySQL. Esta plataforma consiste num ambiente online, de acesso livre, multiplataforma e compatível com a maioria dos navegadores, via internet.

O gene da sua construção surgiu devido ao alto custo dos kits didáticos de hardware disponíveis para o ensino de Robótica, bem como a carência de laboratórios equipados com tais Kits. Diante deste panorama, o uso de laboratórios remotos acessados via internet disponível para professores e alunos 24 horas, 7 dias por semana pode ser uma alternativa didática viável. Este é exatamente o caminho abordado pelo presente trabalho.

Esta dissertação apresenta o panorama mundial do uso de laboratórios remotos no ensino de ciências e robótica, através de uma revisão de literatura consistente. Também serão descritas avaliações de processo e produto (avaliação final), obtidas através da análise das Interfaces Humano – Computador, constituída na plataforma do LabVad.

Os trabalhos futuros sobre o LabVad já estão em curso, e foram contemplados por um edital da Rede Nacional de Ensino e Pesquisa (RNP) em outubro de 2014.

Uma associação de pesquisadores, entre a Universidade Federal do Rio de Janeiro (UFRJ) e a Universidade Federal do Rio Grande do Norte (UFRN), propõe a utilização de laboratórios remotos no ensino de ciências e Robótica Educacional no sistema escolar brasileiro.

Palavras-chave: Arduino, hardware livre, laboratórios remotos, robótica educacional, ensino a distância.

## ABSTRACT

---

---



SOUZA, Paulo Roberto de Azevedo. **Labvad: Desenho e Implementação do Laboratório Virtual De Atividades Didáticas Com Robótica**. 2015. 131 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

Keywords: Arduino, Open-source Electronic Platform, Remot Labs, Educational Robotics, Distance Education Course.

This work aims the proposition, development and conceptual validation of a virtual laboratory with remote access for educational activities in Robotics. It is an environment that was built based on the Arduino Project and connected to some electronic devices, such as displays, motors, sensors, LED and cameras.

The design of the platform is completely based on open source software, using technologies like: PHP, HTML, CSS, JavaScript and MySQL. This platform consists of an online environment, with free access, cross-platform and compatible with most browsers via the Internet.

The gene of its construction came about due to the high cost of educational hardware kits available for teaching Robotics, as well as the lack of laboratories equipped with such kits. Faced with this viewpoint, the use of remote laboratories accessed via the Internet, available for teachers and students 24 hours, 7 days a week is a viable alternative teaching. This is exactly the way approached by this work.

This dissertation presents the global overview of using remote laboratories in Science teaching and Robotics, through a consistent literature review. Evaluations of process and product are also described (final evaluation), obtained by the analysis of the Human-Computer Interfaces, constituted in LabVad platform.

The future work about the LabVad is already underway, and was covered by an official announcement of the National Network of Education and Research (RNP) in October 2014.

An association of researchers, from the Federal University of Rio de Janeiro (UFRJ) and the Federal University of Rio Grande do Norte (UFRN) proposes the use of remote laboratories in Science teaching and Educational Robotics in the Brazilian school system.

## LISTA DE FIGURAS

Figura 1. A placa Arduino Uno. Fonte: (ARDUINO TUMBR, 2014).....	26
Figura 2. Ambiente de Programação Arduino. Fonte: (ARDUINO TUMBLR, 2014). .....	28
Figura 3. Exemplo de um projeto construído no Arduino. Fonte: (ARDUINO TUMBLR, 2015).....	29
Figura 4. ProgrameFácil – Linguagem Computacional Icônica para Aprendizado de Robótica. Fonte: (MIRANDA, 2006).....	31
Figura 5. Layout do Ambiente DuinoBlocks. Fonte: (ALVES, 2013).....	32
Figura 6. LabVad 1.0. Fonte: (VICTORINO et al., 2009).....	33
Figura 7. (a) Diagrama esquemático do sistema do trilho de ar; (b) Experimento trilho de ar; (c) Controlador VB; (d) Detalhe trilho de ar com photogates e carrinho no trilho. Fonte (VICTORINO et al., 2009).....	34
Figura 8. ProtoLabVad Eletrônica. Fonte: (SAMPAIO et al., 2015).....	35
Figura 9. Classmate PC. Fonte: (S2RD2, 2015) .....	36
Figura 10: Execução de um experimento no Arduino Remote Lab. ....	41
Figura 11: Hardware do Arduino Remote Lab. ....	42
Figura 12: Transmissão de vídeo de uma das estações pertencentes ao CLEM.....	43
Figura 13: Interface do Microelectronics WebLab. ....	44
Figura 14: Aplicação para Arduino no LabVIEW. ....	46
Figura 15: Arquitetura do HExLab. Fonte: (SILVA et al, 2013).....	47
Figura 16: Interface do Codebender. Fonte: (CODEBENDER, 2015). ....	49
Figura 17: Diagrama de Caso de Uso. ....	57
Figura 18: Diagrama de Classes conceitual.....	60
Figura 19. Exemplo de plataforma de correio eletrônico. ....	67
Figura 20. Plataforma de correio eletrônico PPGI UFRJ.....	67
Figura 21. Design responsivo: compatível com dispositivos móveis. ....	68
Figura 22. Parte física do LabVad. ....	71
Figura 23. Diagrama completo do hardware do LabVad. ....	72
Figura 24. Menu Principal.....	74
Figura 25. Transmissão dos Experimentos em tempo real. ....	74
Figura 26. Painel Meus Arquivos Arduino. ....	75
Figura 27. Barra de Ferramentas. ....	75

Figura 28. Área de codificação.....	76
Figura 29. Processo de compilação. ....	77
Figura 30. Arquivo temporário, com os blocos dos códigos da multiplexação inseridos. ....	78
Figura 31. Mensagem de compilação do Servidor LabVad NCE .....	79
Figura 32. Processo de compilação exibido no Navegador do usuário.....	79
Figura 33. Configurações do usuário e LOG.....	80
Figura 34. Ajuda no LabVad.....	81
Figura 35. Atributos de aceitabilidade de sistemas (NIELSEN E MACK, 1994).....	85
Figura 36. Barra de Ferramentas, com exemplos de programas Arduino.....	90
Figura 37. Gráfico sobre a Inspeção dos avaliadores. ....	101
Figura 38. Gráfico com os erros e a severidade média da gravidade das heurísticas violadas. ....	103
Figura 39. Subconjuntos do menu Executar.....	104
Figura 40. Tutorial das principais funções da Interface .....	110
Figura 41. Tutorial dos Exemplos e Experimentos. ....	110

## LISTA DE TABELAS

---

---

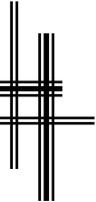
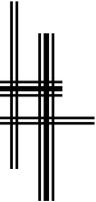


Tabela 1: Caso de Uso – Solicitar Cadastro.....	57
Tabela 2: Caso de Uso – Efetuar Login .....	58
Tabela 3: Caso de Uso – Manter Arquivos.....	58
Tabela 4: Caso de Uso – Agendar Experimentos .....	58
Tabela 5: Caso de Uso – Executar Experimentos.....	58
Tabela 6: Caso de Uso – Acessar Ajuda.....	59
Tabela 7: Caso de Uso – Manter Usuários .....	59

## LISTA DE QUADROS

---

---

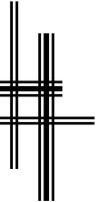


Quadro 1. Versões das placas do projeto Arduino. Fonte: (ARDUINO, 2015). .....	27
Quadro 2. Classificação dos laboratórios virtuais e remotos.....	40
Quadro 3: Ferramentas e recursos utilizados.....	69
Quadro 4: Componentes do LabVad.....	73
Quadro 5. Configurações das variáveis de ambiente do processo de compilação ...	78
Quadro 6. Escala de gravidade da avaliação heurística .....	88
Quadro 7. Inspeção dos avaliadores.....	100
Quadro 8. Análise de gravidade das heurísticas violadas.....	102

## LISTA DE ABREVIATURAS E SIGLAS

---

---



BAT	<i>Batch</i>
bps	Bits por segundo
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CLEM	Cloud E-learning for Mechatronics
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CSS	<i>Cascading Style Sheets</i>
EAD	Educação a Distância
GPL	<i>General Public License</i>
GTE	Guia de Tecnologias Educacionais
HOU	Hellenic Open University
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IHC	Interação Humano-Computador
IoT	<i>Internet of Things</i>
INEP	Instituto Nacional de Estudos e Pesquisas
NCE	Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais
LabVad	Laboratório Virtual de Atividades Didáticas
LCD	<i>Liquid Crystal Display</i>
LDR	<i>Light Dependent Resistor</i>
LED	<i>Light Emitting Diode</i>
MEC	Ministério da Educação
MIT	<i>Massachusetts Institute of Technology</i>
OCDE	Organização para Cooperação e Desenvolvimento Econômico
OLPC	<i>One Laptop per Child</i>
PPGI	Programa de Pós-Graduação em Informática
PISA	<i>Programme for International Student Assessment</i>

PROUCA	Programa Um Computador por Aluno
RE	Robótica Educacional
REBC	Robótica Educacional de Baixo Custo
RP	Robótica Pedagógica
SEED	Secretaria de Ensino a Distância
TI	Tecnologia da Informação
TIAE	Tecnologias da Informação Aplicadas à Educação
TIC	Tecnologia da Informação e Comunicação
UCA	Um Computador por Aluno
UFRJ	Universidade Federal do Rio de Janeiro
UFRN	Universidade Federal do Rio Grande do Norte
URL	<i>Uniform Resource Locator</i>
USB	<i>Universal Serial Bus</i>
V	<i>Volt</i>
$\Omega$	<i>Ohm</i>

# SUMÁRIO

<b>1.</b>	<b>Introdução .....</b>	<b>17</b>
1.1	A Robótica Educacional.....	18
1.3	Relevância.....	20
1.4	Motivações, Objetivos e Justificativas .....	21
1.5	Organização da Dissertação .....	22
<b>2.</b>	<b>O Projeto Arduino e o GINAPE.....</b>	<b>24</b>
2.1	O Projeto Arduino .....	25
2.1.1	O hardware do Arduino.....	25
2.1.2	Ambiente de programação do Arduino .....	28
2.2	A linha de pesquisa de Robótica do GINAPE.....	30
2.2.1	O Kit RoboFácil .....	30
2.2.2	O DuinoBlocks.....	31
2.2.3	A Origem do LabVad .....	32
2.2.4	A Arquitetura Pedagógica do LabVad .....	35
2.3	O PROUCA e o estágio atual do LabVad.....	36
<b>3.</b>	<b>Revisão de Literatura .....</b>	<b>38</b>
3.1	Laboratórios Remotos na Educação .....	39
3.2	Trabalhos Relacionados.....	40
3.2.1	Arduino Remote Lab (HOU) .....	41
3.2.2	Cloud E-learning for Mechatronics: CLEM.....	42
3.2.3	iLab.....	43
3.2.4	LabVIEW .....	45
3.2.5	HEXLab.....	46
3.2.6	O Arduino Online Compiler e o CodeBender.....	48
3.3	Reflexão sobre os Trabalhos Relacionados .....	49
<b>4.</b>	<b>Requisitos do Sistema .....</b>	<b>51</b>
4.1	Levantamento de Requisitos .....	52
4.2	Principais Requisitos .....	52
4.2.1	Similaridade com um ambiente real de RE.....	53
4.2.2	Requisito de portabilidade: Ambiente Multiplataforma.....	53
4.2.3	Requisitos de eficiência e desempenho .....	54

4.2.4	Ambiente Multiusuário .....	54
4.3	Diagrama de Casos de Uso.....	56
4.4	Diagrama de Classes .....	59
<b>5.</b>	<b>Metodologia de Desenvolvimento.....</b>	<b>61</b>
5.1	Metodologia Extreme Programming (XP) .....	62
5.2	Desenvolvimento em camadas.....	64
5.3	Softwares utilizados na concepção do LabVad .....	64
5.4	API Web Storage e a Interface Responsiva .....	66
5.5	Relação de Softwares.....	68
<b>6.</b>	<b>Implementação do LabVad .....</b>	<b>70</b>
6.1	Arquitetura Física do LabVad .....	71
6.2	Experimentos.....	74
6.3	O processo de compilação .....	76
6.4	Foco no Usuário .....	80
<b>7.</b>	<b>A Avaliação do LabVad .....</b>	<b>82</b>
7.1.	Comunicabilidade e Usabilidade .....	84
7.2.	Métodos de avaliação.....	86
7.3	Avaliações de processo e produto aplicadas ao LabVad .....	89
7.4	O Processo de avaliação dos Especialistas em IHC .....	90
7.4.1	A visão dos avaliadores.....	99
7.5	Conclusão sobre os dados aferidos na avaliação IHC .....	102
7.6	Avaliação docente .....	105
7.7	Conclusões sobre a avaliação docente .....	107
7.8	Contribuição da Avaliação Docente.....	109
<b>8</b>	<b>Conclusões e Trabalhos Futuros.....</b>	<b>111</b>
8.1	Artigo Acadêmico sobre o Tema da Dissertação Submetido e Aprovado	112
8.2	Trabalhos Futuros .....	112
	<b>REFERÊNCIAS.....</b>	<b>116</b>
	<b>ANEXOS .....</b>	<b>123</b>
	Anexo 1 - Instalação do LabVad no Linux .....	124
	<b>APÊNDICES .....</b>	<b>128</b>
	Apêndice A – Tarefas de avaliação (Método de Percurso Cognitivo). .....	129



# 1. Introdução

Este capítulo estrutura-se, basilarmente, em quatro momentos. No primeiro momento, ele apresenta, de forma geral, uma visão da Robótica Educacional (seção 1.1); no segundo, expõe as questões relacionadas às barreiras na aquisição de kits educacionais de robótica nas escolas públicas (seção 1.2); no terceiro, explica a relevância do presente trabalho (seção 1.3), seguido dos seus objetivos, motivações e justificativas (seção 1.4); no quinto, finaliza com uma explanação a respeito da estrutura organizacional desta dissertação (seção 1.5).

## 1.1 A Robótica Educacional

A Robótica Educacional (RE) é ministrada normalmente em laboratórios com práticas que enlaçam raciocínio lógico, trabalho em equipe e conhecimento multidisciplinar. Desta forma, conteúdos de disciplinas como Eletrônica, Física, Informática, Matemática, Línguas, Química, Biologia, entre outros, também são empregados para o desenvolvimento dos projetos tornando a prática da robótica uma atividade única em relação à construção do conhecimento.

Alves (2013) afirma que a atividade com Robótica Educacional é dinâmica e motivadora e, nela, o esforço do educando é empregado na criação de soluções, sejam essas advindas da construção de hardware e/ou de software. Os experimentos ou exercícios visam à resolução de um problema proposto, podendo o mesmo ser do cotidiano, promovendo assim a transformação do ambiente escolar em uma oficina de inventores.

No Brasil, diferentes trabalhos são direcionados à aplicação da Robótica Educacional de Baixo Custo (REBC) em sala de aula. É possível citar, como exemplos mais marcantes desses trabalhos, os esforços empreendedores de Sampaio e Elia (2011); Alves et al. (2012); Pinto et al. (2012); Zanetti (2014). Além desses mencionados, há uma tendência promissora a partir do uso de Simuladores Virtuais (CHELLA, 2012; FERNANDES, 2013) e Laboratórios Remotos (VICTORINO et al., 2009; CRUZ et al., 2009; SOUZA et al., 2011) objetivando estudos a distância e sem altos custos iniciais na aquisição de kits robóticos.

Mesmo com a existência desses trabalhos direcionados, convém apontar, conforme Miranda et al. (2010), uma lacuna no desenvolvimento da RE. Ainda há poucos trabalhos preocupados em apresentar alternativas às barreiras iniciais da programação em Robótica Educacional.

## 1.2 Caracterização do Problema

Para Souza et al. (2014), uma das adversidades com que se deparam alunos e professores da rede pública de ensino, no uso da robótica educacional, tem sido o alto custo de aquisição de kits de hardware para o desenvolvimento das atividades

práticas nas escolas e fora delas. Nesse aspecto, nos últimos anos, o movimento do hardware livre, através de projetos como o Arduino (ARDUINO, 2005) e o Raspberry Pi (RASPBERRY PI, 2012), vem colaborando para uma maior disseminação da robótica em ambientes de ensino-aprendizagem.

Outro aspecto que também debilita a existência de experiências profícuas de RE no Brasil é o pouco conhecimento dos professores e dos estabelecimentos de ensino sobre as suas propriedades multidisciplinares (ALVES et al., 2012). O treinamento de professores pode ser viabilizado pelos Laboratórios Remotos por serem ferramentas projetadas ou adaptáveis ao ensino a distância, aliando um baixo investimento inicial e reutilização de recursos a possibilidades de multiplicação do processo de formação de outros professores de forma otimizada (SOUZA et al., 2014).

Um Kit didático físico é composto por diversos artefatos de hardware. Pode ser frustrante para o aprendiz ter que lidar e memorizar dezenas de entradas e/ou saídas analógicas, digitais, bem como sua interface com os computadores. Nos laboratórios remotos, como veremos na revisão de literatura do capítulo 3, os dispositivos eletrônicos, mecânicos e outros atuadores já estão conectados prontamente para sua utilização. Este fato pode motivar o aluno a explorar tais dispositivos sem que haja, de imediato, o conhecimento prévio de eletrônica.

Diante deste panorama, o uso de laboratórios remotos acessados via web e disponíveis para professores e alunos 24 horas, 7 dias por semana, pode ser uma alternativa didática viável (SOUZA et al., 2014). Este é exatamente o caminho abordado como objeto de estudo desta dissertação, denominada LabVad, Laboratório Virtual de Atividades Didáticas, com origem em 2008 (VICTORINO et al., 2009) quando foi então formulada sua arquitetura sustentada em dois pilares: um pedagógico e outro tecnológico. Esta dissertação aborda este segundo pilar, o desenho e a implementação do Laboratório Virtual de Atividades Didáticas com robótica, com ênfase no acesso remoto e adotando como opção de hardware a plataforma livre Arduino.

### 1.3 Relevância

O aluno incipiente em um curso técnico de informática, ou na faculdade, já detém um grau de experiência com tecnologia em seu cotidiano, seja no trabalho, seja com diversos dispositivos eletrônicos, que povoam os lares. Outro perfil detectado é dos nativos digitais, que já são usuários de celulares, tablets e laptops, desde a infância, tanto no ambiente familiar quanto na educação infantil formal. Essa proximidade dos estudantes com a tecnologia torna o ambiente escolar desmotivador para os discentes, uma vez este ainda possui as mesmas características do século passado, onde os alunos, na maior parte do tempo, ficam restritos a conteúdos teóricos, sem aplicar tais conceitos nas experimentações proporcionadas pela Robótica Educacional (ZANETTI et al., 2012).

Em relação aos alunos da nossa atualidade, a escola e o ensino de programação estão descontextualizados. Os cursos se estruturam em módulos ou períodos compartimentados, como pacotes desarticulados. Os alunos aprendem as mesmas estruturas de dados, decisão (condições lógicas) e controle (loops/laços de repetição) em introdução à programação, simbolizada por algoritmos; avançando ao próximo estágio de aprendizado em programação, simbolizada, por exemplo, pela de programação linguagem C e, no estágio avançado, onde lidamos com programações orientadas a objetos, simbolizada por linguagens como Java, C#, C++ ou Python. Atualmente, o ensino de programação pode formar um aluno detentor de um conhecimento rico em tecnologias, porém pobre em lógica e técnicas de programação.

Essa situação desencadeia um choque no momento da transição desse aluno do final do curso ao ingresso no mercado de trabalho, circunstância essa que segue o caminho inverso do pensamento de Papert (2008) cuja proposta é a de que a escola deveria preparar os jovens, de forma que os mesmos sejam inseridos na sociedade, acompanhando suas transformações tecnológicas.

A Robótica Educacional (RE), ou Robótica Pedagógica (RP), pode ser utilizada para motivar e promover um envolvimento maior do aluno no ensino de programação, devido à sua forte experimentação prática. Major et al. (2011), por

intermédio da SLR (Revisão de Literatura Sistemática), apresentaram vários dados significativos sobre RE.

Foram analisados 34 artigos completos que apresentavam relatos de experiência e análise de dados. De todos os artigos analisados, 25, ou seja, 74% mostraram que o uso da robótica educativa é uma ferramenta eficiente no ensino da programação de computadores. Os mesmo trabalhos apresentaram um crescimento significativo no uso de laboratórios virtuais e remotos. Considerando-se as conclusões desses trabalhos, é crível que o uso desses laboratórios remotos será potencialmente objeto de estudo e pesquisa no ensino de lógica e programação nos próximos anos.

#### **1.4 Motivações, Objetivos e Justificativas**

O Programa Internacional de Avaliação de Estudantes (PISA) é uma avaliação aplicada pela Organização para Cooperação e Desenvolvimento Econômico (OCDE) para aferir o nível de habilidades de estudantes de diferentes países em três áreas do conhecimento: Matemática, Leitura e Ciência. O exame ocorre a cada triênio para alunos na faixa etária dos 15 anos.

O Brasil participa do PISA desde o ano 2000 e o seu desempenho evoluiu nas últimas edições, mas os alunos brasileiros ainda ocupam as últimas posições do ranking, precisamente 58º lugar, entre 65 países (OCDE, 2013).

A cada ano existe maior ênfase em uma área; neste ano, 2015, será a área de ciências sobre três dimensões: identificação de questões científicas; explicação científica de fenômenos e utilização de evidências científicas (INEP, 2013).

O ensino de robótica não está restrito à programação de computadores. A Robótica Educativa articula também as disciplinas de Eletrônica, Mecânica e Ciências, ou seja, o universo que será avaliado no PISA.

Zanetti et al. (2012) afirmam que alunos do Ensino Médio passam maior parte do seu tempo estudando conteúdos de Matemática, Física, Mecânica, Termodinâmica, Eletricidade e outros, sem aplicar os conceitos teóricos de tais disciplinas no mundo real das experimentações.

A julgar pelo quantitativo de tecnologias livres disponíveis atualmente, tanto na área de software quanto na de hardware, pode ser viável investigar se os laboratórios remotos, de baixo custo, devem ser usados também em outros contextos pedagógicos, como no ensino de Ciências, bem como os conteúdos acima citados.

## **1.5 Organização da Dissertação**

Esta dissertação está organizada em sete capítulos, além desta introdução. No segundo capítulo, explica-se o projeto Arduino e se faz um breve histórico da linha de pesquisa em Robótica do GINAPE. O terceiro capítulo trata da revisão da literatura, no que diz respeito ao uso de laboratórios remotos no ensino de Robótica e Ciências.

O quarto, o quinto e o sexto capítulos abordam respectivamente o levantamento de requisitos até a implementação da plataforma WEB do LabVad.

O sétimo capítulo apresenta resultados sobre métodos de avaliações de processo e produto, ou seja, a avaliação final da plataforma WEB do LabVad, baseada na sua Interação humano-computador. No último capítulo, conclui-se o trabalho, tecendo as considerações finais e propondo trabalhos futuros. Em seguida expõem-se a bibliografia, os apêndices e os anexos.





## **2.O Projeto Arduino e o GINAPE**

O alicerce deste capítulo é uma explanação sobre o Projeto Arduino, uma abordagem da sua influência na linha de pesquisa de Robótica do GINAPE (Grupo de Informática Aplicada à Educação) da UFRJ e algumas pesquisas que influenciaram na concepção do LabVad. Ele está estruturado em três seções. A seção 2.1 trata da apresentação do projeto Arduino como uma tecnologia de hardware livre e de baixo custo para a implementação de trabalhos em Robótica Educacional; a seção 2.2 destaca alguns trabalhos da linha de pesquisa de Robótica do GINAPE e a seção 2.3 trata do Programa “Um Computador por Aluno”.

## 2.1 O Projeto Arduino

Criado em 2005 na Itália, o projeto Arduino é uma plataforma de prototipagem eletrônica de baixo custo e código aberto destinado a educadores, artistas, designers, *hobbyistas* e outros interessados em criar objetos ou ambientes interativos (ARDUINO, 2015).

Segundo McRoberts (2011), a Plataforma Arduino é estruturada por dois componentes: o hardware ou placa Arduino, baseado em microcontrolador programável e o Ambiente Integrado de Desenvolvimento (IDE) utilizado para construção dos programas que irão controlar os protótipos desenvolvidos.

### 2.1.1 O hardware do Arduino

O hardware Arduino (Figura 1) é uma placa eletrônica microcontrolada e desenvolvida sobre uma biblioteca escrita em C. O microcontrolador<sup>1</sup> da família ATMEL AVR (2013) presente na placa pode ser programado com a linguagem de programação Arduino, baseada na linguagem Wiring (2013) e com o ambiente de desenvolvimento Arduino, baseado no ambiente Processing (2013).

---

<sup>1</sup> Microcontrolador é um circuito integrado que contém todas as funções básicas de um computador.

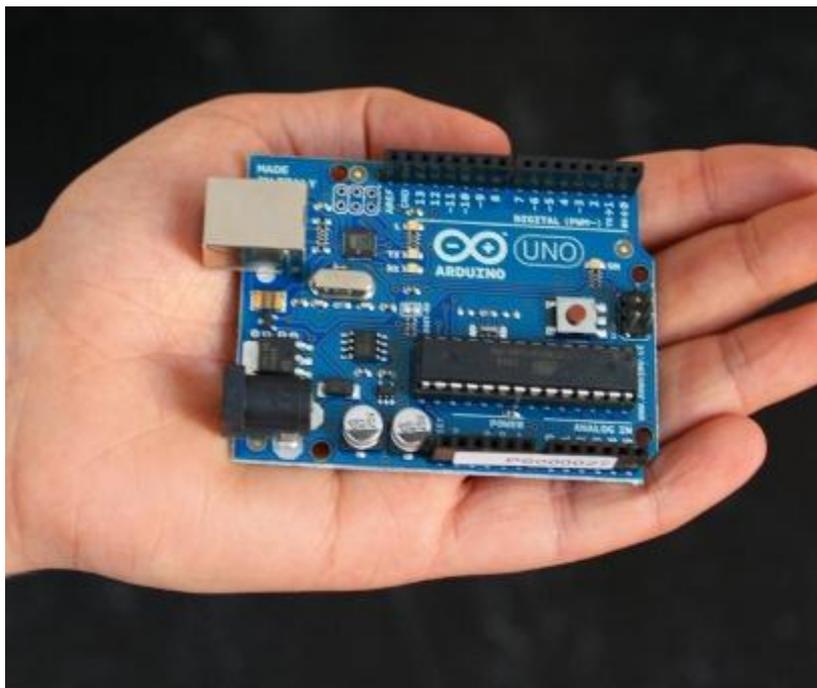


Figura 1. A placa Arduino Uno. Fonte: (ARDUINO TUMBR, 2014).

Com dez anos de existência, o projeto Arduino cresceu e se expandiu pelo mundo. Veja abaixo o Quadro 1 contendo as especificações das 20 placas que atualmente são homologadas pelo projeto Arduino (ARDUINO, 2015).

Nome da placa	Processor	Tensão/Entrada de Tensão	Velocidade da CPU	Entrada/Saída Analógica	Digital IO/PWM
ArduinoBT	ATmega328P	5 V / 2.5-12 V	16 MHz	6/0	14/6
Due	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12
Esplora	ATmega32U4	5 V / 7-12 V	16 MHz	-	-
Ethernet	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/4
Fio	ATmega328P	3.3 V / 3.7-7 V	8 MHz	8/0	14/6
Gemma	ATtiny85	3.3 V / 4-16 V	8 MHz	1/0	3/2
Leonardo	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7

LilyPad	ATmega168V ATmega328P	2.7-5.5 V / 2.7-5.5 V	8MHz	6/0	14/6
LilyPad SimpleSnap	ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	4/0	9/4
LilyPad USB	ATmega32U4	3.3 V / 3.8-5 V	8 MHz	4/0	9/4
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6
Pro	ATmega168 ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6
Pro Mini	ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6
Yùn	ATmega32U4 AR9331 Linux	5 V	16 MHz 400MHz	12/0	20/7
Zero	ATSAMD21G18	3.3 V / 7-12 V	48 MHz	6/1	14/10

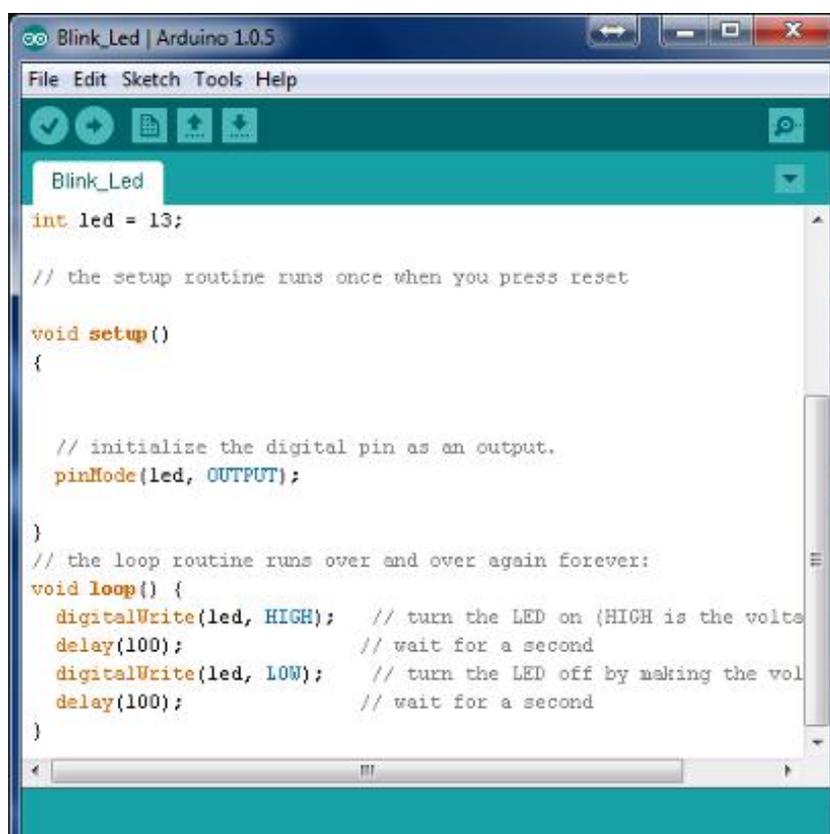
**Quadro 1. Versões das placas do projeto Arduino. Fonte: (ARDUINO, 2015).**

Vale ressaltar que, além das placas homologadas supramencionadas, existem outros projetos baseados na filosofia do hardware livre do Arduino, como o NetDuino. Walker (2012) explica que a placa NetDuino foi criada para o

desenvolvimento de projetos em eletrônica, seguindo a tendência do movimento IoT (Internet das Coisas) e utilizando, em seu desenvolvimento, a linguagem C# do framework<sup>2</sup> “.NET” da Microsoft.

### 2.1.2 Ambiente de programação do Arduino

O ambiente de programação Arduino Wiring (Figura 2) é uma aplicação multiplataforma desenvolvida em linguagem Java e liberada sob a licença GPL (GENERAL PUBLIC LICENSE, 2013), ele inclui um conjunto de opções que auxiliam na depuração e comunicação (upload) com a placa de hardware Arduino.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink\_Led | Arduino 1.0.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for file operations and execution. The main text area contains the following C++ code:

```
int led = 13;

// the setup routine runs once when you press reset

void setup()
{

  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);

}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(100);              // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(100);              // wait for a second
}
```

Figura 2. Ambiente de Programação Arduino. Fonte: (ARDUINO TUMBLR, 2014).

Para conexão do hardware Arduino com o computador, que possui o ambiente de desenvolvimento, é utilizada uma interface serial com conexão USB<sup>3</sup>.

---

<sup>2</sup> Um framework (ou arcabouço), em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.

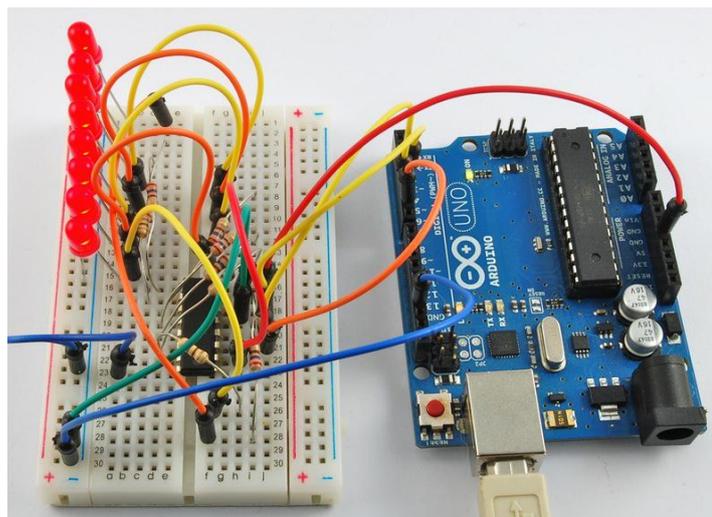
<sup>3</sup> Universal Serial Bus (USB) é um tipo de conexão "ligar e usar" que permite a fácil conexão de periféricos sem a necessidade de desligar o computador.

Na plataforma Windows, a conexão do Arduino é identificada como uma porta serial tipo COM (p. ex. COM3) e na plataforma Linux, como um dispositivo serial ttyUSB (p. ex. ttyUSB1). O cabo utilizado para interconexão é do tipo USB, o mesmo utilizado em impressoras e scanners (McRoberts, 2011).

A alimentação elétrica padrão da placa Arduino é 5 volts (necessária ao funcionamento do microcontrolador) e pode ser obtida de duas formas: um cabo USB ligado a um computador ou através de uma fonte externa, tais como baterias ligadas diretamente ao conector de energia externa da placa.

Nos casos de alimentação elétrica com fonte externa, podem-se aplicar tensões entre 7 e 20 volts, pois a placa possui um dispositivo regulador de voltagem que executa a adequação necessária para entregar a tensão de 5 volts ao microcontrolador.

A prototipagem de um projeto ou um projeto básico de robótica de hardware livre consiste na conexão da placa Arduino a uma protoboard<sup>4</sup>. Nesta é possível conectar e desconectar facilmente LED (Diodo Emissor de Luz), display (mostradores), botões, interruptores, motores, sensores de temperatura, sensores de distância, receptores de sistemas de posicionamento global (GPS) entre outros, como na Figura 3.



**Figura 3. Exemplo de um projeto construído no Arduino. Fonte: (ARDUINO TUMBLR, 2015).**

---

<sup>4</sup> Protoboard é uma placa com orifícios e conexões condutoras utilizadas no processo de montagem de circuitos eletrônicos experimentais.

Após o desenvolvimento e compilação do seu programa, o Arduino passa a controlar os dispositivos conectados à protoboard. O usuário carrega estes programas na placa Arduino, via porta USB, que deve estar conectada a um computador.

A beleza deste sistema é que você pode criar alguma coisa com um Arduino e depois transformá-lo em algo permanente, dessa forma, você terá um dispositivo embarcado personalizado. O Arduino é um dispositivo incrível e possibilitará que você crie de tudo, desde obras de arte interativas até robôs. Com um pouco de entusiasmo para aprender como programar um Arduino fazendo-o interagir com outros componentes, tendo um pouco de imaginação, você poderá construir tudo o que quiser. (MCROBERTS, p. 25, 2011)

## **2.2 A linha de pesquisa de Robótica do GINAPE**

Iremos citar alguns trabalhos da linha de pesquisa de Robótica do GINAPE que são destaques ou objetos de estudos influenciadores na concepção do LabVad. Estes trabalhos, mesmo apresentando temas e objetivos diferentes, possuem em comum a disseminação da Robótica Educativa de Baixo Custo (REBC).

### **2.2.1 O Kit RoboFácil**

De acordo com Miranda et al. (2010) as propostas de artefatos computacionais didáticos exploram, em sua maioria apenas software. Portanto, o RoboFácil foi uma proposta de pesquisa que objetivava a criação de um kit de robótica, com fins pedagógicos, composto tanto por hardware quanto por software.

Uma das principais motivações para se construir o Kit RoboFácil foi a escassez de Kits didáticos robóticos que apresentassem uma boa relação de custo benefício no Brasil e principalmente pelo fato de o público alvo serem os estudantes do Ensino Médio.

A implementação do RoboFácil sempre priorizou a criação de um ambiente intuitivo, agradável de usar e eficiente para manipular o hardware deste Kit. O ProgrameFácil é uma linguagem de programação visual (VPL) que possibilita controlar os dispositivos eletrônicos e eletromecânicos da parte física do RoboFácil.

A parte física do RoboFácil provê a utilização de circuitos externos, através de um plug-in acoplado ao hardware que deve controlar, por exemplo, motores de passo, sensores de luminosidade e temperatura, entre outros elementos eletrônicos. A fim de facilitar a manipulação dos objetos, os mesmos foram divididos em três categorias para melhor identificação das suas funcionalidades: a) objetos de hardware; b) objetos de programação e c) objetos de apoio. Na linguagem ProgrameFácil, os objetos são representações icônicas, que podem ser manipulados tanto no “Meu Mundo” como no “Meu Programa”, como está ilustrado na Figura 4 (MIRANDA, 2006).

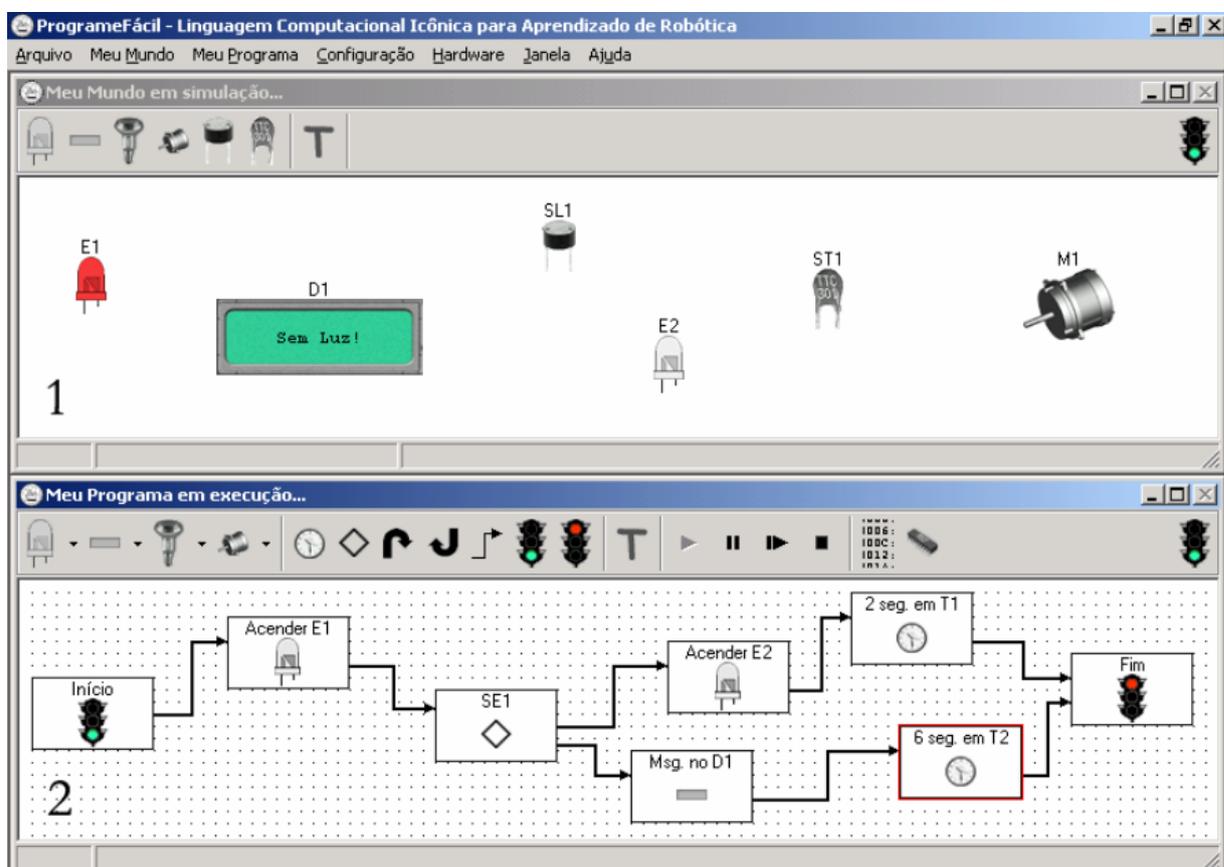


Figura 4. ProgrameFácil – Linguagem Computacional Icônica para Aprendizado de Robótica.

Fonte: (MIRANDA, 2006).

### 2.2.2 O DuinoBlocks

O DuinoBlocks é um ambiente de desenvolvimento de programas (IDE), semelhante ao Scratch, que incorpora uma VPL (Visual Programming Language) interativa e amigável utilizando blocos de códigos na construção de programas

(SCRATCH BRASIL, 2015). O diferencial é que o DuinoBlocks é um ambiente direcionado para a programação em Robótica para placas Arduino. (ALVES et al., 2013).

O principal objetivo dos elementos de programação utilizados no ambiente DuinoBlocks (Figura 5) é a redução do esforço cognitivo dos estudantes novatos em programação (PASTERNAK, 2009).

Dargains (2015) vem explorando o potencial didático desta ferramenta educacional através de metodologias de ensino alicerçadas nas Taxonomias de Bloom (JESUS e BRITO, 2010).

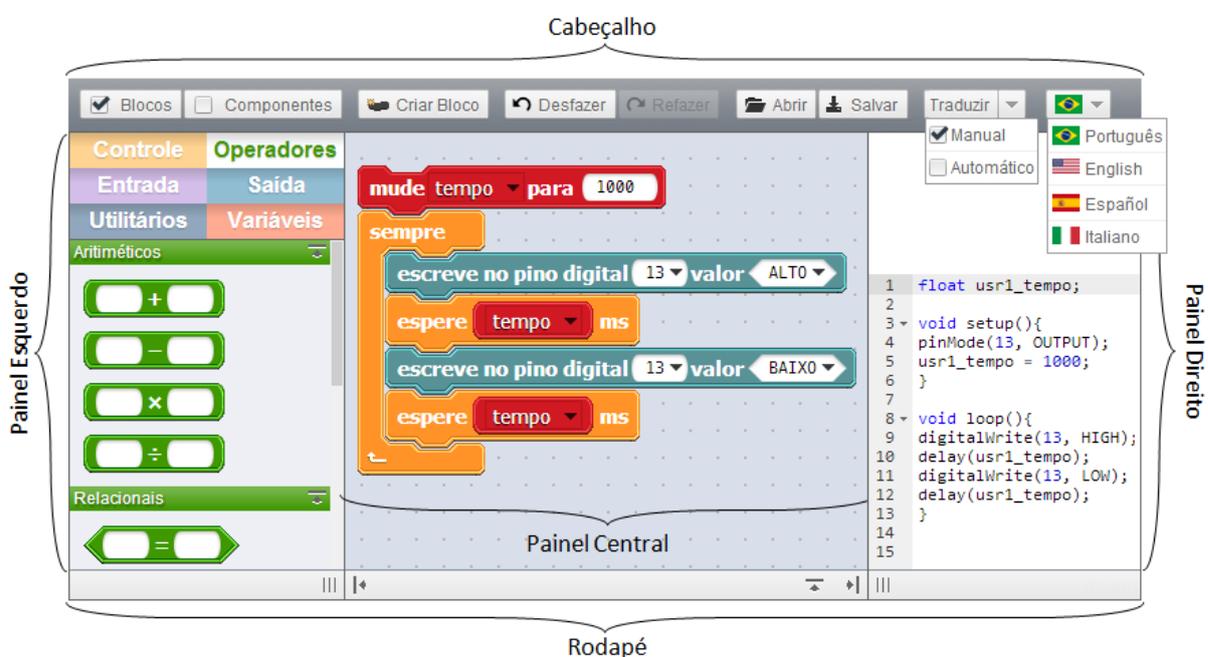
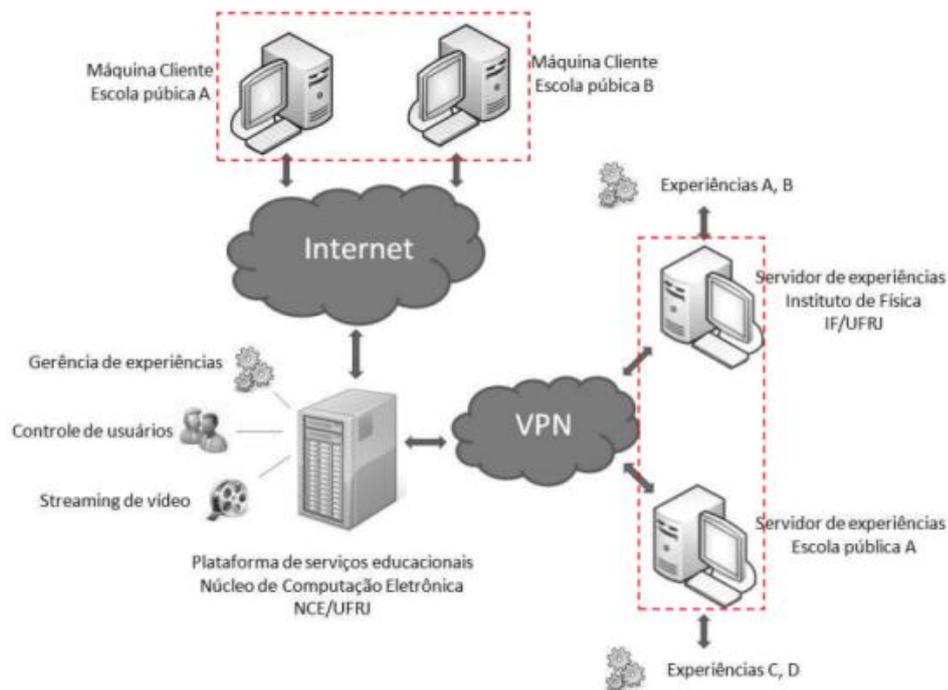


Figura 5. Layout do Ambiente DuinoBlocks. Fonte: (ALVES, 2013).

### 2.2.3 A Origem do LabVad

Victorino et. al (2009) relataram que os levantamentos de requisitos, sobre o ambiente da plataforma WEB de serviços educacionais do LabVad, estavam em construção desde 2008. O ponto de partida é a arquitetura tecnológica ilustrada na Figura 6.



**Figura 6. LabVad 1.0. Fonte: (VICTORINO et al., 2009).**

Baseado nas Tecnologias da Informação e Comunicação (TIC) oferecidas pela WEB 2.0<sup>5</sup> foi proposto o desenvolvimento de plataforma WEB em consonância com a modelo de desenvolvimento WEB em três camadas MVC (Modelo, Visualização e Controle).

Esta plataforma WEB seria responsável por vários serviços destacando a essência do projeto: a visualização de imagens dos experimentos de robóticos executados pelos alunos e obtidas pelas câmeras instaladas sobre uma placa de robótica conectada a um servidor localizado no NCE.

Portanto, o primeiro passo foi uma revisão de literatura sobre as interfaces que controlavam remotamente experimentos como, por exemplo, a interface do LabVIEW da National Instruments (VICTORINO et al., 2009).

---

<sup>5</sup> Web 2.0 é um termo usado para designar uma segunda geração de comunidades e serviços oferecidos na Internet, tendo como conceito a Web e através de aplicativos baseados em redes sociais e tecnologia da informação. A Web 2.0 foi criada em 2004 pela empresa americana O'Reilly Media.

Sampaio et al. (2015) descrevem que após esta investigação sobre acesso remoto, utilizou-se uma solução trivial através do protocolo RDP<sup>6</sup> numa Rede Particular Virtual (VPN). A partir dessa infraestrutura, foi desenvolvida uma aplicação desktop na linguagem de programação Visual Basic 6.0 e outra aplicação em Active Server Page<sup>7</sup> (ASP) que, em conjunto, viabilizam a manipulação, via internet, de alguns dispositivos eletrônicos montados em um trilho de ar. Este trilho foi projetado para o desenvolvimento de atividades didáticas, envolvendo conceitos básicos de física.

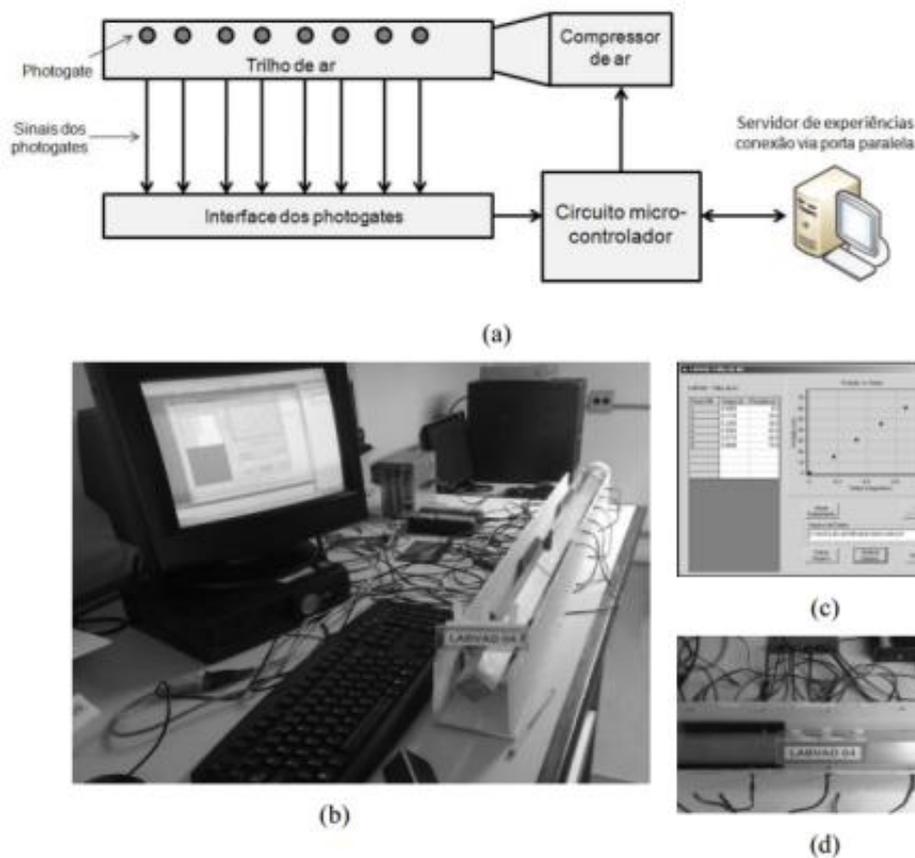


Figura 7. (a) Diagrama esquemático do sistema do trilho de ar; (b) Experimento trilho de ar; (c) Controlador VB; (d) Detalhe trilho de ar com photogates e carrinho no trilho. Fonte (VICTORINO et al., 2009).

A partir de um estudo piloto direcionado a docentes do ensino técnico em eletrônica, pode-se dizer que uma nova fase do LabVad iniciou-se em 2010. De acordo com Sampaio et al. (2015) essa pesquisa objetivava que alunos e

<sup>6</sup> Remote Desktop Protocol (ou somente RDP) é um protocolo multicanal que permite que um usuário se conecte a um computador rodando utilizando tecnologias de acesso remoto da Microsoft.

<sup>7</sup> Active Server Page, da Microsoft, é uma tecnologia de codificação de scripts de servidor que pode ser utilizada para criar aplicativos de web dinâmicos e interativos.

professores realizassem, remotamente, a programação de um micro controlador dotado dos seguintes periféricos: display de 7 segmentos, LCD e uma sequência de LEDs. Neste momento nasce o ProtoLabVad Eletrônica.

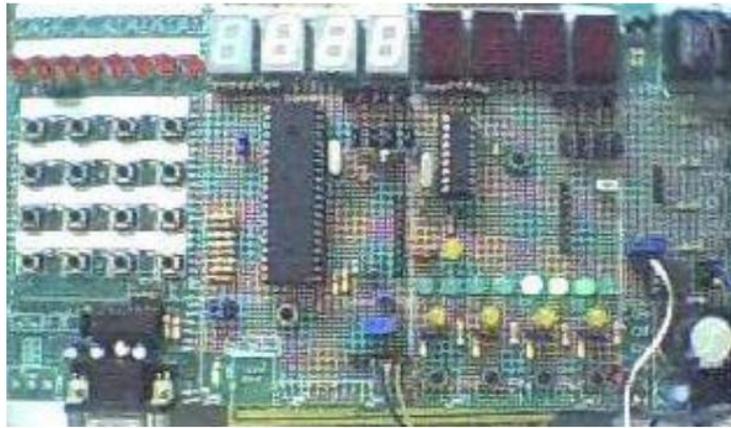


Figura 8. ProtoLabVad Eletrônica. Fonte: (SAMPAIO et al., 2015).

#### 2.2.4 A Arquitetura Pedagógica do LabVad

Mediante a arquitetura do LabVad, preconizada por Victorino et al. (2009), Pinto (2011) desenvolveu a aplicação de uma arquitetura pedagógica com a proposta do curso de Robótica Educacional usando hardware livre.

Pinto (2011) defende que no ensino remoto de RE, onde se usam ambientes virtuais de aprendizagem, deve-se favorecer a produção coletiva de atividades didáticas. Assim foi construída uma arquitetura pedagógica que obedece a três fases ou camadas sequenciais, denominada Modelo Hierárquico de Interatividade em três camadas (MHI-3C).

Os atores envolvidos neste processo interativo são pesquisadores, professores e alunos. Na primeira camada, pesquisadores e professores orientam propostas de projetos através de fóruns virtuais. Na segunda camada, os docentes discutem entre si as propostas de atividades didáticas seguindo as diretrizes das propostas dos projetos, e na terceira camada, os docentes aplicam as atividades didáticas junto aos seus alunos, ou seja, discutem com os discentes o que construir dentro de um projeto pedagógico previamente desenhado pelos professores.

Sampaio e Elia (2012) ressaltam que nas camadas 2 e 3 permanece a participação da equipe de pesquisadores do projeto, porém gradativamente menos incisiva.

Segundo Pinto (2011) a arquitetura MHI-3C deverá contemplar:

- A contextualização sociocognitiva;
- A produção colaborativa dos professores com a equipe do projeto;
- Por fim, o efeito multiplicador, onde material didático construído de forma colaborativa.

### 2.3 O PROUCA e o estágio atual do LabVad

O Programa “Um Computador por Aluno”, PROUCA (SAMPAIO e ELIA, 2012) é um projeto do Governo Federal com o propósito de promover a inclusão digital das crianças brasileiras da rede pública de ensino mediante a aquisição de computadores portáteis, novos, de baixo custo (Figura 9) e com conteúdos pedagógicos.



Figura 9. Classmate PC. Fonte: (S2RD2, 2015)

No final de 2011, o Ministério da Educação realizou uma chamada via edital (SEED, 2010) para que grupos de pesquisa no Brasil apresentassem propostas que contemplassem o uso dos laptops adquiridos pelas escolas parceiras do PROUCA. O projeto Uca na Cuca (SAMPAIO e ELIA, 2012), de pesquisa científica, tecnológica

e inovação pedagógica na área de RE, foi um dos projetos selecionados pelo referido Edital.

O projeto prevê cinco metas e o presente trabalho está inserido na meta quatro do projeto supracitado. A conclusão desta meta do edital, bem como a popularização das placas de baixo custo do projeto Arduino, como foi explicado na seção 2.1 deste capítulo, deram origem do desenho e implementação do LabVad, ou seja, a proposta da construção de uma plataforma online para viabilizar o ensino remoto de RE.

Esta plataforma tem como objetivo controlar um hardware personalizado, criado pela equipe do GINAPE e localizado no Servidor NCE LabVad. A plataforma em questão é denominada LabVad e seus principais requisitos e características de implementação serão descritos nos capítulos 4, 5 e 6.



## **3.Revisão de Literatura**

Este terceiro capítulo organiza-se em três segmentos, todos eles contemplam diferentes aspectos no trabalho com os laboratórios abordados nesta dissertação. No primeiro segmento serão apresentados modos de utilização de laboratórios remotos no campo educacional (seção 3.1); no segundo, serão descritos trabalhos relacionados ao LabVad (seção 3.2) e por último (seção 3.3), será feita uma reflexão e conclusão sobre os trabalhos relacionados.

### 3.1 Laboratórios Remotos na Educação

O ensino de ciências e Robótica pode se apoiar em laboratórios virtuais ou remotos. Chao et al. (2014) explicam que, nos laboratórios virtuais, os alunos utilizam modelos baseados em dados matemáticos e estatísticos. Esses modelos aceitam a entrada e o ajuste de variáveis, sem o contato ou manipulação de dispositivos reais. Deste modo, os alunos visualizam e manipulam gráficos simulando diversas situações com o propósito de imitar um processo do mundo real.

Nos laboratórios remotos, os alunos controlam dispositivos reais operados por instruções fornecidas por software. Além disso, câmeras são utilizadas para monitorar o funcionamento de diversos dispositivos em tempo real. Portanto, em função da gama de recursos oferecidos pelos laboratórios remotos, podemos afirmar que estes estão mais próximos dos laboratórios físicos do que os laboratórios virtuais.

Silva et al. (2013) apontam que a experimentação remota, mesmo a distancia, permite ao aluno interação prática com os mecanismos físicos. Os laboratórios remotos permitem uma imersão real, o que os diferenciam qualitativamente de simples simuladores ou laboratórios virtuais, que disponibilizam apenas experiências gravadas e resultados simulados.

Abaixo, temos o Quadro 2 contendo a relação e classificação dos projetos baseados em laboratórios virtuais e remotos, estes últimos citados na revisão de literatura.

Categoria	Projeto
Laboratórios virtuais	NetLogo <sup>8</sup> , MatLab <sup>9</sup> , Gizmos <sup>10</sup> e Easy Java Simulations <sup>11</sup> .
Laboratórios com acesso remoto	iLab, CLEM, LabView, Arduino Remote Lab (HOU), HexLab e LabVad.

**Quadro 2. Classificação dos laboratórios virtuais e remotos.**

De acordo com Ertugrul (2000), a principal vantagem no uso de laboratórios remotos na educação é o fato de que, com esses laboratórios, os discentes passam a ter acesso a equipamentos que não teriam devido ao custo, tempo, distância e escassez.

Nedic et al. (2003) apontam que os laboratórios remotos representam a melhor alternativa para trabalhar com alunos, caso não haja disponibilidade de uso ou a existência de laboratórios físicos no ambiente escolar. Nadic et al. (2003) explicam que além da experimentação remota, estes laboratórios são projetados para oferecer criação de grupos de discussão e fóruns, através de vídeo conferência; estímulo da aprendizagem colaborativa; aprendizado por intermédio de tentativa e erro; possibilidade de o aluno manipular e analisar dados reais dos experimentos realizados e flexibilidade na escolha da hora e local para a realização de experimentos.

### 3.2 Trabalhos Relacionados

Neste segmento, será realizada uma explanação a respeito do franco desenvolvimento da utilização de laboratórios remotos na área de treinamento e

---

<sup>8</sup> É um ambiente de desenvolvimento integrado baseada em agentes.

<sup>9</sup> É um programa interativo de alto desempenho voltado para o cálculo numérico avançado.

<sup>10</sup> É um repositório de simulações interativas, com ênfase em matemática e ciências.

<sup>11</sup> É uma ferramenta utilizada na criação de simulações interativas em Java ou Java script, para fins de ensino ou aprendizagem.

educação na modalidade a distância, além de um enfoque na relevância dessa categoria de laboratório no ensino de Ciências e Robótica.

### 3.2.1 Arduino Remote Lab (HOU)

O Arduino Remote Lab é um projeto da Hellenic Open University – Grécia (ARDUINO REMOTE LAB, 2013). Os procedimentos para sua utilização se constituem de ações simples. Após se cadastrar, o usuário deve reservar um horário e depois acessar a página principal, onde quatro painéis são exibidos simultaneamente no navegador WEB. O primeiro painel é o editor de código; nele, o usuário pode escrever o seu programa Arduino com destaque nas palavras reservadas da linguagem Wiring (VASSILIS, 2013). Há, também, três botões de comando e um botão para compilar o código. O resultado é mostrado na janela de saída de compilação.

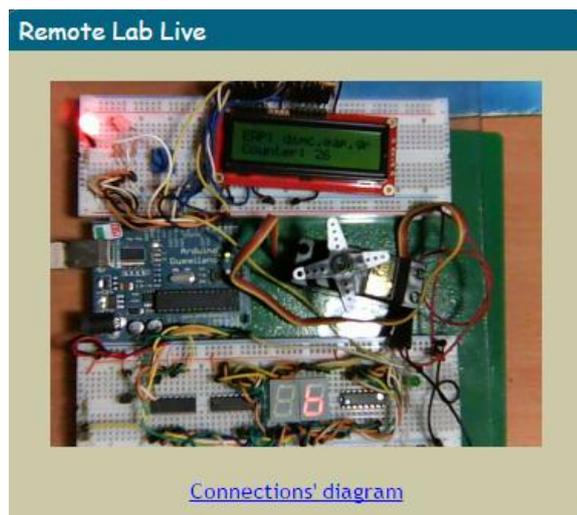


Figura 10: Execução de um experimento no Arduino Remote Lab.

A visualização dos experimentos exige a instalação do Adobe Flash Player (um reproduzidor multimídia para navegadores WEB).



Figura 11: Hardware do Arduino Remote Lab.

### 3.2.2 Cloud E-learning for Mechatronics: CLEM

O CLEM é um consórcio formado por algumas universidades europeias. Graças a essa relação de parceria, o aluno tem acesso a diversos laboratórios remotos, para executar seus experimentos, e a repositórios didáticos, para estudar Mecatrônica (CLEM, 2012).

As universidades que compõem a tessitura deste consórcio são: a Universidade de Cluj-Napoca (Romênia); a Universidade de Cassino e a Southern Lazio (Itália); a Universidade de Ruse (Bulgária), o Institut Français de Mécanique Avancée (França), a Hildebrand da Inglaterra e a ESTA Bildungswerk (Alemanha).

O projeto emerge de estudos reveladores da iminente e gigantesca carência de mão de obra qualificada nas áreas de Eletrônica, Mecânica e Ciência da Computação (CHAO et al., 2014). O escopo principal do projeto CLEM, calcado nas previsões desses estudos, é promover o estudo da Mecatrônica, em toda a Europa, através de vários laboratórios remotos baseados na tecnologia da Computação em Nuvem.

O CLEM ainda conta com uma plataforma com códigos de exercício e materiais didáticos armazenados em um banco de dados centralizado. A infraestrutura em nuvem permite que vários laboratórios possam se unir virtualmente para criar um ecossistema para educadores e alunos. A plataforma pode suportar múltiplos programas, cursos e exercícios inclusos num sistema centralizado (JAMES et al., 2014).



Figura 12: Transmissão de vídeo de uma das estações pertencentes ao CLEM.

### 3.2.3 iLab

Em nossa revisão de literatura, o projeto mais citado, entre os laboratórios remotos com aplicações educacionais é o iLab, pertencente ao MIT (Instituto de Tecnologia de Massachusetts).

O projeto foi iniciado em 1998 por Jesus del Alamo (HARWARD et al., 2008). O incipiente motivo para criação desse projeto foi a ausência de laboratórios para experimentação prática dos cursos em dispositivos semicondutores do MIT. Uma pequena bolsa de estudos, financiada pela Microsoft, possibilitou a contratação de um estudante que produziu um Applet Java<sup>12</sup>. Este Applet Java permitiu que os alunos do curso de Engenharia Elétrica utilizassem navegadores web para manipular semicondutores a partir de um servidor. O laboratório ficou conhecido

---

<sup>12</sup> Applets em Java são pequenos programas que podem ser executados em um navegador WEB, usando uma Java Virtual Machine (JVM).

como Microelectronics WebLab e, no ano seguinte, já atendia a quase 100 alunos do curso de engenharia elétrica (HARDISON et al., 2005).

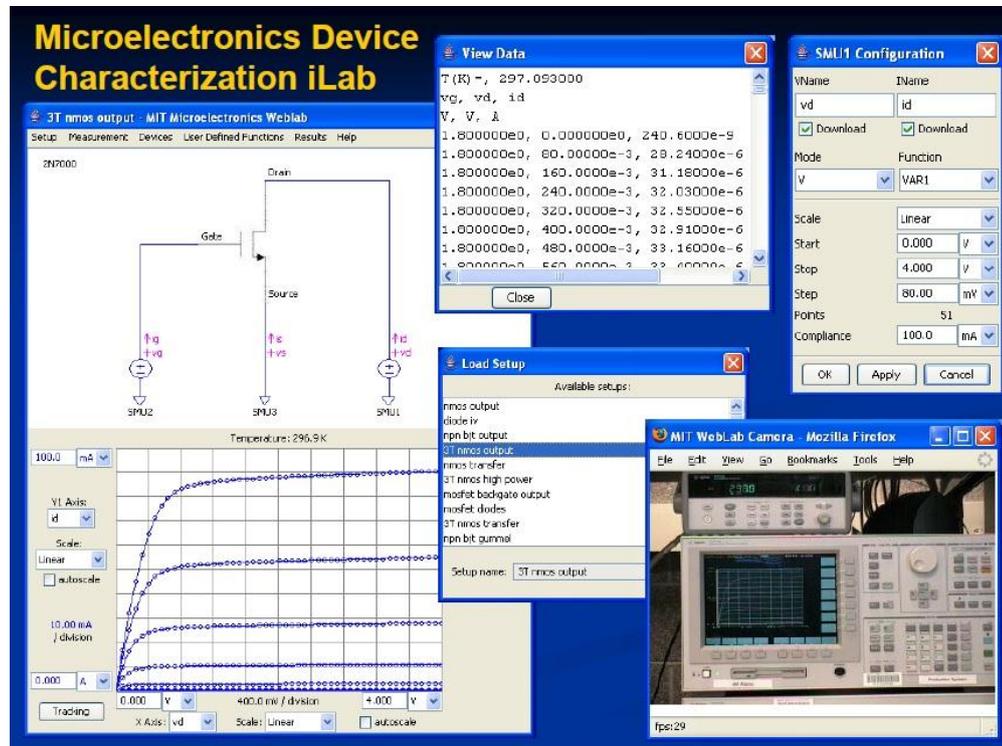


Figura 13: Interface do Microelectronics WebLab.

De 1999 a 2006, através o Projeto iCampus, a Microsoft financiou várias pesquisas baseadas no iLab (HARWARD et al., 2008). Estima-se que foram desenvolvidos mais de 50 projetos focados em três áreas distintas: Ciências, Matemática e Engenharia. Esses trabalhos foram conduzidos pelos principais pesquisadores do corpo docente do MIT (iCAMPUS, 2006) e, devido a isso, as tecnologias da Microsoft sempre foram utilizadas na arquitetura de todas as versões do iLab, os WebLabs.

Segundo Hardson et al. (2005), o WebLab 5.0 possibilitou o apoio a cursos com até 350 alunos e já foi utilizado por cerca de 700 alunos em diferentes disciplinas de instituições acadêmicas em três continentes.

O escopo do Projeto iLabs é criar um conjunto de recursos experimentais para facilitar a utilização destes laboratórios pelos docentes no currículo de seus cursos, e que os desenvolvedores construam seu próprio iLab, através da arquitetura compartilhada e colaborativa do projeto. (HARDISON et al., 2005).

Harward et al. (2008) afirmam que, nos últimos anos, a arquitetura do iLab foi adotada por um número considerável de universidades nos continentes europeu, africano e asiático e em países como Austrália e Estados Unidos. Nesses lugares o projeto iLab ratificou que o uso de laboratório on-line pode atingir milhares de alunos em diversos continentes.

### 3.2.4 LabVIEW

Uma das plataformas pioneiras em relação a laboratórios remotos é o LabVIEW – Laboratory Virtual Instrument Engineering Workbench (LABVIEW, 2015), que desde 1986 oferece um ambiente de desenvolvimento de sistemas para medições, controle de dispositivos eletrônicos e até criação de protótipos eletrônicos avançados (ERTUGRUL, 2000).

O LabVIEW foi utilizado em diversas aplicações tanto na indústria quanto na educação, inclusive no MIT (Instituto de Tecnologia de Massachusetts) em uma das versões mais atuais e robustas do iLab (HARWARD et al., 2008).

Ertugrul (2000) defende que o LabVIEW é o melhor laboratório de acesso remoto para ser utilizado nos cursos superiores nas diversas áreas da engenharia, por causa das seguintes características: custo benefício em relação a outros softwares de engenharia; desenvolvimento modular; portabilidade (ambiente multiplataforma); compatibilidade do código com as versões anteriores do software; compatibilidade com diversos tipos de hardware; posse de biblioteca de códigos extensível; depuração do código com técnicas avançadas; encapsulamento dos códigos executáveis; desempenho e interface gráfica intuitiva.

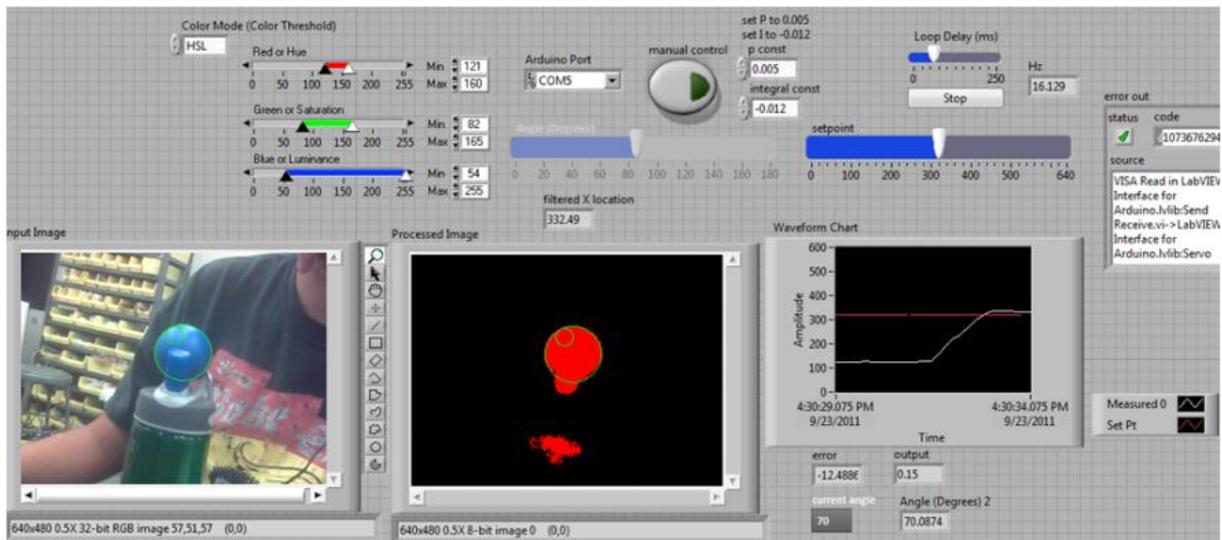


Figura 14: Aplicação para Arduino no LabVIEW.

Ertugrul (2000), ainda, acredita que os instrumentos virtuais podem fornecer uma interface com o usuário altamente interativa e que os laboratórios remotos, como o LabVIEW, podem prover análises avançadas que não são possíveis nos métodos convencionais no ensino de Engenharia.

As redes locais das universidades também podem ser usadas para trazer, em tempo real, dados dos laboratórios remotos. O possível corolário disso é que, no futuro, haverá maior ênfase em acessar componentes instrucionais de laboratório virtuais através de Intranets (ERTUGRUL, 2000).

### 3.2.5 HExLab

Assim com o LabVad, o HexLab é um projeto também em desenvolvimento no Brasil. O HExLab possibilita a integração entre os dispositivos móveis, ambiente virtual de aprendizagem (AVA) e experimentos direcionados à disciplina de Física no Ensino Médio.

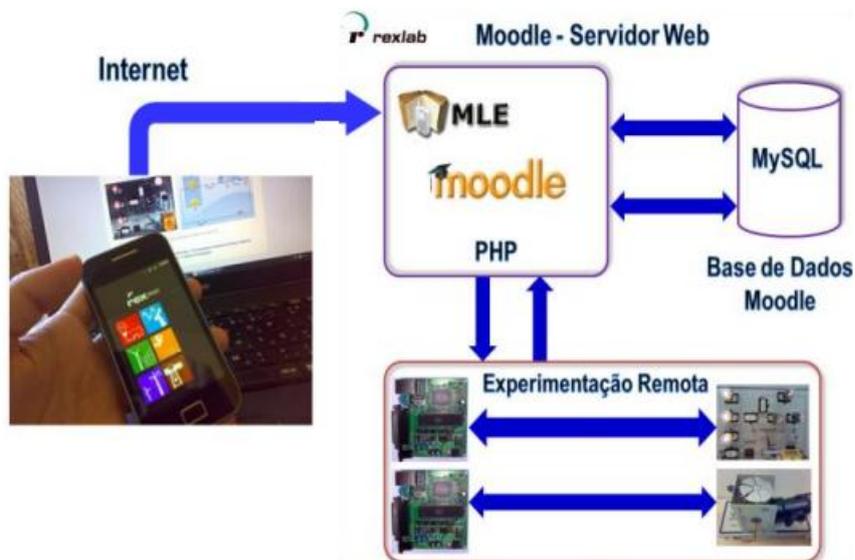


Figura 15: Arquitetura do HExLab. Fonte: (SILVA et al, 2013).

De acordo com Silva et al. (2013), a motivação para elaborar o projeto é a carência de profissionais de engenharia, bem como o desinteresse de jovens pelas áreas de ciências exatas. A justificativa apontada foi que o ensino em laboratórios remotos, intermediado pela Mobile Learning<sup>13</sup>, pode aproximar os jovens da escola porque smartphones, tablets e laptops são dispositivos que possuem diversos recursos multimídia, conectividade e mobilidade, além de fazerem parte do cotidiano desses jovens.

Silva et al.(2013) relataram que a aplicação deste projeto recebeu apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Cerca de 150 alunos do 2º ano do Ensino Médio de uma escola básica da região sul de Santa Catarina têm utilizado os recursos disponibilizados pelo RExLab. Os alunos têm à disposição o material das aulas e realizam suas atividades através do Ambiente Virtual de Aprendizagem. Neste espaço foram ainda disponibilizadas tarefas e questionários sobre os conteúdos abordados nos experimentos.

<sup>13</sup> Mobile learning, ou aprendizagem móvel é uma das modalidades da educação a distância, ou e-learning. Acontece quando a interação se dá através de dispositivos móveis, tais como smartphones, tablets, laptops, entre outros.

### 3.2.6 O Arduino Online Compiler e o CodeBender

O Arduino Online Compiler é um dos seis projetos relacionados à plataforma Arduino em desenvolvimento pela empresa ArduinoDev (ARDUINODEV, 2015). É um ambiente que pode ser acessado via web, sem a necessidade de instalação de extensões ou drivers. Neste ambiente, o usuário tem a possibilidade de compilar seu código Wiring num espaço da web e, posteriormente, carregar o código numa placa Arduino conectada ao seu computador.

Para utilizar o Arduino Online Compiler, o usuário deve copiar o código do seu computador e colar diretamente no ambiente. Uma vez compilado com sucesso, o ambiente permite o download do código com extensão hex<sup>14</sup>, o qual pode ser utilizado no simulador Protheus (WANG et al., 2012). Todavia, na situação de ocorrência de erros, as informações prestadas pelo ambiente são pouco explicativas para os usuários novatos.

Além dessa última característica negativa, é pertinente salientar outra característica: no referido ambiente, a inexistência de ferramentas de edição / alteração / armazenamento do código dificultam ainda mais o trabalho do usuário na criação de programas. Para tal, o usuário terá que utilizar os recursos de edição da sua máquina local.

Passemos agora ao Codebender. Ele é uma plataforma web que tem como principais objetivos o compartilhamento de arquivos e a aprendizagem colaborativa (CODEBENDER, 2015). A interface é bem desenhada e o usuário possui dois canais simultâneos de suporte e monitoramento. O ambiente permite que o usuário crie suas próprias bibliotecas de programas, facilitando a cooperação entre eles. Disponibiliza também um serviço de tutoriais que facilita a construção de programas / projetos mais complexos.

---

<sup>14</sup> São arquivos que estão no formato do sistema numérico hexadecimal. Este é um sistema de numeração que representa os números na base 16, portanto empregando 16 símbolos.

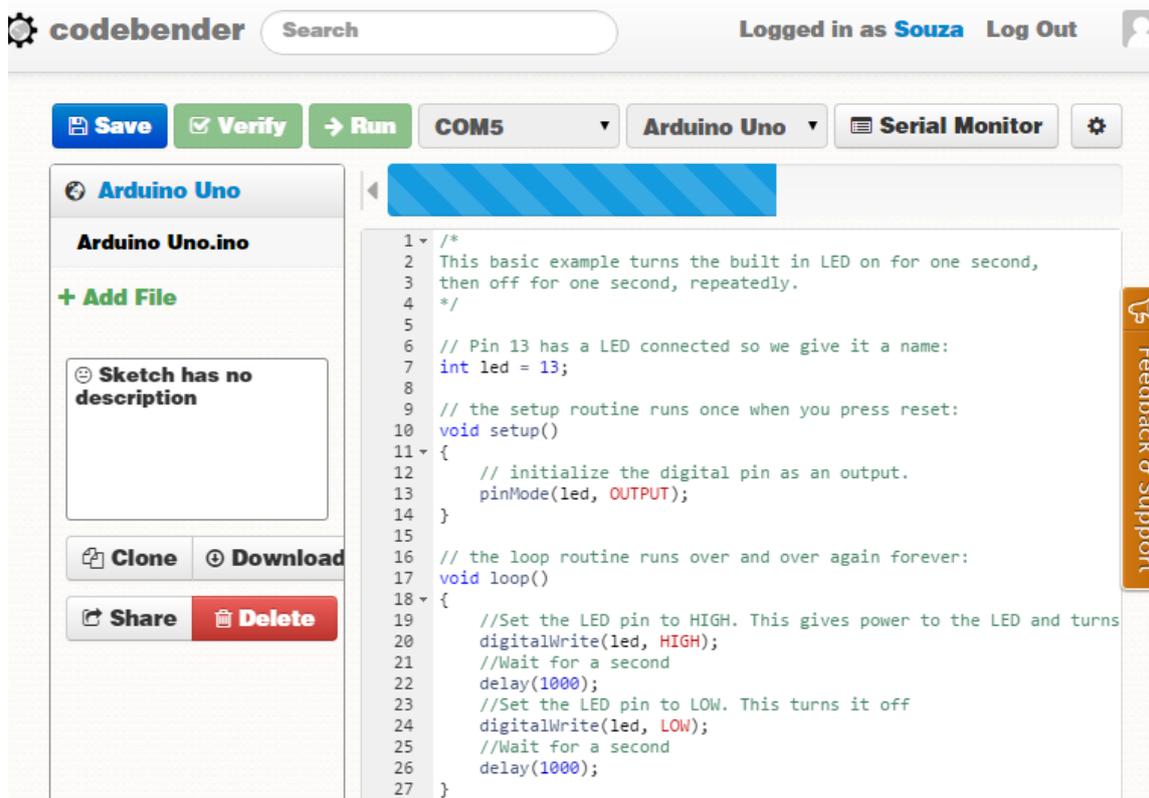


Figura 16: Interface do Codebender. Fonte: (CODEBENDER, 2015).

O Codebender (Figura 16) é uma plataforma mais completa que o Arduino Online Compiler, no que se refere às funções de programação e compilação. Suas mensagens de verificação / compilação apontam exatamente onde está o erro. Contudo, se pensarmos em usuários em estado incipiente, perceberemos que essas informações não serão suficientemente explicativas. Outro ponto que pode prejudicar a funcionalidade dessa plataforma é a necessidade de uma conexão de banda larga, com alta velocidade.

### 3.3 Reflexão sobre os Trabalhos Relacionados

A partir da revisão de literatura e dos trabalhos relacionados foi possível levantar as características operacionais e de implementação, que o ambiente didático do LabVad deve possuir. Algumas características relevantes dos laboratórios estudados justificaram a escolha dos requisitos funcionais da plataforma WEB do LabVad, não obstante existem alguns pontos que merecem um olhar mais atento.

O LabVIEW foi apresentado com o custo benefício vantajoso em relação a seus concorrentes (ERTUGRUL, 2000), mas o custo final ainda é alto. Além disso, o software possui um ciclo de atualizações anuais, que exigem alto investimento financeiro (LABVIEW, 2015), uma condição que pode comprometer o planejamento e gestão de um projeto em longo prazo, levando em consideração que se leva pelo menos cinco anos para formar um engenheiro no Brasil.

Existem versões do iLab que atendem diversas instituições de ensino no mundo, mas toda sua arquitetura, bem como a tecnologia empregada em seu desenvolvimento é proprietária, exclusivamente da Microsoft Corporation (HARWARD et al., 2008). Exceto pelas tecnologias da linguagem Java, que constituíram uma pequena parte do projeto, os softwares da Microsoft acarretam um alto investimento financeiro. Estas questões foram determinantes para que o LabVad fosse projetado para plataformas de software e hardware livres.

Os ambientes Arduino Online Compiler e o CodeBender influenciaram o LabVad nos requisitos de desempenho, eficiência, compilação de comandos, contexto na manipulação de dados, persistência dos arquivos e, principalmente, na construção de um suporte eficaz e com enfoque no usuário.

Do ponto de vista tecnológico, as plataformas do Arduino Remote Lab, CLEM e o HexLab são muito semelhantes à proposta do LabVad, mas o escopo do nosso projeto está delineado em dois objetivos principais: um ambiente de programação semelhante a IDE do Arduino, idêntico em sua sintaxe, à linguagem de programação Wiring e uma coleção de artefatos robóticos, análogos a um kit didático convencional de robótica, como será apresentado nos capítulos subsequentes.



## 4. Requisitos do Sistema

Nesta etapa do presente trabalho, descreve-se os procedimentos realizados para o levantamento dos requisitos do ambiente LabVad (seções 4.1 e 4.2), seguidos da apresentação dos diagramas de casos de uso do sistema (seção 4.3) e seu diagrama conceitual de classes (seção 4.4).

## 4.1 Levantamento de Requisitos

Crinnion J. (1991) afirma que a compreensão completa dos requisitos é fundamental para um desenvolvimento de um software bem-sucedido. No entanto, é pertinente considerar que a identificação inicial de todos os requisitos, na realidade, não é somente um momento de uma sequencial, ou seja, ela não se insere numa fase de uma linearidade. Neste projeto, a identificação das necessidades dos usuários foi processual, ou seja, realizada durante todo o seu ciclo de vida, mas com uma maior intensidade no seu começo.

Nesse ciclo, o levantamento de requisitos foi feito em diferentes momentos, tendo seu início com as condições de contorno contextuais descritas por Alves (2013) e com a revisão da literatura de softwares e com a revisão da literatura de softwares e ambientes de robótica na web.

Os “trabalhos relacionados” sobre uso de laboratórios remotos influenciaram e fortaleceram características funcionais do LabVad, mas sobretudo influenciados pelo modelo tecnológico preconizado por Victorino et al (2009). A equipe do GINAPE, em diferentes seminários e reuniões, discutiu sobre as principais características a serem implementadas no LabVad. Por fim, pesquisas de campo realizadas durante os cursos de Robótica Educacional de Baixo Custo – REBC - promovidos pelo projeto, também foram de grande importância na identificação das necessidades do público-alvo.

## 4.2 Principais Requisitos

A coleta de opiniões do público estratégico, como professores em processo de formação em RE e especialistas na área de Tecnologias no Ensino e Robótica, ajudou a compreender as reais necessidades dos potenciais usuários. Os quatro principais requisitos levantados que o ambiente LabVad deve prover são:

- Similaridade com um ambiente real de RE (subseção 4.2.1);
- Requisito de portabilidade: Ambiente Multiplataforma (subseção 4.2.2);

- Requisitos de eficiência e desempenho: adequação às limitações do Classmate (subseção 4.2.3).
- Ambiente Multiusuário (subseção 4.2.4).

#### **4.2.1 Similaridade com um ambiente real de RE**

Este requisito, definido ainda na fase de proposta do projeto, trata da similaridade que o ambiente na internet deve prover, seu principal intuito é se aproximar ao máximo da situação em que se encontra um usuário numa bancada com computador, IDE e hardware Arduino conectado à sua máquina (ambiente local).

Em suma, todo o ambiente de programação do LabVad deve ter correspondência nas funcionalidades existentes no ambiente local. Deve ser possível também ao usuário visualizar a execução dos seus experimentos no LabVad de forma muito semelhante ao que visualizaria, caso tivesse um hardware Arduino conectado ao seu computador.

O hardware do LabVad deve ser semelhante a um Kit educacional de Robótica, do mesmo modo que a plataforma deve fornecer toda a funcionalidade da linguagem Wiring ao usuário.

#### **4.2.2 Requisito de portabilidade: Ambiente Multiplataforma**

Um aspecto notável levado em consideração no desenvolvimento do LabVad foi a existência de diferentes sistemas operacionais disponíveis para alunos e professores nos diferentes equipamentos utilizados por eles. Os laptops das escolas do PROUCA possuem o sistema operacional MeeGo (METASYS MEEGO, 2010) ou alguma outra distribuição Linux. Por sua vez, os computadores, que os alunos, porventura, possuem em suas residências, têm – com uma boa chance – o sistema operacional Windows. Já os computadores distribuídos aos professores da rede pública de ensino nos diferentes estados do Brasil vêm também com alguma versão Linux. A alternativa econômica, e alinhada às tendências atuais no desenvolvimento de software (ZHANG et al., 2010) para equacionar tal problema, foi a de implementar um ambiente multiplataforma que fosse executado na nuvem.

O requisito definido acima traz no seu bojo três outros importantes aspectos positivos. Primeiro, a eliminação da complexidade de uma eventual instalação, configuração ou atualização do sistema, possibilitando aos usuários o acesso ao LabVad sem a necessidade de conhecimento sobre a tecnologia utilizada. Segundo, a redução da necessidade de espaço de memória ou disco nos laptops, uma vez que os projetos podem ser salvos na nuvem e as máquinas dos usuários não necessitam ter alta capacidade de processamento (SOUZA et al., 2010). Terceiro, a introdução de facilidades para o compartilhamento e colaboração dos projetos desenvolvidos pelos alunos e professores.

### 4.2.3 Requisitos de eficiência e desempenho

As funcionalidades do sistema devem ser adequadas às limitadas especificações de hardware e periféricos do laptop Classmate<sup>15</sup> (2013). As principais limitações da última versão deste equipamento são:

- O tamanho reduzido da tela de 8.9" - o layout do sistema deverá ser adequado a essa pequena dimensão e de preferência ajustável para que também se adeque às telas de alta resolução.
- A ausência de mouse, a usabilidade do sistema deverá considerar o uso do touchpad<sup>16</sup>.
- As limitações de processamento, de armazenamento e de banda de internet: no desenvolvimento do sistema deve-se atentar para o tempo de execução das ações, a quantidade de memória requerida e a baixa largura de banda, situação hoje (ainda) existente nas escolas.

### 4.2.4 Ambiente Multiusuário

Ao se considerar que o LabVad é uma ferramenta educativa, foi determinado que o mesmo fosse multiusuário em todas as suas funções, exceto pela transmissão

---

<sup>15</sup> O ClassMate é um laptop de configuração simples, comprado pelo Governo Federal e presente em boa parte das escolas parceiras do PROUCA.

<sup>16</sup> Touchpad é uma área sensível ao toque que realize as mesmas tarefas de um mouse comum.

dos experimentos, porque as imagens são geradas por um servidor de transmissão de vídeo, contendo um IP fixo<sup>17</sup>.

Mesmo quando um usuário estiver agendado, qualquer outro terá acesso a todas as outras funções, principalmente à lista de agendamentos, onde ele pode visualizar os horários vagos e quais usuários estão agendados para que no futuro eles se comuniquem, seja por e-mail ou através de fórum, que pode ser implementado no sistema.

Outra função, que faz parte da propriedade multiusuário é o compartilhamento de arquivos entre usuários através do acionamento de um simples ícone. Essa função facilita, assim, a aprendizagem colaborativa ou cooperativa, proposta por alguns educadores e discutida já há alguns anos em diversas vertentes da educação.

Neste arrolamento de pontos positivos que agora fazemos, vale apontar para um outro igualmente importante: a possibilidade de o professor planejar suas aulas, mesmo fora dos horários de agendamento, pois a edição, manipulação e gravação de arquivos devem ser feitas e gravadas diretamente no servidor LabVad NCE.

Por último, o LabVad também pode ser usado de forma coletiva em sala de aula. Além de visualizar o experimento em execução, o professor pode criar um e-mail que toda a turma possa acessar e manipular esse experimento simultaneamente. Assim, como agente facilitador da aprendizagem, o mesmo deve saber conduzir esse processo, bem como trabalhar, em aula, metodologias de programação como o Coding Dojo.

Um Coding Dojo é uma reunião onde um grupo de estudantes ou programadores se encontra para trabalhar em um desafio de programação. Os alunos participam para aprender de forma lúdica, a fim de melhorar suas habilidades (CODING DOJO, 2015). Nessa metodologia, um computador é conectado a um projetor a fim de que todos possam, simultaneamente, ver uma programação par a par, com a participação ativa de outros alunos, conduzidos por um professor.

---

<sup>17</sup> É um conjunto de número que identifica, de forma única, cada dispositivo em uma rede. IP fixo significa que o IP será sempre a mesmo, toda vez que o dispositivo ou hardware se conectar a rede.

### **4.3 Diagrama de Casos de Uso**

Booch et al. (2006) afirmam que uma das principais fases de engenharia de software é o Levantamento de Requisitos apresentado nas seções anteriores e, a partir desta etapa, a Análise de Requisitos constitui a principal etapa da modelagem do Projeto / Sistema.

Em todas as etapas na modelagem do sistema, o LabVad foi modelado para dois perfis diferentes de diferentes usuários: superusuários (professores) e usuários (alunos).

No contexto escolar, o professor é o superusuário, com autonomia de ser mediador dos seus alunos e detentor de acesso a funcionalidades importantes, como o registro das atividades dos alunos no sistema. Com o intuito de apresentar as funcionalidades propostas e a interação usuários-sistema, elaborou-se o diagrama e

caso de uso mostrado na Figura 17.

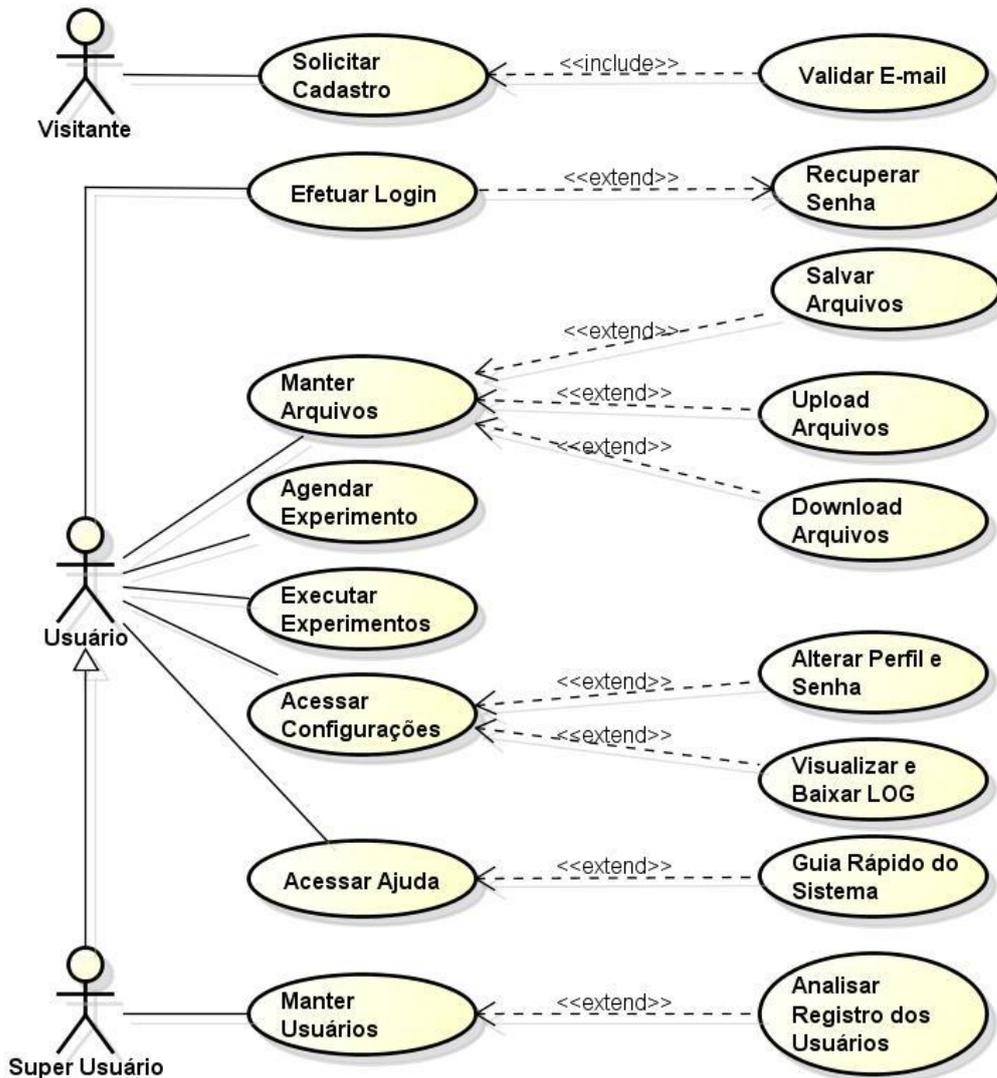


Figura 17: Diagrama de Caso de Uso.

A seguir são detalhados os casos de uso que o ambiente LabVad proverá e seu comportamento em certas circunstâncias, descrevendo como os usuários interagem com o sistema.

Tabela 1: Caso de Uso – Solicitar Cadastro

<b>Ator</b>	Visitante.
<b>Precondições</b>	Possuir e-mail válido.
<b>Fluxo Normal</b>	1. O usuário informa dados para cadastro. 2. A senha do primeiro acesso é enviada por e-mail

Tabela 2: Caso de Uso – Efetuar Login

<b>Ator</b>	Usuário.
<b>Precondições</b>	1. Ter executado o caso de uso “Solicitar Cadastro”
<b>Fluxo Normal</b>	1. O usuário informa e-mail e senha.
<b>Fluxo Alternativo: Esqueceu Senha</b>	1. O usuário informa o e-mail e clica em “Esqueci senha”. 2. O sistema envia um e-mail ao usuário com a senha.

Tabela 3: Caso de Uso – Manter Arquivos

<b>Ator</b>	Usuário.
<b>Precondições</b>	1. Ter executado o caso de uso “Efetuar Login”.
<b>Fluxo Normal</b>	1. O usuário cria um novo arquivo, abre exemplos, edita arquivos.
<b>Fluxo Alternativo: Salvar Arquivos</b>	1. O usuário salva um arquivo que foi editado.
<b>Fluxo Alternativo: Upload Arquivos</b>	1. O usuário envia um arquivo do seu computador para o servidor LabVad.
<b>Fluxo Alternativo: Download Arquivos</b>	1. O usuário baixa um arquivo do servidor LabVad para seu computador.

Tabela 4: Caso de Uso – Agendar Experimentos

<b>Ator</b>	Usuário.
<b>Precondições</b>	1. Ter executado o caso de uso “Efetuar Login”.
<b>Fluxo Normal</b>	1. O usuário escolhe uma data do mês atual para executar seus experimentos. 2. O usuário escolhe os horários na data desejada. 3. Sistema envia mensagem que o agendamento foi executado com sucesso. 4. O usuário pode desmarcar agendamentos.
<b>Fluxo Alternativo: Horário já reservado.</b>	2. O sistema envia aviso de que o horário está reservado. Retorna ao passo 2.

Tabela 5: Caso de Uso – Executar Experimentos

<b>Ator</b>	Usuário.
-------------	----------

<b>Precondições</b>	1. Ter executado o caso de uso “Efetuar Login”. 2. Ter executado o caso de uso “Agendar Experimentos”.
<b>Fluxo Normal</b>	1. O usuário cria e edita um arquivo Arduino. 2. O usuário executa seu Experimento. 3. O sistema informa que o arquivo foi compilado com sucesso. 4. O sistema informa que o arquivo não pode ser compilado.
<b>Fluxo Alternativo:</b> Erro na compilação.	4. Erro na compilação Retornar ao passo 1.

Tabela 6: Caso de Uso – Acessar Ajuda

<b>Ator</b>	Usuário
<b>Precondições</b>	1. Ter executado o caso de uso “Efetuar Login”.
<b>Fluxo Normal</b>	1. O usuário acessa a seção “Ajuda”. 2. O usuário pode visualizar ou baixar um guia rápido para aprender a programar no LabVad.

Tabela 7: Caso de Uso – Manter Usuários

<b>Ator</b>	Superusuário
<b>Precondições</b>	1. Ter executado o caso de uso “Efetuar Login”.
<b>Fluxo Normal</b>	1. O superusuário pode cadastrar Usuários. 2. O superusuário pode editar nome, senha e status de todos os usuários. 2. Visualizar e baixar os registros dos usuários para analisar dados.

#### 4.4 Diagrama de Classes

Após utilizarmos o diagrama de casos de uso para explicar de forma elucidativa e dinâmica o sistema, mostraremos o diagrama de classes do LabVad. De acordo com Booch et al., (2006) o diagrama de classes serve como base para todos os outros diagramas do sistema e apresenta a visão estática de como as classes estão organizadas logicamente, com seus respectivos atributos e métodos.

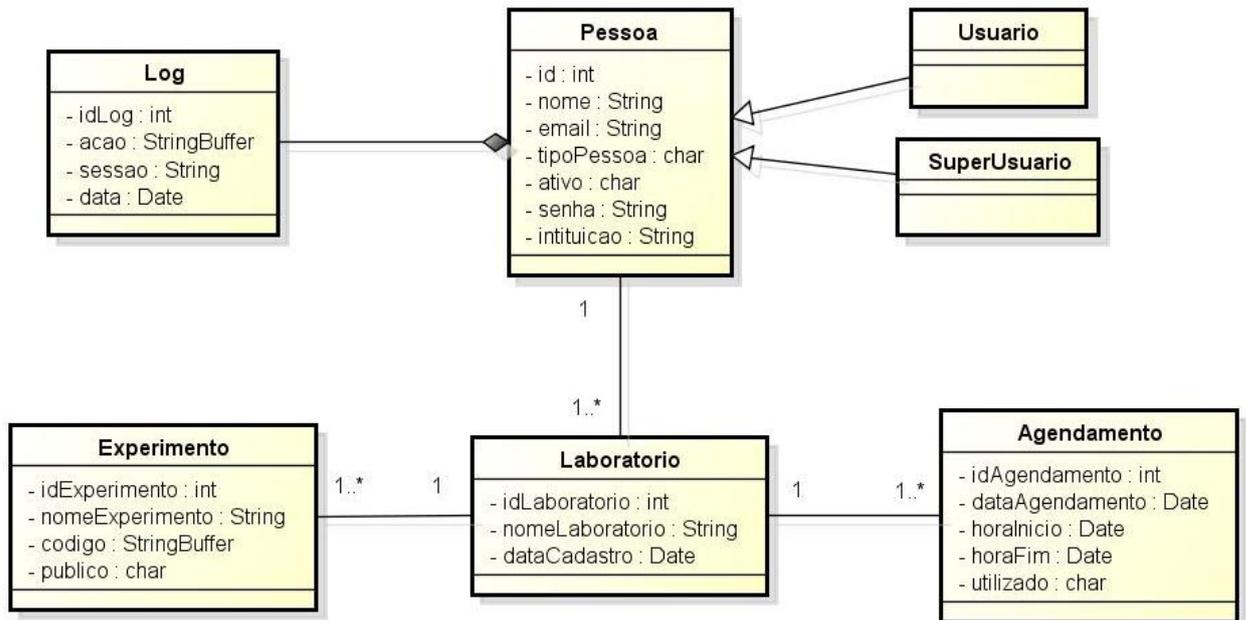


Figura 18: Diagrama de Classes conceitual



## **5. Metodologia de Desenvolvimento**

Os propósitos deste capítulo organizam-se em torno das seguintes ações: explicar a adaptação da metodologia Extreme Programming (XP) utilizada na concepção do LabVad (seção 5.1); expor a aplicação do desenvolvimento em camadas (seção 5.2); abordar os softwares utilizados na concepção desse ambiente (seção 5.3); descrever e justificar sua interface responsiva (seção 5.4) e, por último, explicitar a relação dos softwares utilizados na construção da plataforma WEB do LabVad (Seção 5.5).

## 5.1 Metodologia Extreme Programming (XP)

Flower (2005) define a UML como uma família de notações gráficas que ajuda na descrição e na construção dos projetos de softwares orientados a objetos. No capítulo imediatamente anterior a este utilizou-se a UML (Linguagem de Modelagem Unificada) para especificar o produto da análise de requisitos que serão exploradas neste capítulo.

Para desenvolver o LabVad, foi escolhida uma adaptação da metodologia ágil Extreme Programming, por ser compatível com o escopo e tamanho do projeto e por fornecer uma versão adaptável ao desenvolvimento individual, definida como eXtreme Solo. Cronin (2001) apresenta os quatro valores fundamentais da XP:

- **Comunicação:** deve conter o que for necessário para a explicação do código fonte, diagramas em UML, descrição de casos de uso, etc.
- **Simplicidade:** as práticas XP, que reduzem a complexidade do sistema, são projeto com especificações bem definidas em seus algoritmos, tecnologias e técnicas.
- **Feedback:** quanto mais cedo forem detectados os problemas, mais rápido serão resolvidos ou contornados. O objetivo é tentar resolver problemas no código, processo, interface, requisitos, etc.
- **Coragem:** em comparação às metodologias clássicas de desenvolvimento de software, o desenvolvedor fica mais confiante e seguro para: alterar o código que está funcionando a fim de torná-lo mais simples; jogar o código fora quando for necessário; alterar o código a qualquer momento e pedir ajuda a seus colaboradores e usuários.

Cronin (2001) ainda arrola um conjunto de sugestões de atitudes e de práticas que auxiliam nos valores acima citados sobre a XP:

- **A Equipe:** deve ser composta pelo desenvolvedor e pelo usuário;
- **Testes de Usuário:** uma bateria de testes deve ser elaborada pelo usuário e implementada automaticamente;

- Versões Pequenas: cada versão do software é funcional. O usuário pode utilizar o sistema na medida em que seus módulos ficarem prontos e não apenas na fase final;
- Jogo de Planejamento: estimativas e prioridades devem ser estimadas de acordo com prazo e custo;
- Padrão de Codificação: um padrão deve ser seguido como se uma só pessoa tenha elaborado todo o sistema;
- Integração Contínua: a integração dos módulos do sistema deve ser feita com frequência para eliminar o mais rápido possível os erros de integração;
- Metáfora: uso de analogias para facilitar o desenvolvimento;
- Ritmo sustentável: prazos adequados que não afetem a produtividade, a comunicação e a disciplina;
- Projeto Simples: codificar apenas o necessário;
- Orientação a Testes: os testes devem ser executados constantemente;
- Refatoramento: Testes frequentes e melhoria contínua facilitam a manutenção e qualidade do software;
- Programação em Par: Infelizmente, não é possível na modalidade eXtreme Solo. A programação em pares significa que toda a programação é feita como um par de programadores num único terminal. Pesquisas indicam que um código elaborado por duas pessoas possui uma melhora significativa na qualidade;
- Posse Coletiva: esta também é uma propriedade original da XP que não se aplica a eXtreme Solo, onde toda equipe seria responsável pelo projeto em que todos são donos do código fonte;

## 5.2 Desenvolvimento em camadas

Optou-se pelo desenvolvimento em camadas com o padrão Model-View-Controller para cumprir o requisito de eficiência e desempenho (seção 4.2.3 do capítulo 4). A reutilização das classes de negócio, bem como seus componentes WEB, faz com que a plataforma do LabVad seja executada no navegador do usuário, levando em consideração possíveis problemas de banda larga recorrentes nas escolas públicas brasileiras.

Outro fator decisivo foi a opção por um código organizado que viabilizasse a sua manutenção e escalabilidade<sup>18</sup>, portanto o LabVad foi projetado para funcionar fracionado em três distintas camadas a conhecer:

- **Modelo (Negócio):** é a camada que controla a persistência dos dados do sistema. No LabVad, existem dados que “vivem” temporariamente e é imperativo que outros dados existam até o término da aplicação. Para isso, é necessário armazená-los em um meio físico permanente. Assim, nesta camada, encontrar-se-ão esses dados “persistidos” em simples arquivos temporários ou no banco de dados fisicamente;
- **Visão (Apresentação):** engloba as classes que serão responsáveis pela interação do usuário com o próprio sistema. Utiliza os dados do modelo para apresentar as informações, interage com o usuário e envia dados do modelo controlador;
- **Controlador:** define a maneira que a interface do usuário reage às entradas fornecidas pelo mesmo. Processa os dados, usando funções de negócio do modelo e escolhe a visão que será apresentada aos usuários;

## 5.3 Softwares utilizados na concepção do LabVad

Pinto et al. (2012) apontam o caráter interativo e colaborativo de projetos com tecnologias livres. Eles estão associados à internet e às comunidades de usuários

---

<sup>18</sup> Escalabilidade é uma característica desejável em todo o sistema, que indica sua capacidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer.

que se auxiliam mutuamente no sentido de esclarecimento, soluções de problemas e fortalecimento de projetos.

A partir destas premissas, e procurando minimizar os custos de construção e manutenção do projeto, optou-se, no desenvolvimento deste último, pela utilização de softwares livres consolidados.

Desta forma, utilizou-se em sua concepção as linguagens de programação PHP<sup>19</sup>, HTML5<sup>20</sup>, CSS3<sup>21</sup>, jQuery<sup>22</sup> e o banco de dados MySQL<sup>23</sup>. O banco de dados usado do LabVad está configurado com a criptografia do algoritmo MD5 de chave de 128 bits (PHD, 2010). Este algoritmo transforma um conjunto de caracteres legível em um código complexo, ou seja, ele transforma uma palavra simples em 32 dígitos hexadecimais. Exemplo: selecionar MD5 ('teste'). Resultado: 'Ae2b1fca515949e5d54fb22b8ed95575'.

Para garantir o funcionamento de todas estas tecnologias supramencionadas, necessita-se utilizar o Servidor Apache. Este servidor é instalado e configurado através do pacote multiplataforma denominado **XAMPP**, este é um acrônimo em que: **X** representa qualquer sistema operacional; **A** (Apache); **M** (MySQL); **P** (PHP) e **P** (Perl).

O PHP permite processar, diretamente no servidor do NCE, o conteúdo dinâmico das páginas e exibi-lo dinamicamente nos navegadores dos computadores do usuário (PHD, 2010). A plataforma do LabVad também executa um servidor de *stream* de vídeo, que é responsável pela transmissão das imagens dos experimentos de robótica em tempo real.

---

<sup>19</sup> PHP: Hypertext Preprocessor, originalmente Personal Home Page.

<sup>20</sup> Hypertext Markup Language ou Linguagem de Marcação de Hipertexto, versão 5.

<sup>21</sup> Cascading Style Sheets ou linguagem de folhas de estilo, versão 3.

<sup>22</sup> É uma biblioteca JavaScript desenvolvida para simplificar interação do computador do cliente com a aplicação, construída em HTML.

<sup>23</sup> É um sistema de gerenciamento de banco de dados relacional que utiliza a linguagem SQL - Linguagem de Consulta Estruturada.

Na transmissão de vídeo, priorizaram-se a iluminação adequada, a qualidade da imagem e a redução nos atrasos de visualização. O software responsável pela transmissão das imagens é o VideoLAN - VLC (FALON et al., 2004), uma vez que o mesmo não necessita da instalação de players de vídeos, nem direciona a função da transmissão de vídeo para outros serviços como o Videolog ou o Youtube. Tais características permitem uma melhor geração de imagens, sem a necessidade de instalação de software especial nas máquinas dos usuários.

#### **5.4 API Web Storage e a Interface Responsiva**

Antes do HTML5, os dados de uma aplicação eram armazenados em cookies. Cookie é um arquivo de texto muito simples, cuja composição depende do conteúdo do endereço da WEB visitado. A maioria dos sites armazenam informações básicas, como endereços IP e preferências sobre idiomas, cores, etc.

O armazenamento local é mais rápido, seguro e não afeta o desempenho da aplicação. Para garantir a liberdade e segurança do usuário durante a navegação pelos diferentes módulos da plataforma do LabVad, foi utilizada a API Web Storage (W3C, 2014). Optamos por trabalhar com muito mais espaço de armazenamento, onde os dados não são transitados a cada requisição de página e que, ainda por cima, não têm horário predeterminado para expirar.

A liberdade e segurança que o usuário possui em relação ao seu arquivo Arduino são garantidas. O usuário pode alternar sua navegação entre guias ou janelas, executar outras funções dentro do LabVad, como agendar horários, consultar ajuda e alterar sua senha. Mesmo depois de transitar por todos estes estados, o código Arduino de seu experimento continuará ileso. O usuário mesmo não é interrompido diversas vezes a cada ação, como, por exemplo, na plataforma da Figura 19, onde o usuário é obrigado a salvar seu e-mail como rascunho ou excluir.

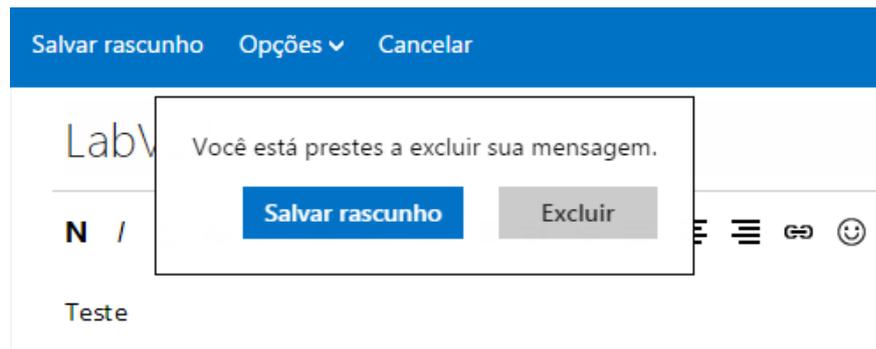


Figura 19. Exemplo de plataforma de correio eletrônico.

Outra função descartada foi o salvamento automático do código Arduino, como se processa em outras plataformas e/ou serviços WEB. Observa-se na Figura 20 que o usuário não é obrigado a salvar ou descartar seu texto (mensagem de e-mail), mas com o tempo de uso ele pode acumular centenas de arquivos de rascunho, sem alguma utilidade, o que pode comprometer a usabilidade do sistema. Rocha e Baranauskas (2003) dizem que as interfaces atuais seguem a tendência do fenômeno “fatware”, sobrecarregadas com funções desnecessárias. Ter somente opções necessárias na execução de uma tarefa significa melhor usabilidade, pois o usuário pode se concentrar em poucas funções para executar a tarefa necessária.

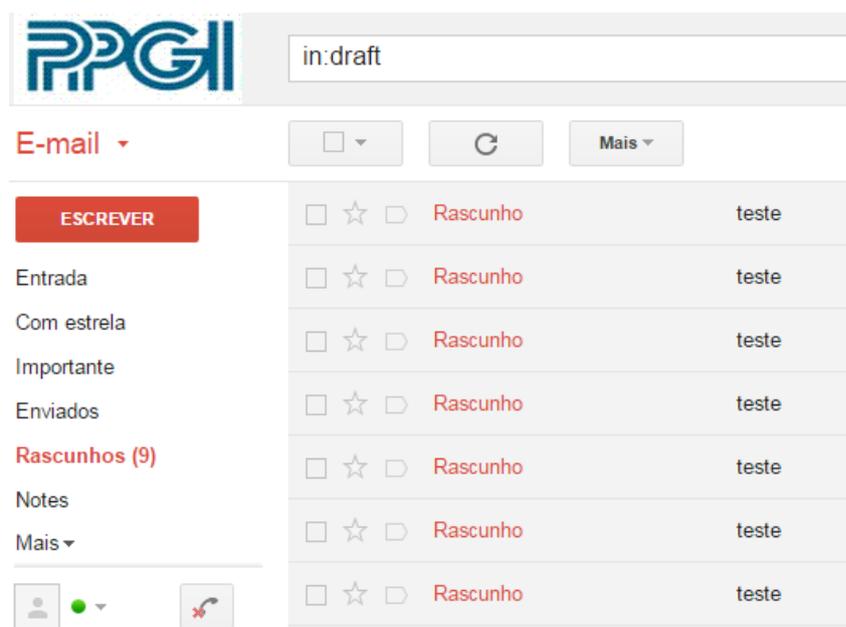


Figura 20. Plataforma de correio eletrônico PPGI UFRJ.

Para Silva et al. (2013), a presença expressiva dos dispositivos móveis na vida das pessoas tem alterado de forma significativa os estilos de vida da sociedade, mais especificamente a vida dos jovens. A ubiquidade dos dispositivos móveis nos momentos da vida cotidiana e a sua contínua utilização fazem com que esta tecnologia se constitua num elemento frequente na vida das pessoas através de múltiplas práticas, que incluem a gestão e manuseio da tecnologia.

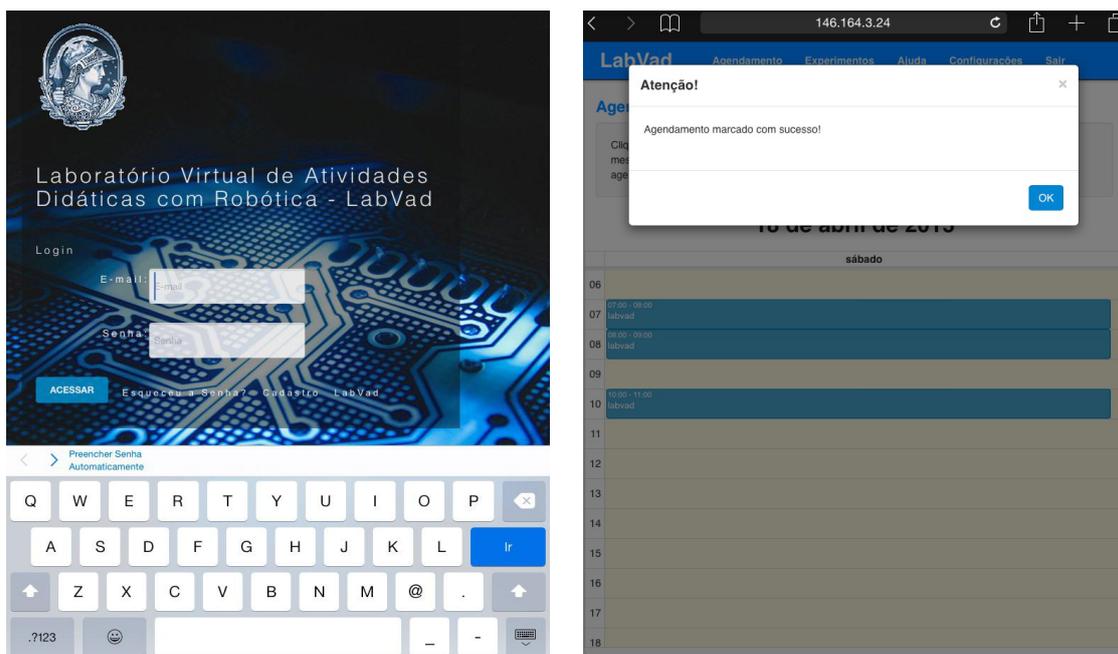


Figura 21. Design responsivo: compatível com dispositivos móveis.

## 5.5 Relação de Softwares

No Quadro 3 fazemos um arrolamento de todos os softwares utilizados na concepção, manutenção e funcionamento da plataforma do LabVad.

Recurso	Finalidade	End. Eletrônico
NetBeans	IDE de desenvolvimento do PHP.	<i>netbeans.org</i>
XDebug	Ferramenta CASE utilizada para depuração de código PHP.	<i>xdebug.org</i>
jQuery	Biblioteca Java Script.	<i>http://jquery.com/</i>
Bootstrap	Framework para desenvolvimento responsivo.	<i>http://getbootstrap.com/</i>
Storage	Permite o armazenamento local na WEB.	<i>http://dev.w3.org/html5/storage/</i>
XAMPP	Servidor para uso síncrono com o	<i>https://www.apachefri</i>

	MySQL.	<i>ends.org/</i>
GIMP	Editor de imagens.	<i>www.gimp.org</i>
Fritzing	Utilização das imagens dos componentes.	<i>fritzing.org</i>
Astah	Criação de caso de uso e diagrama de classes.	<i>astah.net</i>
Chrome	Navegador escolhido para teste.	<i>http://www.google.com.br/chrome/</i>

**Quadro 3: Ferramentas e recursos utilizados.**



## **6. Implementação do LabVad**

Este capítulo tem como objetivo a apresentação do layout das principais funcionalidades e da arquitetura física e o portal (plataforma WEB), que foi desenvolvido neste trabalho. A seção 6.1 explica minuciosamente a arquitetura física do LabVad; a seção 6.2 apresenta a plataforma WEB que foi contruída (com ênfase na execução de experimentos); a seção 6.3 descreve o processo de compilação no Servidor do NCE e no Navegador WEB do usuário e a seção 6.4 destaca as configurações pessoais dos usuários.

## 6.1 Arquitetura Física do LabVad

Esta seção descreve as tecnologias empregadas na construção do LabVad através do levantamento de requisitos (capítulo 4) e da metodologia empregada no desenvolvimento do LabVad (capítulo 5).

Como foi visto anteriormente, o LabVad é constituído por dois componentes distintos: um hardware e uma plataforma WEB, que é controladora das funcionalidades desse hardware. O hardware é composto por circuito eletrônico, desenvolvido por pesquisadores do GINAPE, acoplado a uma placa Arduino (Figura 22).

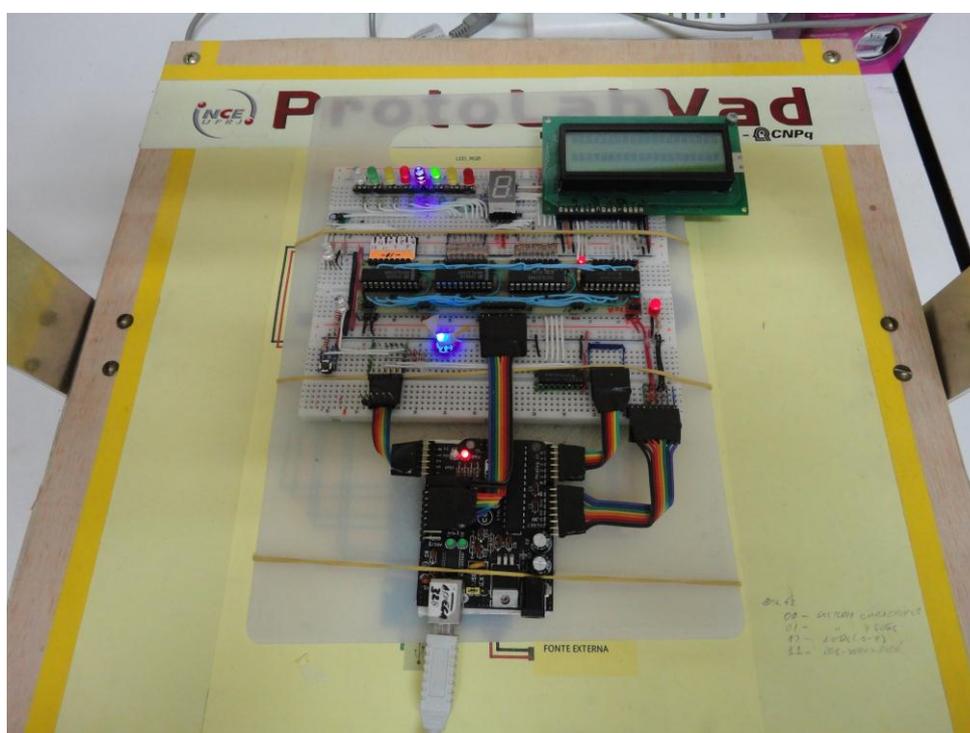


Figura 22. Parte física do LabVad.

A eletrônica cumpre o papel de permitir, de forma transparente para o usuário, a multiplexação<sup>24</sup> de diferentes experimentos no Arduino, organizando-os em grupos e ampliando as possibilidades pedagógicas do projeto.

A conexão entre os dispositivos periféricos (LED, RGB, servo-motor, display, etc.) e os pinos da placa Arduino já estão previamente definidos. Por exemplo, o

---

<sup>24</sup> Multiplexação é a técnica que utiliza um dispositivo que codifica as informações de duas ou mais fontes de dados em um único canal.

RGB e o servo-motor estão conectados no mesmo pino 11 do Arduino. Ao criar seu programa, o usuário seleciona, previamente na plataforma WEB do LabVad, qual dos dois componentes pretende utilizar no seu experimento. Na Figura 23, temos o diagrama completo do hardware do LabVad.

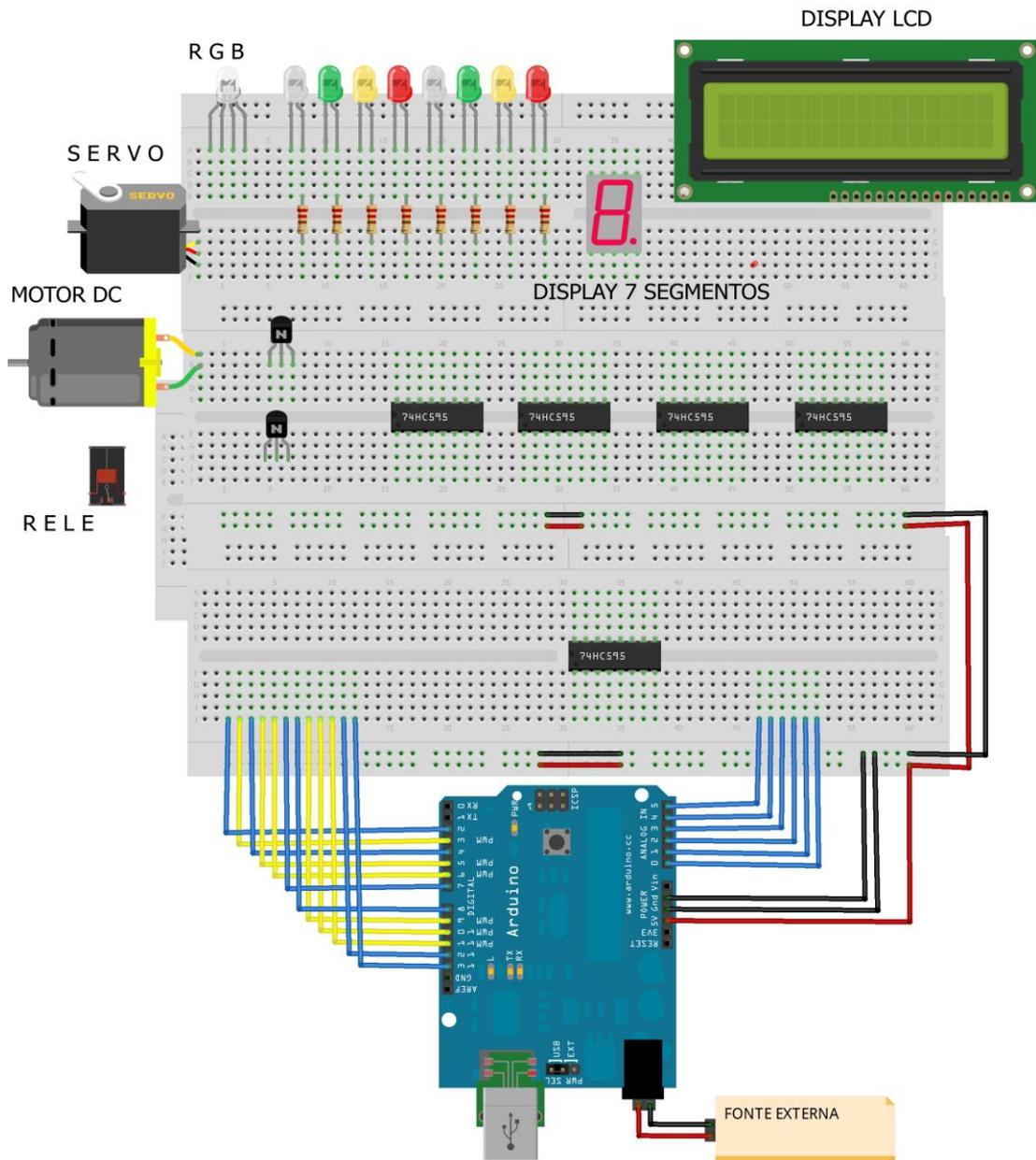


Figura 23. Diagrama completo do hardware do LabVad.

Componente	Quantidade
Webcam	1
Placa Arduino Uno	1
Cabo USB	1
Fonte DC 9 volts	1
Protoboard	3
LED (Light Emitting Diode)	8
LED RGB	1
Display de Caracteres (LCD)	1
Servo Motor	1
Display de 7 segmentos	1
Motor DC 5 volts	1
Relé	1
Resistores 680 $\Omega$	8
Resistores de 1k $\Omega$	8
Resistores de 220 $\Omega$	4
Transistores NPN	2
Circuito integrado 74LS244	4
Circuito integrado 74LS239	1
Fios diversas cores	5 metros
Conector de 3 pinos para cargas de 110 volts (Relé)	1

Quadro 4: Componentes do LabVad.

O funcionamento do hardware LabVad depende de um programa criado pelo usuário com instruções que permitam o funcionamento dos dispositivos eletromecânicos e/ou eletroeletrônicos. A visualização da execução de um programa do usuário se efetua por intermédio da transmissão de imagens capturadas por uma câmera acoplada ao hardware.

## 6.2 Experimentos

Na parte superior temos um menu com as principais funções do sistema (Figura 24).

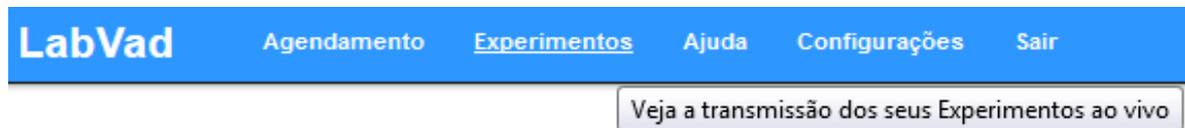


Figura 24. Menu Principal.

Logo abaixo a **transmissão dos Experimentos** executados pelos usuários e transmitidos em tempo real pelo servidor do LabVad NCE. A interface foi meticulosamente projetada para que a programação dos seus dispositivos seja intuitiva. Os LEDs estão conectados sequencialmente como em uma protoboard; o relé, por ser um dispositivo que trabalha com alta voltagem, foi associado à imagem de uma cafeteira e o servo motor está disposto em ângulos de 0° a 180°, como podemos ver na Figura 25.

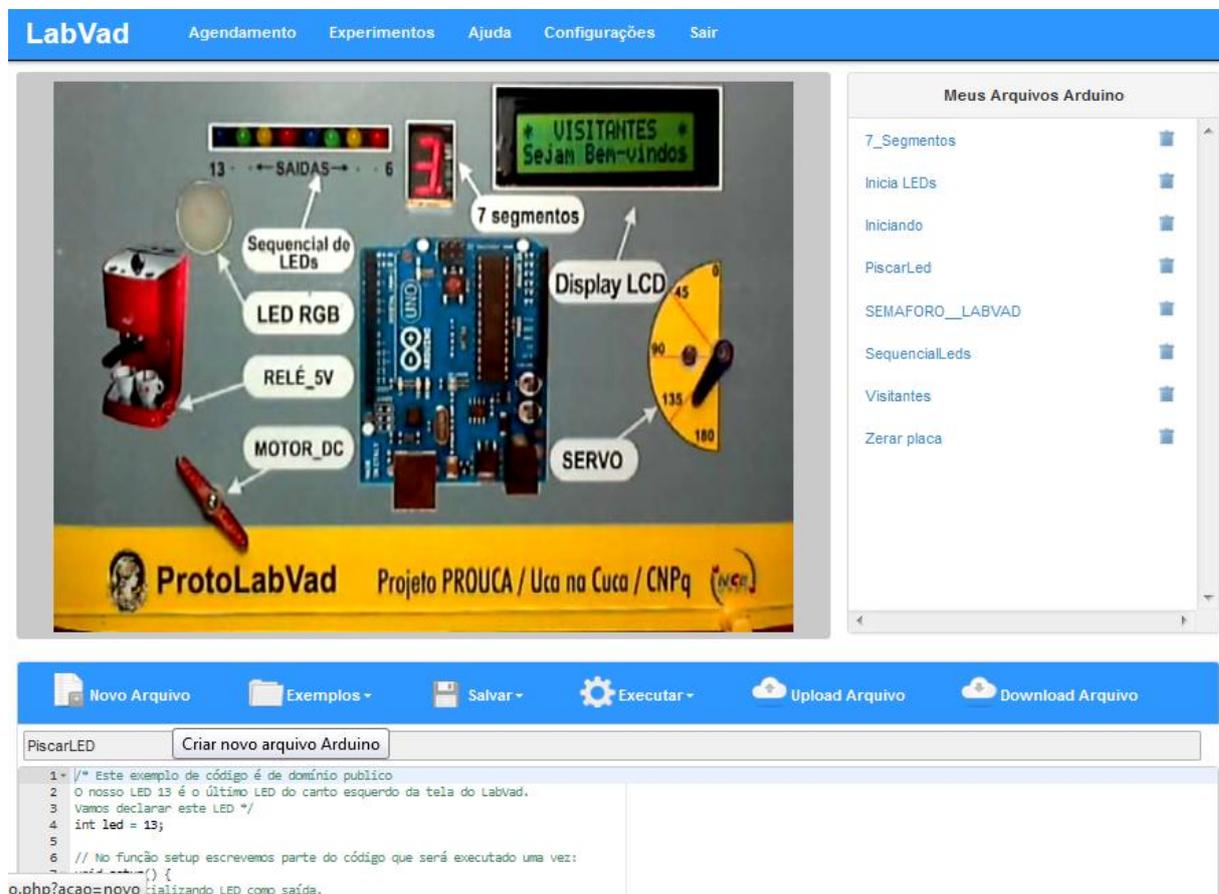


Figura 25. Transmissão dos Experimentos em tempo real.

Ao lado da transmissão dos Experimentos, temos o painel “Meus Arquivos Arduino” (Figura 26). A função deste painel é exibir os programas salvos (gravados) pelos usuários no servidor. O usuário possui o conforto de não se preocupar com dispositivos de armazenamento ou restrições de memória do seu próprio computador, como foi descrito no capítulo 4, onde foi tratado do requisito de desempenho e eficiência.



Figura 26. Painel Meus Arquivos Arduino.

Os arquivos estão indexados por ordem alfabética e para abri-los basta clicar no hiperlink desejado, conforme os padrões de comunicabilidade da internet.

Existem também as possibilidades de fazer **upload de arquivos** com extensão “**ino**” diretamente para servidor; realizar **download de arquivos do servidor** para o computador do usuário ou, simplesmente, fazer **exclusão de arquivos** desnecessários.

Abaixo da transmissão dos Experimentos, está a **Barra de Ferramentas** (Figura 27), que exibe os principais comandos para manipulação dos arquivos, incluindo uma pasta de exemplos de programas Arduino e o botão **Executar**, com o qual o usuário poderá compilar seus programas para executar seus experimentos.

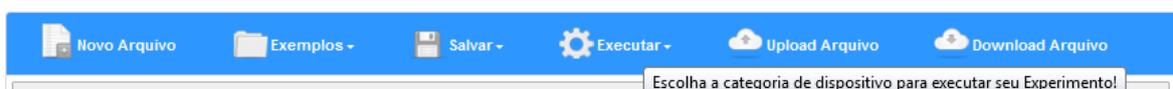


Figura 27. Barra de Ferramentas.

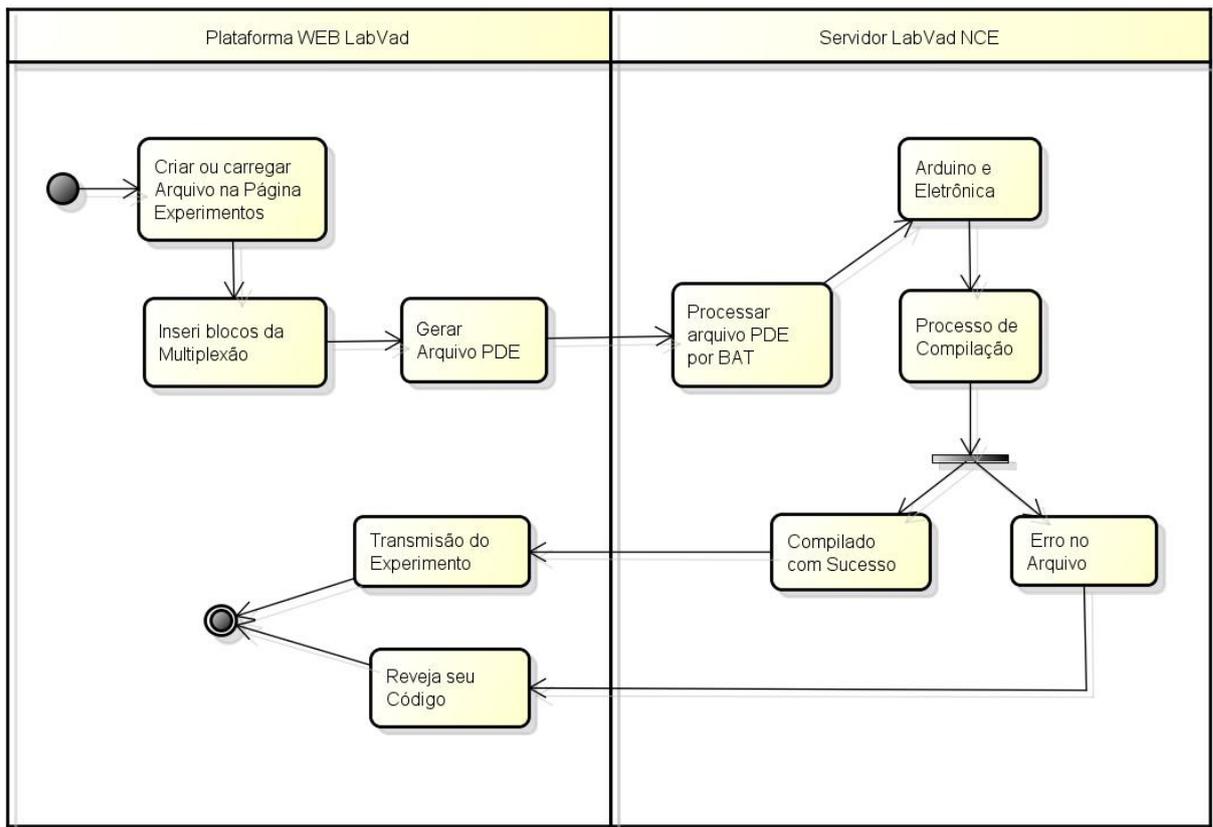
O último item do módulo de Experimentos é a área de codificação (local onde os programas são escritos), muito similar a IDE nativa do Arduino, contendo destaque na sintaxe de comandos e controles interativos para as duas principais funções da linguagem Wiring: **void setup ( )** e **void loop ( )**. A área de código ainda oferece a numeração de linhas, recurso que a IDE oficial do Arduino não oferece ao usuário, como podemos ver na Figura 28.

```
1- /* Este exemplo de código é de domínio publico
2  O nosso LED 13 é o último LED do canto esquerdo da tela do LabVad.
3  Vamos declarar este LED */
4  int led = 13;
5
6  // No função setup escrevemos parte do código que será executado uma vez:
7- void setup() {
8      // Inicializando LED como saída.
9      pinMode(led, OUTPUT);
10 }
11
12 // O loop rodará parte do código até que o mesmo seja interrompido ou zerado.
13- void loop() {
14     digitalWrite(led, HIGH); //Acende o LED
15     delay(1000); // Espera um segundo. Para esperar meio segundo o valor atribuido seria 500
16     digitalWrite(led, LOW); // Apaga o LED
17     delay(1000); // Espera um segundo
18 }
```

Figura 28. Área de codificação.

### 6.3 O processo de compilação

O processo de compilação é muito simples, claro e interativo, de forma a exibir se o programa foi ou não compilado com sucesso. Este processo é iniciado pelo usuário na plataforma WEB, passa pelo Servidor LabVad NCE e retorna ao usuário através da plataforma WEB, como podemos visualizar no diagrama de atividades abaixo:



**Figura 29. Processo de compilação.**

Curiosamente, o processo de compilação nasceu de algumas deficiências que o Sistema Operacional Windows apresentava para compilar arquivos Arduino (ARDUINO PLAYGROUND, 2008).

Don Cross, engenheiro elétrico e programador, escreveu três arquivos BAT na linguagem C, que compilam os arquivos diretamente no sistema operacional, utilizando a linguagem Wiring, sem a necessidade do uso de uma IDE. A seguir a temos a funcionalidade cada arquivo:

- O arquivo em lotes `abuild.bat` serve para construir e enviar o arquivo Arduino, com extensão "pde"<sup>25</sup>, que será compilado.
- O `upload.bat` é usado para carregar um arquivo Arduino depois de tê-lo compilado. O `abuild.bat` chama `upload.bat` após compilar o arquivo.

<sup>25</sup> As versões do ambiente de programação Arduino, anteriores a 1.0, usavam a extensão "pde" em seus arquivos.

- O agetpref.bat é invocado pelo abuild.bat e aupload.bat para extrair as variáveis de ambiente corretamente.

Basta fazer o download destes três arquivos e configurar as variáveis de ambientes do Windows, como no Quadro 5.

Nome da variável	Valor da Variável
ARDUINO_PATH	Onde se instalou o Arduino no computador do usuário (por exemplo: C: \ ARDUINO-0011).
ARDUINO_MCU	O nome do microcontrolador (por exemplo: atmega168).
ARDUINO_PROGRAMMER	O nome do programador que o usuário deseja usar (geralmente stk500)
ARDUINO_FCPU	A frequência de clock do microcontrolador (geralmente 16 milhões para ATmega168).
ARDUINO_COMPORT	A porta na qual a placa está conectada (por exemplo, COM1, COM2 , etc.).
ARDUINO_BURNRATE	Baud <sup>26</sup> (taxa de transmissão) em que o download estiver a ocorrer (19200 parece ser um bom ponto de partida).

**Quadro 5. Configurações das variáveis de ambiente do processo de compilação**

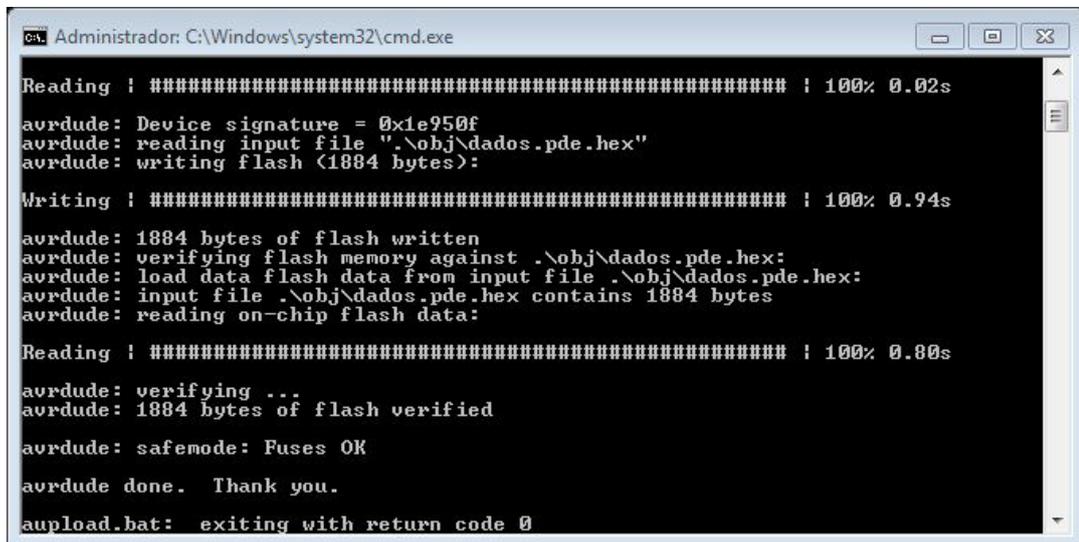
A plataforma utiliza um arquivo temporário para inserir a multiplexação (técnica que utiliza um dispositivo que codifica as informações de duas ou mais fontes de dados em um único canal), depois envia este arquivo ao Servidor LabVad NCE. Este arquivo é compilado pelo arquivo de lote abuild.bat.

```
//inicio Decodificador do MUX
int PIN_MUX_0 = 2;
int PIN_MUX_1 = 4;
//fim Decodificador do MUX
void setup() { //Habilita MUX - 01 - Conjunto de LEDs - LED6 a LED13
//Decodificador do MUX.*****
pinMode( PIN_MUX_0,OUTPUT); // Pino_2 - Decodificador do Conjunto
bit_0
pinMode(PIN_MUX_1,OUTPUT); // Pino_4 - Decodificador do Conjunto
bit_1
digitalWrite(PIN_MUX_0, HIGH); // MUX_0
digitalWrite(PIN_MUX_1, HIGH); // MUX_1
```

**Figura 30. Arquivo temporário, com os blocos dos códigos da multiplexação inseridos.**

<sup>26</sup> Um baud é uma medida de velocidade de sinalização e representa o número de mudanças na linha de transmissão (seja em frequência, amplitude, fase, etc) ou eventos por segundo.

Quando o arquivo de lote `upload.bat` retorna o valor 0, o arquivo foi compilado com sucesso (Figura 30) e quando o mesmo retorna o valor maior que 0, os erros de compilação são apresentados.



```
Administrador: C:\Windows\system32\cmd.exe
Reading | ##### | 100% 0.02s
avrdude: Device signature = 0x1e950f
avrdude: reading input file ".\obj\dados.pde.hex"
avrdude: writing flash (1884 bytes):
Writing | ##### | 100% 0.94s
avrdude: 1884 bytes of flash written
avrdude: verifying flash memory against .\obj\dados.pde.hex:
avrdude: load data flash data from input file .\obj\dados.pde.hex:
avrdude: input file .\obj\dados.pde.hex contains 1884 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 0.80s
avrdude: verifying ...
avrdude: 1884 bytes of flash verified
avrdude: safemode: Fuses OK
avrdude done. Thank you.
upload.bat: exiting with return code 0
```

Figura 31. Mensagem de compilação do Servidor LabVad NCE

## Atenção!



**Compilado com sucesso!**

**Em alguns segundos seu Experimento será executado...**

OK

Figura 32. Processo de compilação exibido no Navegador do usuário.

A inovação está no fato de utilizar as linguagens apresentadas na seção 5.3 para capturar os parâmetros dos arquivos de lotes (BAT) e, posteriormente, exibir no navegador do usuário uma mensagem continente de conceitos típicos sobre o processo de compilação da maioria dos ambientes de programação.

## 6.4 Foco no Usuário

Além de contar com uma IDE repleta de recursos, o usuário tem acesso às suas configurações de perfil para editar nome, e-mail e senha. Em Configurações, o usuário tem autonomia para visualizar e baixar seu registro de atividades no sistema (LOG).

**Edição de Usuário**

**Nome**  
Paulo Souza

**Email**  
prasouza@ufrj.br

**Senha**

**Nova Senha**  
Informe sua nova senha

**Ativo**

**Gravar** **Voltar**

**Visualização de Log do usuário: Paulo Souza**

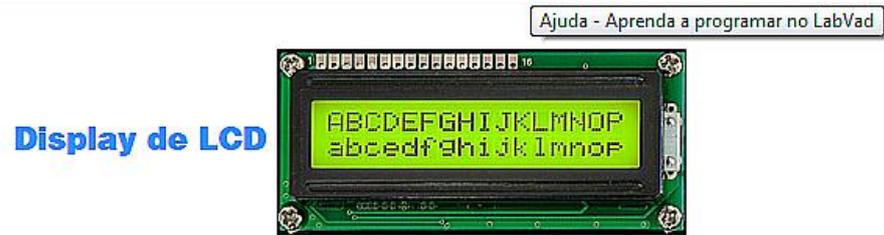
Ação	Data
Acesso ao sistema	02/26/2015 - 10:50
Controle de usuário	02/26/2015 - 10:50
Criando novo experimento	02/26/2015 - 10:51
Controle de usuário	02/26/2015 - 10:51
Controle de usuário	02/26/2015 - 11:04
Acesso ao sistema	04/30/2015 - 21:56
Alterações configurações	04/30/2015 - 21:56
Acesso ao sistema	04/30/2015 - 21:56
Alterações configurações	04/30/2015 - 21:56
Acesso ao sistema	04/30/2015 - 22:00
Alterações configurações	04/30/2015 - 22:00

**Download log** **Fechar**

Figura 33. Configurações do usuário e LOG.

As mensagens de retorno sempre informam os usuários da consequência de suas ações; com essa ação, elas previnem erros e apresentam alternativas com mensagens simples, garantindo a flexibilidade e eficiência de uso.

O módulo de ajuda não é alicerçado em funções técnicas do sistema, mas sim na programação dos dispositivos dos eletrônicos. A ajuda é padronizada e apresentada da seguinte forma: definição do dispositivo (**o que é**); funcionamento (**o que faz**); como o dispositivo está conectado fisicamente ao LabVad (**como funciona no LabVad**) e um **exemplo** de aplicação no módulo de Experimentos. O usuário ainda pode baixar um documento que contém um guia rápido sobre o funcionamento e programação no LabVad.



### O que é? O que faz?

Um display de cristal líquido, acrônimo de LCD (em inglês *liquid crystal display*). Mostra dados lidos pelo LabVad em letras e números, de uma forma inteligível.

### Como funciona no LabVad?

Os display está conectado ao LabVad nos pinos 6, 7, 8, 9, 10, 11, 12. Veja na imagem abaixo:

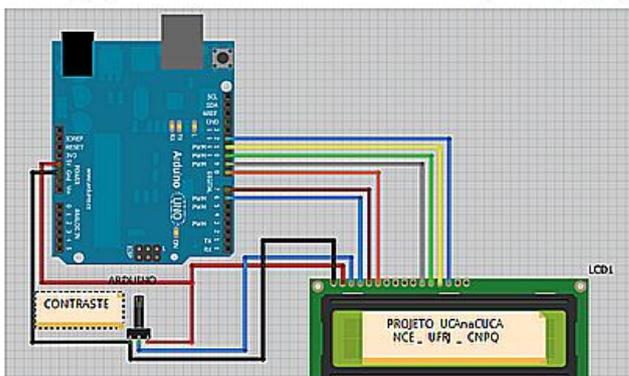


Figura 34. Ajuda no LabVad.



## 7. A Avaliação do LabVad

Neste capítulo, são apresentados os conceitos de comunicabilidade/usabilidade (seção 7.1); os métodos de avaliação de Interfaces Humano-Computador (IHC) que foram aplicados ao LabVad (seção 7.2); o redesenho da plataforma WEB, oriundo de problemas apontados nas avaliações de processo (desenvolvimento) e produto (final) aplicadas à plataforma do LabVad (seções 7.3 e 7.4); as conclusões, análises e gráficos acerca das avaliações propostas (seção 7.5); a avaliação docente aplicada a um grupo de professores (seção 7.6) e, por último, as conclusões e contribuições desta avaliação (seção 7.7).



## 7.1. Comunicabilidade e Usabilidade

No contexto do presente trabalho, a interação é entendida como o processo de comunicação entre humanos e sistemas interativos (PREECE et al., 1994) e interface pode ser definida como algo discreto e tangível, que se pode desenhar, mapear, projetar e implementar, "encaixando-o" posteriormente a um conjunto já definido de funcionalidades (ROCHA e BARANAUSKAS, 2003).

Muito embora, ainda não exista uma definição estabelecida para IHC acreditamos que a seguinte definição incorpora o espírito da área no momento: IHC é a disciplina preocupada com o design, avaliação e implementação de sistemas computacionais interativos para uso humano e com o estudo dos principais fenômenos ao redor deles. (ROCHA e BARANAUSKAS, 2003, p.14).

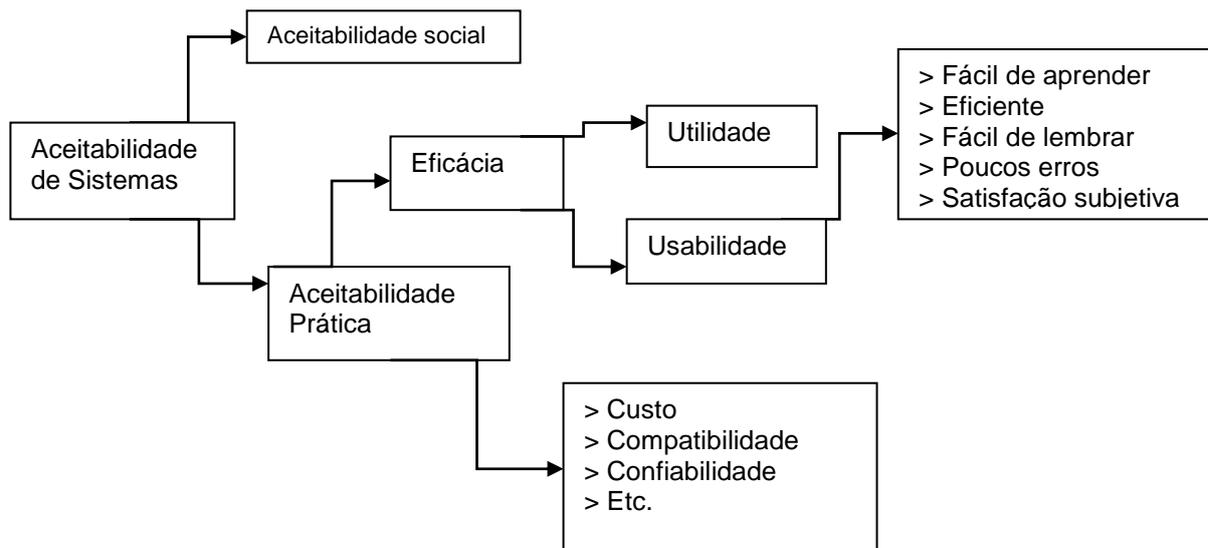
Comunicabilidade é a qualidade do ato comunicativo otimizado, no qual a mensagem é transferida de forma integral, correta, rápida e economicamente. Para Salgado et al. (2006), a comunicabilidade se refere à capacidade de os usuários entenderem a lógica do design do sistema tal como construído pelos desenvolvedores. Caso o usuário entenda as decisões que o desenvolvedor tomou na construção do sistema, conseqüentemente, entenderá a utilidade do sistema, sua funcionalidade, as vantagens ao utilizá-lo e seus princípios gerais de interação.

A Norma ISO 9241 (2014) conceitua a usabilidade como a capacidade de interatividade que uma interface oferece ao seu usuário para a realização de tarefas de maneira eficaz e agradável. O grande desafio é construir interfaces sem sobrecarga cognitiva, capazes de minorar o esforço mental para realizar interações e tarefas.

Outro conceito importante apontado por Salgado et al. (2006), sobre usabilidade, é a facilidade de entender, memorizar e utilizar um software, acompanhada de satisfação e produtividade do usuário. Fica ostensivo que a comunicabilidade contribui para a usabilidade, embora não possa garanti-la se não fizer parte do design, como, por exemplo, tornar o usuário mais produtivo (SALGADO et al., 2006). Deve-se aproveitar o conhecimento que o usuário possui a respeito do mundo que o cerca. O uso de analogias com artefatos familiares a ele pode contribuir para atingir uma comunicabilidade alta. "Uma interface com boa

comunicabilidade permite que o usuário formule um modelo mental compatível com o do desenvolvedor” (PRATES et al., p. 5, 2003).

Nielsen e Mack (1994) exploram o design e propõem princípios que levem a uma majoração da usabilidade de acordo com os princípios definidos na aceitabilidade de um sistema (Figura 35).



**Figura 35. Atributos de aceitabilidade de sistemas (NIELSEN E MACK, 1994).**

De acordo com Preece et al. (2002) e Nielsen e Mack (1994), existem fatores típicos que envolvem o conceito de usabilidade, fatores como: facilidade de aprendizado, facilidade de uso, eficiência de uso e produtividade, satisfação do usuário, flexibilidade, utilidade e segurança no uso.

Para Nielsen e Mack (1994), a facilidade de aprendizado é o atributo mais eminente da usabilidade. O usuário precisa interagir rapidamente com o sistema em um nível razoável de competência e desempenho. Na análise da facilidade de aprendizado, é necessária a consciência de que o usuário, com o tempo de uso do sistema, transita por níveis que vão do intermediário ao avançado até atingir um grau de proficiência satisfatório na execução das tarefas.

A facilidade de uso está relacionada com o esforço cognitivo para interagir com o sistema. A interface precisa ser facilmente reconhecida/relembrada, de forma que o usuário, ao retornar à sua utilização, não encontre necessidade de reaprendê-la.

A facilidade de utilização também está relacionada aos erros. O erro é definido como uma ação que leva a um resultado inesperado. O usuário não deve cometer muitos erros durante o uso do sistema e, caso isso ocorra, deve ser de fácil recuperação e sem a perda do trabalho, ou seja, do esforço despendido para executar determinada tarefa.

A eficiência na utilização deve refletir o escopo do sistema. Uma vez aprendido, o usuário deve executar de forma rápida e eficaz uma tarefa, obtendo elevados níveis de produtividade. A produtividade deve ser avaliada pelo número de passos que o usuário precisou para concluir seu trabalho, ou seja, o percurso cognitivo.

A satisfação subjetiva ou satisfação do usuário é determinante para o sucesso do sistema. Nielsen e Mack (1994) ressaltam que satisfação subjetiva vai além da usabilidade, porque está ligada à satisfação, à emoção e aos sentimentos que as pessoas têm em relação a computadores, que afetam, e em muito, a interação com sistemas. Nielsen e Mack (1994) ainda afirmam que estas interações podem ser negativas, como tédio e frustração, ou positivas, como prazer e diversão.

A flexibilidade se refere à forma peculiar que cada usuário lida com as interfaces dos sistemas. Pessoas percorrem caminhos distintos para executar a mesma tarefa. De acordo com Preece et al. (2002), os usuários utilizam estratégias de soluções para problemas completamente distintos e, por isso, a interface deve ser preparada para acomodar esta diversidade condições, de caminhos, de possibilidades.

Para Prates et al. (2003), a utilidade de um sistema está relacionada ao conjunto de funcionalidades que ele oferece para que os usuários realizem suas tarefas. Em outras palavras, a pluralidade de opções para se realizar uma ação, diferentes caminhos oferecidos para executar uma tarefa ou rotina.

## **7.2. Métodos de avaliação**

Pressman (2006) afirma que nas etapas de testes e implantação, o desenvolvedor busca compreender as necessidades do usuário e o que ele deseja

que o sistema realize. Isto é feito, sobretudo, com entrevistas, nas quais o desenvolvedor tenta determinar as reais necessidades do sistema de informação.

A Avaliação Heurística é um método baseado na verificação de uma pequena lista de regras (heurísticas) ou na própria experiência dos avaliadores que visam de forma econômica, fácil e rápida, descobrir grandes problemas potenciais da interface (NIELSEN, 1994). Nielsen (1994) propôs um conjunto de dez heurísticas; nele, a usabilidade dos elementos de uma interface deve ser analisada para verificar sua conformidade com cada uma das regras abaixo:

1. **Visibilidade do estado do sistema:** manter o usuário informado sobre o que está acontecendo, através de feedback imediato. Ex.: Caixas de Diálogos.
2. **Correspondência entre o sistema e o mundo real:** utilizar conceitos, vocabulários e processos familiares aos usuários.
3. **Controle e liberdade do usuário:** fornecer alternativas e “saídas e emergência”. O usuário deve sentir-se seguro ao navegar pelas telas de um site mediante possibilidades patentes de avanço ou retrocesso.
4. **Consistência e padronização:** usuários não precisam adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. Seguir convenções de plataforma computacional.
5. **Prevenção de erros:** melhor que uma boa mensagem de erro é um sistema capaz de prevenir o erro.
6. **Reconhecimento ao invés de relembração:** o usuário não deve ter que lembrar informação de uma para outra parte do diálogo. Instruções para uso do sistema devem estar visíveis e facilmente recuperáveis quando necessário.
7. **Flexibilidade e eficiência de uso:** prover aceleradores com o objetivo de aumentar a velocidade da interação. Permitir a usuários experientes "cortar caminho" em ações frequentes, ou seja, utilizar atalhos.

8. **Estética e design minimalista:** diálogos não devem conter informação irrelevante ou raramente necessária. Qualquer unidade de informação extra no diálogo irá competir com unidades relevantes de informação.
9. **Ajudar os usuários a reconhecer, diagnosticar e corrigir erros:** mensagens de erro devem ser expressas em linguagem clara (sem códigos) indicando precisamente o problema e, construtivamente, sugerindo uma solução.
10. **Ajuda e documentação:** o ideal é projetar um sistema que possa ser usado sem documentação, entretanto, é necessário prover ajuda e documentação. Essas informações devem ser fáceis de encontrar, focalizadas na tarefa do usuário.

Quando uma heurística é violada, os problemas devem ser documentados e analisados (BERTINI et al. 2006). Os problemas de interface são analisados por valores, de acordo com uma escala de gravidade. Esta escala será nossa matriz de referência em um questionário semiestruturado. A avaliação heurística deve ser realizada com 3 a 5 especialistas em IHC.

Escala	Tipo de problema	Descrição do problema
0	Sem problema	O avaliador reconhece que não existem problemas.
1	Problema Cosmético	Não precisa ser consertado a menos que haja tempo extra no cronograma.
2	Problema Pequeno	O conserto é desejável, mas deve receber baixa prioridade.
3	Problema Grande	Alta prioridade de correção
4	Catastrófico	Imperativo consertar.

**Quadro 6. Escala de gravidade da avaliação heurística**

A matriz de referência na realização das avaliações da plataforma WEB do LabVad será a combinação da Avaliação Heurística com outro método conceituado como Percurso Cognitivo (WHARTON et al., 1994). O Percurso Cognitivo avalia uma

interface pela sua facilidade de aprendizado, em particular, pela exploração do sistema através dos usuários, no papel de avaliadores.

Elaboramos um cenário com um conjunto de tarefas, que exploram as principais funções do sistema, para avaliarmos as hipóteses sobre os usuários e sobre o conhecimento que eles têm a respeito da tarefa e da interface (Apêndice A).

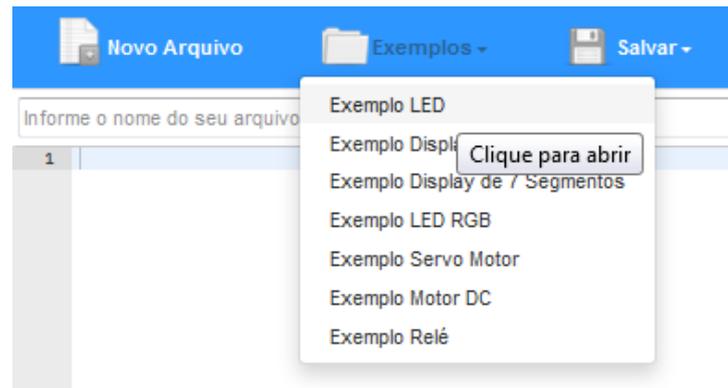
### 7.3 Avaliações de processo e produto aplicadas ao LabVad

Preece et al. (1994) defendem que a avaliação de uma interface deve ser feita em diferentes etapas do desenvolvimento do software. As avaliações de processo são aquelas que são feitas durante a construção do design, portanto, em momentos anteriores ao término da fase final de implementação da plataforma do LabVad. Aproveitamos avaliações de processo para ter embasamento no redesign da plataforma.

Preece et al. (1994) ainda afirmam que o avaliador pode prever tais problemas através da interpretação de dados obtidos nas avaliações. O intuito das avaliações de processo é a identificação de problemas na interface com vistas à reparação desses problemas antes que o sistema seja liberado para uso e, assim, oferecer ao usuário um sistema de melhor qualidade.

As avaliações de processo foram feitas por um especialista em IHC, um membro do GINAPE e um professor. Estas avaliações foram fundamentais para termos uma versão “estável” do LabVad. Por se tratar de um portal foram redefinidas ou desativadas várias funções nativas dos navegadores WEB que prejudicavam a usabilidade. Por termos uma área de codificação de programas e utilizarmos excessivamente o teclado, a função de navegação da tecla voltar ou *Backspace* foi desativada, o mesmo aconteceu com a tecla *Enter*, entre outras.

As avaliações de processo também trouxeram melhorias que não estavam definidas no escopo do projeto, baseado no levantamento de requisitos apresentado no capítulo 4. Dentre as melhorias, destaca-se a inclusão de alguns exemplos de programas em Arduino; inclusão de legendas e a padronização dos ícones, como vemos na Figura 36.



**Figura 36. Barra de Ferramentas, com exemplos de programas Arduino.**

O LabVad também passou por avaliações de produto, ou seja, avaliações feitas após a interface ficar pronta. Na próxima seção deste capítulo, apresentaremos a coleta de dados de opinião e as observações dos usuários especialistas, que deram origem a uma análise de dados baseada na severidade média dos números de problemas apresentados. Além disso, através da entrevista e questionário pode-se colher a opinião pessoal do usuário sobre sua experiência, sua satisfação com o sistema e sugestões (PRATES et al., 2003).

#### **7.4 O Processo de avaliação dos Especialistas em IHC**

Para se obter o retorno desejado, seguimos a recomendação de escolher 5 avaliadores, com experiência sólida em IHC, para efetuar a avaliação heurística (SALGADO et al., 2006; NIELSEN, 1994). Os avaliadores detêm experiência prática em usabilidade e comunicabilidade. Todos são analistas e projetistas de sistemas para Internet, dentre eles, dois participantes também são desenvolvedores de sistemas inclusivos.

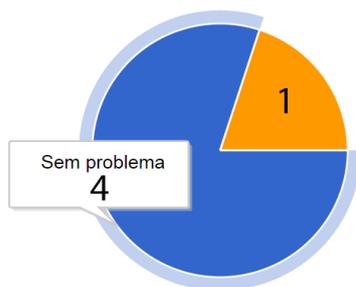
Os avaliadores permitiram, previamente, que todo o processo avaliação fosse registrado em gravações de vídeo. Fez parte do planejamento de coleta de dados a inclusão do método de protocolo verbal “Pensar Alto”. Barbosa e Prates (2003) defendem que os métodos de protocolos verbais ajudam na compreensão dos processos mentais dos participantes da avaliação porque os mesmos expressam comportamentos, sentimentos e atitudes na execução de tarefas, o que não ocorre, com riqueza de detalhes, nos métodos de coleta de dados que utilizam somente registro de documentos escritos ou digitais.

As avaliações duraram, em média, de 40 minutos a 1 hora. Primeiramente foi explicado o objetivo principal do LabVad e, tecnicamente, como ocorre o acesso remoto ao Servidor LabVad NCE.

Os avaliadores executaram um conjunto de tarefas exploratórias (Apêndice A), que englobam as principais funcionalidades do sistema desde o cadastro na plataforma WEB do LabVad até o uso da seção de Ajuda. Após a conclusão destas tarefas, os mesmos responderam um questionário.

A seguir apresentaremos tal questionário, seus gráficos, a escala de gravidade, caso haja a violação de uma heurística, e as explicações sobre os problemas apontados pelos avaliadores.

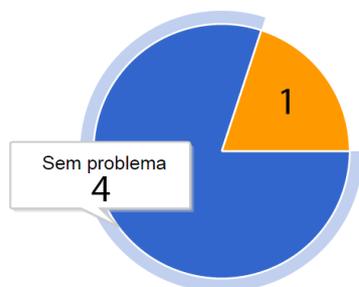
*A) Existem problemas na página de acesso ao LabVad?*



Sem problema	4
Problema cosmético	0
Problema pequeno	1
Problema grande	0
Problema catastrófico	0

Análise: um dos avaliadores relatou que o e-mail de confirmação de cadastro pode ir para o “lixo eletrônico” ou spam, violando a heurística sobre a prevenção de situações de erros.

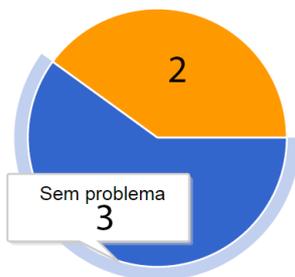
*B) Existe problema no processo de recuperar senha?*



Sem problema	4
Problema cosmético	0
Problema pequeno	0
Problema grande	1
Problema catastrófico	0

Análise: Novamente, um dos avaliadores relatou que o e-mail de redefinição de senha pode ir para o “lixo eletrônico” ou *spam*, violando a heurística sobre a prevenção de situações de erros.

*C) Ao acessar o LabVad ficam claros os objetivos da plataforma?*



Sem problema	<b>3</b>
Problema cosmético	<b>0</b>
Problema pequeno	<b>2</b>
Problema grande	<b>0</b>
Problema catastrófico	<b>0</b>

Análise: Dois avaliadores apontaram que no primeiro contato com a interface do sistema, não fica claro que o LabVad está diretamente relacionado ao Projeto Arduino, violando a heurística sobre a ajuda e documentação.

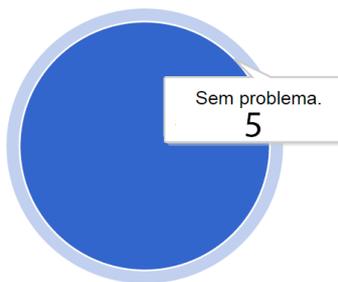
*D) Os links do LabVad são intuitivos e instrutivos?*



Sem problema	<b>5</b>
Problema cosmético	<b>0</b>
Problema pequeno	<b>0</b>
Problema grande	<b>0</b>
Problema catastrófico	<b>0</b>

Análise: Não ocorreu violação na heurística sobre consistência e padronização dos ícones da interface do sistema.

*E) Você detectou algum problema visual no tamanho ou redimensionamento da tela?*



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: Não ocorreu violação na heurística sobre a flexibilidade e eficiência de uso.

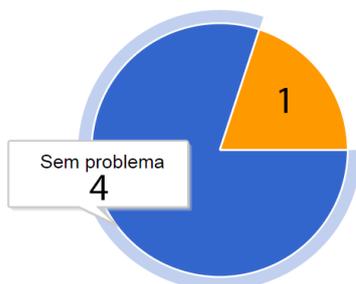
F) *Você detectou algum problema de navegação nas telas do LabVad?*



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: Não ocorreu violação na heurística sobre a flexibilidade e eficiência de uso.

G) *O ambiente deixa claro que para executar os experimentos é necessário efetuar agendamento prévio?*



Sem problema	4
Problema cosmético	0
Problema pequeno	1
Problema grande	0
Problema catastrófico	0

Análise: um dos avaliadores relatou que esta informação deve ficar mais clara ao acessar a interface, violando a heurística sobre consistência e padrões.

H) *Efetuar ou excluir agendamentos é intuitivo?*

Análise: Não ocorreu violação na heurística sobre o design estético e minimalista do sistema.

- I) *Avalie se os ícones da barra de ferramentas possuem um padrão em seus nomes e suas legendas.*



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: Não ocorreu violação na consistência e padronização dos ícones.

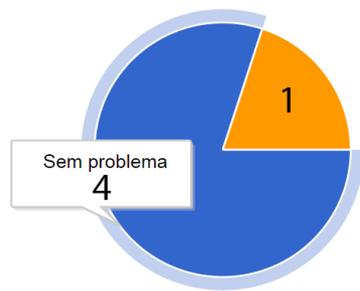
- J) *Os ícones da barra de ferramentas apontam para reconhecimento imediato, ou somente com o uso do software o usuário irá memorizá-los?*



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: Não ocorreu violação na heurística sobre o reconhecimento ao invés de memorização.

- K) *Durante a manipulação e edição de arquivos, existe uma correlação clara entre as funções da Barra de Ferramentas e o Painel Meus Arquivos Arduino?*



Sem problema	4
Problema cosmético	1
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: Um dos avaliadores disse que não percebeu que o painel Meus Arquivos Arduino é um repositório dos arquivos que são manipulados pelo usuário. A heurística sobre a prevenção de erros foi violada.

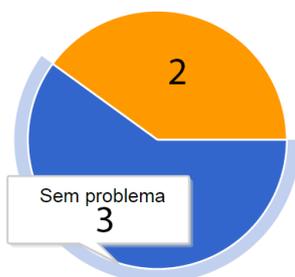
*L) Na execução dos experimentos, avalie se o botão “Executar” facilita o acesso aos dispositivos (LED, Display de Caracteres, 7 Segmentos...).*



Sem problema	3
Problema cosmético	1
Problema pequeno	0
Problema grande	0
Problema catastrófico	1

Análise: Dois avaliadores relataram a heurística sobre o design estético e minimalista foi violada.

*M) Você se sente seguro em relação ao seu programa na área de código?  
Você acha que pode perder suas informações?*



Sem problema	3
Problema cosmético	0
Problema pequeno	2
Problema grande	0
Problema catastrófico	0

Análise: Dois avaliadores relataram que uma função de auto salvamento deve ser incorporada a área de código do sistema. A heurística sobre controle e liberdade do usuário foi violada.

N) Avalie o processo de gravação dos arquivos (menu salvar).



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: A heurística sobre correspondência do sistema com o mundo real não foi violada.

O) As mensagens de respostas das caixas de diálogos sobre salvar ou descartar estão numa linguagem adequada para os usuários (sem termos técnicos)?



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: As mensagens de retorno não violam a heurística sobre ajuda aos usuários para diagnósticos, ou seja, foram escritas numa linguagem simples, adequada e familiar ao usuário.

P) Avalie os processos de upload (levar um arquivo do seu computador para o LabVad) e download (baixar um arquivo do LabVad para seu computador).



Sem problema	3
Problema cosmético	1
Problema pequeno	0
Problema grande	0
Problema catastrófico	1

Análise: Os avaliadores relataram que nas funções de upload e download, o usuário deve receber mais orientações sobre o processo. A heurística sobre correspondência do sistema e o mundo real foi violada.

Q) *A seção Ajuda foi útil na realização da tarefa em que a mesma foi solicitada?*



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: A ajuda e documentação do sistema estão com fácil acesso e focadas no usuário.

R) *Na seção Ajuda, fica claro como os dispositivos estão conectados na parte física do LabVad?*

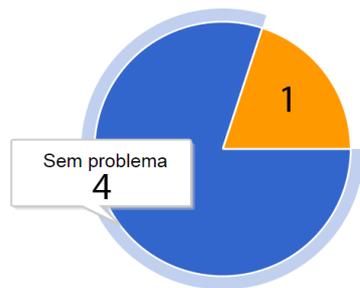


Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: A documentação do sistema está focada no usuário, ensinando-o como programar os dispositivos eletromecânicos e eletroeletrônicos do LabVad.

S) Na seção Configurações, é fácil acessar seu próprio LOG, ou seja, seu registro de atividades?

Análise: Um avaliador sugeriu que as informações do registro de atividades devem ser apresentadas de forma decrescente. A heurística sobre consistência e padrões foi violada.



Sem problema	4
Problema cosmético	1
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

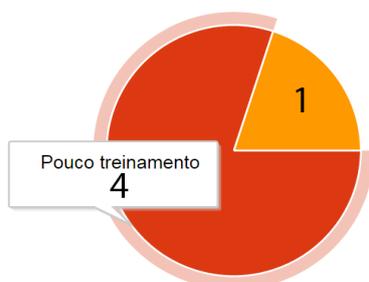
T) Avalie se existem problemas na edição de dados como o nome e senha.



Sem problema	5
Problema cosmético	0
Problema pequeno	0
Problema grande	0
Problema catastrófico	0

Análise: A heurística sobre visibilidade e estado do sistema e correspondência do mesmo com o mundo real.

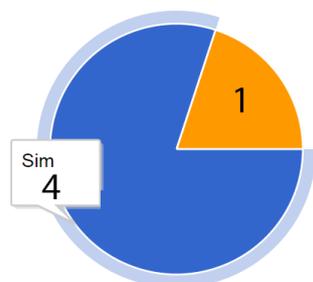
U) Quanto tempo de treinamento seria necessário para usar o LabVad?



Nenhum treinamento.	0
Pouco treinamento.	4
Tempo razoável de treinamento.	1
Muito treinamento.	0
Treinamento intensivo.	0

Análise: A maioria dos avaliadores relatou que a interface exige pouco treinamento, mas todos enfatizaram a necessidade de ter conhecimentos prévios sobre robótica.

V) *Do ponto de vista de interface, você acredita que o LabVad está pronto para ser utilizado por usuários não-especialistas em computação?*



Sim. 4

Não. 0

Não sei. 1

Análise: Apenas um dos avaliadores afirmou que o LabVad ainda não está pronto para a utilização para o aprendizado de Robótica. Sua justificativa e/ou sugestão foi a inclusão de mais funções para que o ambiente também fosse utilizado para programadores na linguagem de programação C.

#### 7.4.1 A visão dos avaliadores

No quadro e no gráfico desta subseção, temos a visão de cada avaliador durante o processo da avaliação heurística.

Questão	Avaliador 1	Avaliador 2	Avaliador 3	Avaliador 4	Avaliador 5
A	0	2	0	0	0
B	0	3	0	0	0
C	2	0	0	2	0
D	0	0	0	0	0
E	0	0	0	0	0
F	0	0	0	0	0
G	0	0	2	0	0
H	0	0	0	0	0
I	0	0	0	0	0
J	0	0	0	0	0

K	0	0	0	1	0
L	0	4	0	0	1
M	2	0	2	0	0
N	0	0	0	0	0
O	0	0	0	0	0
P	0	0	0	3	1
Q	0	0	0	0	0
R	0	0	0	0	0
S	0	0	1	0	0
T	0	0	0	0	0

Quadro 7. Inspeção dos avaliadores.

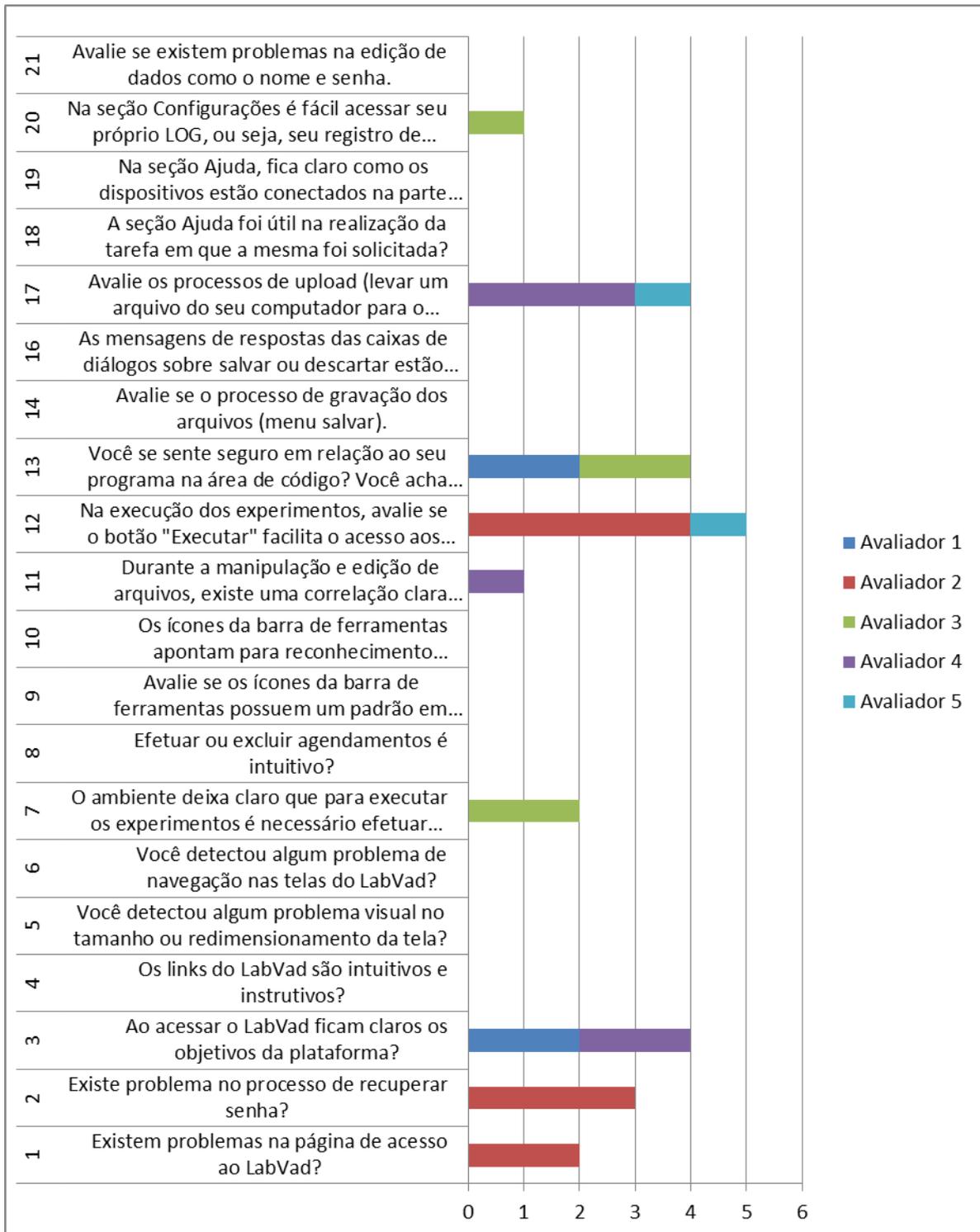


Figura 37. Gráfico sobre a Inspeção dos avaliadores.

De acordo com o gráfico, podemos notar que a inspeção dos avaliadores apontou erros em comum, com destaque para a questão L ou 12, em que a mesma

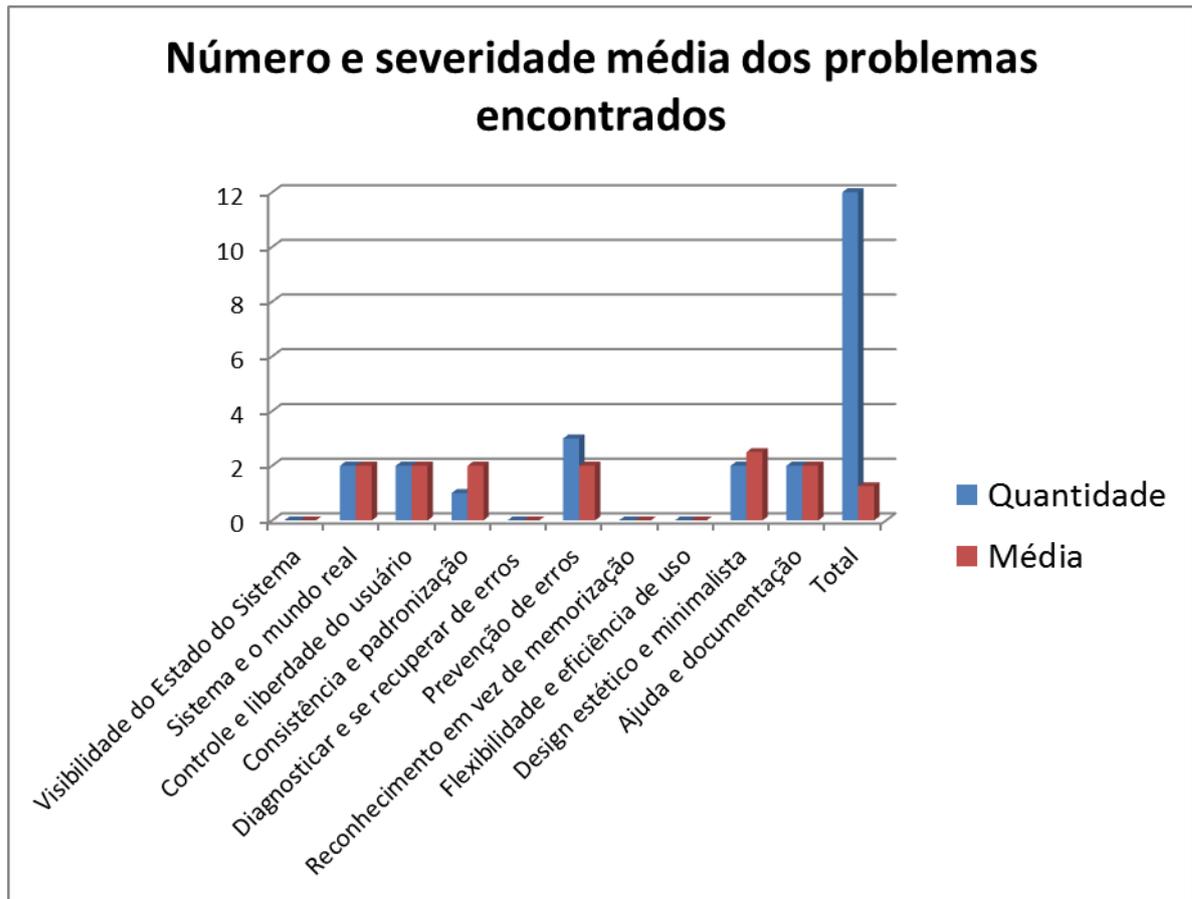
atingiu o grau máximo de severidade, portanto comentaremos este fato na conclusão dos dados aferidos.

### 7.5 Conclusão sobre os dados aferidos na avaliação IHC

Através dos resultados de um questionário, apresentaremos um quadro sobre as heurísticas que foram violadas, o número de problemas encontrados e uma relação do grau de severidade com problemas encontrados na interface.

Heurística	Número de problemas encontrados	Severidade média
Visibilidade do Estado do Sistema	0	0
Correspondência entre o sistema e o mundo real	2	2
Controle e liberdade do usuário	2	2
Consistência e padronização	1	2
Ajudar os usuários a reconhecer, diagnosticar e corrigir erros	0	0
Prevenção de erros	3	2
Reconhecimento em vez de memorização	0	0
Flexibilidade e eficiência de uso	0	0
Design estético e minimalista	2	2,5
Ajuda e documentação	2	2
Total	12	1,25

**Quadro 8. Análise de gravidade das heurísticas violadas.**



**Figura 38.** Gráfico com os erros e a severidade média da gravidade das heurísticas violadas.

Os problemas relacionados à previsão de situações de erro, em relação cadastro e recuperação de senha, foram resolvidos com mensagens de retorno mais explicativas para o usuário como, por exemplo, “*Cadastro efetuado com sucesso; Verifique sua senha no e-mail cadastrado!; Se não receber o e-mail em sua caixa de entrada, por favor, não deixe de verificar em seu Lixo Eletrônico ou Spam*”.

Em relação às informações contidas na página de recepção do LabVad (introdução), foram inseridas informações sobre a influência do projeto Arduino e enfatizado que existem lições sobre funcionamento dos dispositivos na sessão Ajuda.

Como vimos na seção 5.4, a área de código foi construída de acordo com métodos inovadores para o armazenamento local (API Storage) ao invés do auto salvamento ou salvamento automático. Esta decisão foi tomada para não obrigar o usuário a salvar sempre seu conteúdo e evitar que porções de informações extras competissem com informações funcionais e relevantes na execução de uma tarefa.

O botão de Executar prejudica o design estético e minimalista no processo de compilação execução dos experimentos. O usuário tem que escolher a opção Executar e depois um dos subconjuntos que refletem a categoria do dispositivo: a) Executar para LED; b) Executar para Display LCD; c) Executar para Display de 7 Segmentos; d) Executar para Led RGB; e) Executar para Servo Motor; f) Executar para Motor DC e g) Executar para Relé.



Figura 39. Subconjuntos do menu Executar.

No capítulo 8, apresentaremos, na seção de trabalhos futuros, uma solução que implica a substituição de um dos componentes de hardware, assim todos os subconjuntos do menu Executar serão suprimidos.

No ambiente de desenvolvimento nativo do Arduino, a função Upload serve para enviar e compilar o código nas placas Arduino, portanto os avaliadores apontaram a falta de correspondência entre o sistema e tal ambiente. Como já foi explicado, a plataforma do LabVad, agrega mais funções que a IDE nativa do Arduino, além não deve exigir do usuário um computador com alto poder de processamento. No entanto, estas funções se tornam indispensáveis e essenciais para enviar (upload) e baixar (download) os arquivos Arduino do usuário.

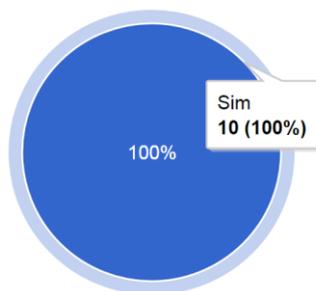
A alteração na exibição do conteúdo dos registros dos arquivos dos usuários foi indexada de forma decrescente. O nome do arquivo do download do LOG também foi alterado com o intuito de ser mais significativo. Atualmente, o nome do arquivo de LOG contém o usuário e data em que o mesmo foi baixado.

## 7.6 Avaliação docente

A plataforma WEB do LabVad também foi avaliada por um grupo de 10 docentes. O pré-requisito foi o conhecimento em linguagem de programação e robótica destes professores. Foram selecionados 3 professores de física e 7 professores da área de informática.

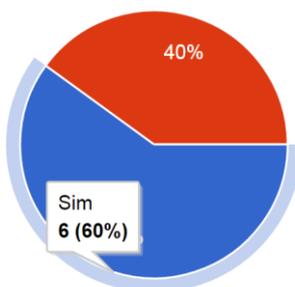
Os professores receberam, por e-mail, a mesma lista de tarefas dos avaliadores IHC, o guia rápido de ajuda do LabVad, um questionário e dois arquivos com aplicações didáticas sobre estruturas de repetição e condição na linguagem Wiring. Abaixo está o questionário semiestruturado que foi respondido pelos docentes.

*Você consegue imaginar uma situação de sala de aula presencial onde o LabVad traria contribuições?*



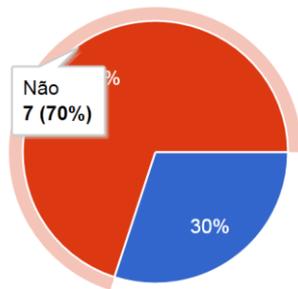
Sim	10	100%
Não	0	0%

*Você explicaria a estrutura de repetição "for" e "if" com o LabVad?*



Sim	6	60%
Não	4	40%

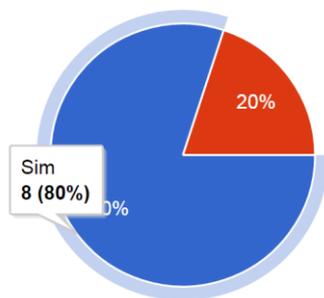
*Um estudante utilizando o LabVad não manipula diretamente os dispositivos da placa robótica. Você considera isso prejudicial à formação do aluno(a)?*



Sim 7 70%

Não 3 30%

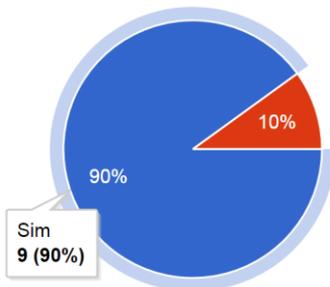
Na seção ajuda (help) do LabVad temos o desenho de como os dispositivos estão conectados. Você acredita que o material lá descrito é suficiente para o aluno entender como funcionam os mesmos dispositivos numa placa Arduino?



Sim 8 80%

Não 2 20%

Você se imagina ministrando um treinamento, na modalidade EAD, utilizando o LabVad?



Sim 9 90%

Não 1 10%

Que outras funcionalidades faltam ao LabVad para ele ficar pronto para ser utilizado em sala de aula?



A ferramenta está pronta 5 50%

Acrescentar mais funções 5 50%

A ferramenta está pronta para ser utilizada em sala de aula  
5 (50%)

## 7.7 Conclusões sobre a avaliação docente

Todos os professores afirmaram que o LabVad geraria contribuições no ambiente escolar, tanto no ensino de programação quanto em eletrônica. Eles ressaltaram a vantagem do aluno poder testar e visualizar o funcionamento de um experimento (programa) em tempo real, em relação à simples análise de uma mensagem de retorno para verificar se o programa foi compilado com sucesso ou não, fato que acontece nas linguagens tradicionais utilizadas na introdução à programação.

Segundo os professores, numa simples programação de um LED intermitente, o aluno também começa a passar por um processo de construção do conhecimento, porque existem outras variáveis a serem trabalhadas, como por exemplo, manipular o tempo que o LED ficará aceso e a possibilidade de simular um semáforo, visto que o aluno dispõe de uma sequência de LED.

Também foi apontado que o LabVad pode resolver o problema da escassez dos Kits didáticos de Robótica. Neste sentido, o LabVad poderá ser utilizado simultaneamente com kits didáticos do projeto Arduino, com o objetivo de replicar programas que foram executados remotamente e as conexões apresentadas na seção Ajuda e, assim, construir protótipos mais complexos.

O aluno que aprende a controlar e programar um motor DC, possivelmente não terá dificuldades num outro momento em que estiver construindo um robô. Outro ponto positivo relatado é a oportunidade de o aluno testar todas as possibilidades do display de 7 segmentos ou LCD, para concluir se este dispositivo é adequado para o uso em um projeto futuro.

Quatro professores responderam que optariam por explicar as estruturas de repetição (for) e condicional (if) de forma clássica, ou seja, através de fluxogramas, lógica matemática ou silogismo, para somente depois utilizar o LabVad como ferramenta didática, mas a maioria apontou que trabalhar com estruturas de repetição e condicionais, no LabVad, seria lúdico e aceleraria o conhecimento dos alunos sobre tais assuntos.

A questão 4 gerou as algumas opiniões opostas e, principalmente, levantou a questão que é de fundamental importância a atuação do professor como facilitador no processo de aprendizagem.

Três professores afirmaram que somente o ensino remoto não é suficiente para a formação de um profissional, levando em consideração que nos cursos profissionalizantes e superiores o aprendiz deve apresentar um projeto final concreto, composto por componentes físicos. Os outros afirmaram que o ensino remoto, da forma que é apresentada no LabVad, pode fazer com que o aluno torne-se autodidata, logo se interesse em adquirir dispositivos robóticos e, conseqüentemente, apresente trabalhos de conclusão de curso inovadores.

Sobre o seção Ajuda do LabVad, os professores de física afirmaram que é necessário que as informações sobre os dispositivos sejam gradualmente detalhadas até chegar num nível avançado. Os outros professores elogiaram a estrutura de tópicos: “o que é”, “o que faz” e “como funciona no LabVad”. Estes defendem que os desenhos são suficientes para os iniciantes, porque as informações estão acessíveis a qualquer aluno.

Um dos docentes disse que seria a ferramenta educacional ideal no seu local de trabalho, o Consórcio das Universidades Públicas do Estado do Rio de Janeiro (CEDERJ). Apenas um professor afirmou que o LabVad não seria adequado para o ensino a distância, por ser monousuário na execução de Experimentos.

Metade dos professores afirmou que a deficiência do LabVad é ser monousuário na execução de Experimentos, fato que pode ser resolvido pelo baixo custo para clonar ou replicar o ambiente. Pode-se criar várias estações do LabVad para atender a uma demanda maior de usuários, como foi apresentado no trabalho relacionado CLEM (subseção 3.2.2).

Um dos docentes realizou uma oficina com o LabVad com uma turma de programação inicial, com cerca de 30 alunos do curso de Engenharia Civil e Engenharia de Produção. A aula foi numa sala convencional, com o uso de um projetor como recurso didático, para abordar o projeto Arduino, bem como o LabVad.

O tema da aula foi sobre estruturas condicionais, utilizando experimentos para exemplificar o uso do comando "if". Segundo o professor, o resultado da apresentação foi positivo, porque mesmo sem ele poder levar os alunos ao laboratório, os mesmos se interessaram pelos exemplos e apresentaram algumas sugestões sobre alterações no código dos programas, mostrando que entenderam o conceito de desvio condicional.

Num segundo momento, dois alunos, que estão fazendo o trabalho de conclusão de curso com Arduino, testaram a maioria das funções do ambiente e se interessaram pela ferramenta. O docente relatou que ferramentas como o LabVad podem gerar outra visão sobre o ensino de programação nos cursos de Engenharia, por causa da contextualização da robótica com a área de atuação deles.

## **7.8 Contribuição da Avaliação Docente**

O objetivo principal da avaliação docente foi coletar dados para melhorar a usabilidade da plataforma WEB do LabVad. No questionário, também havia espaço para sugestões e melhorias. Os professores solicitaram mais informações na página inicial e vídeos instrucionais.

Um dos riscos de trabalhar com vídeos é prejudicar a interatividade do usuário com a interface. Rocha e Baranauskas (2003) afirmam que os usuários podem ficar perdidos com enorme material numa seção de help e que é sempre melhor que o usuário opere o sistema sem utilizar a ajuda. Portanto, foi decidida a inclusão de um tutorial interativo nas principais funções do sistema e na página de Experimentos.

O usuário pode seguir todo o tutorial ou simplesmente encerrá-lo no momento que desejar, clicando na opção "Finalizar".



Figura 40. Tutorial das principais funções da Interface

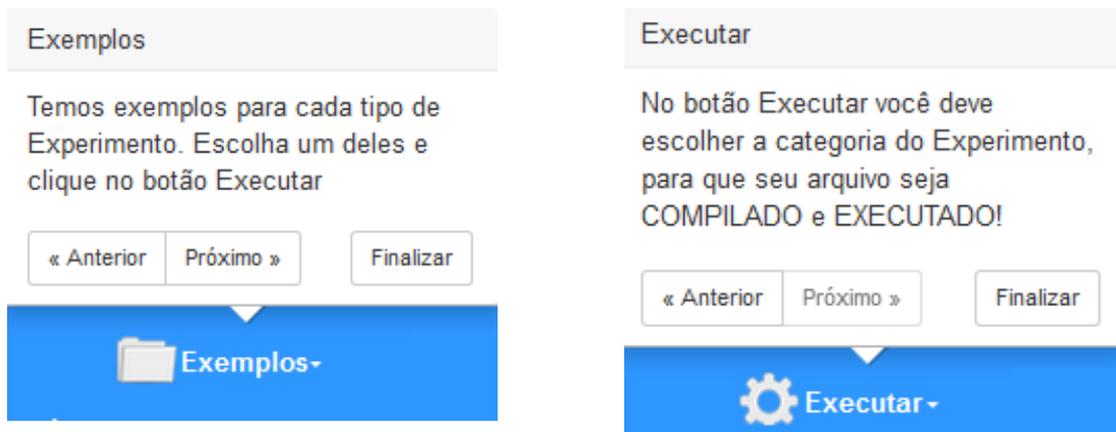


Figura 41. Tutorial dos Exemplos e Experimentos.



## **8 Conclusões e Trabalhos Futuros**

Este capítulo apresenta as conclusões da pesquisa e tece as considerações finais com propostas de trabalhos futuros.

O LabVad consolida mais de 10 anos de pesquisas bem sucedidas em robótica do GINAPE, iniciadas por Miranda em 2004, com construção do Kit didático RoboFácil.

Como vimos no capítulo 2, a versão atual do LabVad tem sua origem no edital PROUCA de 2012, sendo em seguida aprimorado pela equipe com o aporte de recursos da RNP (RNP, 2014).

Sampaio, Miranda e Elia (2014) propuseram o uso dos laboratórios didáticos de ciências e de Robótica Educacional no sistema escolar brasileiro. Esses laboratórios, na concepção desses autores, ocupariam um papel de polo virtual integrador e roteador de montagens experimentais nestas áreas do conhecimento, objetivando parceria com outras instituições, criando assim outros polos provedores de laboratórios com acesso remoto. Este projeto, o GT LabVad, será utilizado pela Secretaria Estadual de Educação (PORTAL NCE, 2015).

O estado atual do LabVad encontra-se em condições de utilização em espaços de ensino-aprendizagem. Para tanto, é importante que novas parcerias sejam construídas entre a equipe do GINAPE e escolas e/ou secretarias de educação.

## **8.1 Artigo Acadêmico sobre o Tema da Dissertação Submetido e Aprovado**

- **LabVad: Laboratório Remoto para o Desenvolvimento de Atividades Didáticas com Robótica".** Congresso Internacional de Informática Educativa – TISE. [SOUZA *et al.*, 2014].

## **8.2 Trabalhos Futuros**

### **8.2.1 Quanto aos aspectos tecnológicos**

Durante as avaliações IHC do LabVad, percebeu-se que ainda existem funcionalidades relevantes a serem implementadas, porém que fugiriam do escopo e/ou do tempo disponível para implementação, sendo apontados como problemas

cosméticos do projeto. Dentre as funcionalidades previstas que não foram desenvolvidas, devem ser priorizadas:

- A migração do hardware **Arduino Uno** para **Arduino Mega**. Em uma breve comparação, a versão Mega R3 possui 54 pinos digitais I/O e 16 pinos analógicos de entrada, enquanto que a versão Uno R3 possui, 14 DIGITAIS I/O e 6 pinos analógicos de entrada (ARDUINO, 2015). Pelo fato do Arduino Mega ser superior ao Uno, a interface não seria sobrecarregada de informações no botão Executar, utilizado na complicação e execução dos experimentos.
- Melhoria na mensagem de retorno do erro de compilação. Por enquanto, o LabVad só retorna se o código do Experimento foi compilado com sucesso ou não. Precisamos informar onde o erro está precisamente localizado dentro da área de código.
- O desenvolvimento de uma comunidade em torno do ambiente LabVad voltada para o incentivo do compartilhamento de programas.
- Liberar o projeto em repositório de código fonte online para que outros desenvolvedores possam propor e criar novas funcionalidades.
- Tradução da interface para outras línguas, como inglês e espanhol, de forma que o projeto ganhe visibilidade em eventos científicos internacionais.

### 8.2.2 Quanto aos aspectos pedagógicos

À medida que os projetos de robótica são aplicados em sala de aula, novas demandas vão surgindo e diferentes hipóteses podem ser levadas como objetos de estudo. Uma delas seria estudar a fusão do ambiente DuinoBlocks (ALVES et al., 2013) com o LabVad, formando uma “suíte” para o desenvolvimento de projetos na robótica educacional de baixo custo. Os dois ambientes, tanto para desenvolvimento quanto para implantação, utilizam uma infraestrutura baseada em softwares e tecnologias livres, não sendo portanto difícil replicá-la numa rede local de uma escola baseada no sistema operacional Linux (Anexo 1).

Outro aspecto importante a ser investigado é, em que medida, o uso de laboratórios remotos se diferenciam do uso de laboratórios presenciais em questões relativas ao aprendizado e interesse dos alunos.



# REFERÊNCIAS

---

---



ALVES, R. M. DuinoBlocks: **Desenho e Implementação de um Ambiente de Programação Visual para Robótica Educacional**. Dissertação de Mestrado. UFRJ/IM/NCE, Rio de Janeiro, 2013.

ALVES, R. M.; da SILVA A. L. C.; PINTO, M. de C.; SAMPAIO, F. F.; ELIA, M. da F. **Uso do hardware livre Arduino em ambientes de ensino-aprendizagem**. In: **JAIE - Jornada de Atualização em Informática e Educação**. In: CBIE - Congresso Brasileiro de Informática da Educação. Rio de Janeiro, Brasil, 2012.

ALVES, R. M.; SAMPAIO, F. F.; ELIA, M. da F. DuinoBlocks: **Um Ambiente de Programação Visual para Robótica Educacional**. In: SEMISH - XL Seminário Integrado de Software e Hardware. In: CSBC - XXXIII Congresso da Sociedade Brasileira de Computação. Maceió, Brasil, 2013.

ARDUINO. **What is Arduino?** Disponível em: <<http://arduino.cc>>. Acesso em: abril de 2015.

ARDUINO REMOTE LAB. Disponível em: <<http://pc2.dsmc.eap.gr>>. Acesso em: abril de 2015.

ARDUINODEV. **Arduino Online Compiler**. Disponível em: <[arduinodev.com/software/builder](http://arduinodev.com/software/builder)>. Acesso em: abril de 2015.

ARDUINO PLAYGROUND. **Windows command line build**. Disponível em: <<http://playground.arduino.cc/Code/WindowsCommandLine>>. Acesso em: janeiro de 2015.

ARDUINO TUMBLR. Disponível em: <<https://www.tumblr.com/tagged/arduino>>. Acesso em: abril de 2015.

ATMEL AVR. Disponível em: <<http://www.atmel.com/products/microcontrollers/avr/>>. Acesso em: maio de 2015.

BERTINI E.; GABRIELLI S.; KIMANI S. **Appropriating and assessing heuristics for mobile computing**. Proceedings of the working conference on Advanced visual interfaces, Pag 119-126, 2006.

BOOCH, G.; RUMBAUGH J.; JACOBSON I. **UML Guia do Usuário**. Ed. Campus, 2006.

CHAO K. M.; James A. E.; NANOS A.G.; CHEN J. H. **Cloud E-learning for Mechatronics: CLEM**. Future Generation Computer Systems, Volume 48, Pg. 46-59. 2014.

CHELLA, M. T. **SimRobô: simulador para robótica com propósito educacional**. In: III Workshop de Robótica Educacional, Fortaleza, 2012.

CLASSMATE. Disponível em: <http://www.instituicaodoseculo21.com.br/classmate.html>. Acesso em: julho de 2015.

CLEM. **Cloud Services for E-learning in Mechatronics**. Disponível em:<<http://clem.coventry.ac.uk/arduino/index.php>>. Acesso em: janeiro de 2015.

- CODEBENDER. **The Largest Arduino Playground In The World**. Disponível em: <<https://codebender.cc/>>. Acesso em: janeiro de 2015.
- CODING DOJO. **What is Coding Dojo**. Disponível em: <http://codingdojo.org/>. Acesso em: maio de 2015.
- CRINNION J. **Evolutionary systems development: a practical guide to the use of prototyping within a structured systems methodology**. Pitman Publishing. London, 1991.
- CRONIN G. **eXtreme Solo: A Case Study in Single Developer eXtreme Programming**. University of Auckland, 2001.
- CRUZ, M. K.; HAETINGER, W.; HORN, F.; de CARVALHO, D. V.; ARAÚJO, G. H. **Controle de Kit de Robótica através de Laboratório Remoto pela Internet: uma Aplicação para a Formação Docente e para a Educação Básica**. In: XX Simpósio Brasileiro de Informática na Educação, Florianópolis, 2009.
- DARGAINS A. R. **Estudo exploratório sobre o uso de robótica no ensino de programação**. Rio de Janeiro: Dissertação de Mestrado. UFRJ/IM/NCE, 2015.
- ERTUGRUL N. Towards virtual laboratories: a survey of LabVIEW-based teaching/learning tools and future trends. *International Journal of Engineering Education*, vol. 16, 2000.
- FALLON H.; LATTRE A.; BILLEN J.; DAOUD A.; GAUTIER M.; STENAC C. **VLC User Guide**, 2004. Disponível em: <<http://images.videolan.org/doc/vlc-user-guide/en/vlc-user-guide-en.pdf>>. Acesso em: fevereiro de 2015.
- FERNANDES, C. **Um Simulador de Ambiente de Robótica Educacional em Plataforma Virtual**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, 2013.
- FOWLER, M.; **UML Essencial – Um breve guia para a linguagem padrão de modelagem de objetos**. Bookman, 2005.
- GENERAL PUBLIC LICENSE. Disponível em: <<http://www.gnu.org/licenses/gpl.html>>. Acesso em: maio de 2015.
- HARDISON J.; ZYCH D.; ALAMO J. A.; HARWARD V. J.; LERMAN S. R.; WANG S. M.; YEHA K.; VARADHARAJAN C. **The Microelectronics WebLab 6.0 – An Implementation Using Web Services and the iLab Shared Architecture**. iCEER, Exploring Innovation in Education and Research, 2005.
- HARWARD V.J.; DEL ALAMO, J.A.; LERMAN S.R. ; BAILEY P.H.; CARPENTER J. ; DELONG K.; FELKNOR, C. ; HARDISON J. ; HARRISON B. ; JABBOUR I. ; LONG,P.D.; TINGTING MAO; NAAMANI, L. ; NORTHRIDGE, J.; SCHULZ, M. ; TALAVERA, D.; VARADHARAJAN C.; WANG S.M.; YEHA, K. ; RABIH Z.; ZYCH, D. **The iLab Shared Architecture: A Web Service Infrastructure to Build Communities of Internet Accessible Laboratories**. Proceedings of the IEEE, Vol. 96, No. 6, 2008.
- ICAMPUS. **Commemoration of the successful conclusion of the iCampus program**. 6 Massachusetts Institute of Technology, 2006.
- INEP. **Sobre o PISA**. Disponível em: <<http://portal.inep.gov.br/pisa/sobre-o-pisa>>. Acesso em: março de 2015.

ISO 9241. **Design de Interfaces.** Disponível em <<http://www.design.eti.br/content/view/67/2/>>. Acesso em: dezembro de 2014.

JAMES A. E.; CHAO K. M.; LI W.; MATEI A.; NANOS A.G.; STAN S.D; FIGLIOLINI G.; REA P., BOUZGARROU C. B.; BRATANOV D., COOPER J.; WENZEL A., CAPELLE J.V.; STRUECKMEIER K. **An Ecosystem for E-Learning in Mechatronics: The CLEM Project.** In **e-Business Engineering (ICEBE)**, 2013 IEEE 10th International Conference.pp. Disponível em: <<https://www.deepdyve.com/lp/institute-of-electrical-and-electronics-engineers/an-ecosystem-for-e-learning-in-mechatronics-the-clem-project-tM53cHDrXa/3>>. Acesso em: fevereiro de 2015.

JESUS A. e BRITO G.C. **Concepção de ensino-aprendizagem de algoritmos e programação de computadores: a prática docente.** Revista Varia Scientia v.09 , n.16, p. 149-158, 2010.

LABVIEW. **Versões do LabVIEW.** Disponível em: <<http://sine.ni.com/np/app/main/p/ap/lvtn/lang/pt/pg/1/ps/1000/sn/n21:28,n2:20167>>. Acesso em 20 de abril de 2015.

MAJOR L.; KYRIACOU T.; BRERETON O. P. **Systematic Literature Review: Teaching Novices Programming Using Robots.** Proceedings of EASE, 2011.

MCROBERTS M. **Arduino Básico.** São Paulo: Novatec Editora, 2011.

METASYS MEEGO. Disponível em: [http://www.metasys.com.br/index.php?option=com\\_content&view=article&id=417&Itemid=173&lang=pt](http://www.metasys.com.br/index.php?option=com_content&view=article&id=417&Itemid=173&lang=pt). Acesso em: julho de 2015.

MIRANDA, L. C. de. **RoboFácil: especificação e implementação de artefatos de hardware e software de baixo custo para um kit de robótica educacional.** Dissertação de Mestrado. UFRJ/IM/NCE, Rio de Janeiro, 2006.

MIRANDA, L. C. de; SAMPAIO, F. F.; BORGES, J. A. dos S. **RoboFácil: Especificação e Implementação de um Kit de Robótica para a Realidade Educacional Brasileira.** Revista Brasileira de Informática na Educação, Volume 18, Número 3, 2010.

NEDIC Z.; MACHOTKA J.; NAFALSKI A. **Remote laboratories versus virtual and real laboratories.** IEEE, vol. 1. 2003.

NIELSEN, J. **Heuristic Evaluation.** New York, NY: John Wiley & Sons, 1994.

NIELSEN, J. e MACK R.L. **Usability Inspection Methods.** New York, NY: John Wiley & Sons, 1994.

OECD. **Results from PISA 2012.** Disponível em: <[http://download.inep.gov.br/acoes\\_internacionais/pisa/resultados/2013/country\\_note\\_brazil\\_pisa\\_2012.pdf](http://download.inep.gov.br/acoes_internacionais/pisa/resultados/2013/country_note_brazil_pisa_2012.pdf)>. Acesso em: março de 2015.

PAPERT S. **A máquina das crianças: repensando a escola na era da informática.** Edição Revisada. Porto Alegre: Artmed, 2008.

PASTERNAK, E. **Visual Programming Pedagogies and Integrating Current Visual Programming Language Features.** Carnegie Mellon University Robotics Institute. Thesis

Master's Degree. 2009. Disponível em: <[http://www.ri.cmu.edu/pub\\_files/2009/8/Thesis-1.pdf](http://www.ri.cmu.edu/pub_files/2009/8/Thesis-1.pdf)>. Acesso em: fevereiro de 2015.

PHD. **The definitive guide to PHP's DocBook Rendering System**, 2010. Disponível em: <http://doc.php.net/phd/docs>. Acesso em: fevereiro de 2015.

PINTO, M. de C.; ELIA, M. da F.; SAMPAIO, F. F. **Formação de professores em robótica educacional com Hardware Livre Arduino no contexto Um Computador por Aluno**. In: XVIII Workshop de Informática na Escola. Rio de Janeiro, 2012.

PINTO, M. de C. **Aplicação de arquitetura pedagógica em curso de robótica educacional com hardware livre**. UFRJ/IM/NCE, Rio de Janeiro, 2011.

PORTAL NCE. Disponível em: <[http://portal.nce.ufrj.br/index.php?option=com\\_content&view=article&id=589%3Aparceria&catid=34%3Afique-por-dentro&Itemid=67](http://portal.nce.ufrj.br/index.php?option=com_content&view=article&id=589%3Aparceria&catid=34%3Afique-por-dentro&Itemid=67)>. Acesso em: outubro de 2015.

PRATES, R.O.; BARBOSA, S.D.J. **Avaliação de Interfaces de Usuário - Conceitos e Métodos** Anais do XXIII Congresso Nacional da Sociedade Brasileira de Computação. XXII Jornadas de Atualização em Informática (JAI). SBC 2003.

PREECE, J.; ROGERS, Y.; SHARP, E.; BENYON, D.; HOLLAND, S.; CAREY, T. **Human-Computer Interaction**. England: Addison-Wesley, 1994.

PREECE, J.; ROGERS, Y.; SHARP, E. **Interaction Design: Beyond Human-computer Interaction**. New York, NY: John Wiley & Sons. 2002.

PRESSMAN, R. S. **Engenharia de Software**. 5ª edição, Rio de Janeiro, McGraw-Hill, 2006.

PROCESSING. Disponível em: <<http://processing.org/>>. Acesso em: fevereiro de 2015.

PROUCA. **Programa Um Computador por Aluno**. Disponível em: <[http://www.nce.ufrj.br/GINAPE/publicacoes/Projetos/Proj\\_UCAnaCUCA.pdf](http://www.nce.ufrj.br/GINAPE/publicacoes/Projetos/Proj_UCAnaCUCA.pdf)>. Acesso em: maio 2015.

RASPBERRY PI. **What is a Raspberry Pi?** Disponível em: <<http://www.raspberrypi.org/>>. Acesso em: maio de 2015.

RNP EM REVISTA. **Uma publicação da Rede Nacional de Ensino e Pesquisa**. Ano 3. 2015. Disponível: < [http://www.rnp.br/sites/default/files/rnp\\_revista\\_2015.pdf](http://www.rnp.br/sites/default/files/rnp_revista_2015.pdf)>. Acesso em: setembro de 2015.

SALGADO L.C.C.; BIM S.A.; SOUZA C.S. **Comparação entre os Métodos de Avaliação de Base Cognitiva e Semiótica**. Anais do IHC, 2006.

SAMPAIO, F. F.; ELIA, M. F. **Projeto Uca na Cuca: robótica educacional na sala de aula**. Disponível em: <[http://www.nce.ufrj.br/GINAPE/publicacoes/Projetos/Proj\\_UCAnaCUCA.pdf](http://www.nce.ufrj.br/GINAPE/publicacoes/Projetos/Proj_UCAnaCUCA.pdf)>. Acesso em: novembro 2014.

SAMPAIO, F. F.; MIRANDA, L.C.; ELIA M. F. **LabVad: Laboratório Virtual de Atividades Didáticas em Ciências e Robótica**. Rede Nacional de Pesquisas, 2014. Disponível em: <[http://www.rnp.br/sites/default/files/labvad\\_sdf.pdf](http://www.rnp.br/sites/default/files/labvad_sdf.pdf)>. Acesso em: janeiro de 2015.

SAMPAIO, F.F.; MIRANDA, L.C.; ELIA, M.F.; PINTO, M.C.; BOMFIM, M.N.C.; PINTO, S.B.; QUEIROZ, R.L.; ROCHA, R.N.C.B.; BENTO, M.A., LabVad: **Uma Proposta de Consórcio Nacional para Laboratórios Didáticos a Distância de Ciências e Robótica**. XLII Seminário Integrado de Software e Hardware (SEMISH'15) / XXXV Congresso da Sociedade Brasileira de Computação (CSBC'15), 2015.

SCRATCH BRASIL. Disponível em: <<http://www.scratchbrasil.net.br/index.php/materiais/tutoriais.html>>. Acesso em: abril de 2015.

SILVA, J.B.; ROCHADEL W.; SIMÃO J.P.S.; FIDALGO A.V.S. **Uso de dispositivos móveis para acesso a Experimentos Remotos na Educação Básica**. VAEP-RITA. Vol. 1, Núm. 2, 2013.

SOUZA, M. B.; NETTO, J. F. M.; ALENCAR, M. A. S.; SILVA, M. M. **Arcabouço de um Ambiente Telerobótico Educacional Baseado em Sistemas Multiagentes**. In: XXII Simpósio Brasileiro de Informática na Educação. Aracajú, 2011.

SOUZA, P.R.A.; SAMPAIO, F.F.; ELIA, M.F. **Laboratório Remoto para o Desenvolvimento de Atividades Educacionais com Robótica** – LabVad. XIX Conferência Internacional sobre Informática na Educação, 2014. Disponível em: <[http://www.tise.cl/volumen10/TISE2014/tise2014\\_submission\\_218.pdf](http://www.tise.cl/volumen10/TISE2014/tise2014_submission_218.pdf)>. Acesso em: abril de 2015.

VASSILIS F.; ANASTASIOS S.I.; ANASTASIOS F. **Preparing a Remote Conducted Course for Microcontrollers based on Arduino**. 7th International Conference in Open and Distance Learning (ICODL 2013), Ellinogermaniki Agogi, Athens, Greece, 2013.

VICTORINO, L.; ELIA, M.; GOMES, A.; CASTRO, M. E BASTOS, C. **Laboratório Virtual de Atividades Didáticas – LabVad**, In: XV Workshop de Informática na Escola – WIE. Bento Gonçalves, Rio Grande do Sul, 2009. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wie/2009/022.pdf>>. Acesso em: dezembro de 2014.

W3C. **Web Storage**, 2014. Disponível em <<http://dev.w3.org/html5/webstorage/>>. Acesso em: janeiro de 2015.

WALKER, C. **Primeiros passos com o NetDuino**. São Paulo: Novatec Editora, 2012.

WANG X; ZHOU F.; ZHANG H. (2012). **The Research and Practice of Teaching Method Reform for Single-chip Microcomputer Course**. International Conference on Education Technology and Computer, 2012. Disponível em: <http://www.ipcsit.com/vol43/046-ICETC2012-T2096.pdf>. Acesso em: de abril de 2015.

WHARTON, C.; RIEMAN, J. ;Lewis, C.; POLSON, P. **The Cognitive Walkthrough Method: A Practitioner's Guide**. John Wiley & Sons, New York, NY, 1994.

ZANETTI, H. A. P. **Análise semiótica do uso de Robótica Pedagógica no ensino de Programação de Computadores**. Dissertação de Mestrado. Faculdade Campo Limpo Paulista. Campo Limpo Paulista – SP, 2014.

ZANETTI, H. A. P.; SOUZA A.L.S.; d'ABREU J.V.V.; BORGES M.A.F. **O Uso da robótica em jogos digitais como sistema de apoio ao aprendizado**. In: JAIE - Jornada de Atualização em Informática e Educação. In: CBIE - Congresso Brasileiro de Informática da Educação. Rio de Janeiro, Brasil, 2012.

ZHANG S.; CHEN X.; HUO X. **Cloud Computing Research and Development Trend**. In: Second International Conference on Future Networks, 2010.



# ANEXOS



## Anexo 1 - Instalação do LabVad no Linux

**1a parte** – Instalação do Arduino e do script compilação e upload via linha de comando.

*1) Instalar o Arduino no Ubuntu 12.04 ou mais recente (na pasta /usr/share/arduino)*

```
sudo apt-get update  
sudo apt-get install arduino arduino-core
```

*2) Instalar o Ino*

```
sudo apt-get install python-pip  
sudo pip install ino  
sudo apt-get install picocom
```

*3) Criar a pasta do projeto*

```
mkdir ~/someproject  
cd ~/someproject  
ino init # initializes an empty directory for use by Ino
```

*4) Conectar a placa Arduino e verificar se a mesma foi reconhecida com o comando dmesg*

**Observação:** caso o Arduino não tenha aparecido em /dev/ttyUSB0 ou /dev/ttyACM0, utilize os comandos:

```
sudo rmmod cdc_acm  
sudo modprobe cdc_acm
```

Aparecerá no dmesg uma linha indicando que o hardware foi reconhecido:

```
[76.390720] cdc_acm 6-2:1.0: ttyACM0: USB ACM device
```

**5) Configurar a placa Arduino (criar arquivo *ino.ini* dentro do projeto) fazendo referência ao modelo da placa Arduino (uno) e a porta serial.**

```
[build]
```

```
board-model = uno
```

```
[upload]
```

```
board-model = uno
```

```
serial-port = /dev/ttyACM0
```

```
[serial]
```

```
serial-port = /dev/ttyACM0
```

**6) Copiar o arquivo *.ino* (a ser compilado e carregado na placa) para a pasta *src* do projeto.**

**7) Compilar e carregar o programa através da linha de comando:**

```
ino build # compiles your code
```

```
ino upload # uploads to your board
```

**2a Parte** – Instalação da infra-estrutura web (apache, PHP e MySQL) e de aplicação LabVad.

**1) Instalar o Apache, o PHP e o Mysql na máquina Ubuntu:**

```
sudo apt-get install apache2 php5
```

```
sudo apt-get install php5-mysql # instala a conexão php-mysql (PDO)
```

```
sudo service apache2 restart # é necessário restartar o apache após instalar o PDO senão vai dar erro qdo for conectar no banco.
```

```
sudo apt-get install mysql-server
```

```
sudo apt-get install mysql-workbench
```

### **2) Copiar o sistema labvad da máquina Windows para o Ubuntu**

Copiar a pasta labvad para o Apache (na pasta /var/www ou /var/www/html, dependendo da versão do apache instalada)

### **3) Criar o projeto ino dentro da pasta labvad**

```
cd /var/www/labvad
mkdir proj_arduino
cd proj_arduino
ino init
```

Serão criadas as pastas src e lib dentro de proj\_arduino.

Apagar o arquivo sketch.ino criado dentro da pasta src;

### **4) Copiar o banco de dados LabVad**

Entrar no MySql Workbench e conectar em localhost;

Clicar no ícone "Create a new schema in the connected server"

Criar o database labvad;

### **5) Definir permissões de acesso ao labvad**

Abrir a permissão de leitura das seguintes pastas do labvad (chmod 755)

```
/var/www/labvad
/var/www/labvad/app.ado
/var/www/labvad/app.include
/var/www/labvad/set
/var/www/labvad/set/css
/var/www/labvad/set/img
/var/www/labvad/set/js
/var/www/labvad/css/*
/var/www/labvad/img/*
/var/www/labvad/js/*
```

O usuário www-data precisa ter permissão de escrita sobre o proj\_arduino para que ele possa salvar o código fonte em src e o compilado em .build

```
cd /var/www/labvad/proj_arduino  
sudo chown -R www-data .
```

Para que o usuário `www-data` possa acessar o hardware do Arduino, ele tem que pertencer ao grupo `dialout`. É preciso editar o arquivo `/etc/group` e acrescentar `",www-data"` na linha onde o grupo `dialout` é definido:

```
dialout:x:20:administrador,www-data
```

# APÊNDICES



## Apêndice A – Tarefas de avaliação (Método de Percurso Cognitivo).

Nome do avaliador: \_\_\_\_\_

Data e hora: \_\_\_\_\_

**Objeto de avaliação: LabVad**

**Endereço de acesso:** <http://146.164.3.24/ucanacuca/labvad>

### 01.Tarefa - *Login* e outras funções relacionadas ao Acesso.

1. Acesse <http://146.164.3.24/ucanacuca/labvad> e se **cadastre**.
2. Após efetuar **login** (<http://146.164.3.24/ucanacuca/labvad>) altere sua senha.
3. Navegue livremente, para conhecer a plataforma do LabVad.
4. Saia do sistema. Volte à tela página de acesso (<http://146.164.3.24/ucanacuca/labvad>) e recupere sua senha.
5. Faça login para validar sua nova senha.

### 02.Tarefa – Agendamento

1. Tente agendar um horário a partir da tela de Experimentos.
2. Efetue um agendamento.
3. Agende mais 3 horários no LabVad, sendo um deles de período de pelo menos 90 minutos.
4. Exclua um dos horários reservados.

### 03.Tarefa – Experimentos

1. A partir da pasta “Exemplos” execute experimentos para:
  - a. LED
  - b. Display de LCD
  - c. Display de 7 Segmentos
  - d. LED RGB
  - e. Servo Motor
  - f. Motor DC.
  - g. Relé

2. Salve, no LabVad, 3 cópias de um exemplos diferentes.
3. Copie um exemplo da página oficial do Arduino e teste se funciona no LabVad.
  - a. Link: <http://arduino.cc/en/Tutorial/Blink?action=sourceblock&num=1>
4. Execute-o como experimento.
5. Salve este arquivo.
6. Faça um upload de dois arquivos, que estão na pasta Uploads.
7. Faça download dos arquivos que desejar.
8. Exclua um dos arquivos que foi criado no passo 2 desta Tarefa.

#### **04.Tarefa – Utilizar a Ajuda.**

1. Selecione a opção Ajuda
2. Escolha um dos tópicos
3. Copie e cole um dos nossos exemplos na área de código dos Experimentos.
4. Execute e salve o arquivo.

#### **05.Tarefa – Configurações.**

1. Clique em Configurações para acessar seu LOG.
2. Visualize e faça download do seu LOG.



**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

CCMN - Bloco C - Cidade Universitária - Ilha do Fundão  
Rio de Janeiro - RJ CEP: 21941-916  
[www.pggi.ufrj.br](http://www.pggi.ufrj.br)