



UNIFACS

UNIVERSIDADE SALVADOR

LAUREATE INTERNATIONAL UNIVERSITIES'

**UNIFACS UNIVERSIDADE SALVADOR
MESTRADO ACADÊMICO EM SISTEMAS E COMPUTAÇÃO**

CRÍJINA CHAGAS FLORES

**LERO: UM LABORATÓRIO REMOTO DE ROBÓTICA EDUCACIONAL
EXTENSÍVEL E ADAPTÁVEL**

Salvador
2015

CRÍJINA CHAGAS FLORES

**LERO: UM LABORATÓRIO REMOTO DE ROBÓTICA EDUCACIONAL
EXTENSÍVEL E ADAPTÁVEL**

Dissertação apresentada ao Mestrado de Sistemas e Computação da UNIFACS Universidade Salvador, Laureate International Universities, como requisito parcial para obtenção do título de Mestre.

Orientador: Prof. Dr. Jorge Alberto Prado de Campos.

Salvador
2015

FICHA CATALOGRÁFICA

Elaborada pelo Sistema de Bibliotecas da UNIFACS Universidade Salvador, Laureate International Universities

Flores, Críjina Chagas

LERO: um laboratório remoto de robótica educacional extensível e adaptável. / Críjina Chagas Flores. – Salvador, 2015.

143 p.: il.

Dissertação apresentada ao Curso de Mestrado em Sistemas e Computação, UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Dr. Jorge Alberto Prado de Campos.

1. Robótica. 2. Robótica Educacional. 3. Laboratório Remoto de Robótica. 4. Programação em Blocos. I. Campos, Jorge Alberto Prado de, orient. II. Título.

CDD: 629.892

TERMO DE APROVAÇÃO

CRÍJINA CHAGAS FLORES

LERO: UM LABORATÓRIO REMOTO DE ROBÓTICA EDUCACIONAL
EXTENSÍVEL E ADAPTÁVEL

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, da UNIFACS Universidade Salvador, Laureate International Universities, pela seguinte banca examinadora:

Jorge Alberto Prado de Campos – Orientador _____
Doutor em Spatial Information Science and Engineering, University of Maine at
Orono
UNIFACS Universidade Salvador, Laureate International Universities

Artur Henrique Kronbauer _____
Doutor em Ciência da Computação
Universidade Federal da Bahia - UFBA

Roque Mendes Prado Trindade _____
Doutor em Engenharia Elétrica e de Computação
Universidade Federal do Rio Grande do Norte - UFRN

Salvador, 9 de outubro de 2015

Dedico este trabalho à minha mãe Corina,
ao meu esposo Igor e as minhas filhas
Tatyana e Júlia.

AGRADECIMENTOS

Agradeço a Deus por me carregar no colo durante esse caminho tão pesado, mas extremamente compensador.

Agradeço a minha mãe e meus irmãos por serem os pilares de toda a minha vida.

Agradeço ao meu amado esposo Igor Luiz e as minhas filhas lindas Tatyana e Júlia por tanto apoio, compreensão, respeito e principalmente denovo e amor.

Agradeço aos meus sogros e minha cunhada pelo apoio incondicional.

Agradeço a toda minha família tão maravilhosa, em especial ao meu primo amado Diego, por ser a minha inspiração de coragem e força.

Agradeço com todo carinho aos meus anjos na Terra tia Vina e Luiza.

Agradeço ao meu orientador Prof. Jorge Campos por ter abraçado a minha causa depois de tantos percalços, por ter me orientado com tanta competência, disponibilidade, exigência, sem perder os valores humanos de compreensão e solidariedade. Não esquecerei os seus ensinamentos e sempre os passarei adiante.

Agradeço a todos os amigos da UESB, em especial do projeto LARA, que sempre me incentivaram, me apoiaram e me deram toda a estrutura física para a realização deste projeto.

Agradeço a Rodrigo, Railane e Iago por tanta dedicação, empenho, compromisso. Vocês foram essenciais nessa jornada.

Agradeço pela participação mais que especial de Alan Yorinks, que criou o s2a. Suas orientações foram muito importantes em um momento crucial desse projeto.

E a todos que contribuíram para a realização deste trabalho.

“Agradeço a todas as dificuldades que enfrentei; não fosse por elas, eu não teria saído do lugar. As facilidades nos impedem de caminhar. Mesmo as críticas, nos auxiliam muito.” (Chico Xavier).

RESUMO

A robótica educacional é uma ferramenta que quando utilizada em ambientes acadêmicos favorece o aprendizado através do lúdico. Os grandes desafios relacionados à implantação de ambientes de robótica educacional são os custos para aquisição de equipamentos, a complexidade da infraestrutura robótica e capacitação dos profissionais envolvidos no processo. Uma forma de minimizar esses desafios é disponibilizar acesso a laboratórios remotos de robótica. Nesse contexto o presente trabalho propõe o desenvolvimento de um Laboratório Remoto de Robótica Educacional (LERO). O ambiente LERO tem como objetivo contribuir com o ensino e aprendizagem de programação e robótica para diferentes faixas etárias, estimulando e diversificando a aquisição de conhecimentos. O LERO é um ambiente baseado na Web, extensível e adaptável e que conta com um ambiente integrado de desenvolvimento especializado em programação de robôs. O termo extensível é usado para caracterizar a capacidade que o sistema possui de permitir que novos blocos de construção sejam incorporados. Blocos com complexidade crescente são desenvolvidos e incorporados ao sistema através da utilização do próprio ambiente de desenvolvimento LERO. Já o termo adaptável ilustra a característica de atender a um público com nível de conhecimento e experiência variada, tanto no que se refere à lógica de programação, quanto aos conhecimentos em robótica.

Palavras chave: Robótica. Robótica Educacional. Laboratório Remoto de Robótica. Programação em Blocos.

ABSTRACT

Educational robotics is a tool that when used in academic environments encourage learning through playful activities. The major challenges related to the implementation of educational robotics environments are the costs to purchase equipment, the complexity of robotic infrastructure and the training of professionals involved in the process. One way to minimize these challenges is to provide access to remote laboratories of robotics. In this context, this work proposes the development of a Remote Laboratory of Educational Robotics (LERO). LERO aims to contribute to the teaching and learning of programming skills and robotics to different groups, stimulating the acquisition of knowledge. The LERO is a Web-based environment, extensible and adaptable and which includes an integrated development environment specialized in robot programming. The extensible term is used to characterize the ability of the system to incorporate new functionalities. The term adaptable term highlights that the environments can be used by a broad audience with varied level of knowledge and experience in terms of programming logic and robotics.

Key Words: Robotic. Educational Robotic. Robotics Remote Laboratory. Programming Blocks.

LISTA DE FIGURAS

Figura 2.1 Ambientes do projeto GrubiBots.....	24
Figura 2.2 – Arquitetura do projeto ROMEU III.....	26
Figura 2.3 – Estrutura de <i>software</i> e <i>hardware</i> do projeto ROMEU III	27
Figura 2.4 – Arquitetura da Aplicação Telemática	28
Figura 2.5 – Estrutura de <i>software</i> da Aplicação Telemática.....	28
Figura 2.6 – Estrutura de <i>hardware</i> da Aplicação Telemática	29
Figura 2.7 – Arquitetura do Ambiente Telerobótico Educacional.....	30
Figura 2.8 – Sala de experimentação do ATE	31
Figura 2.9 – Arquitetura para o controle remoto do projeto de pesquisa.....	32
Figura 2.10 – Cenários para movimento do robô	32
Figura 2.11 – Ambiente de programação do Scratch	35
Figura 2.12 – Comparação de um algoritmo de soma escrito em Scratch, C e Java.....	36
Figura 2.13 – Ambiente gráfico do Snap!	37
Figura 2.14 – Código no S4A que faz piscar um LED no Arduino	39
Figura 2.15 – Blocos adicionais do S2A para Scratch e Snap!	40
Figura 3.1 – Arquitetura do Ambiente LERO	43
Figura 3.2 – Componentes do Ambiente Cliente	44
Figura 3.3 – Página inicial do projeto LERO.....	45
Figura 3.4 – Tela do Meu Perfil do cliente usuário	46
Figura 3.5 – Tela Usuários do cliente administrador	47
Figura 3.6 – Ambiente Cliente do LERO Criança	48
Figura 3.7 – Grupos Movimento e Aparência do LERO Criança	50
Figura 3.8 – Grupos Controle e Som do LERO Criança.....	51
Figura 3.9 – Roteiro construído no LERO Criança	51
Figura 3.10 – Ambiente Cliente do LERO Adolescente.....	53
Figura 3.11 – Grupos Variáveis e Controle do LERO Adolescente	54
Figura 3.12 – Grupos Sensores e Operadores do LERO Adolescente	55
Figura 3.13 – Roteiro construído no LERO Adolescente	56
Figura 3.14 – Ambiente Cliente do LERO Extremo	57
Figura 3.15 – Grupos Movimento e Sensores do LERO Extremo	58
Figura 3.16 – Roteiro construído no LERO Extremo	59
Figura 3.17 – Funcionalidade Upload do Ambiente LERO Extremo.....	60
Figura 3.18 – Componentes do Ambiente Servidor.....	61

Figura 3.19 – Fluxo das solicitações entre os servidores	62
Figura 3.20 – Componentes do Servidor Web.....	63
Figura 3.21 – Componentes do Servidor Lero.....	64
Figura 3.22 – Componentes da Plataforma Robótica	65
Figura 3.23 – Imagem do Lerobô	66
Figura 3.24 – Planta do Lerobô	68
Figura 3.25 – Robô e Arena da Plataforma Robótica	69
Figura 3.26 – Desenho e medidas da arena labirinto	69
Figura 3.27 – Imagem da LifeCam HD-3000 da Microsoft	70
Figura 3.28 – Visão da arena no ângulo da câmera de vídeo	70
Figura 4.1 – Diagrama de Caso de Uso do Ambiente LERO	74
Figura 4.2 – Tecnologias Utilizadas no Servidor Web	79
Figura 4.3 – Mais Tecnologias Utilizadas no Servidor Web	80
Figura 4.4 – Integração entre as Tecnologias Utilizadas no Servidor Web	81
Figura 4.5 – Adaptações do Snap! para o Snap! for LERO	83
Figura 4.6 – Sistema de conversão de projetos LERO.....	85
Figura 4.7 – Esquema geral da comunicação Snap! for LERO e o Robô.....	87
Figura 4.8 – <i>Script</i> e Comando interno de um bloco s2a.....	87
Figura 4.9 – Fluxo do processamento do bloco s2a no projeto LERO	89
Figura 4.10 – Trecho de código do método <i>digital_write</i>	90
Figura 5.1 – Tela Meu Perfil do ambiente Web LERO.....	92
Figura 5.2 – Tela da mensagem de alerta da reserva do laboratório	93
Figura 5.3 – Tela com <i>script</i> sem sucesso no roteiro do LERO Criança	94
Figura 5.4 – Tela com <i>script</i> sem sucesso sendo executado na Plataforma Robótica	95
Figura 5.5 – Tela com mensagem de erro no LERO Criança.....	96
Figura 5.6 – Tela com <i>script</i> com sucesso no roteiro do LERO Criança	97
Figura 5.7 – Tela com ações iniciais do <i>script</i> com sucesso na Plataforma Robótica	97
Figura 5.8 – Tela para criar um novo bloco no LERO Extremo	99
Figura 5.9 – Tela com o <i>script</i> para funcionamento do novo bloco no LERO Extremo	100
Figura 5.10 – Tela com a mensagem de sucesso do Upload do novo projeto no LERO Extremo	101
Figura 5.11 – Tela com <i>script</i> utilizando o novo bloco no roteiro do LERO Criança	102

Figura 5.12 – Tela com ações iniciais do *script* utilizando o novo bloco do LERO
Criança 103

SUMÁRIO

1 INTRODUÇÃO	13
1.1 JUSTIFICATIVA	14
1.2 OBJETIVO	15
1.3 METODOLOGIA	16
1.4 CONTRIBUIÇÕES.....	16
1.5 AUDIÊNCIA	17
1.6 APRESENTAÇÃO	17
2 LABORATÓRIOS REMOTOS DE ROBÓTICA COM FINS EDUCACIONAIS	19
2.1 ROBÓTICA NO CONTEXTO EDUCACIONAL.....	20
2.2 LABORATÓRIOS REMOTOS	22
2.3 LABORATÓRIOS REMOTOS E A ROBÓTICA EDUCACIONAL	26
2.4 LINGUAGENS DE PROGRAMAÇÃO EM BLOCOS	34
2.5 LINGUAGENS DE PROGRAMAÇÃO EM BLOCOS E A ROBÓTICA.....	38
2.6 CONSIDERAÇÕES FINAIS.....	40
3 A ARQUITETURA DO AMBIENTE LERO	42
3.1 O AMBIENTE CLIENTE	43
3.1.1 Páginas WEB	45
3.1.2 LERO Criança.....	48
3.1.3 LERO Adolescente.....	52
3.1.4 LERO Extremo.....	56
3.2 O AMBIENTE SERVIDOR.....	60
3.2.1 O Servidor Web.....	62
3.2.2 O Servidor Lero.....	64
3.3 A PLATAFORMA ROBÓTICA	65
3.4 CONSIDERAÇÕES FINAIS.....	70
4 DESENVOLVIMENTO DO AMBIENTE LERO	72
4.1 O DESENVOLVIMENTO DO AMBIENTE CLIENTE LERO	72
4.1.1 Requisitos Funcionais e Não Funcionais.....	73
4.1.2 Métricas da Usabilidade	75
4.2 O DESENVOLVIMENTO DO AMBIENTE SERVIDOR LERO	79
4.2.1 Servidor Web.....	79
4.2.2 Servidor Lero.....	86
4.3 CONSIDERAÇÕES FINAIS.....	91
5 O USO DO AMBIENTE LERO.....	92

5.1 LERO EM AÇÃO	93
5.2 DE LERO PARA LERO	98
5.3 MAIS LERO EM AÇÃO.....	102
5.4 CONSIDERAÇÕES FINAIS.....	103
6 CONCLUSÕES E TRABALHOS FUTUROS.....	104
REFERÊNCIAS.....	106
ANEXO A – AVALIAÇÃO DE USABILIDADE.....	110
APENDICE A – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS DO LERO	115

1 INTRODUÇÃO

A robótica tem sido uma área de muitos avanços no panorama mundial e está presente em soluções de automação residencial, médicas, de resgate em situação de risco e, em destaque neste trabalho, para fins educacionais.

O uso da robótica em ambientes educacionais já demonstrou ser uma ferramenta apropriada em atividades lúdicas que favorecem o processo de ensino-aprendizagem e que ampliam a relação entre diferentes áreas de conhecimento (BAGNALL, 2007). Na robótica educacional, os desafios propostos para a solução de problemas direciona o estudante a trabalhar com conceitos multidisciplinares e a vivenciar, na prática, o método científico, simulando mecanismos do cotidiano através de construção de projetos controlados por computador (DE SOUZA PIO; DE CASTRO; DE CASTRO JÚNIOR, 2006). Além disso, a robótica favorece o desenvolvimento do raciocínio lógico, incentiva tanto o trabalho em grupo quanto o autônomo, estimula a criatividade de uma forma lúdica e motivadora e possibilita a associação da teoria e dos ensinamentos apresentados em sala de aula com a solução de problemas práticos. O desafio atual é desenvolver aplicações inovadoras e garantir a inclusão computacional-robótica a um maior número de crianças e adolescentes. Neste contexto, é importante que pesquisadores da área de educação, computação e robótica se unam no processo de democratizar o acesso ao conhecimento, à produção e interpretação dessas tecnologias (BRITO; PURIFICAÇÃO, 2006).

Os maiores desafios relacionados à popularização de ambientes de robótica educacional estão relacionados aos custos envolvidos para aquisição dos equipamentos, a complexidade de implantação da infraestrutura robótica e a capacitação de profissionais para propor e orientar práticas e experimentos. Uma forma de minimizar o custo e a complexidade de implantação dessa infraestrutura nas escolas é disponibilizar, através dos recursos computacionais existentes, o acesso a laboratórios virtuais ou remotos de robótica. Em uma realidade em que muitos laboratórios de informática de escolas públicas se encontram subutilizados, esta estratégia pode contribuir para o desenvolvimento de atividades práticas que auxiliem o processo de aprendizagem do aluno (FIORIO, et al., 2014)

Diante desse cenário, este trabalho apresenta o Laboratório Remoto de Robótica Educacional (LERO). O LERO é um ambiente extensível e adaptável

direcionado a diferentes públicos e conta com uma plataforma robótica real e um ambiente de programação em blocos especializado para a programação de robôs. A adaptabilidade do LERO é obtida através do oferecimento de diferentes visões do sistema, onde cada visão apresenta níveis diferentes de abstração para as funcionalidades disponíveis. A extensibilidade do ambiente vem da capacidade de alunos mais experientes ou de educadores mais preparados expandirem o arcabouço de ações robóticas básicas através da criação de novas funcionalidades utilizando a metáfora de desenvolvimento e recursos disponibilizados no próprio ambiente LERO.

1.1 JUSTIFICATIVA

O presente trabalho foi motivado pela possibilidade educativa na inserção de crianças e adolescentes no mundo da robótica. Uma pesquisa foi feita para verificar soluções em ambiente WEB direcionadas à aprendizagem desse público alvo utilizando a robótica. Verificou-se que a escassez dessas soluções dificulta o acesso desse público a esse mundo tão motivador e lúdico para estimular o conhecimento. Além disso, os kits de robótica disponíveis no mercado são caros e as peças para montagem de robôs, que têm os preços mais acessíveis, necessitam de conhecimentos avançados para manipulação. Mesmo escolas que já possuem esses kits de robótica têm restrições de dias e horários limitados para o acesso dos estudantes a esses materiais.

Paralela à pesquisa acima, a autora vivenciava uma experiência em um projeto de pesquisa direcionado ao desenvolvimento de soluções inovadoras no campo das Tecnologias da Informação e Comunicação (TIC's) na Universidade Estadual do Sudoeste da Bahia (UESB). O referido projeto, denominado de Laboratório Remoto em Ambiente Virtual (LARA), visa o desenvolvimento de um laboratório remoto agregado a um ambiente virtual de aprendizagem, o Moodle. O LARA foca no ensino de computação em disciplinas como linguagem de programação, robótica, inteligência artificial, entre outras (SILVA et al 2014). O objetivo do projeto LARA contempla a concepção de uma metodologia pedagógica para a utilização de um laboratório remoto de robótica em um ambiente virtual de aprendizagem (AVA) destinado ao ensino dos alunos da graduação de Ciência da

Computação da UESB. Durante a execução dos trabalhos de pesquisa a autora, que ministrava aulas para alunos do quarto ano do ensino médio integrado em informática, identificou a possibilidade de desenvolver um laboratório remoto de robótica livre e gratuito na WEB para o acesso de diferentes faixas etárias. A ideia contemplava também uma interface de desenvolvimento direcionada ao público específico, permitindo uma linguagem adequada a esse público. O incentivo à interação desse público com o mundo da robótica e seu potencial educativo desde cedo pode despertar o interesse e vocação na área computacional ou áreas afins.

1.2 OBJETIVO

Considerando os problemas da escassez de soluções em ambiente WEB para o uso da robótica, além do custo envolvido na aquisição de kits de robótica e, caso tenha acesso na escola, existe a restrição de dias e horários para utilizá-los, este trabalho tem como objetivo geral o desenvolvimento de um laboratório remoto de robótica educacional, extensível e adaptável. Extensível por permitir ao usuário contribuir com o crescimento do laboratório submetendo novas funcionalidades à sua biblioteca. Adaptável por direcionar o usuário a um perfil compatível com sua faixa etária ou habilidade na área de conhecimento.

Para alcançar este objetivo, faz-se necessário:

- a) Adaptação de um ambiente de programação em blocos para codificação dos experimentos;
- b) Criação de um ambiente Web que permita o gerenciamento de usuários e experimentos;
- c) Criação de uma infraestrutura computacional que permita a integração do ambiente de programação com a plataforma robótica;
- d) Validação da usabilidade do sistema desenvolvido através de métricas de IHC;
- e) Garantia da extensibilidade do sistema através de recursos que permitam a inclusão de novas funcionalidades;
- f) Garantia da adaptabilidade do sistema através do oferecimento de diferentes visões do ambiente de programação.

1.3 METODOLOGIA

Inicialmente realizou-se uma revisão bibliográfica sobre robótica, em especial sobre robótica com finalidade educacional e voltada para o público alvo de 8 a 16 anos. Em um segundo estágio, as pesquisas foram direcionadas para identificação de trabalhos relacionados na área de laboratórios remotos de robótica, com interesse em especial para os laboratórios remotos voltados para o ensino de técnicas de programação para criança e adolescentes.

Uma vez identificado o estado da arte e os principais trabalhos relacionados, iniciou-se o processo de especificação e implementação do ambiente LERO. Este processo começou com o levantamento dos requisitos da aplicação. Logo após essa etapa foi realizado o processo de análise em que foi destacada a arquitetura das aplicações focando a usabilidade e confiabilidade. A etapa de implementação das aplicações do ambiente LERO envolveu conhecimentos na área de programação e, além disso, alguns conhecimentos específicos da área de eletrônica, dentre eles o controle de motor de corrente contínua usando ponte H e confecção de placas de circuito.

Objetivando o sistema desenvolvido quanto a sua usabilidade, foi aplicado um questionário para verificação das heurísticas de Jakob Nielsen. Por fim, foi realizada uma prova de conceitos para explorar na prática algumas funcionalidades criadas no projeto dessa dissertação.

1.4 CONTRIBUIÇÕES

A concepção e o desenvolvimento do ambiente LERO se caracterizam como as contribuições mais relevantes desse trabalho. O LERO é uma aplicação em ambiente WEB que tem como finalidade contribuir com o processo de ensino e aprendizagem de programação e robótica para diferentes faixas etárias, estimulando e diversificando a aquisição de conhecimentos de forma inovadora através de um sistema adaptável e extensível. São utilizados, nesse contexto, conceitos e metáforas de programação amplamente testadas e relatadas na literatura da área educacional.

O fato de oferecer um ambiente adaptável e extensível é uma característica saliente desse trabalho. O caráter adaptável do ambiente permite fornecer ao

usuário blocos com ações abstratas e compatíveis com o seu nível de conhecimento. O caráter extensível permite que novas funcionalidades possam ser adicionadas ao ambiente LERO, utilizando-se para isso o próprio ambiente LERO. Estas características mostram-se promissoras, pois além de permitir a agregação de novas funcionalidades ao ambiente, permitem que outros experimentos físicos possam ser explorados na plataforma robótica e disponibilizados em versões futuras da ferramenta.

1.5 AUDIÊNCIA

Esse trabalho tem como audiência um público diversificado, pois contempla profissionais das áreas de computação, robótica e educação. Dentre esses profissionais, destacam-se: pesquisadores interessados no desenvolvimento de laboratórios remotos de robótica; pesquisadores e educadores interessados no uso da robótica como ferramenta de auxílio no processo de ensino-aprendizado de crianças e adolescente, em geral, e no ensino de lógica de programação, em particular; ou qualquer outra pessoa interessada no uso sinérgico da robótica e computação aplicadas à educação.

1.6 APRESENTAÇÃO

O trabalho está organizado em cinco capítulos. O segundo capítulo aborda os conceitos relativos à área de robótica. Além disso, discorre sobre laboratórios remotos e ambientes de programação em blocos, assuntos que são a base na construção desta pesquisa.

O terceiro capítulo aborda os elementos determinantes na elaboração da arquitetura do ambiente LERO. Estes elementos foram utilizados no desenvolvimento da arquitetura para o ambiente proposto neste trabalho. A arquitetura é um fator determinante na qualidade de qualquer aplicação desenvolvida. Na estrutura da arquitetura, são apresentados também os componentes lógicos e físicos utilizados na construção desse trabalho.

O quarto capítulo apresenta os principais processos de desenvolvimento do ambiente LERO baseados na arquitetura apresentada no capítulo anterior.

O quinto capítulo explora o potencial do ambiente LERO através de um estudo de caso que serve ao mesmo tempo como prova de conceito e validação da ferramenta.

Por fim, o sexto capítulo apresenta as conclusões da pesquisa e as propostas para trabalhos futuros.

2 LABORATÓRIOS REMOTOS DE ROBÓTICA COM FINS EDUCACIONAIS

A robótica está em evidência e é uma realidade no contexto das mudanças tecnológicas da sociedade. A robótica é tida como a ciência multidisciplinar dos sistemas que se relaciona com o mundo real, com pouca ou até sem a intercessão do ser humano (MARTINS A., 2006). A robótica objetiva a realização de uma determinada tarefa no mundo real de uma forma independente e inteligente. Sendo assim, a sua utilização no cotidiano das pessoas pode trazer grandes benefícios, pois tarefas mecânicas que não contribuem na evolução cognitiva do ser humano podem ser executadas por robôs, permitindo assim que a humanidade direcione seus maiores esforços ao crescimento do seu intelecto.

Os robôs estão presentes na área da saúde, na realização de cirurgias à distância, em ambientes insalubres ou de difícil acesso como resgate em incêndios ou desativação de bombas atuando para fins militares. Além disso, seu uso vai desde os casos clássicos como os braços robóticos industriais até os robôs dotados de inteligência artificial que simulam o comportamento humano e animal através de mecanismos de inteligência artificial. No contexto educacional, a robótica vem sendo utilizada como ferramenta para motivar o ensino-aprendizagem em diversas instâncias acadêmicas. Dentro dessa diversidade de áreas de aplicações da robótica, ganha destaque o seu uso na área educacional, pois beneficia o processo de organização do pensamento, estimula o raciocínio lógico, propicia o desenvolvimento de atividades multidisciplinares, incentiva a capacidade do aluno de resolver problemas e motiva o trabalho em grupo.

Paralelas às vantagens da robótica educacional existem também dificuldades e desafios. Infelizmente, os kits de robótica mais conceituados são importados e não são acessíveis a grande parte das instituições de ensino. Os custos desses kits são elevados para a realidade das escolas do país, principalmente para as instituições públicas. Esse contexto fomenta o meio acadêmico nacional a desenvolver projetos na área. Alguns kits de robótica nacionais já são reconhecidos, mas estão sujeitos às limitações de *hardwares* e *softwares*. Além disso, quando a escola procura editais para financiamento desses recursos os processos burocráticos são extensos e desmotivadores.

Diante deste cenário, são procuradas soluções alternativas para tornar o ambiente de robótica educacional uma realidade nas escolas com poucos recursos.

Dentre as opções, o laboratório remoto de robótica educacional vem despontando como uma alternativa de grande valor, pois apresenta características que favorecem a superação de grande parte das dificuldades apontadas.

As próximas seções descrevem a robótica e seu uso no contexto educacional, além de conceitos, vantagens e desvantagens dos laboratórios remotos. Descrevem também projetos na área de laboratórios remotos aplicados na robótica educacional. Além disso, apresenta algumas linguagens de programação em blocos e soluções existentes para a conversão de programas em blocos para a plataforma robótica do Arduino.

2.1 ROBÓTICA NO CONTEXTO EDUCACIONAL

Em um mundo que a tecnologia é presença constante no dia-a-dia das pessoas, as discussões sobre a sua importância no ambiente escolar ganham evidência. Segundo Brito e Purificação (2006, p.23) é importante o envolvimento da educação na democratização do acesso ao conhecimento, à produção e interpretação das tecnologias. Esse envolvimento deve ser mediado pelos professores, que tem um papel fundamental no direcionamento de como as tecnologias devem atuar como facilitadoras do processo de ensino-aprendizagem. A combinação das tecnologias de informação abrem oportunidades sem precedentes para ações que visam melhorar a qualidade do ambiente de aprendizagem, pois produz variadas oportunidades para moldar a aprendizagem não somente na escola e no trabalho como também em atividades lúdicas (PAPERT, 1994).

A robótica em ambientes educacionais evidencia ser uma ferramenta apropriada em atividades que envolvam a criação, o projeto e o planejamento; favorece o processo de ensino-aprendizagem e amplia a relação entre diferentes áreas de conhecimento (BAGNALL, 2007). Os desafios propostos na solução de problemas com a utilização da robótica direciona o aluno a trabalhar com conceitos multidisciplinares e a vivenciar na prática o método científico, simulando mecanismos do cotidiano através de construção de projetos controlados por computador (DE SOUZA PIO; DE CASTRO; DE CASTRO JÚNIOR, 2006). Além disso, a robótica favorece o desenvolvimento do raciocínio lógico, permite o trabalho em grupo, exercita a autonomia, desenvolve a capacidade de resolver problemas

utilizando na prática a teoria e os ensinamentos apresentados em sala de aula e também estimula a criatividade de uma forma lúdica e motivadora.

A implantação em alguns países de um currículo mínimo de computação e robótica em suas escolas já é uma realidade. Em Israel, por exemplo, foi nomeado um comitê que montou um currículo de computação coerente com as suas escolas. Segundo França, tais currículos foram implantados em 1995 (FRANÇA; AMARAL, 2013). A Holanda e Alemanha possuem a robótica pedagógica em todas as suas escolas públicas e outros países da Europa e os Estados Unidos também caminham nessa direção (QUINTANILHA, 2008).

No Brasil, apesar de não fazer parte da formação do currículo mínimo escolar, as experiências com a robótica educacional estão crescendo. Segundo estudo de França e Amaral (2013, p. 426) apesar da implantação do ensino da computação no ensino básico no Brasil ainda ser incipiente, há regiões que se destacam em experiências com robôs no ambiente de aprendizagem. Seu estudo comprovou que entre 2009 e 2010 houve um aumento de interesse de pesquisadores brasileiros em relação ao ensino da computação na escola e esse interesse permaneceu constante nos dois anos seguintes. Dentre as áreas de conhecimentos possíveis de execução no ensino da computação, o estudo mostrou que o ambiente de robótica teve lugar de destaque, vez que, ele foi indicado ou sugerido em 20% dos casos. O estudo mostra também que as instituições de pesquisa do Nordeste se destacam em estudos sobre o ensino de computação na educação básica no Brasil.

Outro estudo exploratório (FABRÍCIO *et al.*, 2014) foi realizado com alunos do ensino médio de uma escola estadual do município de Solânea na Paraíba. O referido estudo objetivava avaliar a metodologia utilizada pelos professores antes e depois da introdução da robótica no processo de ensino e a resposta do aluno frente a esse cenário. O estudo concluiu que a utilização da robótica na escola mostrou-se eficiente trazendo melhoria ao aprendizado dos alunos que participaram de todo processo. Concluiu-se também, que as aulas se tornaram mais atrativas com aprendizagem mais efetiva nas disciplinas que utilizaram a tecnologia, além da disseminação do conhecimento pelos alunos despertando o interesse dos colegas.

Os kits de robótica educacional são fundamentais para que a escola insira esta tecnologia nos seus processos de ensino. Existe atualmente no mercado uma profusão desses kits configurados para diferentes realidades pedagógicas e financeiras. A grande maioria dos kits de robótica educacional, entretanto, é

importada e tem normalmente um custo elevado para os padrões nacionais (MIRANDA, *et al.*, 2010). Diante dessa situação muitos pesquisadores brasileiros têm trabalhado em soluções alternativas e de baixo custo que possam atender essa demanda das escolas. Várias dessas soluções obtiveram êxito e reconhecimento, tais como o RoboFácil (MIRANDA, *et al.*, 2010), Super Robby e Robokit (SILVA; BARRETO, 2011). Silva e Barreto (2011, p.8) fizeram uma análise comparativa entre vários kits de robótica educacionais, nacionais e importados, abordando vários critérios, dentre eles o custo dos mesmos. Foram encontrados preços variando entre R\$300,00 a R\$1.600,00. Dentro dessa realidade, adquirir esses kits em quantidade para manter um laboratório de robótica torna-se um desafio para as escolas, principalmente as públicas. Normalmente os modelos importados têm o custo mais elevado e os nacionais possuem limitações de *hardware* ou *software* para algumas aplicações (MIRANDA *et al.*, 2010). Até mesmo quando existem editais públicos para o fomento à instalação de infraestrutura computacional e robótica, os projetos esbarram em grandes dificuldades devido aos trâmites burocráticos (ZILLI; LAMBERT, 2010).

Uma possível solução para minimizar as dificuldades técnicas e os custos de implantação de uma infraestrutura robótica é a utilização de laboratórios remotos e virtuais. A próxima seção descreve o conceito de laboratório remoto, a diferença entre os laboratórios virtuais e remotos e a aplicabilidade dos laboratórios remotos na área de robótica educacional.

2.2 LABORATÓRIOS REMOTOS

O processo tradicional de aprendizagem passa por várias etapas. A aula teórica do professor normalmente é o primeiro contato do aluno com um determinado conhecimento. Em seguida, cabe ao estudante um papel muito importante nesse processo tradicional de aprendizagem que é o estudo complementar, seja em casa, em sala de aula ou em uma biblioteca. A partir do conhecimento teórico, o aluno começa a despertar dúvidas e curiosidades sobre o tema abordado. Nesse momento, os professores utilizam recursos para aulas práticas, como por exemplo, os laboratórios. Os laboratórios vêm para trazer ao ensino um ambiente positivo para conseguir do estudante uma participação ativa, interação e motivação para o

questionamento reconstrutivo na confluência entre teoria e prática aumentando assim a assimilação do conhecimento (SOUZA; COSTA FILHO, 2002).

Os tradicionais laboratórios *hands-on* são aqueles que estão fisicamente localizados na instituição de ensino e que possuem os equipamentos necessários para as atividades laboratoriais de determinada área do conhecimento. Normalmente, esses laboratórios são reservados pelos professores em determinados horários, dentro do funcionamento da instituição, e possuem uma capacidade limite para comportar uma certa quantidade de alunos. Nesse formato, os laboratórios *hands-on* trazem algumas limitações no seu uso como, o compromisso com o tempo e o espaço, e o financiamento para aquisição, instalação e manutenção de equipamentos (SIEVERS JUNIOR *et al.*, 2013).

As tecnologias de informação permitiram que novas propostas fossem elaboradas visando uma solução dessas limitações abordadas através dos laboratórios via internet. Esses laboratórios são estruturas que podem ser acessadas através da internet e que possuem o mesmo objetivo dos laboratórios tradicionais, ou seja, permitir o trabalho prático na aprendizagem. Essas estruturas são diferentes dependendo do tipo de laboratório disponibilizado através da internet, ou laboratórios on-line, podendo ser classificadas em laboratórios remotos e laboratórios virtuais.

A estrutura dos laboratórios virtuais é composta por usuários acessando um computador que disponibiliza um conjunto de experimentos com simulações, modelagens e visualizações que não existem no mundo físico. O laboratório virtual é usado para enriquecer e complementar a função do laboratório real em situações em que não podem ser replicadas, principalmente por apresentarem restrições, tais como: visualização de estruturas internas inacessíveis de equipamentos, situações que apresentam riscos a quem está fazendo o experimento. Além disso, o usuário pode simular acidentes nucleares ou situações que poderiam causar danos humanos e materiais sem oferecer qualquer risco a sua própria segurança ou da comunidade, ou mesmo danificar equipamentos caros. No contexto pedagógico, o consenso é que os laboratórios virtuais podem ser usados como uma etapa preparatória do estudante para realizar as atividades do laboratório real, substituindo a elaboração de trabalhos preparatórios pela realização de uma aula experimental virtual sobre o assunto que será estudado. Dessa forma, o aluno já vai para a aula

no laboratório real com a visualização prática dos conceitos teóricos que serão abordados (SOUZA; OLIVEIRA, 2001).

Um laboratório virtual interessante no contexto da robótica educacional é o projeto GrubiBots (OLIVEIRA *et al.*, 2014). O GrubiBots é um jogo digital que tem como objetivo resolver problemas das diversas áreas do conhecimento através de modelos aplicados à robótica. Este sistema possui uma interface (Figura 2.1) composta por dois ambientes, um ambiente de desenvolvimento e outro de execução

Figura 2.1 Ambientes do projeto GrubiBots



Fonte: Oliveira *et al.* (2014).

O ambiente de desenvolvimento possibilita a construção de algoritmos com uma programação em blocos gráficos (Figura 2.1a), permitindo que o usuário possa montar as ações que serão executadas pelo robô de uma forma sequencial. O ambiente possui também uma área de execução virtual, no qual o usuário visualiza as ações sendo executadas pelo robô (Figura 2.1b).

Os laboratórios remotos, por sua vez, são caracterizados por disponibilizar como estrutura uma interface WEB para que o usuário possa acessar um experimento real, localizado fisicamente em algum local previamente definido. O usuário acessa e controla um computador remoto do laboratório acionando equipamentos, testando condições e fazendo coleta de dados. Câmeras de vídeo são geralmente adicionadas ao ambiente do laboratório para que o usuário consiga ter uma visão em tempo real do que está acontecendo com o experimento. O alcance dessa tecnologia pode ir desde um pequeno experimento de passos sequenciais, como vários experimentos interligados em um ambiente colaborativo. Os laboratórios remotos permitem que os alunos possam ter acesso a experimentos

com materiais de qualidade. Além disso, a acessibilidade é uma forte característica desse formato, pois a restrição do espaço físico e temporal deixa de existir com esse ambiente disponível em horários fora do funcionamento padrão das instituições de ensino. No contexto pedagógico, esse formato de laboratório permite uma interação do aluno com os elementos físicos de uma forma personalizada. Os laboratórios remotos propiciam aos alunos um maior controle do processo de aprendizagem, preservam sua identidade e protegem sua autoconfiança, no caso em que eles se sintam menos capazes que os colegas. O laboratório remoto proporciona também, aos alunos mais interessados, oportunidades ilimitadas para investigar o experimento com maior profundidade, sem se sentir pressionado pelo tempo ou medo de cometer erros embaraçosos. Além disso, esses laboratórios permitem que sejam explorados conhecimentos que não façam parte da vida escolar do aluno e sim da sua curiosidade acadêmica, ampliando, dessa forma, seus conhecimentos interdisciplinares (SIEVERS JUNIOR, *et al.*, 2013).

Os laboratórios virtuais e remotos despertam críticas no meio acadêmico. Alguns autores sugerem que os laboratórios remotos são meramente replicações de um laboratório *hands-on* e que não agregam valor significativo no aprendizado do aluno. Segundo desenvolvedores experientes em sistemas de simulações o uso avançado de simuladores em laboratórios virtuais, é uma tarefa difícil, pois a criação de programas de computador que simulam um processo de forma realista é muito complexa e, conseqüentemente, o avanço dessas experiências é bastante lento (SIEVERS JUNIOR *et al.*, 2013).

De acordo com estudos comparativos entre os três modelos de laboratórios – *hands-on*, remotos e virtuais – os resultados apontam que os mesmos se complementam no contexto pedagógico. Existe um consenso de que um laboratório não substitui o outro, pois existem características significativas em cada um deles que somados podem favorecer uma maior aprendizagem para o aluno (TZAFESTAS *et al.*, 2006). Segundo Corter (2011, p. 2054) o que faz parecer crucial na eficácia da aprendizagem é o tempo total gasto nas atividades laboratoriais. Estudantes altamente motivados passam mais tempo analisando e fazendo seus relatórios e isso pode refletir diretamente em melhores resultados.

O presente projeto de pesquisa escolheu o laboratório remoto como estrutura porque são poucos os laboratórios desse tipo que fornece acesso físico para

manipuladores robotizados. Além disso, o projeto de pesquisa que inspirou a ideia para esse trabalho, o LARA, é um laboratório remoto de robótica disponibilizado através de um ambiente virtual de aprendizagem.

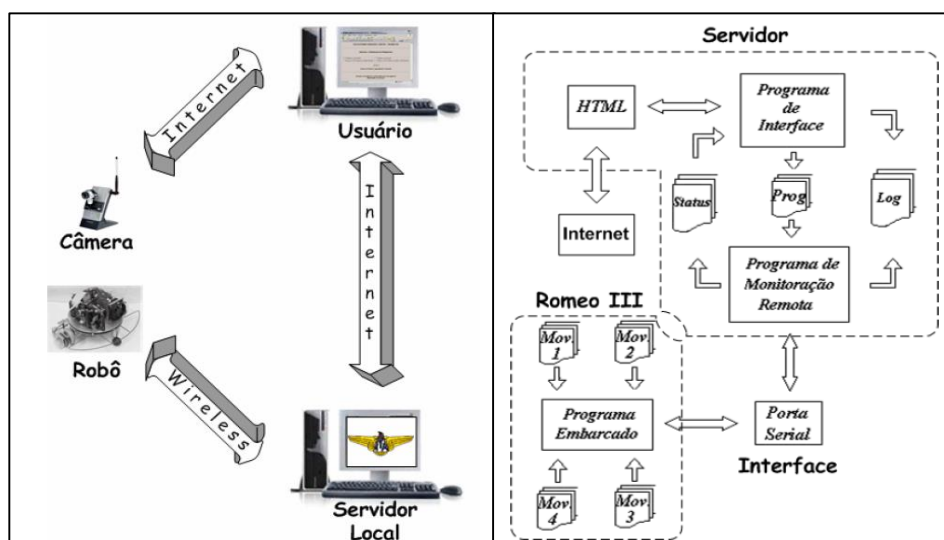
Na próxima seção, serão apresentados alguns laboratórios remotos de robótica que têm em comum o objetivo de estimular o seu público alvo a explorar a aprendizagem através da programação de robôs.

2.3 LABORATÓRIOS REMOTOS E A ROBÓTICA EDUCACIONAL

Os laboratórios remotos podem ser utilizados para promover uma interação com uma infraestrutura robótica localizada fisicamente distante visando a promoção da área educacional e de pesquisa (KHAMIS; RODRÍGUEZ; SALICHS, 2003). Na área educacional, diversos projetos têm explorado essa estrutura de ensino aprendizagem. Apresentamos nessa seção alguns dos principais projetos nesta área.

A pesquisa de (TEIXEIRA *et al.*, 2006) desenvolveu um sistema para o comando e monitoração remota via internet de um robô móvel chamado ROMEO III. A arquitetura do ambiente ROMEO é simples, composta pelo ambiente do usuário, do servidor e da plataforma robótica (Figura 2.2).

Figura 2.2 – Arquitetura do projeto ROMEO III

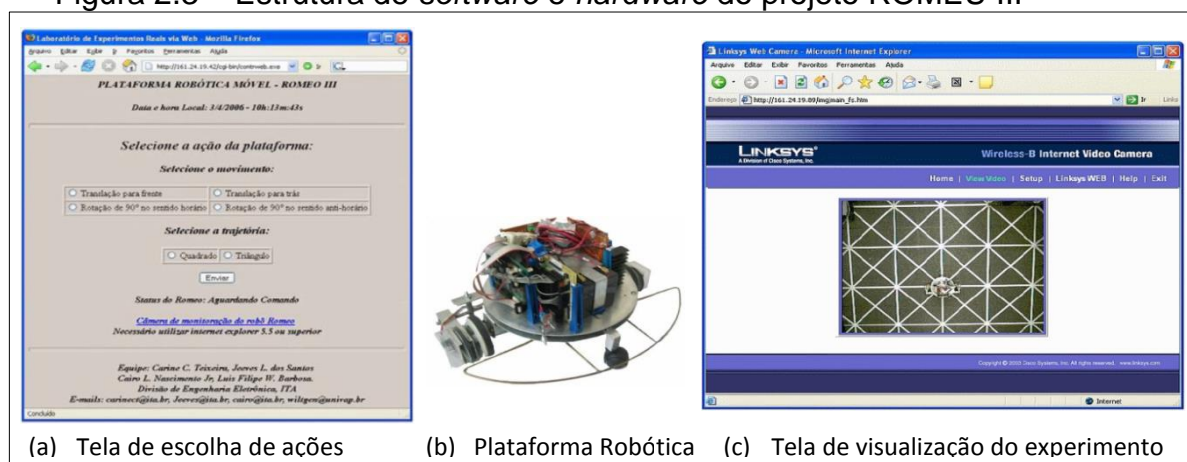


Fonte: Teixeira *et al.* (2006).

A arquitetura do sistema ROMEO não disponibiliza nenhum mecanismo de banco de dados para guardar informações. A interface é simplesmente uma página HTML para os usuários com algumas ações pré-definidas para controlar o robô. Desta forma, não existe a possibilidade do usuário contribuir com a evolução do ambiente propondo novas funcionalidades. O ambiente suporta somente um movimento de cada vez, sem poder criar comandos sequenciais das ações. Esta limitação restringe a criatividade do usuário e minimiza a potencialidade da plataforma robótica disponibilizada. O nível de conhecimento do usuário também não é explorado, pois não oferece cenários distintos que abordem diferentes complexidades.

A Figura 2.3.a mostra algumas ações previstas no ambiente ROMEO. Após selecionar a ação que o robô irá executar, o usuário poderá visualizar o movimento da plataforma robótica (Figura 2.3.b) em uma janela cujo vídeo foi capturado e transmitido por uma câmera IP instalada no local real do experimento (Figura 2.3.b).

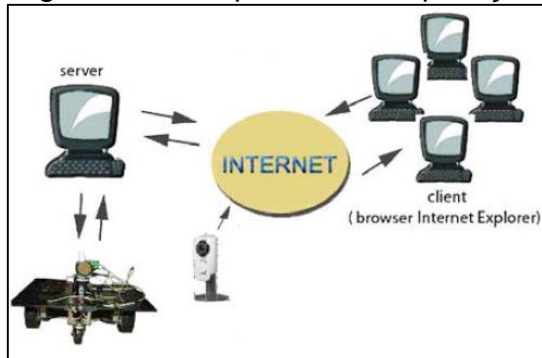
Figura 2.3 – Estrutura de *software* e *hardware* do projeto ROMEO III



Fonte: Teixeira *et al.*, (2006).

Já a pesquisa de Popescu (2007) desenvolveu aplicação telemática que representa um sistema real com acesso remoto, ou seja, o acesso a um experimento real através da internet representada (Figura 2.4).

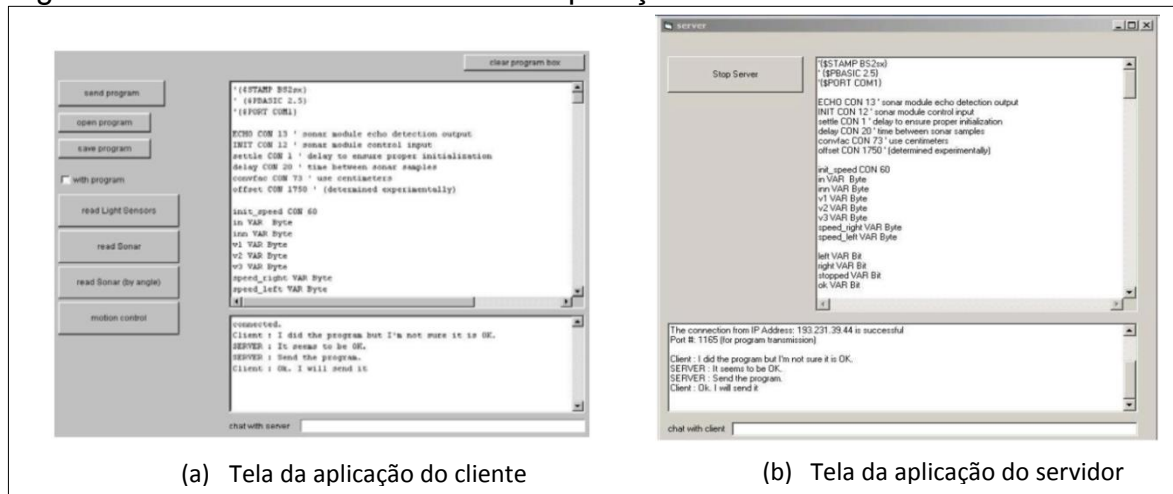
Figura 2.4 – Arquitetura da Aplicação Telemática



Fonte: Popescu *et al.* (2007).

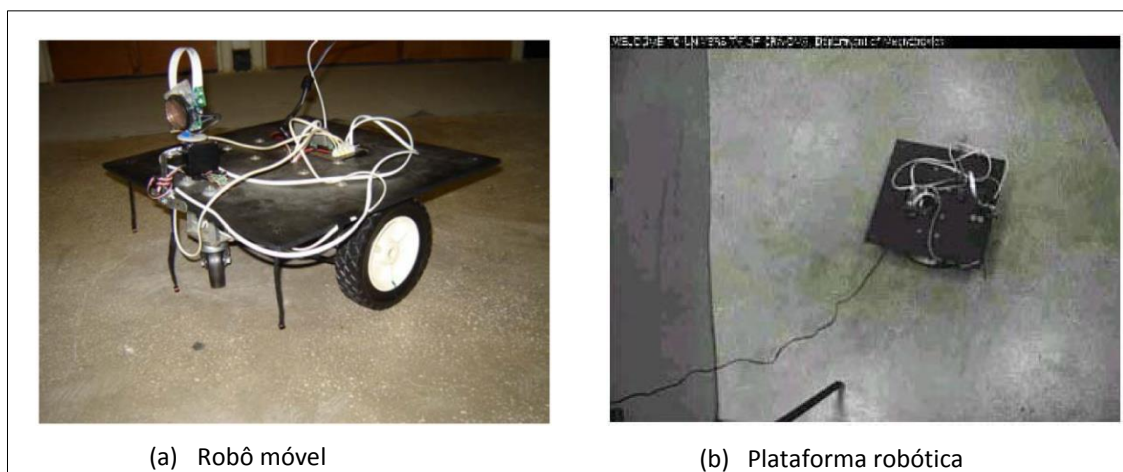
Como mostra a Figura 2.5a, uma interface *applet* Java é utilizada para manipular os dados dos comandos que serão enviados ao servidor do robô móvel. O aplicativo do servidor, Figura 2.5b, é responsável pelo módulo de administração da aplicação que envolve o controle de usuário e a comunicação com a placa de controle do robô.

Figura 2.5 – Estrutura de *software* da Aplicação Telemática



Fonte: Popescu *et al.* (2007).

O robô do experimento tem conexões por cabos com fonte de alimentação (Figura 2.6a). A Figura 2.6b mostra a plataforma robótica da aplicação.

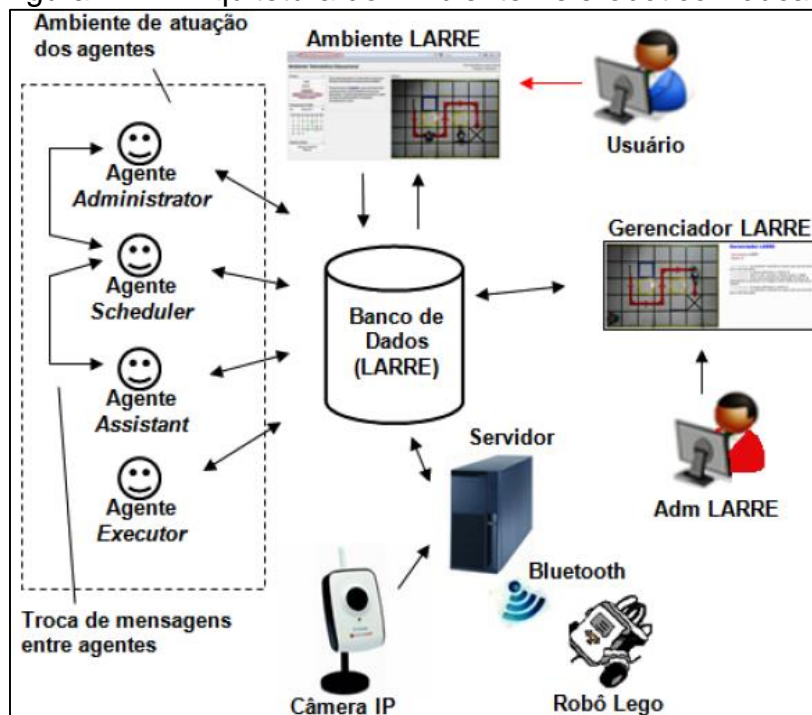
Figura 2.6 – Estrutura de *hardware* da Aplicação Telemática

Fonte: Popescu *et al.* (2007).

A arquitetura possui a mesma estrutura do projeto anterior, isto é, ambiente cliente, servidor e plataforma robótica. A interface o usuário utiliza um *browser* em que poderá programar diretamente os comandos através da linguagem de programação *Visual Basic*. A interface permite ainda utilizar um recurso de controle de movimentos pré-definidos, como ir pra frente, para trás, esquerda e direita. Essas duas formas de interação com o robô estão disponíveis na mesma interface, ou seja, não há um direcionamento de cenários específicos ao provável nível de conhecimento do usuário. Além disso, o ambiente também não permite ao usuário contribuir na construção do sistema, propondo novas funcionalidades, estimulando a criatividade e autonomia do usuário.

O Ambiente Telerobótico Educacional (ATE) é um projeto de pesquisa (SOUZA et al., 2011) que amplia as oportunidades de acesso a laboratório de robótica. Seu objetivo é estimular o interesse dos estudantes de escolas públicas com computadores no desenvolvimento de sistemas automatizados utilizando a robótica educacional como elemento motivador da aprendizagem (SOUZA et al., 2011). Como apresentado na Figura 2.7 a arquitetura desse ambiente é composta de recursos que proporcionam a comunicação entre o usuário, o robô e os agentes que objetivam ajudar o usuário no desenvolvimento de suas atividades na manipulação do robô.

Figura 2.7 – Arquitetura do Ambiente Telerobótico Educacional



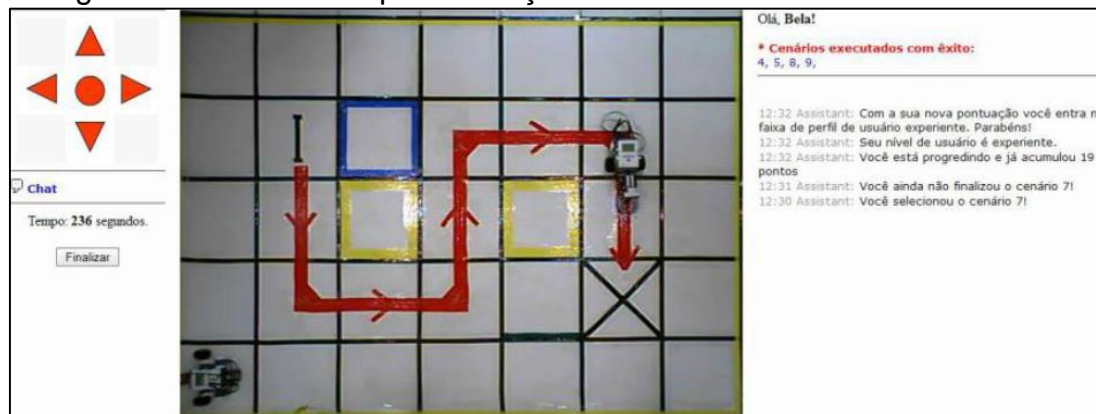
Fonte: Souza *et al.*, (2011).

No Ambiente Telerobótico Educacional o usuário acessa uma sala de experimentos (Figura 2.8), através do ambiente LARRE (laboratório de acesso remoto de robótica educacional). No ambiente LARRE o usuário pode executar ações através de botões que permitem manipular o robô para frente, para trás, para direita e para a esquerda e ainda visualizar o robô *Lego Mindstorms NXT* seguindo as ações. Essas ações dos alunos são gerenciadas e armazenadas em uma base de dados pelo sistema multiagente que ajuda a traçar o perfil de cada um para estimar o seu grau de aprendizagem. Um *link* para *chat* interativo também é disponibilizado para comunicação entre usuários.

Na sua arquitetura, além dos componentes comuns a um laboratório remoto, existe um sistema de multiagente que permite que à medida que o estudante interage com o ambiente, mensagens de seu desempenho vão sendo mostradas e sugestões para melhorá-lo vão sendo apresentadas na tela do sistema. Apesar de se preocupar com a progressão do grau de conhecimento do público alvo, o ambiente ATE não disponibiliza cenários distintos de interfaces com diferentes níveis de complexidade em seu ambiente. Sua interface só permite manipular ações pré-definidas do robô, não oferecendo a possibilidade de montar ações sequenciais para

a sua manipulação. O ATE também não permite ao usuário contribuir na construção do ambiente, propondo novas funcionalidades para explorar a potencialidade dos recursos disponíveis estimulando o conhecimento e a criatividade do público alvo.

Figura 2.8 – Sala de experimentação do ATE

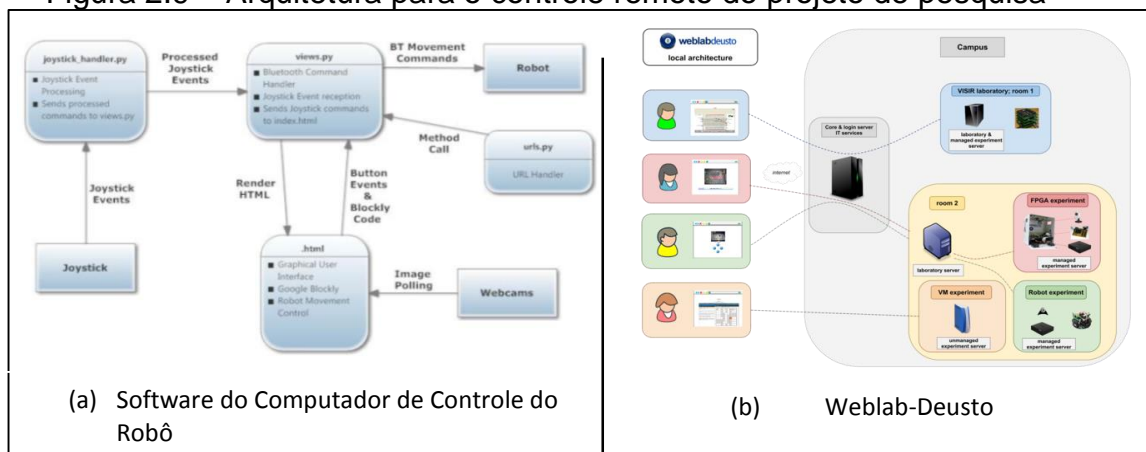


Fonte: Souza, *et al.* (2011).

Outro projeto de pesquisa que merece destaque é um laboratório remoto que permite que o usuário controle um robô móvel em um labirinto e ainda participe de um jogo interativo. Sua arquitetura foi dividida em três principais componentes (ITURRATE *et al.*, 2013):

- Microbot corresponde à parte física do experimento onde existe um robô situado em um labirinto;
- Computador de Controle (Figura 2.9a) é responsável pela comunicação direta com o robô via *Bluetooth*;
- Servidor Weblab-Deusto (Figura 2.9b), que liga o computador de controle ao framework Weblab-Deusto. O servidor do Weblab-Deusto cuida da administração dos acessos do usuário ao laboratório (autenticação, segurança, entre outros).

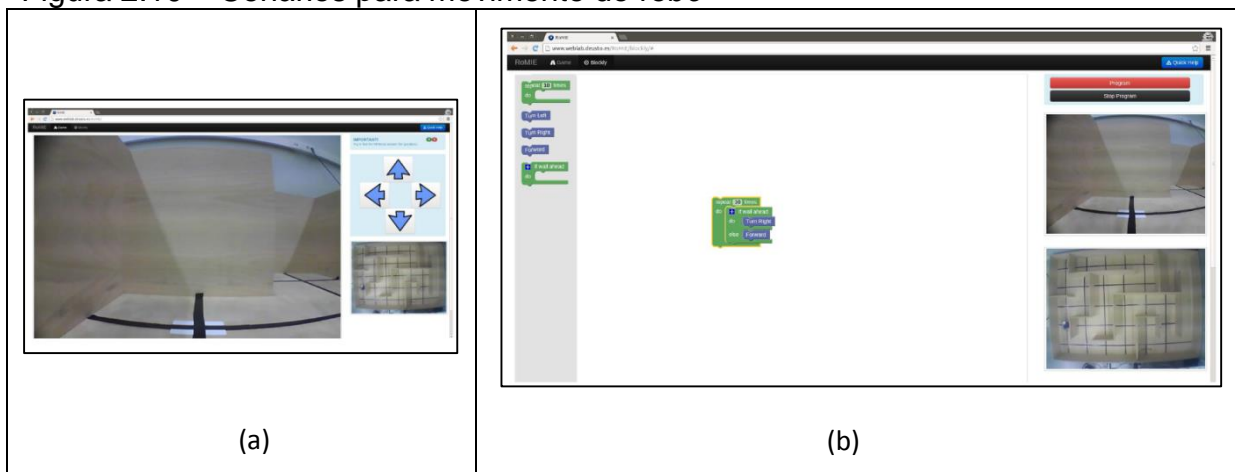
Figura 2.9 – Arquitetura para o controle remoto do projeto de pesquisa



Fonte: Iturrate *et al.* (2013).

Este ambiente fornece dois cenários para os jogadores manipularem o experimento físico. Em um cenário o robô pode ser manipulado diretamente no labirinto através de setas que indicam qual direção seguir (Figura 2.10a). No outro cenário o robô pode ser manipulado através de uma linguagem de programação gráfica, compondo um programa para o robô navegar de forma autônoma (Figura 2.10b).

Figura 2.10 – Cenários para movimento do robô



Fonte: Iturrate *et al.* (2013).

Um diferencial na arquitetura desse projeto é que ela permite o acesso a várias salas de experimentos, aumentando as possibilidades pedagógicas do ambiente. Além disso, sua interface oferece dois cenários para o usuário. Um cenário que possui ações pré-definidas e outro que permite que o usuário trabalhe

com um ambiente de blocos. Desta forma, o ambiente pode ser usado na construção de ações sequenciais visando manipular o experimento físico robótico. Essa abordagem favorece, dentre outros aspectos, a criatividade e a resolução de problemas, incentivando a motivação do público alvo na construção do seu conhecimento. A ideia de um jogo interativo com desafios na plataforma robótica também favorece um ambiente estimulador de aprendizagem. Por essas características o ambiente possui um grande potencial a ser explorado e a própria pesquisa aponta caminhos para isso.

Embora o ambiente desenvolvido por Iturrate (2013) possa ser considerado o mais completo entre os trabalhos discutidos, os próprios autores reconhecem algumas limitações. Eles indicam que vários blocos poderiam, mas não foram criados no ambiente. Além disso, também apontam que não foi criada uma estrutura para separar níveis de dificuldade do jogo: uma com blocos mais gerais e simples e outra com blocos mais específicos e complexos. A possibilidade do próprio usuário acrescentar à interface novas funcionalidades e estender o ambiente não foi considerada.

Os laboratórios remotos de robótica mostrados nesta seção foram motivados pela área educacional. O ambiente LERO se inspirou nesses projetos para agregar funcionalidades básicas já existentes nesse contexto com novas funcionalidades não encontradas e que potencializam a proposta pedagógica desses ambientes. As suas premissas básicas são a adaptabilidade, através dos ambientes de programação em blocos LERO Criança, LERO Adolescente e LERO Extremo que proporcionam diferentes interfaces para diferentes níveis de conhecimento e através da extensibilidade, que permite que o usuário contribua com novas funcionalidades na biblioteca do LERO utilizando o próprio ambiente.

A próxima seção aborda os conceitos de linguagens e ambientes de programação em blocos, que foram utilizadas no projeto LERO. Apresenta o *Snap!*, que foi a interface de desenvolvimento escolhida e adaptada para esse projeto de pesquisa e mostra seu potencial para aplicabilidade na área da robótica educacional.

2.4 LINGUAGENS DE PROGRAMAÇÃO EM BLOCOS

Um dos objetivos desse trabalho é a criação de um ambiente intuitivo, que favoreça e motive a aprendizagem dos conceitos de programação de crianças e adolescentes através de atividades lúdicas. A utilização de linguagens de programação em blocos motiva o estudante a aprender programação de forma criativa e sistemática, sem se preocupar com a sintaxe da programação. Como a intenção é incorporar uma linguagem de programação em bloco como ferramenta de desenvolvimento do laboratório remoto de robótica, apresentamos aqui uma discussão mais detalhada dessa tecnologia através da análise de algumas linguagens de programação em blocos direcionadas ao conceito de construção do conhecimento.

Seymour Papert (1986) é conhecido no mundo como um dos principais pensadores e defensores do uso da tecnologia em processos educacionais. O autor vê a aprendizagem como uma ação de exploração, criação, construção e desenvolvimento do conhecimento na interação com outras pessoas e com o mundo, por isso defende a tecnologia como parte integrante e importante no processo de ensino e aprendizagem (PAPERT, 1986). Visando trabalhar com esses conceitos, Papert liderou um grupo de pesquisadores que criaram no *Massachusetts Institute of Technology de Cambridge* (MIT) a linguagem de programação LOGO. Essa linguagem visa explorar o processo de ensino-aprendizagem e também no contexto em que a criança possa ser vista também como construtora do seu próprio conhecimento.

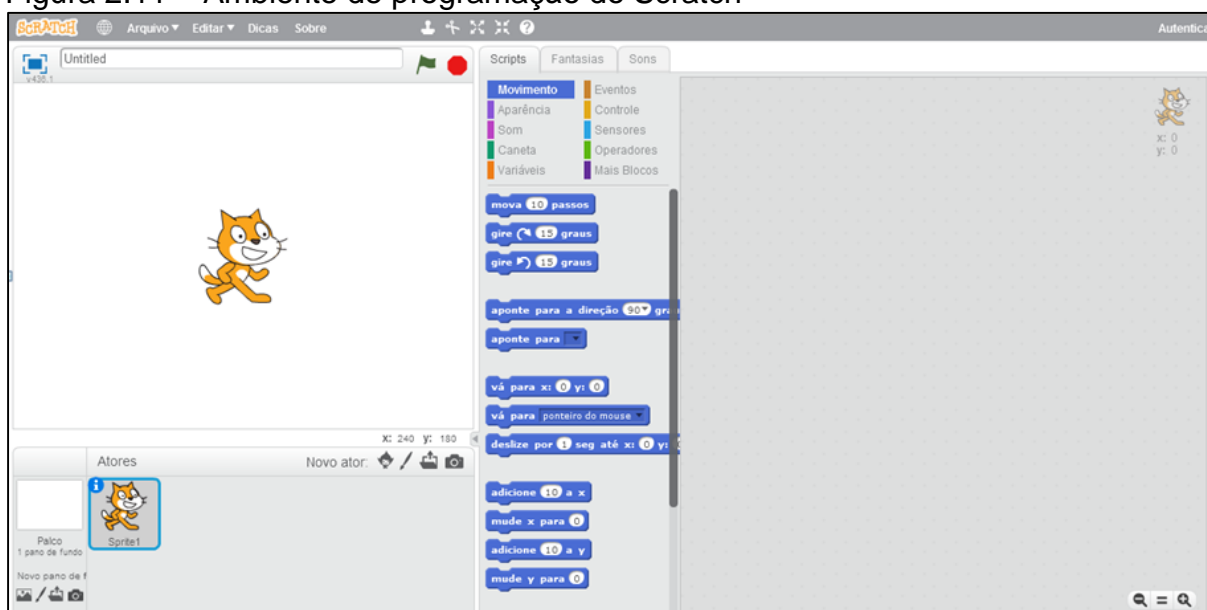
O LOGO tem duas vertentes: a computacional e a pedagógica. A computacional valoriza características que tornem a linguagem de programação fácil de assimilar, explorando termos usados no cotidiano e possibilitando que novos termos ou procedimentos sejam criados. Já a pedagógica valoriza o processo de aprendizagem focado na resolução de problemas. Sendo assim, o LOGO explora todo o ciclo de aprendizagem – descrever, executar, refletir, depurar e descrever – na interação do aprendiz com o computador (FRIEDRICH *et al.*, 2012).

Em 1998, aconteceu uma parceria significativa para a educação tecnológica, principalmente para a robótica educacional, que foi uma adaptação da linguagem LOGO para o LEGO, chamado LEGO Mindstorms (LEGO, 2015). Com essa parceria os elementos do LEGO passaram a ser controlados por um processador

programável. Esse projeto foi inspirado no trabalho de Papert realizado na década 1980, onde ele já apontava como os computadores iriam auxiliar o desenvolvimento intelectual de crianças e jovens.

Uma nova roupagem para o LOGO foi desenvolvida pelo MIT em 2006, chamada Scratch. Esta nova versão permite o desenvolvimento de programas interativos simples, como desenhos animados, jogos, dentre outros. O Scratch é uma linguagem de programação em blocos que oferece um ambiente de aprendizado (Figura 2.11) para diferentes públicos-alvo e utiliza a estratégia de arrastar e soltar blocos de códigos prontos e juntá-los em um conjunto para execução (MAIANI, 2013).

Figura 2.11 – Ambiente de programação do Scratch

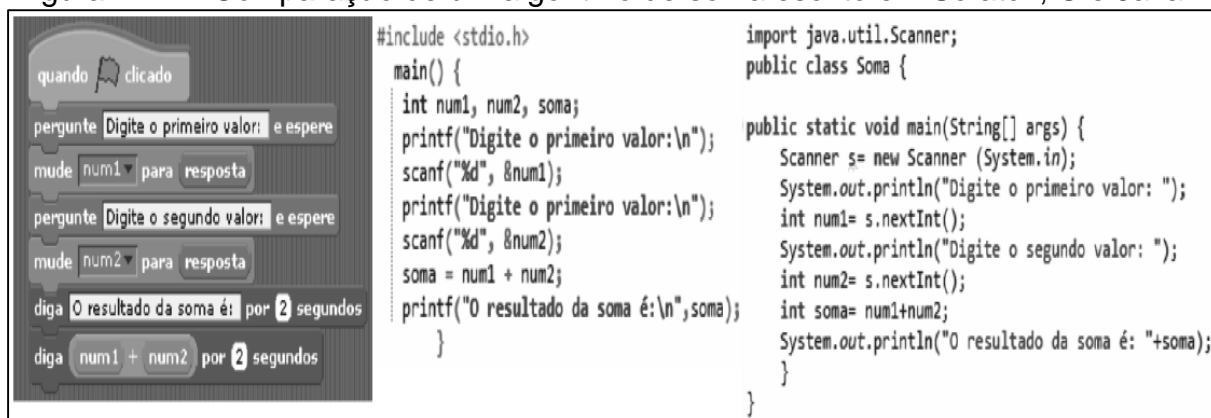


Fonte: Scratch (2015).

Segundo Oliveira² (2014, p.1528) é perceptível o potencial lúdico da ferramenta, pois ela proporciona uma linguagem de programação de alto nível de abstração. O ambiente Scratch foca na concentração, no raciocínio lógico e criatividade do usuário. A programação baseada em blocos representam as estruturas básicas de um algoritmo e se empilham visando a formação de um código executável. Os encaixes dos blocos de construção minimizam possíveis erros de sintaxe, frequentemente encontrados na programação com outras linguagens como C e Java. O elevado nível de abstração de um código utilizando programação em blocos quando comparado com o mesmo código escrito em linguagens de programação tradicionais poder ser observado na Figura 2.12, que mostra uma

mesma funcionalidade escrita nas três linguagens de programação com Scratch, C e Java, respectivamente.

Figura 2.12 – Comparação de um algoritmo de soma escrito em Scratch, C e Java



Fonte: Oliveira² et al. (2014).

O Scratch foi ambiente escolhido inicialmente para ser utilizado no laboratório remoto de robótica LERO. Além de amplamente utilizado no meio acadêmico, o Scratch é um ambiente lúdico, simples, intuitivo e destinado a um público a partir de oito anos. Apesar de muito utilizado no nível básico e fundamental, existem experiências com o uso da ferramenta no ensino de disciplinas de programação para graduandos em computação e também para cursos tecnológicos, como no Instituto Federal de Santa Catarina (IFSC), campus São José (MARTINS, 2012).

A utilização do ambiente Scratch apresentou alguns desafios e dificuldades técnicas que motivaram a utilização de um ambiente alternativo. O primeiro desafio técnico foi o fato do ambiente do Scratch ser desenvolvido em *Squeak*, linguagem visual pouco conhecida e documentada. O segundo desafio técnico relacionado à utilização do ambiente Scratch no LERO é que versão para WEB do Scratch 2.0 (*Editor e Player*), disponibilizada no site do GitHub (GITHUB, 2015), não possuía funcionalidade de criar e editar novos blocos no ambiente. Esta funcionalidade só está disponível atualmente para um grupo restrito de pesquisadores e a disponibilização pública desse recurso ainda não possui data para ocorrer. Como uma das características principais do LERO é permitir que o próprio usuário possa criar e editar o seu bloco, o Scratch foi desconsiderado para ser o ambiente de desenvolvimento do projeto.

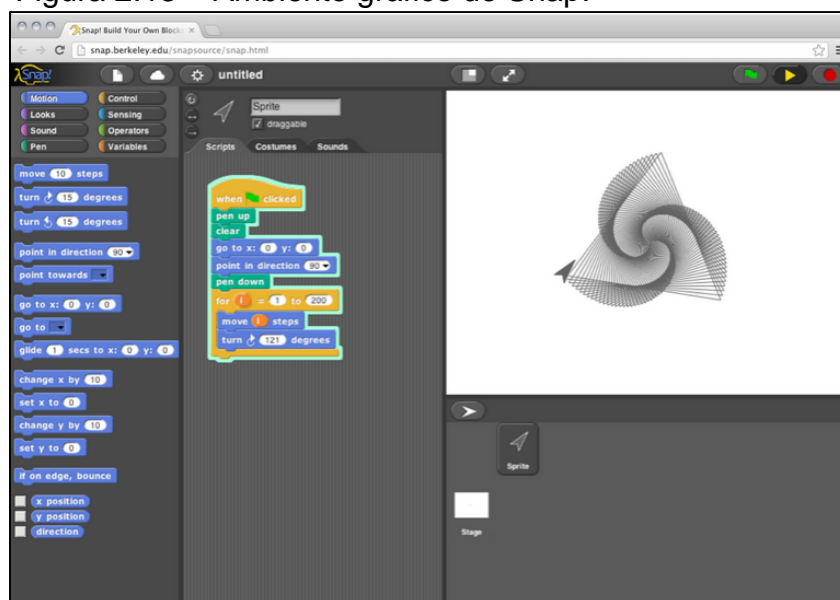
Diante dos problemas para utilização do Scratch, a linguagem de programação em blocos Snap! (SNAP, 2015) passou a ser considerada uma alternativa viável para se tornar o ambiente de desenvolvimento do projeto.

Para apresentar o Snap! é necessário falar primeiramente do BYOB (*Build Your Own Blocks*). O BYOB é uma extensão do Scratch, que incorpora poderosos recursos de programação, tais como considerar blocos como uma classe tipo de dados e permitir blocos de construção recursiva.

Um dos recursos mais importantes incorporados pelo BYOB e decisivo para esse projeto de pesquisa é a possibilidade de criação de blocos de programação personalizados com novas funcionalidades. Estes novos blocos podem ser usados, modificados e excluídos do ambiente facilmente (MAIANI, 2013). Essa funcionalidade é de fundamental importância para o projeto LERO, pois viabiliza a implantação das características de adaptabilidade e extensibilidade do ambiente.

O BYOB teve várias versões e quando passou da versão 3.1 para a 4.0 ela mudou de nome, passando a se chamar Snap! (Figura 2.13). O Snap!, diferente dos seus predecessores que foram escritos em Squeak, foi escrito em JavaScript, uma linguagem que possui farta documentação e facilidade de se encontrar programadores experientes para o desenvolvimento de aplicações.

Figura 2.13 – Ambiente gráfico do Snap!



Fonte: SNAP (2015).

As vantagens do uso de um ambiente de programação em blocos para o ensino de computação já foi evidenciada em diversos projetos de pesquisa. Estas evidências nos dão segurança de que o referido ambiente é o mais adequado para ser incorporado ao LERO. No nosso contexto, um problema que surge é a conexão

dos ambientes de desenvolvimento com a plataforma robótica. A próxima seção apresenta as soluções encontradas para fazer a conversão do código dos ambientes de programação em bloco para a plataforma robótica do Arduino.

2.5 LINGUAGENS DE PROGRAMAÇÃO EM BLOCOS E A ROBÓTICA

A integração dos ambientes de programação com a plataforma robótica é extremamente dependente de ambos os sistemas. Por um lado é preciso saber a linguagem de programação adotada e como converter comandos de alto nível expressos nessa linguagem em comando de baixo nível das plataformas robóticas. Neste trabalho, optou-se pelo ambiente de desenvolvimento em blocos do Snap!, conforme argumentações apresentadas anteriormente. Como plataforma robótica a escolha foi pelo Arduino.

O Arduino é uma plataforma de prototipação eletrônica livre que usa o microcontrolador ATmega 328. Para manipulação do dispositivo eletrônico existe um software que controla os pinos e ações do Arduino, chamado Arduino IDE, com uma interface simples e intuitiva (SMIDT, 2013). A escolha do Arduino para esse projeto foi motivada por ser uma tecnologia livre e de custo acessível para possíveis replicações desse trabalho.

Antes mesmo de definirmos qual o ambiente de programação em blocos utilizar, foi necessário verificar as soluções existentes de conversão dos programas em blocos para a plataforma robótica do Arduino. Esta análise seria fundamental até na definição do próprio ambiente de desenvolvimento. Dentre as opções encontradas duas foram selecionadas para serem analisadas em profundidade: o Scratch for Arduino (S4A, 2015) e o Snap!/Scratch to Arduino (S2A, 2015).

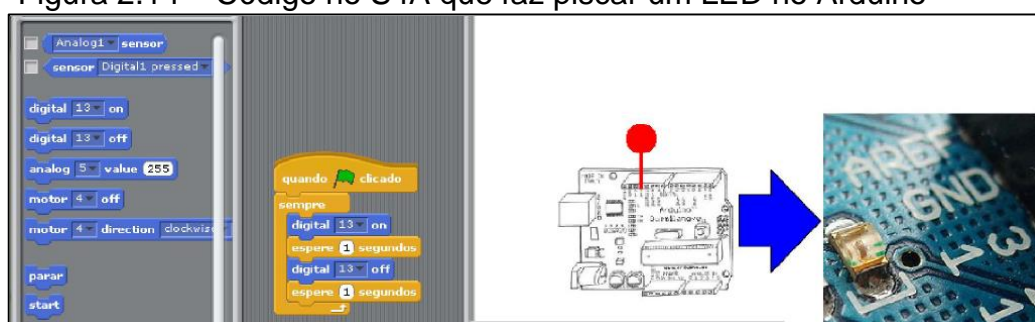
O S4A é uma modificação do Scratch desenvolvida pelo Citilab (CITILAB, 2015), onde é possível realizar a programação do Arduino. O S4A objetiva atrair usuários para o mundo da programação e também fornecer uma interface de alto nível para programadores de Arduino. O S4A oferece blocos tanto para as funcionalidades básicas do Arduino: saídas e entradas digitais e analógicas; quanto blocos para gerenciar servomotores ou de rotação contínua (S4A, 2015). Para que isso aconteça, a placa do Arduino deve estar conectada ao computador que tenha o

S4A instalado. Além disso, o Arduino deverá ser configurado com um programa especial para reconhecer os comandos do S4A.

O S4A é retrocompatível com os projetos feitos no Scratch, ou seja, podem ser lidos no S4A. Essa compatibilidade não acontece com os projetos feitos no S4A em relação ao Scratch (S4A, 2015).

A Figura 2.14 mostra o exemplo de um código que faz piscar um LED conectado ao pino 13 da placa de Arduino (MÉLO *et al.*, 2011).

Figura 2.14 – Código no S4A que faz piscar um LED no Arduino



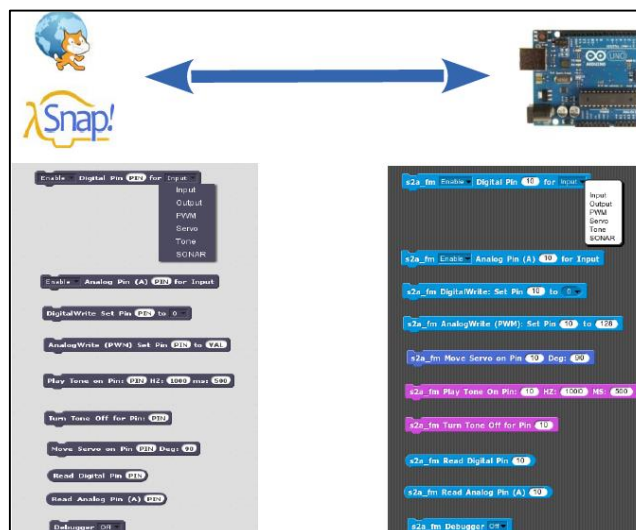
Fonte: Mélo *et al.* (2011).

A análise do S4A determinou a não utilização desse ambiente na interface de desenvolvimento do nosso sistema. Além do S4A ter sido desenvolvido, assim como o Scratch, em Squeak uma linguagem visual pouco conhecida e documentada, o ambiente do S4A é concebido como uma aplicação desktop, sem adaptação para Web, logo inadequada ao uso em um laboratório remoto. Além disso, também não possuía a funcionalidade de criar e editar novos blocos no seu ambiente, ou seja, seus blocos não são extensíveis. Também o S4A apresenta em seus recursos de interface elementos complexos como sensores e atuadores, que precisam de orientações prévias para serem utilizados, podendo limitar o usuário na construção gradativa do seu conhecimento de forma autônoma.

O S2A (S2A, 2015) é uma extensão de hardware que foi desenvolvida por Alan Yorinks (YORINKS, 2014) para ser utilizada de forma integrada aos ambientes de programação em blocos do Snap! e Scratch. A biblioteca do S2A foi elaborada em Python e integra blocos nesses ambientes gráficos para manipular de forma direta a placa Arduino. Sendo assim, o S2A não é um novo ambiente de programação como o S4A e sim uma extensão para os ambientes de programação em blocos citados anteriormente. Com isso, todas as funcionalidades da interface

Snap! / Scratch são preservadas e ainda adicionados mais blocos do S2A (Figura 2.15).

Figura 2.15 – Blocos adicionais do S2A para Scratch e Snap!



Fonte: Yorinks (2014).

Diante do exposto, o S2A é uma solução mais apropriada para integrar o ambiente de programação em blocos Snap! com a placa microcontroladora Arduino. O fato de realizar essa integração e ainda preservar as funcionalidades do Snap! foi fundamental na escolha do S2A para integrar o ambiente LERO.

2.6 CONSIDERAÇÕES FINAIS

Nesse capítulo foram discutidos os benefícios e desafios na inserção da robótica no contexto educacional. Dentre os benefícios estão o processo de organização do pensamento, o estímulo ao raciocínio lógico, o desenvolvimento de atividades multidisciplinares, o incentivo para resolver problemas e o trabalho em grupo. Já dentre os desafios uma das questões abordadas foi o custo envolvido em adquirir kits de robótica, mesmo os nacionais, além das questões burocráticas em adquiri-los.

Com a procura de soluções alternativas para tornar o ambiente de robótica educacional uma realidade nas escolas, o laboratório remoto de robótica educacional foi destaque nesse trabalho. Seus benefícios acadêmicos, dentre as

modalidades de laboratório foram apresentados, evidenciando o fato de acessar através da Internet experimentos físicos reais para serem manipulados sem limitação de tempo nem de espaço. A pouca disponibilidade dessas ambientes para manipular robôs, principalmente para o público-alvo da pesquisa, foi determinante na escolha desse tipo de laboratório para o projeto.

Os laboratórios remotos de robótica educacional analisados nessa pesquisa mostraram ambientes com estruturas diferentes e interface de desenvolvimento com variados níveis de abstração, mas todos objetivando a aprendizagem através do lúdico e da criatividade. Os ambientes de programação em blocos promovem uma interface intuitiva, que favorece e motiva a aprendizagem através do lúdico.

Por fim, foram apresentadas as questões técnicas que culminaram na escolha do Snap! como IDE para o ambiente proposto. Antes de apresentar o desenvolvimento desse ambiente, serão exibidos no próximo capítulo característica e a estrutura da arquitetura definida do laboratório remoto de robótica educacional proposto.

3 A ARQUITETURA DO AMBIENTE LERO

Um laboratório remoto de robótica permite que os usuários tenham acesso a experimentos e equipamentos reais através da Internet. O ambiente LERO é um laboratório remoto voltado para a robótica educacional que utiliza a metáfora de programação em blocos como base de um sistema que tem como funcionalidades inovadoras, ser extensível e adaptável ao nível de conhecimento e experiência do usuário.

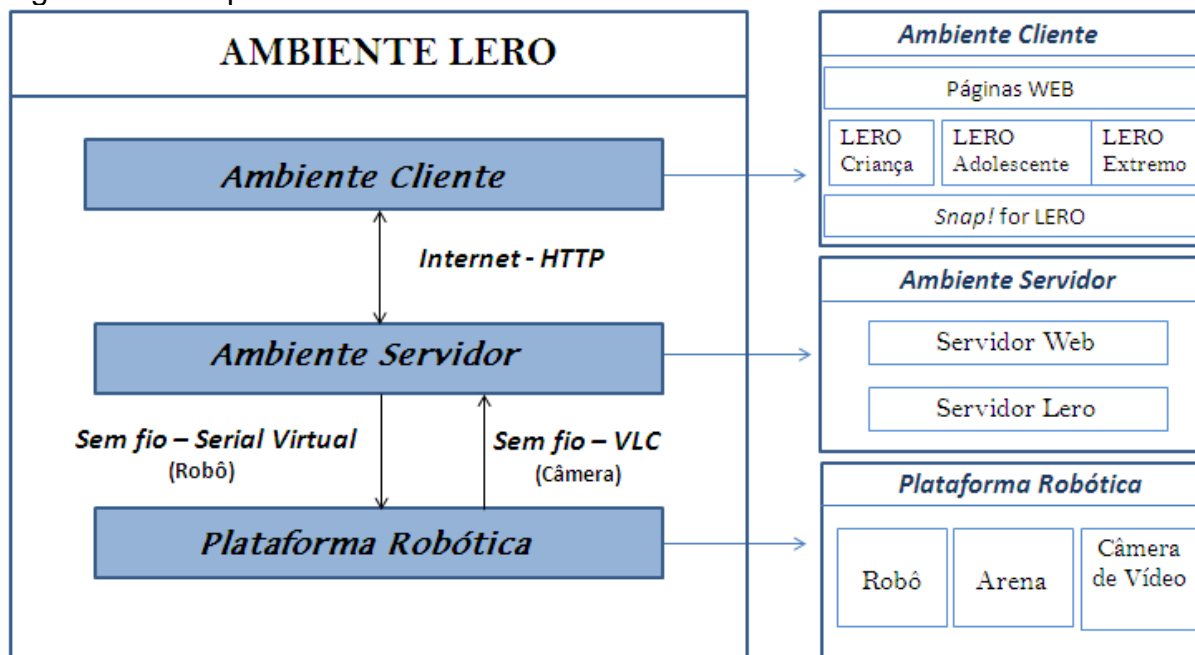
A funcionalidade de ser extensível caracteriza a capacidade que o sistema possui de permitir que novos blocos de construção sejam incorporados. Blocos com complexidade crescente são desenvolvidos e incorporados à ferramenta através da utilização do próprio ambiente de desenvolvimento LERO. Já a funcionalidade adaptável, ilustra a característica de atender a um público com nível de conhecimento e experiência variada, tanto no que se refere à lógica de programação, quanto aos conhecimentos em robótica.

A arquitetura do sistema LERO foi dividida em três componentes: o ambiente cliente, o ambiente servidor e a plataforma robótica, conforme a Figura 3.1.

No nível mais próximo ao usuário se encontra o ambiente cliente. O ambiente cliente é composto por uma aplicação Web. Desta forma, o cliente LERO é naturalmente multiplataforma, podendo ser executado a partir de qualquer navegador, independente do sistema operacional. Após o login, o usuário é direcionado ao ambiente LERO específico à sua faixa etária ou nível de conhecimento. Nesse contexto, são três as possibilidades de ambientes: LERO Criança, LERO Adolescente e LERO Extremo.

O ambiente servidor hospeda o servidor WEB que gerencia a base de dados de usuários, controla o acesso simultâneo e o servidor LERO que traduz a programação realizada pelo cliente para instruções de baixo nível compatíveis com a plataforma robótica. A conversão da programação em blocos para instruções compatíveis com um robô Arduino (Yorinks, 2014) é feita através de uma extensão escrita em Python chamada *Snap! to Arduino* (S2a, 2015). Com a extensão s2a é possível, além de converter as instruções *Snap!*, enviar as ações a serem executadas para o robô através de uma conexão sem fio.

Figura 3.1 – Arquitetura do Ambiente LERO



Fonte: Autoria própria (2015).

A plataforma robótica da arquitetura LERO é composta por um robô, uma câmera de vídeo e uma arena para movimentação do robô. A câmera de vídeo é utilizada para observar a movimentação do robô. A imagem capturada pela câmera é transmitida ao servidor Web e este faz a retransmissão das imagens ao cliente. Detalhamos a seguir os principais componentes do ambiente LERO.

3.1 O AMBIENTE CLIENTE

O ambiente cliente é composto pelas páginas WEB, os três perfis do ambiente – LERO Criança, LERO Adolescente e LERO Extremo – e pelo *Snap! for LERO*, conforme ilustrado na Figura 3.2.

As páginas WEB oferecem a estrutura necessária para o gerenciamento do acesso ao laboratório remoto. Essas estruturas são necessárias para que o usuário seja direcionado, a partir do seu perfil ou habilidade, a uma determinada visão dos recursos e funcionalidades disponibilizadas no sistema.

A visão LERO Criança, desenvolvida para usuários na faixa etária de 8 a 12 anos, possui blocos de construção mais abstratos, com funcionalidades próximas a

comandos ou verbos usuais para esta faixa etária. A visão LERO Adolescente, feita para usuários na faixa etária de 13 a 17 anos, permite trabalhar princípios mais avançados de lógica de programação. As estruturas de controle de fluxo, variáveis e funções estão presentes no ambiente adolescente e podem ser utilizadas para programar comportamentos mais elaborados para o robô disponibilizado no experimento.

Figura 3.2 – Componentes do Ambiente Cliente



Fonte: Autoria própria (2015).

Já a visão LERO Extremo é um ambiente direcionado ao público com idade acima de 18 anos, normalmente estudantes de graduação ou pós-graduação. Esse perfil de usuários tem uma maior experiência em programação e pode explorar o ambiente com ações mais complexas da biblioteca LERO. Nessa visão, o usuário, dependendo do seu grau de conhecimento, pode estender as funcionalidades do ambiente criando seus próprios blocos, tendo como material de apoio pedagógico um tutorial presente no ambiente.

Independente da visão utilizada, o cliente LERO disponibiliza como ambiente de desenvolvimento uma adaptação do ambiente Snap! (SNAP!, 2015). O Snap!, desenvolvido pela *University of California at Berkeley*, é uma linguagem de programação em blocos que utiliza a metáfora de blocos de construção para permitir

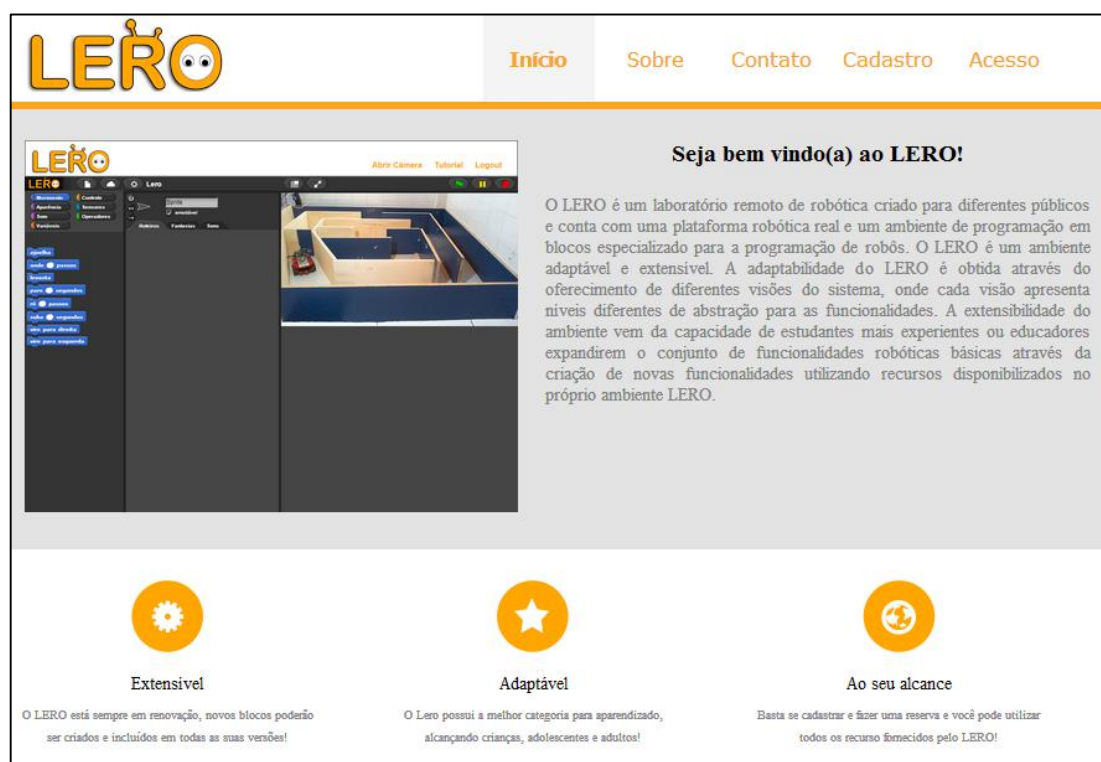
que o aluno possa criar programas, animações e jogos, enquanto assimila conceitos matemáticos e de programação. A adaptação do Snap! para a construção do ambiente LERO foi denominada de Snap! for LERO.

Com as adaptações feitas na interface de desenvolvimento foram criados os três ambientes do LERO para diferentes perfis ou níveis de conhecimento: LERO Crianças, LERO Adolescente e LERO Extremo. As páginas WEB e as diversas versões do cliente LERO são apresentadas e detalhadas nas próximas seções.

3.1.1 Páginas WEB

Ao entrar no site, a página da Figura 3.3 é vista pelos clientes. A partir desta página é possível acessar os links: Sobre, Contato, Cadastro e Acesso. O Sobre é uma página que conta um pouco sobre o projeto e mostra alguns detalhes que podem ser importantes aos navegadores. O Contato foi disponibilizado para permitir o envio de mensagens aos administradores do projeto oportunizando também a comunicação com os criadores do

Figura 3.3 – Página inicial do projeto LERO



Fonte: Autoria própria (2015).

A página de Cadastro solicita algumas informações ao cliente para que seja possível criar a sua conta. No ambiente de acesso do LERO, existem dois tipos de clientes: administradores e usuários. Ao ser realizado o login, dependendo do tipo de cliente, o sistema o redireciona ao seu respectivo ambiente.

Após efetivado o acesso como usuário aparece uma tela que permite acessar os dados do perfil do usuário e da sua conta, realizar reservas para utilizar o laboratório, visualizar dados históricos das reservas já realizadas e acessar o laboratório remoto caso possua uma reserva válida (Figura 3.4).

Figura 3.4 – Tela do Meu Perfil do cliente usuário



Fonte: Autoria própria (2015).

Caso o acesso seja efetivado como administrador, ao ser acessado (Figura 3.5), o ambiente interno do laboratório remoto apresenta algumas funcionalidades complementares às já descritas anteriormente.

Figura 3.5 – Tela Usuários do cliente administrador



LERO [Usuários](#) [Perfis](#) [Reservas](#) [Experimentos](#) [Laboratorio](#) [Sair](#)

Usuarios

Usuarios Cadastrados

Nome	E-mail	Perfil	Alterar	Excluir
Rodrigo Silva Lima	rodrig0_lima@hotmail.com	Extremo		X
teste	teste@teste.com	Criança		X
Rodrigo	rodrigolima2804@gmail.com	Extremo		X
Crijina Chagas	crijina@testes.com.br	Extremo		X

[Novo usuário](#)
[Alterar usuário](#)

Fonte: Autoria própria (2015).

Na opção 'Usuários', os administradores poderão gerenciar todos os usuários cadastrados no LERO. Na página 'Perfis', o administrador pode gerenciar os perfis do ambiente LERO já cadastrados, ou inserir um novo perfil. Na opção 'Reservas', os administradores poderão gerenciar e fazer novas reservas para algum usuário, caso seja necessário. Em caso da necessidade de algum reparo ou manutenção, o administrador pode suspender as reservas realizadas. Na opção 'Experimentos', o administrador pode criar e alterar experimentos.

Na opção 'Laboratório', o administrador pode acessar qualquer perfil em qualquer momento, mesmo sem uma reserva. Essa funcionalidade permite o livre acesso do administrador aos ambientes de desenvolvimento para possíveis monitorações.

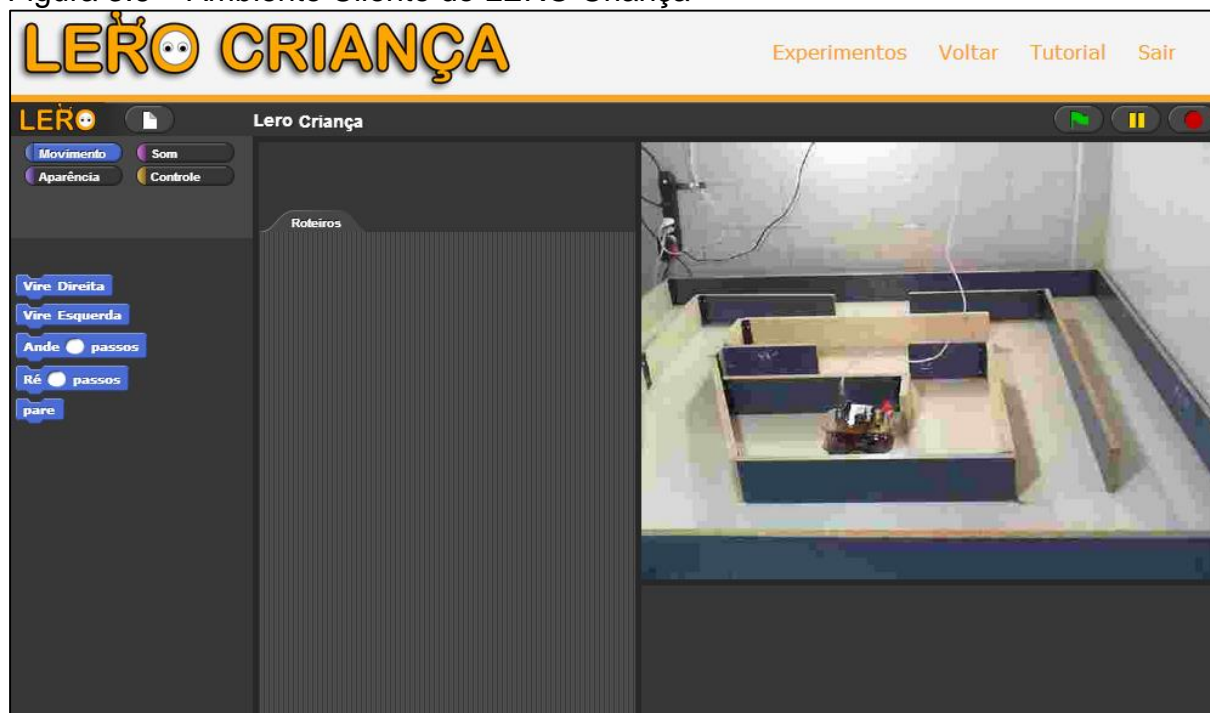
Quando o usuário possui uma reserva válida e seleciona a opção Acessar Laboratório (Figura 3.4), ele é direcionado, em função do seu perfil, para uma dos três ambientes disponíveis no LERO (Criança, Adolescente ou Extremo). As próximas seções discutem as especificidades de cada um desses ambientes.

3.1.2 LERO Criança

O LERO Criança disponibiliza uma interface simples para que o usuário tenha como foco o uso do raciocínio lógico e da percepção espacial e possa desenvolver sua criatividade de uma forma lúdica e motivadora. De forma simplificada, qualquer versão do cliente LERO é dividida em três grandes seções: A seção de funcionalidades, uma seção de roteiros e uma seção de visualização do experimento (Figura 3.6). Estas seções foram adaptadas das seções originais da versão Snap!. (Figura 2.16).

Para a incorporação do ambiente Snap! nas aplicações clientes do LERO foram realizadas algumas modificações na sua estrutura resultando em uma nova versão que foi denominada de Snap! *for* LERO. No Snap! *for* LERO foram retiradas funcionalidades do Snap! consideradas desnecessárias para o ambiente LERO. As principais adaptações realizadas ocorreram nos blocos oferecidos no menu original do Snap! e em algumas estruturas que foram retiradas para melhorar a usabilidade do ambiente. Desta forma só foram disponibilizadas as funcionalidades que efetivamente poderiam ser utilizadas pelo usuário.

Figura 3.6 – Ambiente Cliente do LERO Criança



Fonte: Autoria própria (2015).

A seção de funcionalidades, localizada na aba esquerda da interface do LERO Criança, apresenta somente quatro grupos de funções, a saber: Movimento, Aparência, Som e Controle.

É possível visualizar o grupo Movimento selecionado na Figura 3.7a e os seus blocos com as ações disponíveis para este grupo logo abaixo. O grupo Movimento possui blocos que possibilitam diversos tipos de movimentações do robô, dependendo da modalidade do experimento disponibilizado. Como o experimento disponibilizado foi um labirinto, só existem neste grupo os blocos que são compatíveis com este tipo de experimento. São eles: Vire Direita, Vire Esquerda, Ande N passos, Ré N passos e Pare.

O ambiente LERO atualmente oferece somente blocos de ações abstratas direcionadas à modalidade labirinto. Este bloco de ações pode ser estendido com novas ações, refletindo acréscimos e modificações realizadas na plataforma robótica. Pode-se acrescentar, por exemplo, uma linha preta na arena e ações no grupo Movimento, para oferecer a modalidade de experimento: seguidor de linha. Outra possibilidade é adaptar o robô como novos componentes eletrônicos ou inserir objetos na arena para serem resgatados. Ações para controlar novos equipamentos ou ações específicas para a modalidade resgate podem ser desenvolvidas e incorporadas ao ambiente LERO criança.

O grupo Aparência (Figura 3.7b) possui ações que permitem aos usuários acionarem algum equipamento que produza um efeito visual no robô. Na atual plataforma robótica o único efeito visual disponível é apagar e acender um pequeno farol instalado na parte frontal do robô. Desta forma, o grupo aparência possui somente os blocos Ascender Faróis e Apagar Faróis.

Figura 3.7 – Grupos Movimento e Aparência do LERO Criança



Fonte: Autoria própria (2015).

O grupo Controle (Figura 3.8a) possui um único bloco. Este bloco deve ser colocado sempre no início de cada roteiro e traduz a ação de começar a executar os comandos quando o usuário clicar na bandeira verde. Finalmente, o grupo Som (Figura 3.8b) disponibiliza blocos direcionados a execução de efeitos sonoros. Na plataforma robótica disponibilizada o robô possui uma buzina, desta forma, o único bloco disponível no ambiente é Buzinar, este aciona a buzina por um determinado intervalo de tempo.

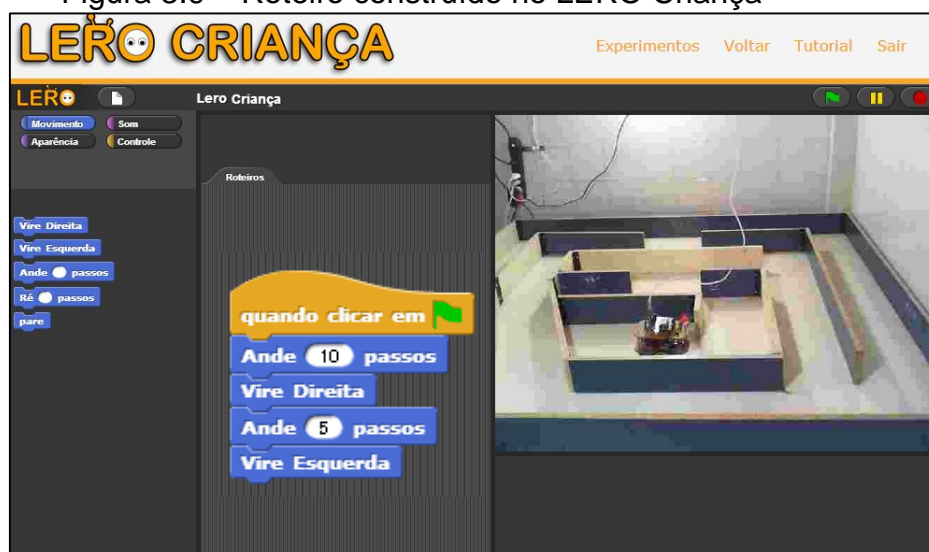
Figura 3.8 – Grupos Controle e Som do LERO Criança



Fonte: Autoria própria (2015).

À medida que o usuário seleciona os blocos de construção e monta o comportamento do seu robô, o diagrama de blocos com as funções selecionadas vai sendo atualizado na aba de roteiros, situada na parte central da interface. A interface do ambiente utiliza o recurso arraste e solte para incluir e posicionar uma nova funcionalidade. A Figura 3.9 ilustra um roteiro que instrui o robô a andar dez passos, virar a direita, andar mais cinco passos e por fim virar à esquerda.

Figura 3.9 – Roteiro construído no LERO Criança



Fonte: Autoria própria (2015).

Para transferir os comandos do roteiro para o robô e visualizar a execução das manobras, o usuário deverá utilizar a região de visualização de experimentos, situada no lado direito da interface. Após clicar na bandeira verde situada no canto superior direito da tela de visualização, o usuário poderá observar o movimento real do robô, na arena tipo labirinto, instalado em um laboratório físico remoto.

É importante salientar que nem todas as ações programadas pelo usuário podem ser executadas. A ação incluída no roteiro que determina o robô a andar dez passos, por exemplo, pode não ocorrer caso o robô detecte algum obstáculo logo a sua frente, isto é, uma parede do labirinto no experimento em questão. Caso isso aconteça, o robô para, o usuário recebe a notificação que o experimento não foi bem sucedido e é convidado a modificar o roteiro para corrigir o problema.

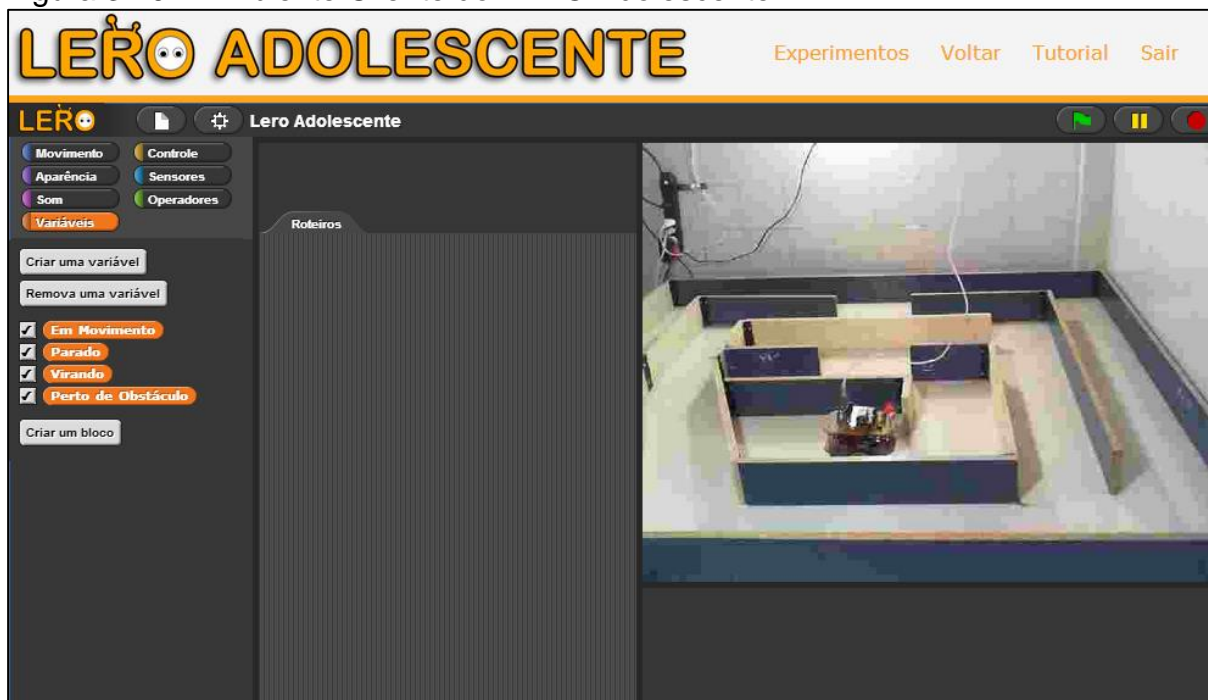
Na próxima seção será apresentado o ambiente de desenvolvimento LERO Adolescente, destinado aos usuários com conhecimento intermediário na área de conhecimento abordada.

3.1.3 LERO Adolescente

Para usuários com diferentes níveis de conhecimento e experiência, o sistema LERO disponibiliza duas outras visões da interface do cliente, as versões LERO Adolescente e LERO Extremo. As diferentes visões do cliente LERO se diferenciam no número de funcionalidades ofertadas, nos níveis de abstração dos comandos e nos recursos de configuração do ambiente. Estas diferenças se baseiam no grau de conhecimento e experiência nos conceitos de programação, geralmente explorados nas diversas faixas etárias suportadas no ambiente.

No cliente LERO Adolescente (Figura 3.10), as funcionalidades do LERO Criança são acrescidas de novos blocos de ações com menor grau de abstração, isto é, com comandos que permitem um controle mais fino do robô. Além disso, são incorporados blocos que trabalham operações aritméticas e lógicas, os conceitos de variáveis e as estruturas de controle de fluxo (estruturas de repetição e condicionais), além de um grupo de comandos básicos que permite operar diretamente os sensores existentes no robô. Estas estruturas são muito utilizadas no ensino de programação e robótica para alunos do ensino médio e como assunto introdutório em cursos técnicos e de graduação.

Figura 3.10 – Ambiente Cliente do LERO Adolescente



Fonte: Autoria própria (2015).

No LERO Adolescente a seção de funcionalidades, localizada na aba esquerda da interface, apresenta sete grupos de funções, três a mais do que o LERO Criança, são eles: Movimento, Aparência, Som, Variáveis, Controle, Sensores e Operadores. Os grupos Movimento, Aparência e Som continuam com os mesmos blocos do LERO Criança. O grupo Controle (Figura 3.11a) possui além do bloco de inicialização (bandeira verde), as usuais estruturas de programação de controle de fluxo (condicionais e de repetição). As estruturas condicionais disponibilizadas são a condicional simples (se-então) e a condicional composta (se-então-senão). As estruturas de repetição disponibilizadas são uma estrutura para repetição de um número conhecido de vezes (repita N vezes), repetição até que uma condição seja atendida (repita até que) e uma estrutura com loop infinito (sempre), nesse caso é necessário incluir uma condição de parada.

No grupo Variáveis (Figura 3.11b), foram pré-definidas algumas variáveis lógicas que representam o estado do robô para serem usadas em tomadas decisões. As variáveis utilizadas são Em Movimento, Parado, Virando e Perto de Obstáculo, que permitem ao usuário trabalhar com valores lógicos (verdadeiro/falso) para possíveis tomadas de decisões, usando o raciocínio lógico.

Figura 3.11 – Grupos Variáveis e Controle do LERO Adolescente



Fonte: Autoria própria (2015).

No grupo Sensores (Figura 3.12a), foram disponibilizados alguns blocos com comandos básicos que permitem operar diretamente os sensores existentes no robô. São eles: ativar pino N como saída, ativar pino N como entrada, valor digital do pino N mudar para N, onde N é um número inteiro indicando o pino na placa Arduino.

Por fim, o grupo de funções chamado Operadores (Figura 3.12b), possui blocos com operações aritméticas (+, -, *, /), lógicas (e, ou, não), relacionais (<, =, >) e constantes lógicas (verdadeiro, falso) que podem ser usadas na construção de códigos avançados para manipulação do robô.

Figura 3.12 – Grupos Sensores e Operadores do LERO Adolescente



Fonte: Autoria própria (2015).

A Figura 3.13 mostra um roteiro construído com o LERO Adolescente, onde o aluno cria um roteiro utilizando estruturas de repetição e condicionais. As ações de movimentação são entremeadas com ações que manipulam valores associados diretamente aos pinos de um sensor no robô. O roteiro apresentado quando executado, irá habilitar o pino doze de um dispositivo LED conectado ao robô. Se o robô não estiver perto de um obstáculo, serão repetidas cinco vezes as ações sequenciais: ascender LED, andar dez passos, apagar LED, virar à direita, ascender LED e andar mais cinco passos.

Figura 3.13 – Roteiro construído no LERO Adolescente



Fonte: Autoria própria (2015).

Na próxima seção será apresentado o ambiente de desenvolvimento LERO Extremo, destinado aos usuários com conhecimento avançado na área de programação e robótica.

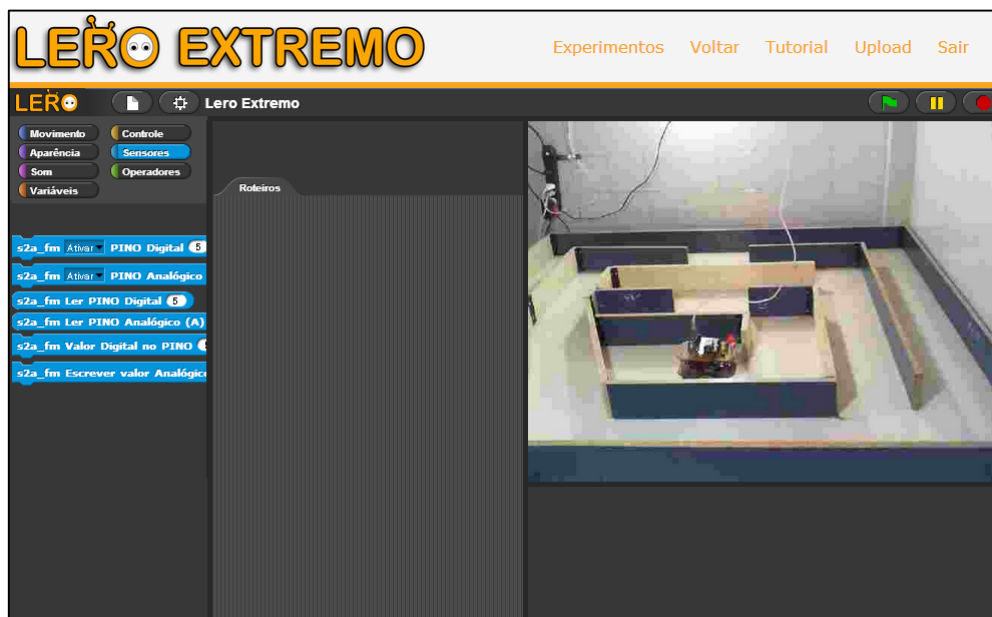
3.1.4 LERO Extremo

O ambiente LERO Extremo, assim como os outros ambientes, é dividido em três seções principais: funcionalidades, roteiros, área para visualização do experimento. O ambiente Extremo além das opções para acesso ao perfil, reserva do laboratório e o tutorial, possui uma opção exclusiva para upload de código, que será discutida mais adiante.

A versão do cliente LERO Extremo foi desenvolvida para usuários experientes e para desenvolvedores LERO. As funcionalidades disponibilizadas para o cliente Extremo são todos os blocos disponíveis na biblioteca LERO Adolescente, acrescidos dos blocos com instruções nativas da biblioteca s2a (S2A, 2015). A biblioteca s2a possui comandos de baixo nível e forma a base de todas as funcionalidades da biblioteca LERO. Os blocos do s2a são disponibilizados no ambiente extremo com o prefixo s2a_fm, que significa *Snap! to Arduino using firmata*. Firmata (FIRMATA, 2015) é uma biblioteca utilizada pelo s2a, que

implementa um protocolo para comunicação entre o Arduino e o software utilizado para controlá-lo.

Figura 3.14 – Ambiente Cliente do LERO Extremo



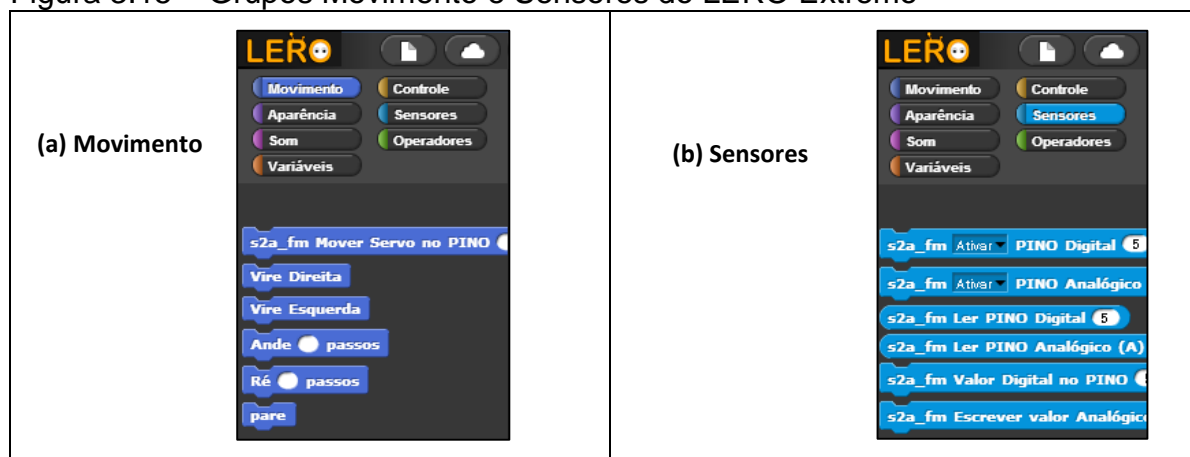
Fonte: Autoria própria (2015).

A seção de funcionalidades do LERO Extremo possui os mesmos sete grupos de funções do LERO Adolescente, a diferença é a presença dos blocos de baixo nível com prefixo `s2a_fm`. Estes blocos estão distribuídos entre os diversos grupos de acordo com a funcionalidade de cada bloco. No grupo Movimento (Figura 3.15a), foi disponibilizado o bloco `s2a_fm Mover Servo no PINO N em Graus: N`. Esse bloco permite controlar a angulação do servo motor, que gira de 0 a 180°, para que o usuário comande diretamente esse equipamento do robô.

No grupo Sensores (Figura 3.15b), foram disponibilizados outros blocos do `s2a` que também permitem operar diretamente os sensores existentes no robô. O primeiro bloco, `s2a_fm () PINO Digital N como ()`, ativa ou desativa um pino digital como uma entrada, saída, servo, tom ou sonar. Já o segundo bloco, `s2a_fm () PINO Analógico N como entrada, ativa ou desativa um pino analógico, que são sempre pinos de entrada`. O terceiro e quarto blocos, `s2a_fm Ler PINO Digital N` e `s2a_fm Ler PINO Analógico N`, servem para leitura do valor de um pino digital e analógico,

respectivamente. O quinto e sexto blocos, s2a_fm Valor Digital no PINO N mudar para () e s2a_fm () PINO Analógico N como entrada, servem para definir valores dos pinos declarados como saída. No caso do pino digital, os valores podem ser 0 ou 1, e no caso do pino analógico os valores podem variar entre 0 a 255. As informações técnicas necessárias estarão no tutorial disponibilizado que mostra o layout das ligações dos pinos do arduino com os componentes eletrônicos do Lerobô e explica as principais partes conceituais.

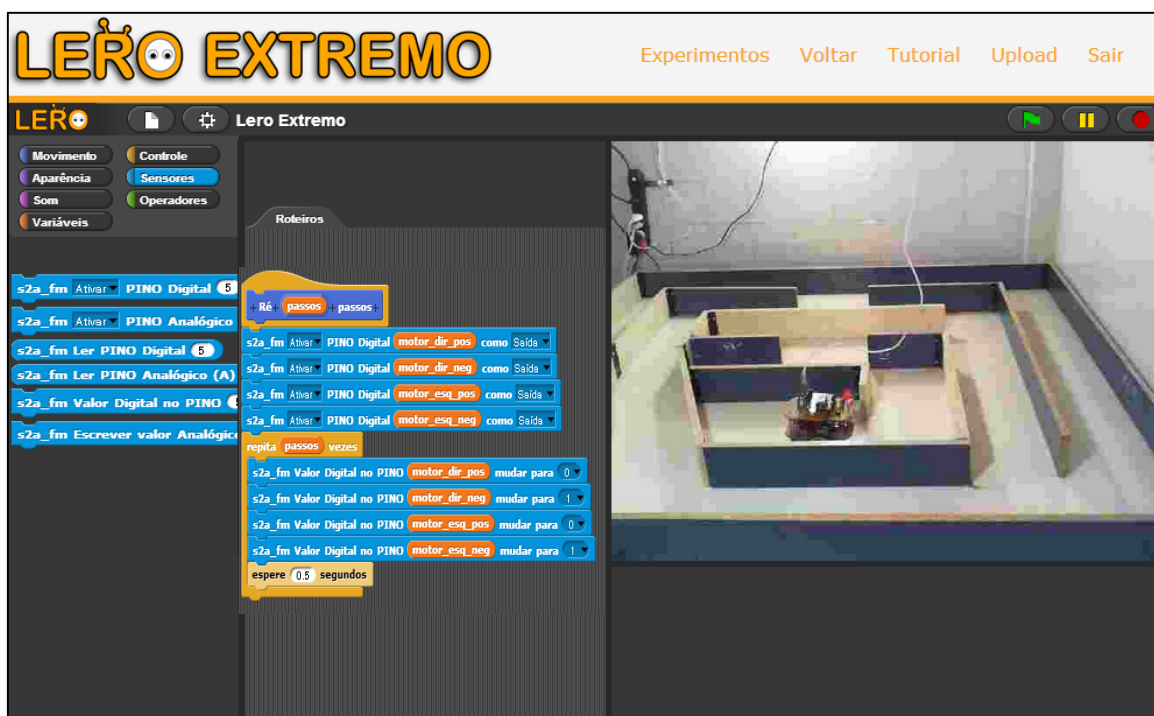
Figura 3.15 – Grupos Movimento e Sensores do LERO Extremo



Fonte: Autoria própria (2015).

O exemplo do roteiro construído com o LERO Extremo (Figura 3.16) ilustra o nível de funcionalidades disponibilizadas para o cliente LERO Extremo. Nesse exemplo encontra-se o bloco construído para o movimento Ré X passos e mostra como os pinos são configurados juntamente com variáveis e organizados em uma estrutura de repetição para realizar a ação necessária.

Figura 3.16 – Roteiro construído no LERO Extremo



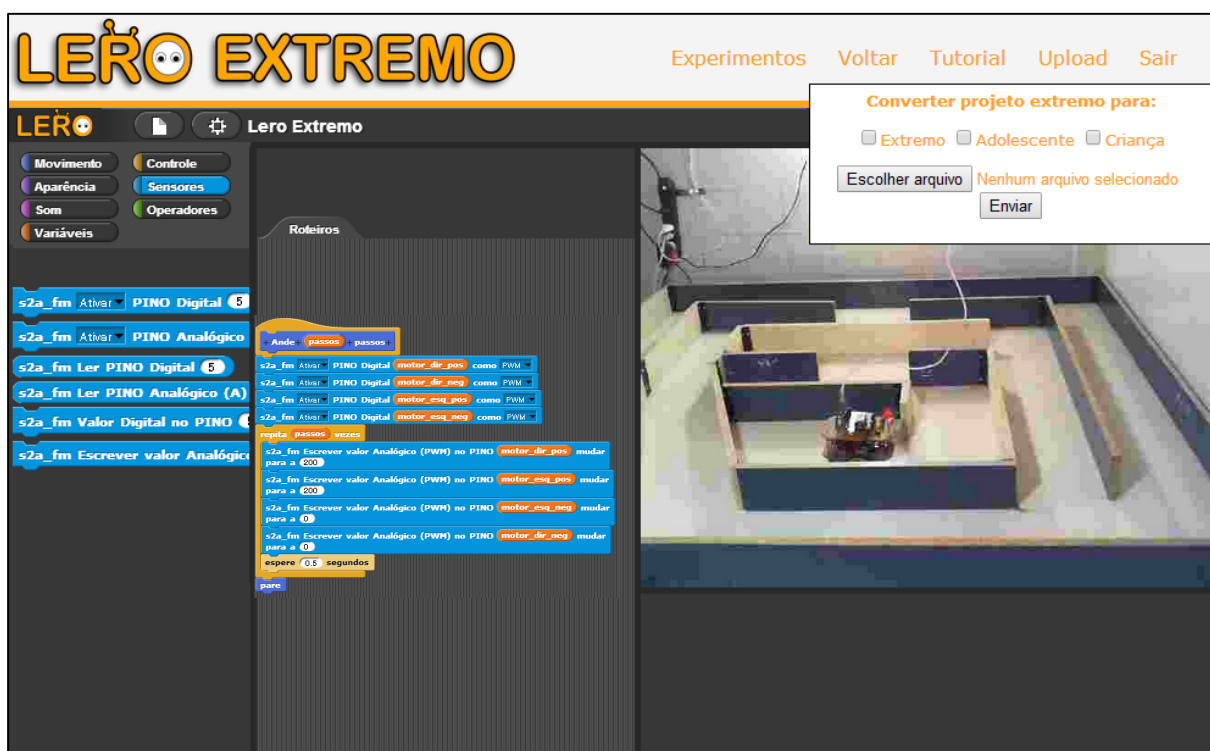
Fonte: Autoria própria (2015).

A característica mais saliente do cliente LERO Extremo, entretanto, é a capacidade de criar novas funcionalidades e agregar estas funcionalidades à biblioteca LERO padrão. De forma a ilustrar o processo de criação e incorporação de novas funcionalidades a biblioteca LERO, considere o desenvolvimento da funcionalidade do bloco Ande X passos apresentado na versão do cliente LERO Criança. O roteiro para esta ação foi codificado na versão LERO Extremo com o auxílio de blocos já existentes no ambiente e blocos da biblioteca s2a.

Na Figura 3.17 observa-se no roteiro o código parcial para construção do bloco Ande X passos. Nesse roteiro, para que o robô possa executar a ação de andar certo número de passos é verificada a existência de algum obstáculo impedindo o movimento. No exemplo, os obstáculos são as paredes do labirinto. Para a codificação da ação que determina a presença de obstáculos foi utilizado blocos que interagem com as leituras do sensor ultrassônico do robô. Após a criação desse bloco, o usuário/desenvolvedor LERO tem a opção de inserir o bloco criado em qualquer ambiente que ele desejar. Para que isso seja feito é necessário utilizar a opção Upload, mencionada anteriormente. No processo de Upload, o

desenvolvedor LERO tem a opção de definir em qual ambiente o novo bloco será disponibilizado. Ele pode incorporar o novo bloco em um ambiente específico ou em todos os existentes, incluindo o próprio LERO Extremo. Esta funcionalidade de produzir novos blocos utilizando o LERO Extremo e disponibilizar estes blocos para os demais ambientes é o que confere o caráter extensível do LERO. Desta forma, a extensibilidade do ambiente além de potencializar a ferramenta também estimula o usuário, pois este passa a ter um papel importante também na construção do ambiente.

Figura 3.17 – Funcionalidade Upload do Ambiente LERO Extremo



Fonte: Autoria própria (2015).

3.2 O AMBIENTE SERVIDOR

O ambiente servidor gerencia a base de dados de usuários, controla o acesso e traduz a programação realizada pelo cliente para instruções de baixo nível compatíveis com a plataforma robótica (Figura 3.18).

Figura 3.18 – Componentes do Ambiente Servidor

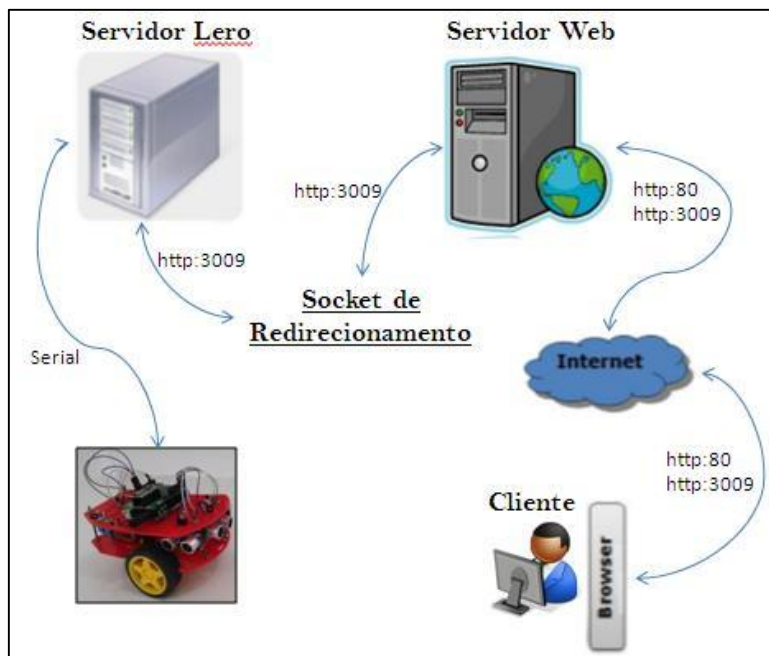


Fonte: Autoria própria (2015).

Os componentes de hardware e software do ambiente servidor se encontram nas instalações da Universidade Estadual do Sudoeste da Bahia (UESB). Sendo assim, esse ambiente teve que se adaptar às regras de segurança da unidade de informática da Universidade. Para o funcionamento do experimento foram utilizados dois servidores no ambiente servidor.

O servidor com acesso externo à Internet foi chamado de Web e o que se encontra na rede interna da UESB foi chamado de Lero. Sendo assim, há necessidade de transmitir as informações de forma bidirecional entre os dois servidores, de forma a permitir que o usuário, através do servidor WEB, consiga enviar dados para o servidor Lero e que obtenha uma resposta do mesmo. Por restrições da rede, não foi possível utilizar a forma mais simples de encaminhamento das solicitações Http feitas pelo usuário ao servidor Lero. Para solucionar essa questão, foi desenvolvido um socket de redirecionamento, que encaminha qualquer solicitação Http do servidor Web para o servidor Lero e que também espera por uma resposta do servidor Lero (,Figura 3.19). Essas respostas são do s2a que envia um pacote ao servidor Web reportando possíveis mensagens de erro ou de sucesso do experimento ao usuário. O socket de redirecionamento faz assim uma mão dupla de comunicação, enviando tudo que o servidor Web recebe dos dois lados fazendo uma ponte entre o servidor Lero e o usuário.

Figura 3.19 – Fluxo das solicitações entre os servidores

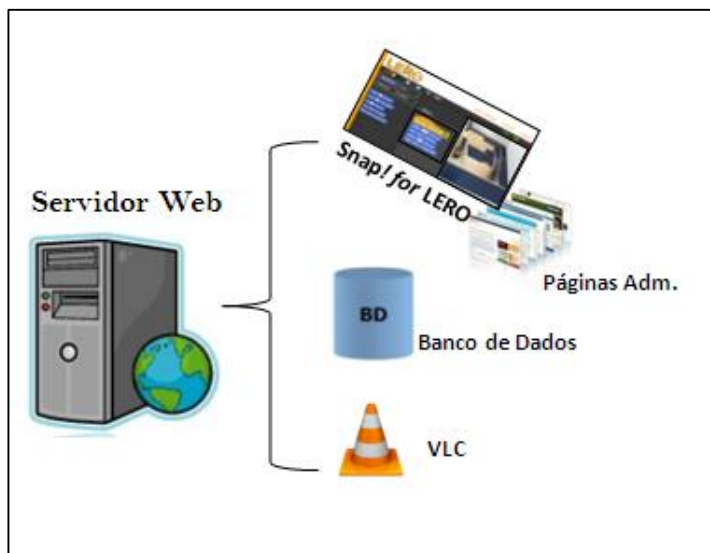


Fonte: Autoria própria (2015).

3.2.1 O Servidor Web

O servidor Web é responsável em hospedar as páginas de administração do sistema, as páginas do Snap! for LERO e o banco de dados (Figura 3.20). Ele tem como sistema operacional o Debian (DEBIAN, 2015).

Figura 3.20 – Componentes do Servidor Web



Fonte: A autoria própria (2015).

Através do servidor Web o usuário acessa a página principal do sistema e pode escolher a opção de se cadastrar para utilizar o laboratório remoto. Após o seu cadastramento, ele poderá acessar funcionalidades relacionadas com o gerenciamento de acesso ao experimento físico. Essas páginas de acesso e gerenciamento do laboratório remoto foram chamadas de páginas administrativas.

O *Snap! for Lero*, por se tratar de uma adaptação do *Snap!* original, também é composto por páginas escritas em *JavaScript*, mas que foram modificadas para serem a interface de desenvolvimento (IDE) do ambiente LERO. Elas estão hospedadas no servidor Web, assim como as páginas administrativas.

Todas as informações cadastradas nas páginas administrativas são alimentadas no banco de dados do sistema. Essas informações são fundamentais para o direcionamento ao ambiente cliente associado ao perfil de quem está acessando o sistema. São importantes também para o controle de reservas para o uso do experimento.

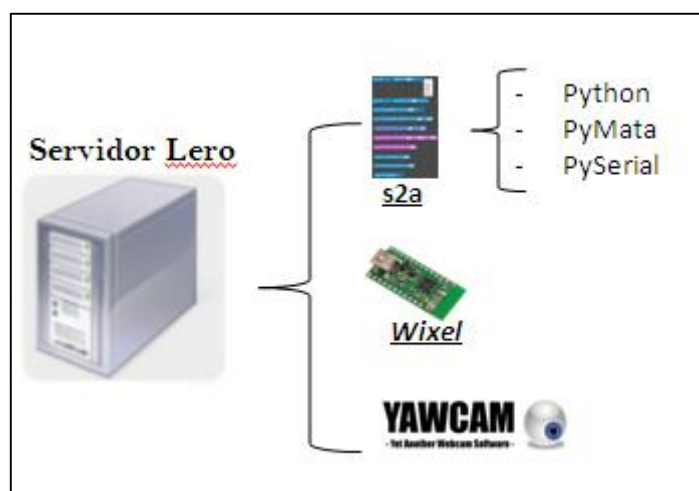
Por fim, mas não menos importante, é parte integrante do servidor Web o VLC (VideoLAN). VLC é um software com múltiplas funções de media e que foi utilizado para criar um fluxo de rede para enviar imagens ao usuário conectado ao ambiente LERO. Este programa foi instalado no servidor Web e é responsável por transmitir a imagem da câmera IP, que vem do servidor Lero, para o cliente.

3.2.2 O Servidor Lero

O servidor Lero, além de hospedar o s2a e o software para gerenciamento da câmera de vídeo, também está conectado a um chip *wixel* que faz a comunicação com a placa Arduino (Figura 3.21). Ele tem como sistema operacional o Windows XP.

No servidor Lero está hospedada a extensão responsável pela comunicação do *Snap! for LERO* com placa Arduino do robô, chamada *Snap! to Arduino (s2a)*, fazendo a conversão da programação em blocos para instruções compatíveis com a placa arduino. Além de converter as instruções da interface adaptada *Snap! for LERO*, ela também é responsável por enviar as ações a serem executadas para o robô através de uma conexão sem fio.

Figura 3.21 – Componentes do Servidor Lero



Fonte: Autoria própria (2015).

A extensão s2a foi desenvolvida em Python e por isso essa linguagem foi instalada no servidor Lero. O arduino é gerenciado pelo s2a através das bibliotecas Firmata, PyMata e PySerial. O s2a recebe uma solicitação via Http, transforma essas solicitações em um vetor que será enviado à biblioteca PyMata. Essa biblioteca faz a validação dessa solicitação, tanto sintática quanto semântica. Se não for validada ela retorna uma mensagem de erro que será repassada ao usuário. Caso contrário, a solicitação será encapsulada e direcionada à biblioteca PySerial,

que é responsável por estabelecer uma conexão com o Arduino e enviar a solicitação para ser executada pelo robô.

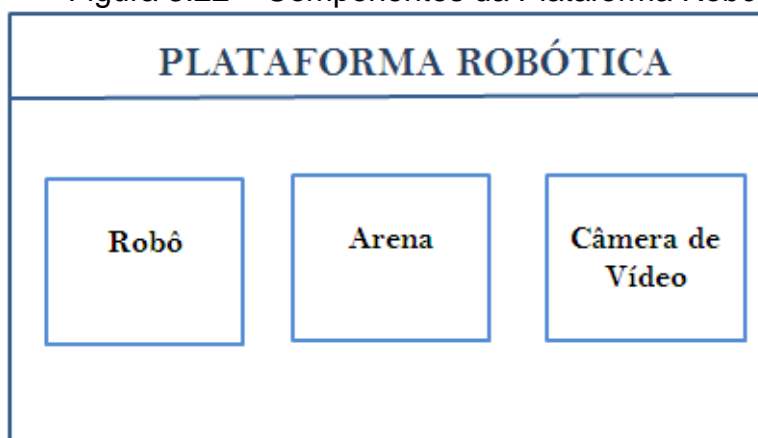
Os dados enviados do servidor Lero ao Arduino utiliza um módulo *Wixel* que se comunica com a placa *Shield Wixel* do robô, através de uma conexão sem fio. Para isso acontecer a biblioteca PyMata e o arquivo Standard Firmata foram modificados para operar na taxa de transmissão de dados por segundos (*baud rate*) do padrão *Wixel*.

A Webcam da plataforma robótica tem uma conexão física com o servidor Lero através de um cabo USB e essa imagem capturada por ela precisa transitar pela rede em formato digital. Um programa chamado YAMCAM (*Yet Another WebCam Software*) precisou ser instalado no servidor Lero com a finalidade de gerenciar os vídeos capturados pela Webcam. Este programa transforma a câmera USB em IP e utilizando o IP da rede UESB, transmite as imagens para o VLC no servidor Web.

3.3 A PLATAFORMA ROBÓTICA

A plataforma robótica da arquitetura LERO é composta por um robô, uma arena para movimentação do robô e uma câmera de vídeo para capturar os movimentos que serão transmitidos ao usuário (Figura 3.22).

Figura 3.22 – Componentes da Plataforma Robótica

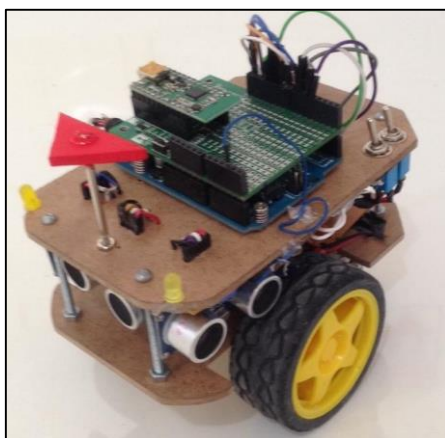


Fonte: Autoria própria (2015).

O robô, denominado Lerobô, foi montado baseado no Kit de Chassi Magician DG007. A diferença da estrutura para o Kit foi a redução de três centímetros nas laterais e altura das bases além do seu material que foi feito de madeira com três milímetros de espessura. Os motores, rodas e roda livre são as mesmas encontradas no Kit. O Lerobô possui dimensões de nove centímetros de altura (contados a partir do piso) limitando-se a altura das paredes do labirinto que possuem 9,5 cm. O robô possui quatro sensores de ultrassom, um *drive* para controle do motor, um Arduino Uno e uma placa *Shield Wixel* de comunicação *Wireless* (Figura 3.23).

O chassi *Magician* é produzido pela empresa DAGU Hi-Tech Eletronic e possui dois motores de corrente contínua acoplados em uma caixa de redução que recebe alimentação de 5volts-6volts e possui torque máximo de 800gf.cm. O Kit possui também duas rodas de plástico, uma roda boba e duas placas de acrílico para a acoplagem dos componentes.

Figura 3.23 – Imagem do Lerobô



Fonte: Autoria própria (2015).

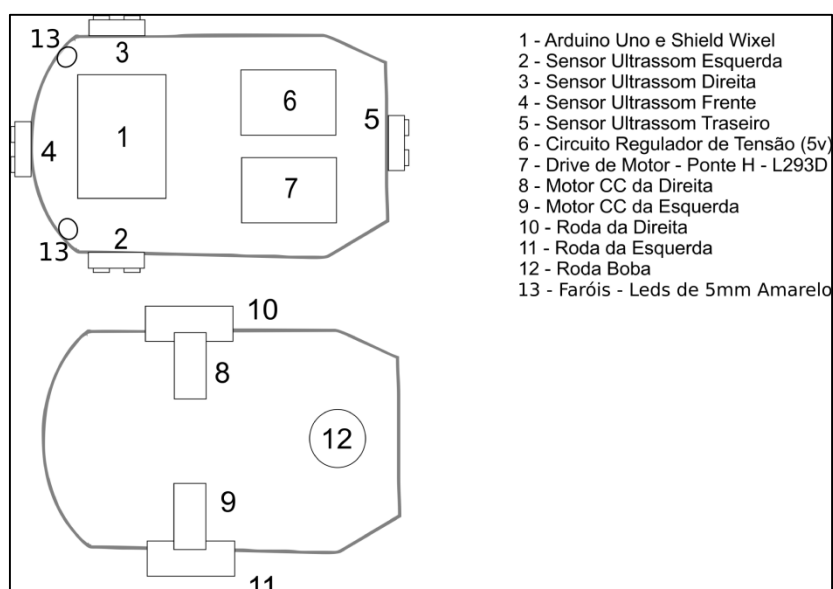
Os sensores de ultrassom (HC-SR04) são componentes eletrônicos utilizados no cálculo de distância entre dois objetos. Este sensor possui dois módulos, um emissor (trig) e um receptor (echo) com abertura angular de 15°. Os sensores de ultrassom utilizam um método chamado Cálculo de Tempo de Voo para aferir a distância do mesmo a um obstáculo. O método baseia-se em calcular o tempo que uma onda é emitida (pelo emissor), refletida (pelo obstáculo) e captada

pelo (receptor). Como a velocidade do som é conhecida e constante, e a posição de origem assume valor zero e então a distância é calculada pela fórmula do movimento retilíneo uniforme ($S=vt$). É importante notar que o tempo resultante do cálculo do sensor deve ser dividido por dois, uma vez que o componente mede o tempo de ida e volta da onda. Os sensores estão localizados na lateral esquerda, na lateral direita, um na frente e outro atrás do robô. Desta forma, o robô tem a noção dos obstáculos que estão nas laterais.

O drive de motor é um componente eletrônico voltado ao controle de motores do robô. O drive em questão possui um circuito integrado L293D conhecido como ponte H dupla, pois permite o controle simultâneo de dois motores. Uma ponte H é formada por um circuito lógico com quatro chaves, que a depender da combinação, permite que o motor gire em um sentido (horário e anti-horário). Desta forma, o robô tem a capacidade de andar para frente e para trás, além de fazer curvas.

Um controlador arduino pode ser entendido como “cérebro” do robô. É nele que o programa com descrição das ações que o robô deverá tomar fica armazenado e é executado. O Arduino Uno é uma plataforma de hardware livre de prototipação que possui um microcontrolador ATmega 328. A plataforma recebe alimentação de 5volts a 20volts e é responsável por executar as ações do robô no labirinto, controlando os motores e fazendo a leitura e cálculo de distância dos sensores de ultrassom. O Arduino possui disponível 14 pinos de entrada e saída digital e 6 pinos de entrada analógica, além disso, possui uma memória Flash de 32KB. Todas essas estruturas técnicas apresentadas podem ser visualizadas na planta do robô, Figura 3.24.

Figura 3.24 – Planta do Lerobô



Fonte: Autoria própria (2015).

A placa *Shield Wixel* de comunicação é produzida pela empresa *Pololu Robotics & Eletronics* e fornece uma interface sem fio para comunicação via serial da plataforma arduino com o computador ou outro dispositivo que seja capaz de comunicar-se via serial. O diferencial dessa *Shield* em relação a outros módulos que fornecem a interface de comunicação sem fio, como o módulo *Bluetooth*, é que o *Shield Wixel* permite que uploads de códigos fontes do Arduino seja feito de modo remoto.

A arena possui a dimensão de 1,5 m x 1,5 m em piso de fórmica na cor branco. Além disso, possui montado sobre ela, um labirinto feito de madeira para o robô atuar (Figura 3.25).

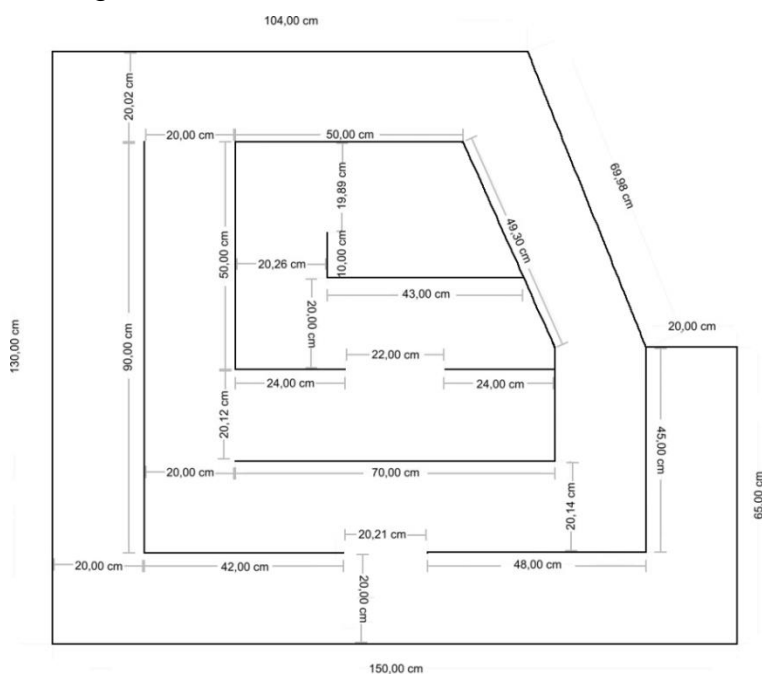
Figura 3.25 – Robô e Arena da Plataforma Robótica



Fonte: Autoria própria (2015).

Devido à restrição de espaço no laboratório físico, uma parte da arena não foi utilizada na criação do labirinto. Foram necessárias medidas precisas para a adaptação do labirinto criado ao tamanho do Lerobô, visando não prejudicar o movimento do mesmo no ambiente (Figura 3.26).

Figura 3.26 – Desenho e medidas da arena labirinto



Fonte: Autoria própria (2015).

A WebCam utilizada para capturar as imagens do robô na arena é a LifeCam HD-3000 da Microsoft (Figura 3.27) .

Figura 3.27 – Imagem da LifeCam HD-3000 da Microsoft



Fonte: MICROSOFT (2015).

Vários testes foram realizados para a escolha do local mais adequado no laboratório físico para uma visão geral do experimento através da câmera. Além disso, duas lâmpadas LED brancas foram instaladas para manter a luminosidade ideal no ambiente físico. O resultado das análises foi uma plataforma robótica mostrada na Figura 3.28 com todos os componentes do seu ambiente integrados para a visualização e manipulação do usuário.

Figura 3.28 – Visão da arena no ângulo da câmera de vídeo



Fonte: Autoria própria (2015).

3.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a arquitetura do ambiente LERO. Foram especificados todos os elementos de cada componente da arquitetura. Além disso, abordou também o papel e as interligações entre os mesmos.

O próximo capítulo irá abordar os métodos e recursos tecnológicos utilizados no desenvolvimento do ambiente LERO.

4 DESENVOLVIMENTO DO AMBIENTE LERO

O desenvolvimento do ambiente LERO demandou a utilização de estratégias e metodologias distintas, pois envolveu o desenvolvimento do ambiente cliente, do ambiente servidor e da plataforma robótica.

No ambiente cliente encontra-se uma aplicação Web que direciona o usuário a um perfil específico que pode ser o LERO Criança, o LERO Adolescente ou LERO Extremo. Os perfis do LERO foram construídos a partir de uma adaptação do ambiente de programação em blocos Snap! e essa nova versão foi denominada Snap! for LERO. A usabilidade da interface com o usuário foi um dos maiores desafios nesta etapa.

No ambiente servidor acontece a conversão da programação em blocos, feitas no ambiente cliente, em instruções compatíveis com o Arduino, para comunicação com o Lerobô na plataforma robótica. Além disso, no ambiente servidor está a infraestrutura necessária para o funcionamento das páginas WEB. A integração de diversas tecnologias e a intermediação das ações programadas pelo cliente com a plataforma robótica também foram bastante desafiadoras nesse estágio.

Na plataforma robótica encontra-se o robô, a arena e a câmera de vídeo, estruturas necessárias para o funcionamento do experimento físico e sua transmissão ao usuário remoto. Os custos financeiros e facilidade de replicação foram os princípios norteadores na especificação da plataforma robótica

Nas próximas seções serão detalhados os principais aspectos metodológicos e tecnológicos envolvidos no desenvolvimento dos diversos componentes da arquitetura LERO.

4.1 O DESENVOLVIMENTO DO AMBIENTE CLIENTE LERO

O ambiente cliente LERO, conforme a Figura 3.2, foi estruturado em três diferentes versões: Criança, Adolescente e Extremo. Estas versões trabalham no topo da adaptação Snap! for LERO, o ambiente de desenvolvimento de programação em blocos do LERO. O desenvolvimento desses componentes se baseou em uma metodologia bastante utilizada no desenvolvimento de software.

A análise dos requisitos, desenvolvimento e validação são processos que fazem parte da construção de qualquer ambiente. Essas atividades compõem o arcabouço da metodologia para o desenvolvimento de um software e são fundamentais na engenharia de software. Segundo Pressman e Lowe (2009, p. 17), esses métodos possuem um conjunto de princípios básicos que incluem atividades de modelagem e outras técnicas descritivas. Sendo assim, essa seção se destina a apresentar atividades realizadas para construção do ambiente cliente.

4.1.1 Requisitos Funcionais e Não Funcionais

A especificação dos requisitos funcionais descreve as ações que são esperadas após a construção do ambiente. Já os requisitos não funcionais apresentam características que irão refletir na qualidade do que foi construído.

Analisando e utilizando outros laboratórios remotos foram levantadas as funcionalidades básicas em comum para iniciar a especificação dos requisitos funcionais do laboratório remoto LERO. Após esse estudo inicial, foram discutidas ações específicas e itens de qualidade da solução a ser implementada. Esse levantamento culminou na documentação da especificação dos requisitos funcionais e não funcionais do ambiente LERO. Esse documento, chamado de documento de especificação de requisitos, está no Apêndice A deste trabalho.

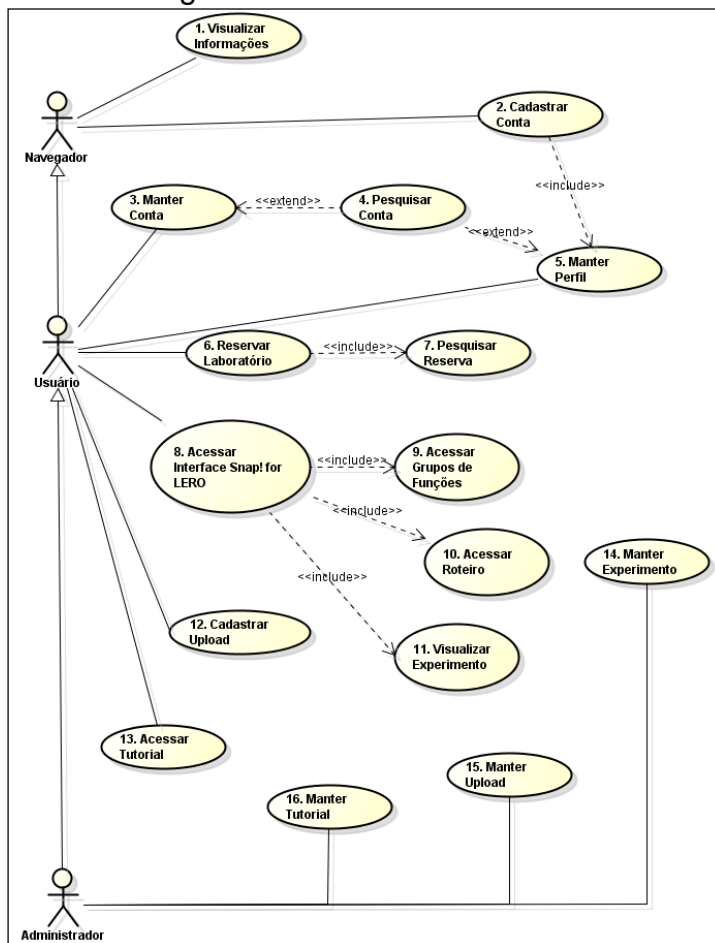
Os clientes, aqueles que interagem com o projeto LERO, foram classificados de acordo com o nível de acesso em: navegador, usuário e administrador. O navegador acessa o site do projeto, visualiza as informações, mas não tem conta cadastrada no ambiente, podendo se cadastrar a qualquer momento. O usuário tem conta cadastrada no ambiente e pode utilizar várias funcionalidades, dentre elas destacam-se:

- **Manter conta:** o usuário pode alterar e excluir a sua conta de acesso ao ambiente LERO;
- **Manter perfil:** o usuário pode alterar o seu perfil definido pelo sistema, de acordo com sua habilidade em robótica e linguagem de programação;
- **Reservar laboratório:** o usuário pode reservar uma data e horário para acessar o laboratório. Além disso, visualizar reservas atuais e/ou futuras, podendo alterá-las ou excluí-las;

- **Acessar a interface Snap! for LERO:** o usuário pode acessar a interface do Snap! for LERO e utilizar os blocos disponibilizados nos grupos de funções referentes ao seu perfil no ambiente. Os blocos podem ser arrastados, editados, alterados, excluídos e organizados em sequência para serem submetidos ao servidor e, conseqüentemente, ao robô na plataforma robótica.
- **Visualizar experimento:** o usuário pode visualizar a ação do robô após o código ser enviado para ser executado;
- **Fazer projeto:** o usuário pode criar um novo bloco e fazer upload do mesmo para ser julgado pelo administrador e caso aprovado, ser disponibilizado na biblioteca do ambiente LERO;
- **Acessar tutorial:** o usuário pode acessar o tutorial disponibilizado no ambiente.

Os principais Requisitos Funcionais do ambiente LERO estão ilustrados no Diagrama de Caso de Uso na Figura 4.1. As relações entre os atores e seus casos de uso e também os casos de uso entre si podem ser observadas nesta figura.

Figura 4.1 – Diagrama de Caso de Uso do Ambiente LERO



Fonte: Autoria própria (2015).

Conforme visto no diagrama de caso de uso, o administrador é uma especialização do usuário, ou seja, pode utilizar as funcionalidades do usuário, mas acessa também funcionalidades específicas e restritas. São elas:

- **Manter experimento:** o administrador pode cadastrar um novo experimento ou alterar e excluir um já existente.
- **Manter projeto:** o administrador pode avaliar as submissões dos projetos enviados pelos usuários e aprová-lo ou não para ser disponibilizado na biblioteca do LERO.
- **Manter tutorial:** o administrador pode cadastrar um tutorial de um determinado perfil do ambiente ou alterar e excluir um já existente

O Requisito Não Funcional, característica que têm como finalidade a qualidade do sistema, em destaque para o ambiente LERO foi a usabilidade. A usabilidade preza a facilidade de uso da interface pelo cliente do ambiente. O fato de o LERO ser um ambiente que interage diretamente com os usuários, a usabilidade teve um papel importante para dar um parecer sobre a qualidade do sistema. As heurísticas de Jakob Nielsen foram as métricas utilizadas para validar a usabilidade do sistema e serão descritas na próxima seção.

4.1.2 Métricas da Usabilidade

O ambiente cliente LERO foi validado através dos chamados testes de unidade e de integração. Diante da dificuldade de aplicar um teste de usabilidade com participantes menores de idade e considerando que a usabilidade da aplicação é importante, mas não o foco dessa pesquisa, decidiu-se aplicar um método de avaliação por inspeção na interface do sistema para tentar identificar problemas que os usuários pudessem ter quando interagissem com o ambiente LERO.

O método utilizado foi uma avaliação heurística, criado para encontrar problemas que pudessem prejudicar a usabilidade. Esta avaliação foi escolhida por ser rápida e de baixo custo. Essas heurísticas resultam da experiência de especialistas em IHC (Interação Humano Computador), dentre eles destaca-se as heurísticas propostas por Jakob Nielsen (BARBOSA; SILVA, 2010). Nielsen

descreve um conjunto inicial de heurísticas utilizadas em seu método de avaliação que analisa se a interface está de acordo com a necessidade do público alvo, neste caso principalmente as crianças e adolescentes.

São dez essas heurísticas de Jakob Nielsen (BARBOSA; SILVA, 2010):

1. Visibilidade do *status* do sistema: ela propõe que o usuário seja informado pelo sistema sobre o que está acontecendo em um tempo razoável através de *feedback*. Sendo assim, qualquer ação que esteja sendo feita pelo sistema em que o usuário tenha que esperar, ele deve ser devidamente informado em um intervalo de tempo satisfatório;
2. Correspondência entre o sistema e o mundo real: propõe que o modelo lógico do sistema seja compatível com o modelo mental do usuário permitindo assim, que o usuário se sinta familiarizado com o software. Para que isto seja cumprido o designer do programa deve estudar as expectativas do usuário, desse modo, o foco do desenvolvimento é o usuário;
3. Liberdade e controle do usuário: ela propõe que o sistema possibilite ao usuário o controle do sistema, isto é, as tarefas podem ser encerradas a qualquer momento, bem como operações serem desfeitas e também refeitas;
4. Consistência e padronização: sugere que o sistema tenha consistência na simbologia, bem como nas plataformas de hardwares e softwares, isto é, a mesma operação deve ser apresentada na mesma localização e deve ser formatada/apresentada da mesma maneira para facilitar o reconhecimento;
5. Prevenção contra erros: propõe que o projeto seja cuidadoso em evitar que problemas e erros aconteçam, caso seja possível;
6. Reconhecimento em lugar de lembrança: propõe que as instruções para o funcionamento do sistema devem estar visíveis para o usuário;
7. Flexibilidade e eficiência de uso: sugere que o sistema tenha atalhos, teclas de função, duplo clique no mouse e funções de volta em sistemas hipertexto;
8. Estética e design minimalista: recomenda que os diálogos do sistema somente apresente informações relevantes ao funcionamento;
9. Ajuda aos usuários no reconhecimento, diagnóstico e correção de erros: adverte que as mensagens devem ser expressas em linguagem, indicando as possíveis soluções;

10. Ajuda e documentação: recomenda que qualquer auxílio em relação ao sistema deve ser facilmente encontrado, contextualizado e não muito extenso.

A avaliação foi feita por uma professora da disciplina de IHC de uma instituição de ensino superior, que se disponibilizou a fazer uma avaliação através de um formulário do ambiente LERO Criança. A seguir será apresentada uma discussão dos principais resultados da avaliação. O documento com a íntegra da avaliação pode ser consultado no Anexo A.

Na avaliação de usabilidade foi utilizada uma escala de erros para classificar o nível de severidade dos problemas encontrados em cada heurística inspecionada. Os erros apontados e as respostas aos mesmos foram:

- Na primeira heurística avaliada foi apontado um erro catastrófico de grau quatro. O erro indicou que a conclusão da execução do bloco no ambiente de desenvolvimento era informada somente através da mudança de cor no bloco, não havendo outro elemento para dar retorno ao usuário sobre o *status* do sistema. A solução proposta no relatório foi implementar uma barra de progresso ou outro elemento que fornecesse ao usuário uma resposta enquanto ocorria a execução da tarefa. Para resolver esse problema inserimos no Snap! for LERO uma área que reporta o status do experimento a medida que o usuário interage com o mesmo;
- Na terceira heurística avaliada foram apontados três problemas, um catastrófico de grau quatro e dois graves de grau três. O primeiro problema, classificado com catastrófico de grau quatro, mostrou que na tela de login, o sistema não oferecia opção de lembrete, dica, pergunta secreta ou qualquer outro mecanismo para recuperação de senha e sugere para solução desse problema implementar a possibilidade de recuperação de senha na tela do login. Esse erro será tratado futuramente na próxima versão do ambiente com a inserção de um mecanismo de recuperação de senha. Já o segundo problema indicava que na tela de reserva do experimento, ao reservar um horário inferior ao atual, o sistema não oferecia mensagem de erro e confirmava a reserva, entretanto, a mesma não se tornava visível na tabela de reservas realizadas, mas aparecia juntamente com as demais reservas feitas, no histórico de reservas realizadas. A solução proposta para esse problema foi que o sistema não permitisse a realização de reservas em

horários/datas inferiores ao atual e que informasse essas regras ao usuário. Para solução desse erro foi feita uma modificação que não permite aparecer nas opções de escolha nem horário ou data inferior ao atual. O terceiro problema aponta que ao excluir uma reserva, antes da execução da tarefa, o sistema não oferece ao usuário uma mensagem de confirmação da exclusão (alerta), que permita desfazer/cancelar a ação. A solução proposta foi inserir uma mensagem de erro permitindo confirmar ou cancelar a ação do usuário. Essa solução proposta foi implementada e a mensagem: “Deseja excluir essa reserva?” aparece antes da exclusão definitiva da reserva pelo usuário.

- Na quarta heurística avaliada foram apontados dois problemas. O primeiro grave de grau três indicou que a opção “Meu perfil” não fica explícita como link, uma vez que as opções “Lero Criança” e “Acessar laboratório” foram desenhadas no mesmo padrão e não são links. Essa ação induz o usuário a passar para outra tela sem clicar na opção de escolha de perfil. A solução proposta para esse problema foi deixar a aba da opção “Alterar perfil” expandida ou colocar uma imagem que sugira que há conteúdo oculto (+). Indicou também corrigir a grafia do nome Perfil na aba superior. O segundo problema apontou que a opção “Acessos ao laboratório” estava “avulsa” na página, não havendo conexão com outros elementos, não é link e não está relacionada a texto ou tabela. Foi explicado à avaliadora que a opção Acessar Laboratório é um link e foi adotada a mesma solução para o primeiro erro desse item. Para resolver esse problema modificamos todas as telas inserindo a imagem (+) em todos os links, inclusive em “Acessar laboratório”, que também é um link;
- Na quinta heurística avaliada foi apontado um problema grave de grau três. No formulário: “Dados Gerais”, a mensagem de erro retornada no preenchimento do campo “idade” não auxiliava o usuário a corrigir o engano cometido, como por exemplo, digitar letras no lugar de números. Como solução foi proposto melhorar a mensagem de forma a auxiliar o usuário a resolver o problema. Esse problema foi resolvido alterando a mensagem de erro para: “A idade é um campo numérico!”.

A próxima seção mostrará as tecnologias envolvidas no desenvolvimento do sistema referentes ao componente ambiente servidor da arquitetura LERO.

4.2 O DESENVOLVIMENTO DO AMBIENTE SERVIDOR LERO

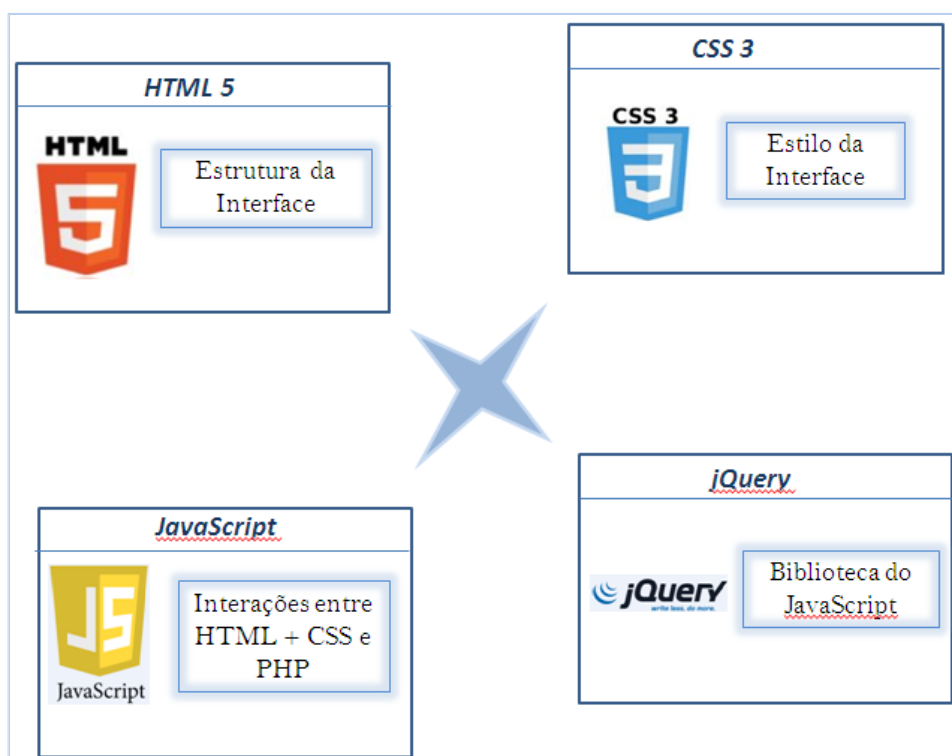
Para criar o laboratório remoto de robótica LERO foram necessárias criações e adaptações de tecnologias já consagradas. Dentre elas destacam-se as tecnologias Web utilizada para o gerenciamento do laboratório remoto e as adaptações realizadas para a comunicação remota entre a interface de desenvolvimento e a plataforma robótica e vice-versa.

As próximas seções discutem as principais tecnologias utilizadas no desenvolvimento dos servidores Web e LERO.

4.2.1 Servidor Web

O servidor Web hospeda estruturas do ambiente LERO, tais como as páginas administrativas e o banco de dados (Figura 3.20). Porém, para o desenvolvimento do ambiente Web do LERO, foram utilizados vários recursos tecnológicos específicos. Alguns desses recursos foram o HTML5 (*HyperText Markup Language*), CSS3 (*Cascading Style Sheets*), JavaScript e jQuery (Figura 4.2).

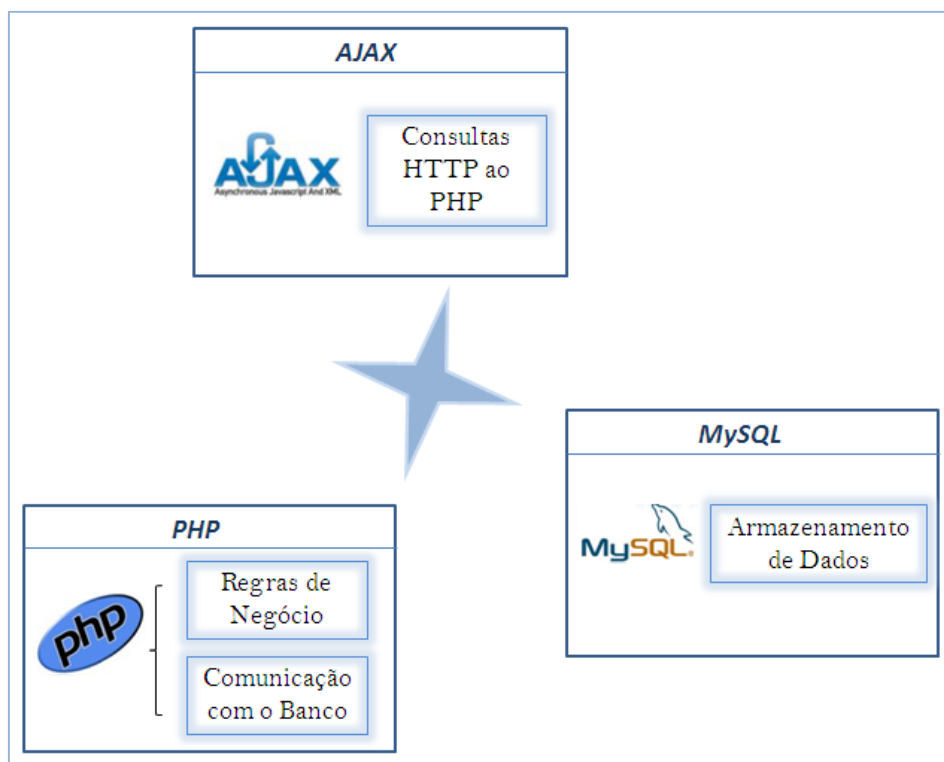
Figura 4.2 – Tecnologias Utilizadas no Servidor Web



Fonte: Autoria própria (2015).

Toda a estrutura da interface para a exibição de conteúdo foi feita utilizando o HTML5, funcionando como esqueleto da página. O HTML é estilizado pelo CSS3, que contém todas as folhas de estilo da interface do LERO. Todas as interações entre a interface (HTML e CSS) e as regras de negócio (PHP) são feitas com JavaScript utilizando a biblioteca jQuery. O jQuery é uma biblioteca do JavaScript que facilita o desenvolvimento de eventos e funções JavaScript.

Figura 4.3 – Mais Tecnologias Utilizadas no Servidor Web

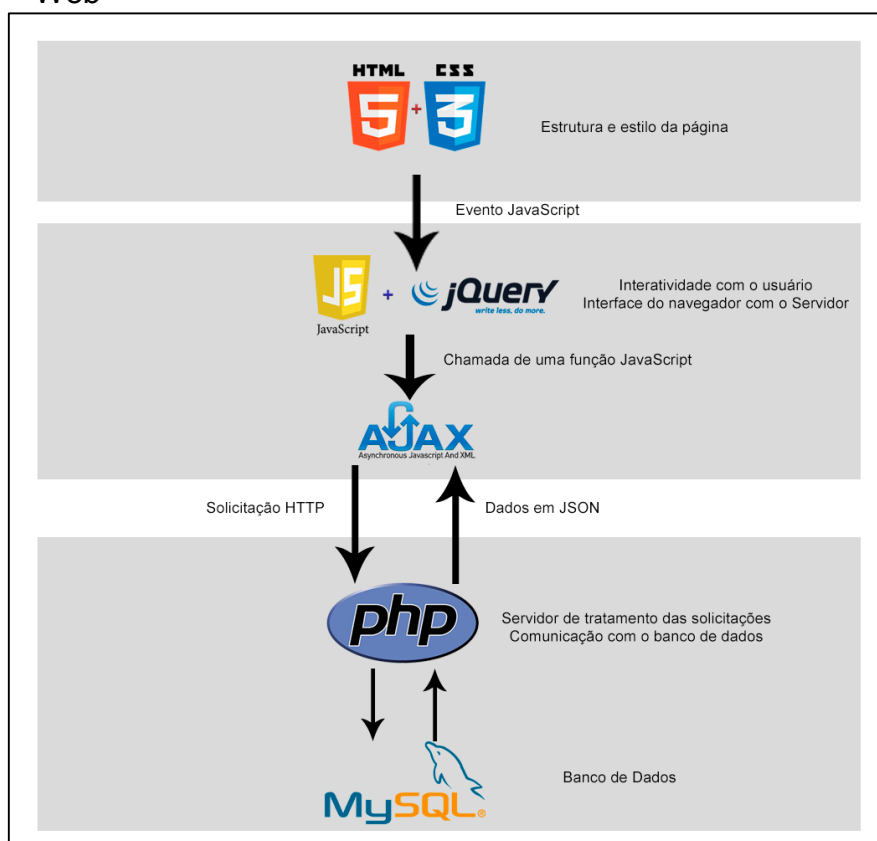


Fonte: Autoria própria (2015).

O AJAX, o PHP e o MySQL são tecnologias que também foram utilizadas no desenvolvimento do projeto LERO. O AJAX (*Asynchronous Javascript and XML*) é uma tecnologia que permite a interação do JavaScript com o servidor. No caso específico do LERO, o AJAX funciona fazendo requisições HTTP ao PHP. O PHP (Hypertext Preprocessor) é uma linguagem de *script* adequada ao desenvolvimento Web e pode ser embutida no HTML. Nessa tecnologia estão codificadas as regras de negócio e as estruturas para comunicação com o banco de dados. O MySQL foi o banco de dados escolhido para armazenar os dados do projeto. Esses dados estão estruturados no banco de acordo com o modelo de dados que pode ser visualizado

no Documento de Especificação de Requisitos do projeto LERO no Apêndice A. A Figura 4.4 exibe a forma como todas estas tecnologias foram integradas para produzir o ambiente servidor Web LERO.

Figura 4.4 – Integração entre as Tecnologias Utilizadas no Servidor Web



Fonte: Autoria própria (2015).

A partir do ambiente Web do LERO o usuário pode acessar o ambiente de desenvolvimento *Snap! for LERO*, que é uma versão adaptada do *Snap!*, para construir experimentos para serem executados na plataforma robótica.

Para fazer as adaptações necessárias no *Snap!* foi preciso entender o seu funcionamento. Cada elemento da interface da ferramenta é desenhado na tela, e todos os seus elementos são interligados, isto é, a remoção de uma parte do código pode resultar em problemas no funcionamento de todo o *Snap!*. Com isso, foi necessário analisar cuidadosamente o que seria removido, para então ser feito o estudo de como remover tais funcionalidades sem comprometer o funcionamento do todo. Desta forma, foi extraído o trecho do código que fazia o desenho de cada elemento da interface e criava a sua referência. Além disso, também foram retiradas

todas as referências a estes elementos, além de verificar se não era um elemento essencial para o funcionamento de toda a ferramenta. Algumas remoções desconfiguraram a disposição dos elementos na tela, assim algumas correções de posicionamento tiveram que ser feitas para manter a estrutura original do Snap!.

O Snap! possui uma série de funcionalidades que foram consideradas dispensáveis a alguns usuários, principalmente o perfil criança. Como os usuários do LERO devem estar focados no aprendizado da lógica de programação e não no aprendizado do sistema. Assim, como forma de direcionar ao uso da biblioteca LERO, criada para o funcionamento do robô, boa parte dessas funcionalidades foi removida na versão *Snap! for LERO*. A versão com mais adaptações foi o LERO Criança. O LERO Adolescente e o Extremo, também sofreram leves adaptações.

Modificações foram feitas em alguns arquivos principais do Snap! (Figura 4.5a). Foi alterado o arquivo responsável por desenhar os elementos na tela onde está codificada toda a estrutura da ferramenta. A maioria das alterações descritas a seguir foi realizada nessa estrutura. Esse arquivo possui um arquivo auxiliar, que também foi modificado, onde guarda informações sobre os blocos, categorias, eventos do sistema e mais algumas funcionalidades. E por fim, o arquivo responsável por gerenciar os processos do Snap! também foi modificado.

Figura 4.5 – Adaptações do Snap! para o Snap! for LERO



Fonte: Autoria própria (2015).

Serão detalhadas algumas das principais modificações. O Snap! possui um palco, e nesse palco eram mantidos atores para serem manipulados pelos usuários. Esses atores não tinham função no *Snap! for LERO* e por isso a criação, modificação, manipulação desses atores foi removidas. Além disso, o palco foi substituído pela câmera perdendo a sua função (Figura 4.5b).

Um espaço focado para a visualização dos atores no palco, também foi removido, e em seu lugar foi criada uma estrutura de status do Lerobô, com o objetivo de disponibilizar ao usuário informações relevantes sobre o robô em relação ao experimento construído (Figura 4.5c).

No Snap! existem três abas de manipulação, uma para os blocos, outra para fantasias dos atores e uma para importar sons do computador. As abas de fantasias e sons foram removidas e foi mantida apenas a aba de roteiro, que é onde o usuário do LERO deverá montar sua estrutura de blocos (Figura 4.5d).

Havia também três opções de acesso na barra superior, contendo uma série de funcionalidades e configurações, as quais foram reconfiguradas. Esses menus foram resumidos, sendo mantidas apenas as funções de exportar e importar um projeto (Figura 4.5e).

A versão do *Snap! for LERO* foi criada a partir dessas principais adaptações apresentadas. Quando o *Snap! for LERO* é solicitado pelo ambiente WEB ele abre e carrega o projeto específico em XML do perfil do usuário. Esse projeto foi previamente configurado para mostrar somente as opções do grupo de função e os blocos da biblioteca LERO vinculados aos seus grupos.

Uma das características mais importante do LERO é a extensibilidade do ambiente. A possibilidade de criação de novos blocos é nativa do Snap!. Esta funcionalidade, entretanto, foi mantida somente na versão LERO Extremo, que é o único ambiente capaz de criar novas funcionalidades. O processo de criação de novos blocos no Snap! é simples e foi mantido da mesma forma na adaptação *Snap! for LERO*. O maior desafio aqui é a disponibilização das novas funcionalidades criadas no ambiente Extremo para todos os demais ambientes do LERO. Esta disponibilização requer alteração nos arquivos de definição dos diversos ambientes.

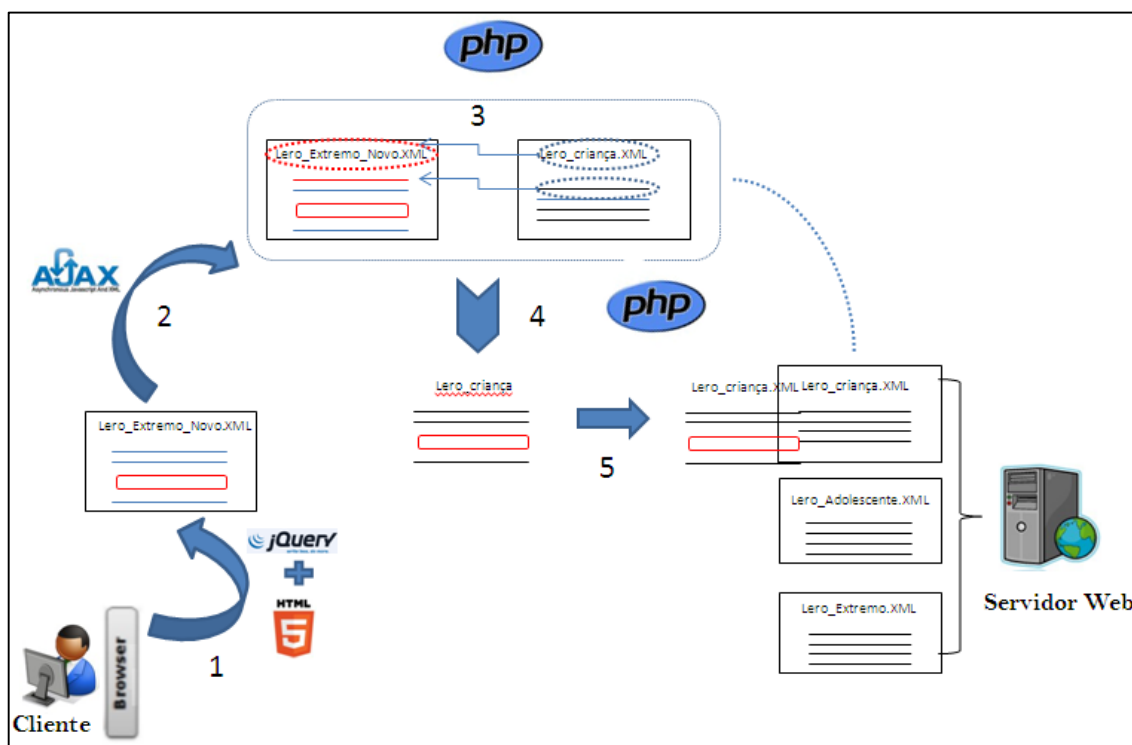
Os três ambientes do projeto LERO (Criança, Adolescente e Extremo) estão definidos em arquivos XML distintos, armazenados no servidor Web. O processo de extensão de novas funcionalidades é definido pela criação de projetos, que como resultado geram novas versões do modelo de definição (arquivo XML) do respectivo ambiente, que por fim, substituem ou sobrescrevem os arquivos de definição originais. A possibilidade de reescrever os arquivos de configuração dos ambientes é exclusiva da versão *Snap! for LERO*, pois não existe na versão original.

Na versão atual do sistema, quando um projeto novo é submetido, através da funcionalidade Upload do LERO Extremo, ele substitui/sobrescreve diretamente o arquivo de definições original do perfil ao qual ele foi submetido através de um sistema de conversão (Figura 4.6). Futuramente esse projeto novo passará primeiramente por uma avaliação do administrador e somente se for aprovado será submetido ao sistema de conversão.

No desenvolvimento do sistema de conversão foram utilizadas as ferramentas: HTML (formulário para submissão), jQuery (biblioteca do JavaScript),

AJAX (envio do formulário sem a necessidade de atualizar a página do LERO), e PHP (realiza as operações no lado servidor Web, convertendo e salvando o arquivo XML que foi enviado pelo usuário).

Figura 4.6 – Sistema de conversão de projetos LERO



Fonte: Autoria própria (2015).

Para a conversão de projetos LERO, o usuário deve escolher um dos perfis do LERO (Criança, Adolescente ou Extremo) (etapa número 1 da Figura 4.6). Ao enviar o novo projeto, o sistema realiza uma validação nas informações do formulário preenchido, e em seguida esse projeto é processado por um *script* específico, de acordo com o perfil previamente do ambiente selecionado, que irá tratar a conversão (etapa número 2 da Figura 4.6).

As descrições abaixo se referem ao processo de conversão para o LERO Criança e LERO Adolescente, visto que o processo do LERO Extremo é mais simples, pois é apenas uma substituição de arquivos.

O processo de conversão tem como objetivo principal a simplificação das estruturas do projeto em criação. O projeto gerado no LERO Extremo é mais completo e possui elementos de interface que devem ser suprimidos nas versões Criança e Adolescente, que requerem um nível maior de abstração. Em termos de

código, essa simplificação se baseia na ocultação de blocos e variáveis. O *script* de conversão processa o arquivo do projeto original para obter os elementos que devem ser ocultados e depois replica estas configurações no projeto novo. O valor do campo nome no código do projeto novo também é mudado e definido como igual ao valor do campo nome do código do projeto base. Essa etapa do processo está representada pelo número 3 da Figura 4.6.

Após essas alterações, o produto resultante é convertido em uma *string*, etapa representada pelo número 4 da Figura 4.6. O projeto original é aberto para escrita, sobrescreve-se o conteúdo pela *string* tratada e o processo de conversão é finalizado. Essa etapa do processo está representada pelo número 5 da Figura 4.6.

O envio de um novo projeto para o LERO Extremo é mais simples, pois apenas verifica se o tipo do arquivo enviado é XML, e esse arquivo é movido ao servidor WEB e substitui o projeto original do perfil Extremo.

A próxima seção apresentará as principais tecnologias presentes no servidor LERO e utilizadas no desenvolvimento do projeto LERO.

4.2.2 Servidor Lero

O servidor Lero é responsável pela tradução das instruções enviadas do ambiente cliente para a plataforma robótica, permitindo dessa forma que o robô possa realizar o roteiro programado pelos usuários (Figura 3.21). O elemento central nessa conversão é a aplicação s2a.

De uma forma geral, a solicitação é feita a partir da interface de desenvolvimento do ambiente LERO, sendo passado via HTTP ao servidor WEB que sempre a redireciona ao servidor LERO, onde o s2a está instalado. O s2a recebe a solicitação, trata, encapsula e envia a solicitação ao Arduino do Lerobô. O Arduino interpreta e executa a solicitação e ao fim do processo retorna uma mensagem ao usuário. Essa estrutura pode ser visualizada na Figura 4.7.

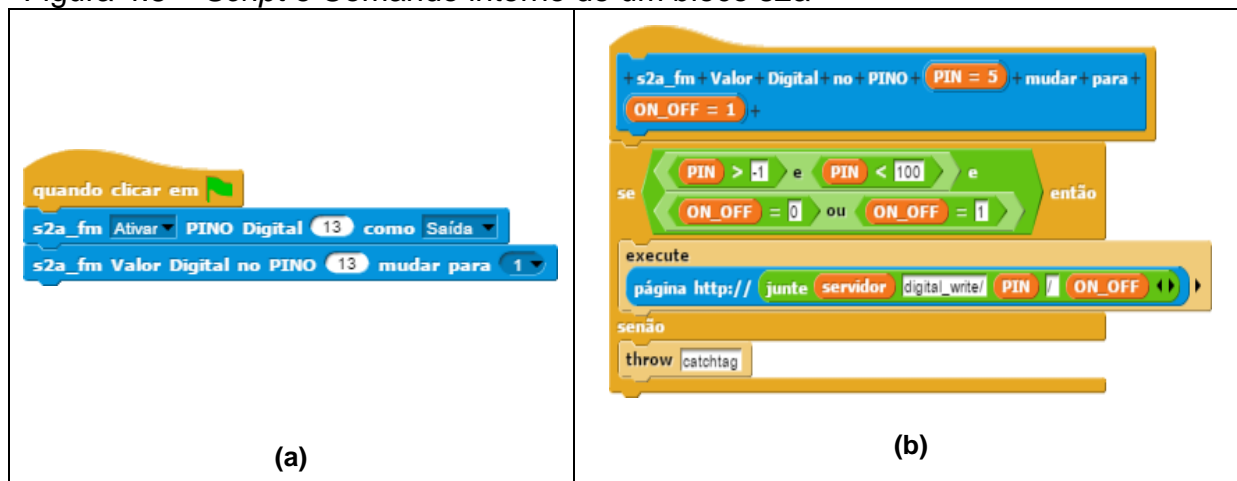
Figura 4.7 – Esquema geral da comunicação Snap! for LERO e o Robô



Fonte: Autoria própria (2015).

Visando facilitar o entendimento de como uma instrução gerada no Snap! for LERO chega ao Arduino foi utilizado o exemplo de uma escrita digital. Nesse exemplo, o pino 13 é ativado como saída e o valor digital mudado para 1, nesse mesmo pino. Os blocos Snap! for LERO dessa ação podem ser visto na Figura 4.8a. O bloco do *script* para a escrita de um valor digital é composto internamente pelos comandos mostrados na Figura 4.8b. Essa informação aparece ao clicar com o botão esquerdo em cima do bloco do s2a e escolher a opção editar.

Figura 4.8 – Script e Comando interno de um bloco s2a



Fonte: Autoria própria (2015).

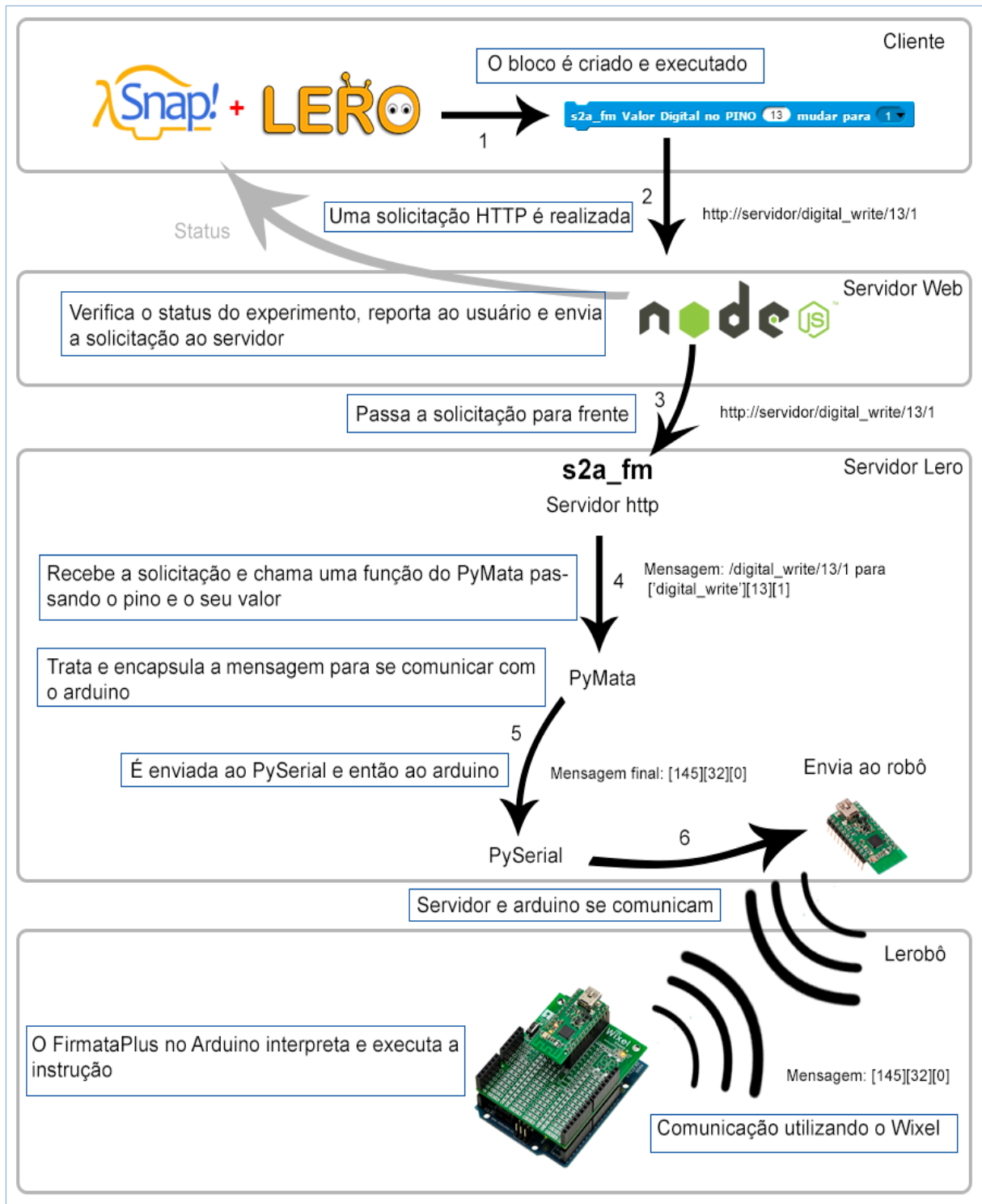
As entradas do bloco são tratadas e encapsuladas para no fim se tornarem uma URL. Os valores atribuídos aos pinos são validados para garantir que estão dentro dos possíveis valores do Arduino, além dos outros parâmetros a serem tratados. Todos os blocos do s2a terminam com uma instrução chamada execute e uma solicitação HTTP. Essa solicitação HTTP é feita a partir da URL que é montada de acordo com o bloco correspondente. Essas informações estão nas etapas um e dois da Figura 4.9.

Essa solicitação é feita ao servidor WEB que a redireciona ao servidor LERO, através do *socket* de redirecionamento desenvolvido em node.js. No servidor LERO está o s2a, que recebe a solicitação e a interpreta, etapa três da Figura 4.9. O servidor Web, além de direcionar a solicitação, verifica a conexão com o servidor LERO e reporta ao usuário o *status* da execução dos seus blocos.

Para essa solicitação poder ser interpretada o s2a_fm.py já deve estar inicializado. Durante a inicialização o s2a_fm.py se comunica com o Arduino fazendo a leitura de todos os pinos presentes na placa e se tudo estiver correto o processo para fazer o tratamento das solicitações é iniciado.

Para entendimento da etapa quatro da Figura 4.9 deve-se saber que o s2a_fm possui duas classes vitais para o seu funcionamento: http_server.py e command_handlers.py. O http_server.py recebe a solicitação, a interpreta e passa para o command_handlers.py que irá tratar as informações da solicitação e passar para a próxima etapa.

Figura 4.9 – Fluxo do processamento do bloco s2a no projeto LERO



Fonte: Autoria própria (2015).

No nosso exemplo, a solicitação que o s2a recebe é `/digital_write/13/1`. O `http_server.py` transforma o caminho em um vetor que ficará no seguinte formato:

['digital_write', '13', '1']. Após isso, o vetor é passado ao `command_handlers.py` que é composto por vários métodos, cada um para uma respectiva solicitação dos blocos do s2a.

O `command_handlers.py` possui um dicionário de comandos e de acordo com a solicitação recebida pelo `command_handlers.py`, o método específico da ação é evocado. Assim, o primeiro método busca em um dicionário o método da ação correspondente, que nesse caso é `'digital_wirte'`, e passa novamente o vetor de comandos como parâmetro. No exemplo que está sendo explicado, como se trata do `digital_write`, o método da Figura 4.10 é chamado.

O vetor de comandos `['digital_write', '13', '1']` é então tratado. Nesse tratamento os parâmetros são validados para verificar se possuem formato e valores corretos. Após a validação é então verificado se o pino foi anteriormente ativado para a escrita. Em caso positivo, é invocado o método `digital_write` do `firmata`, sendo passado como parâmetros o valor do pino e o valor do parâmetro, nesse caso, `(13,1)`.

Figura 4.10 – Trecho de código do método `digital_write`

```
def digital_write(self, command):
    """
    This method outputs a 0 or a 1 to the designated digital pin that has been previously
    been configured as an output.

    If the pin is configured as an INPUT, writing a HIGH value with digitalWrite()
    will enable an internal 20K pullup resistor (see the tutorial on digital pins on arduino site).
    Writing LOW will disable the pullup. The pullup resistor is enough to light an LED dimly,
    so if LEDs appear to work, but very dimly, this is a likely cause.
    The remedy is to set the pin to an output.

    @param command: Command and all possible parameters in list form
    @return: okay
    """
    # test pin as a digital output pin in poll list table

    if not command[self.CMD_PIN].isdigit():
        logging.debug('digital_write: The pin number must be set to a numerical value')
        print 'digital_write: The pin number must be set to a numerical value'
        return 'digital_write: O numero do pino deve ser definido como um valor numerico'

    pin = int(command[self.CMD_PIN])

    if self.digital_poll_list[pin] == self.firmata.OUTPUT:
        self.firmata.digital_write(pin, int(command[self.CMD_VALUE]))
        return 'okay'
    # for pullup - see description above
    elif self.digital_poll_list[pin] == self.firmata.INPUT:
        self.firmata.digital_write(pin, int(command[self.CMD_VALUE]))
        return 'okay'
    else:
        print 'digital write: Pin %d must be enabled before writing to it.' % pin
        logging.debug('digital write: Pin %d must be enabled before writing to it.' % pin)
        return 'digital write: O pino %d deve ser ativado antes de se escrever nele.' % pin
```

Fonte: Autoria própria (2015).

A etapa cinco do processo acontece na biblioteca do firmata que é constituída de três arquivos: o `pymata.py`; `pymata_command_handlers.py` e o `pymata_serial.py`. Essas estruturas fazem uma série de tratamentos com os valores que foram recebidos, dentre elas algumas operações de bits para cálculos do valor do pino e do comando, depois serão escritos no Arduino. Os dados convertidos e escritos no terminal, nesse exemplo, foram: `[145][32][0]`, que se referem ao *digital_write*, ao pino 13 e ao valor passado 1.

O código que gravado no Arduino foi o Firmata Plus que utiliza a biblioteca Firmata e algumas outras bibliotecas. Na etapa sete mostrada na Figura 4.9 é feito o mapeamento da mensagem que foi recebida pelo Arduino, `[145][32][0]`, de acordo com o que foi recebido, uma função irá realizar o que foi solicitado.

Após a execução de todos os métodos pelo python e pelo Arduino, o caminho de volta é feito entre as estruturas anteriormente citadas até chegar no *digital_write* do `s2a_command_handlers.py`, que retorna um *'okay'* ou uma mensagem de erro irá para o usuário. Essa mensagem de erro foi modificada no ambiente LERO. O cabeçalho da mensagem é montado e enviado para o *Snap! for LERO* e se houver algum erro ou mensagens ao usuário ela é enviada junto ao cabeçalho. Se a resposta for *'okay'* nada é escrito e a mensagem enviada é vazia. O *Snap! for LERO* recebe a mensagem e em caso de exceção ele irá parar a execução do código e exibir o erro. No caso de sucesso termina a execução do bloco e inicia outro bloco.

4.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas as tecnologias utilizadas para o desenvolvimento do ambiente LERO. Foram especificados os recursos tecnológicos, seus papéis e como eles interagem entre si.

O próximo capítulo objetiva mostrar o uso do ambiente LERO em algumas situações relevantes. A criação de uma nova funcionalidade para ser realizada pelo robô também será apresentada e por fim a utilização dessa nova funcionalidade no experimento físico.

5.0 USO DO AMBIENTE LERO

Uma funcionalidade importante do ambiente LERO e que não poderia deixar de ser apresentada é a reserva do laboratório remoto. Para que o usuário possa acessar a interface de desenvolvimento e o experimento físico do LERO ele deve ter, obrigatoriamente, uma reserva em um determinado dia e horário, caso contrário, não terá o acesso permitido ao ambiente de desenvolvimento do LERO.

Para um usuário devidamente cadastrado, a primeira tela que aparece, após ter feito o login, é a do Meu Perfil (Figura 5.1).

Figura 5.1 – Tela Meu Perfil do ambiente Web LERO

LERO [Meu Perfil](#) [Minha Conta](#) [Reserva](#) [Histórico](#) [Ajuda](#) [Sair](#)

Meu Perfil

LERO Extremo

A versão do cliente LERO Extremo foi desenvolvida para usuários experientes e para desenvolvedores LERO. As funcionalidades disponibilizadas para o cliente Extremo são todos os blocos disponíveis na biblioteca LERO Adolescente, acrescidos dos blocos com instruções nativas da biblioteca s2a. A biblioteca s2a possui comandos de baixo nível e forma a base de todas as funcionalidades da biblioteca LERO. Nesse ambiente foi disponibilizada uma nova funcionalidade: o upload, que permite ao usuário enviar novos blocos criados por eles e que poderão fazer parte da biblioteca LERO.

[Alterar Perfil +](#)

[Acessar Laboratório +](#)

Experimento:

LERO EXTREMO

Para acessar o laboratório é necessário realizar uma reserva.
Caso não tenha uma, [clique aqui](#).

Fonte: Autoria própria (2015).

Se o usuário já tem uma reserva cadastrada para aquele dia e horário, o link para acessar o ambiente de desenvolvimento do LERO estará ativo e ele precisará clicar no experimento, assim aparecerá o link para o acesso ao seu perfil. Caso ele não tenha feito uma reserva, pode clicar na opção Reserva que mostrará as

informações dos horários reservados, caso tenha alguma. Aparecerão também os campos experimento, data e hora que devem ser informados para fazer uma nova reserva. Cada reserva tem a duração de uma hora e somente permite duas reservas, por dia, para o mesmo usuário. Essa regra visa disponibilizar o laboratório remoto para o maior número de usuários possível, não permitindo a monopolização do experimento por somente um usuário. Caso este tente reservar mais de duas vezes, uma mensagem de alerta será exibida informando que não é permitida mais de duas reservas no mesmo dia (Figura 5.2).

Figura 5.2 – Tela da mensagem de alerta da reserva do laboratório

LERO Meu Perfil Minha Conta **Reserva** Histórico Ajuda Sair

Reserva

Horários reservados +

Data	Hora	Experimento	Excluir
13/08/2015	11:00:00	Labirinto	X
13/08/2015	12:00:00	Labirinto	X

Reservar Experimento +

São permitidas no máximo 2 reservas de horário por dia!

Experimento:

Data:

Hora:

Fonte: Autoria própria (2015).

Após a reserva, o usuário poderá acessar o laboratório remoto no dia e horário solicitado através da tela Meu Perfil. Na próxima seção será apresentada uma simulação de um experimento real no ambiente LERO Criança.

5.1 LERO EM AÇÃO

O LERO Criança permite a execução de movimentos do Lerobô a partir dos blocos abstratos disponibilizados na interface de desenvolvimento Snap! for LERO. Quando o *script* é construído no roteiro e a bandeira verde é ativada o Lerobô

executa o código fazendo um determinado percurso. Foi criado um script no roteiro em que o Lerobô deve andar quatro passos, virar à esquerda e andar vinte passos. Este *script* foi construído para que o Lerobô execute os dois primeiros movimentos e ao tentar executar o terceiro encontre um obstáculo. Nesse momento o robô deverá parar e informar ao usuário o motivo pelo qual parou para o usuário. A Figura 5.3 mostra o Lerobô na posição inicial de execução do *script*.

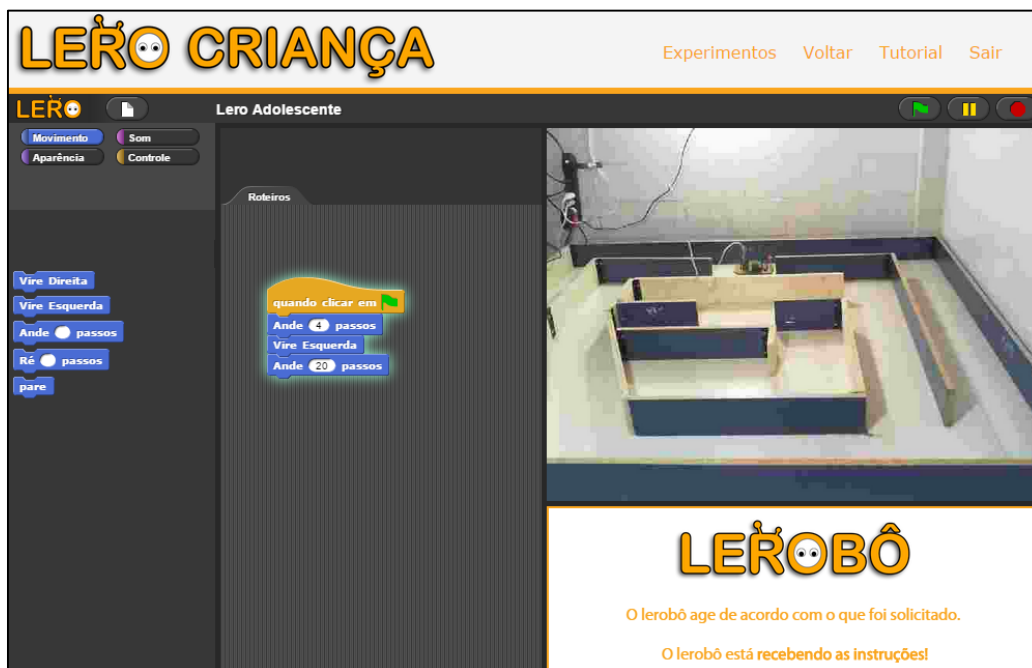
Figura 5.3 – Tela com *script* sem sucesso no roteiro do LERO Criança



Fonte: Autoria própria

A Figura 5.4 mostra a posição do Lerobô após executar o primeiro e segundo bloco do *script* e em direção à execução do terceiro e último bloco. A luz verde ao redor do *script* no roteiro indica que os comandos estão em execução no Snap! for LERO.

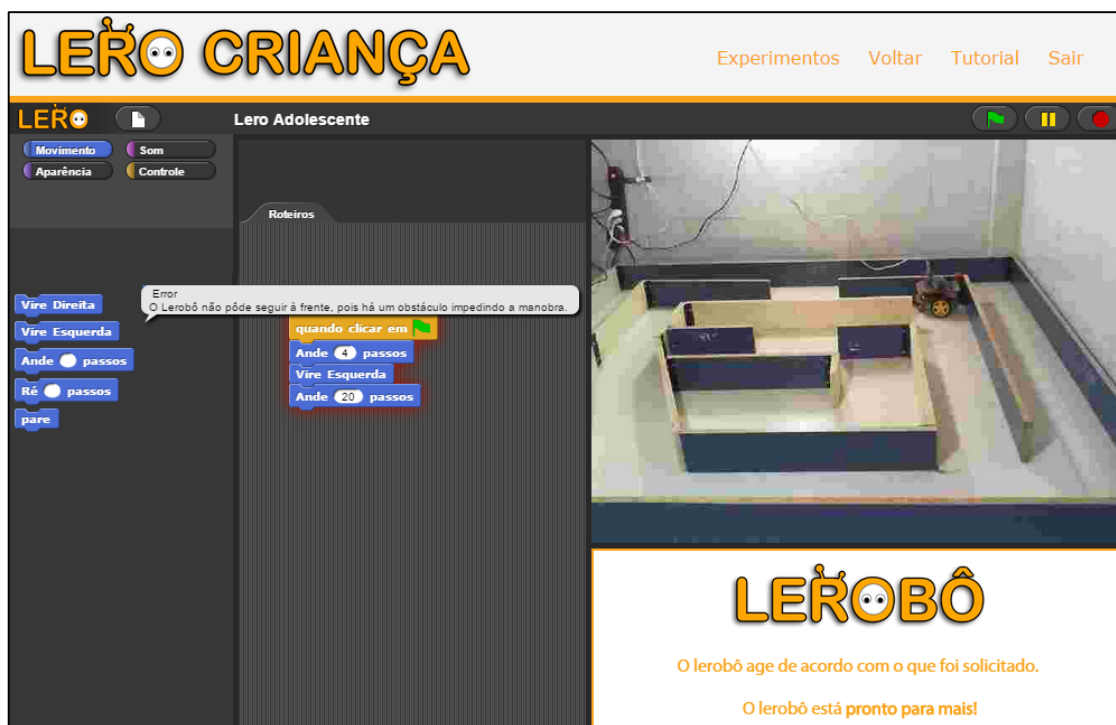
Figura 5.4 – Tela com *script* sem sucesso sendo executado na Plataforma Robótica



Fonte: Autoria própria (2015).

A Figura 5.5 mostra que durante a execução do último bloco foi detectado um obstáculo e o Lerobô parou em uma distância definida como segura para não tocar no obstáculo, e uma mensagem foi exibida ao usuário: “O Lerobô não pode seguir à frente, pois há um obstáculo impedindo a manobra”. A partir daí o *script* fica com uma luz vermelha ao redor indicando que ocorreu uma falha durante a execução do *script*.

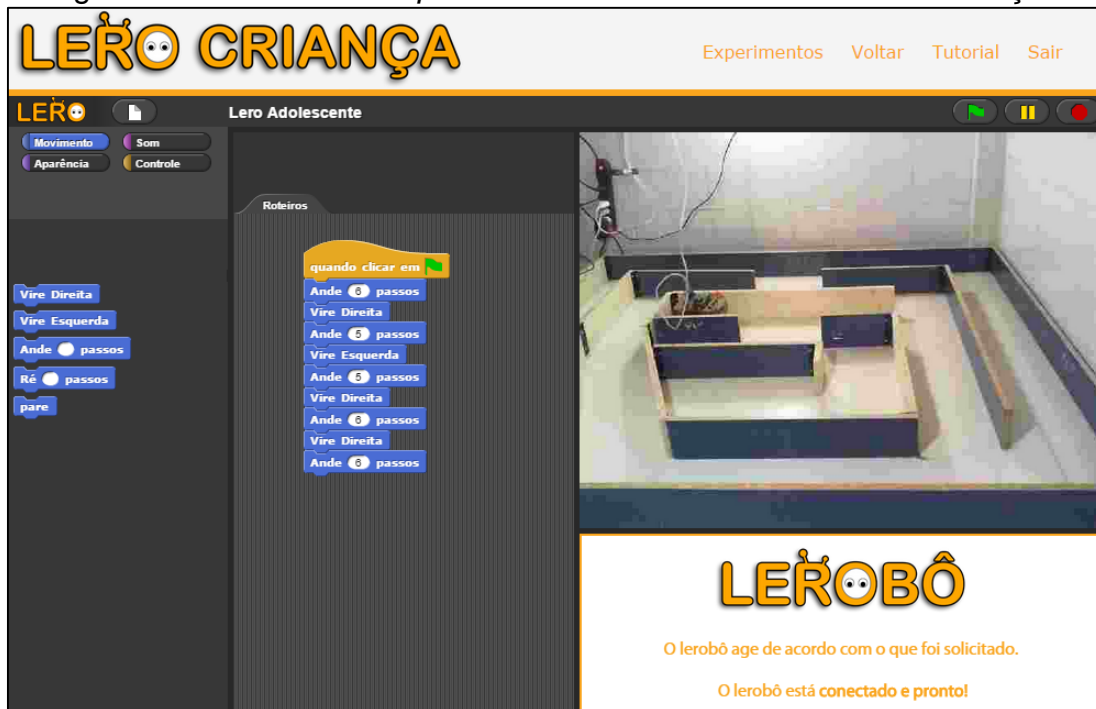
Figura 5.5 – Tela com mensagem de erro no LERO Criança



Fonte: Autoria própria (2015).

Um *script* onde não houvesse nenhuma interrupção nas ações programadas também foi criado para a execução do Lerobô. Nesse *script*, os blocos foram definidos para que o Lerobô ande seis passos, vira à direita, ande cinco passos, vire à esquerda, ande cinco passos, vire à direita, ande seis passos, vire à direita e por fim ande mais seis passos, chegando ao meio do labirinto. A Figura 5.6 mostra o Lerobô na posição de início para a execução dos blocos do *script*, criado no roteiro, assim que a bandeira verde for acionada.

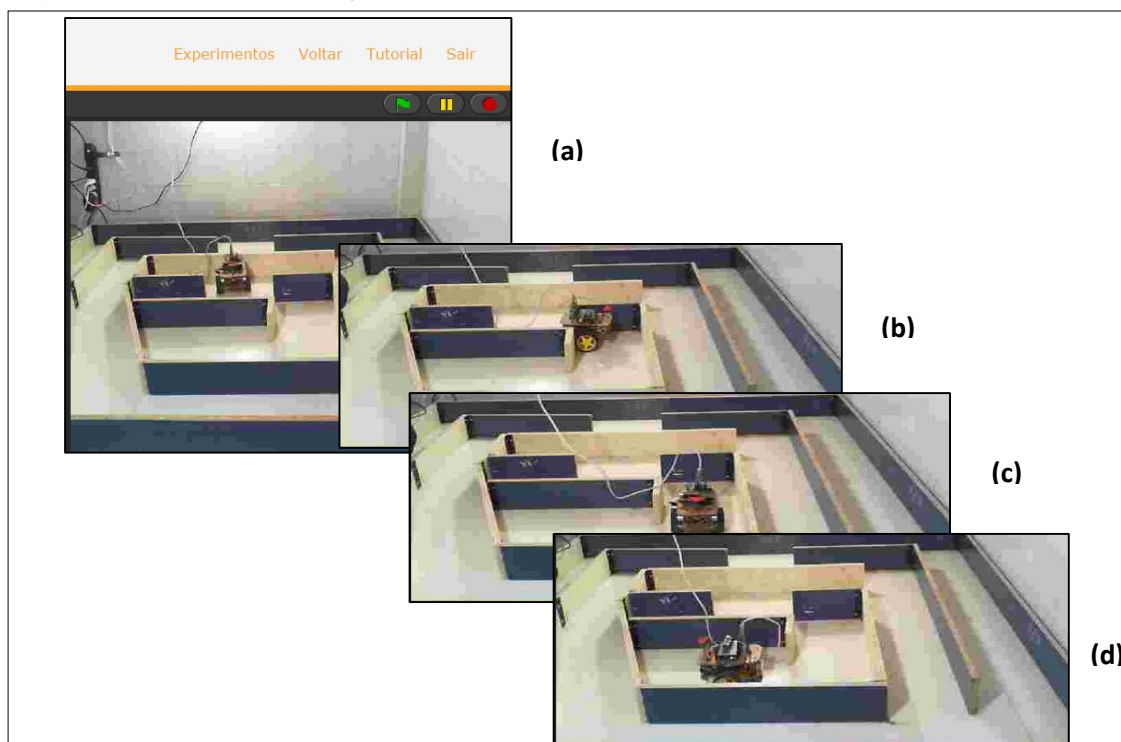
Figura 5.6 – Tela com *script* com sucesso no roteiro do LERO Criança



Fonte: Autoria própria (2015).

A Figura 5.7 mostra a posição do robô após a execução de alguns blocos do *script*: após a execução dos blocos ande seis passos e vire à direita (Figura 5.7a);

Figura 5.7 – Tela com ações iniciais do *script* com sucesso na Plataforma Robótica



Fonte: Autoria própria

A cessão de figuras mostra uma sucessão de execuções dos blocos, são elas: ande cinco passos e vire à esquerda e ande cinco passos (Figura 5.7b); após a execução dos blocos vire à direita e ande seis passos (Figura 5.7c) e finalmente vire à direita e ande seis passos (Figura 5.7d). Após a execução dos comandos solicitados a luz verde apaga ao redor do *script*, indicando assim o final do experimento.

A próxima seção irá mostrar como, de forma prática, um usuário pode criar um novo bloco para a biblioteca do LERO a partir do LERO Extremo e disponibilizá-lo para os demais ambientes.

5.2 DE LERO PARA LERO

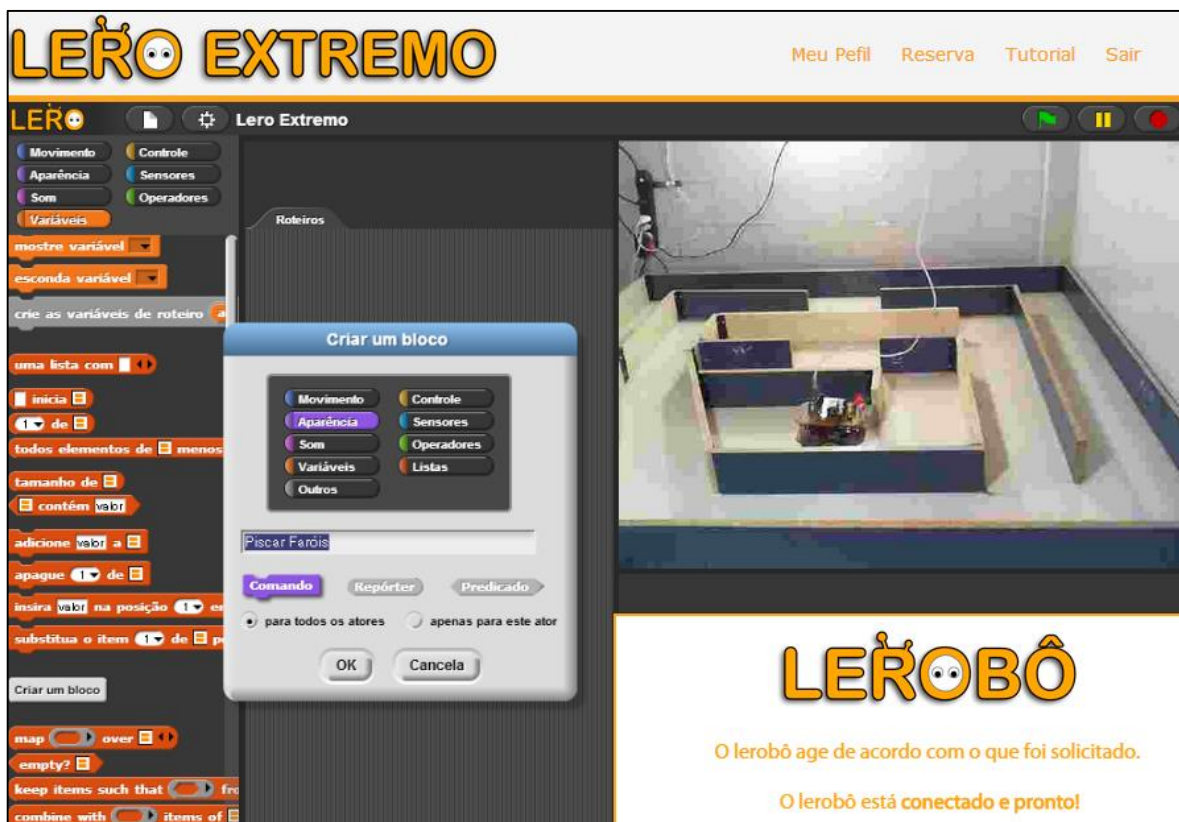
Uma das funcionalidades criativas do projeto LERO é a possibilidade do usuário mais avançado em programação e robótica criar seus próprios blocos no ambiente. O perfil LERO Extremo permite que um *script* possa ser construído no roteiro, testado no experimento físico e disponibilizado como um novo bloco na biblioteca do ambiente LERO. Essa funcionalidade caracteriza o LERO como um ambiente extensível, pois permite ampliar o seu uso a partir da criatividade do usuário.

A título de exemplo serão apresentados os passos para criação de um novo bloco chamado: Piscar Faróis N Vezes. Para criar um novo bloco é necessário que o usuário no ambiente LERO Extremo, vá ao grupo de funções chamado 'Variáveis' e clique em 'Criar um bloco'. Após clicar nessa opção irá aparecer uma nova janela onde o usuário pode escolher em que grupo de funções esse novo bloco ficará alocado, nesse caso ficará no grupo 'Aparência'. O usuário deverá inicialmente, especificar um nome para o bloco (Figura 5.8).

Após criar um novo bloco podem ser acrescentados novos nomes ou parâmetros ao mesmo. No exemplo, foram inseridos o parâmetro "N" e o nome "Vezes", finalizando assim o formato do novo bloco elaborado.

Como o novo bloco foi inserido no grupo de funções 'Aparência', quando ele for selecionado, o novo bloco estará lá, mas ainda sem nenhuma funcionalidade associada. Para inserir os comandos que irão determinar a execução do novo bloco, o usuário deverá clicar com o botão direito em cima do bloco e selecionar a opção editar.

Figura 5.8 – Tela para criar um novo bloco no LERO Extremo



Fonte: Autoria própria (2015).

O processo de edição permite que sejam utilizados blocos de qualquer grupo. No caso do exemplo, para estruturar o *script* do novo bloco Piscar Faróis N Vezes foram utilizados blocos dos grupos ‘Sensores’, ‘Controle’ e Variáveis. Do grupo ‘Variáveis’ foi criada uma variável chamada faróis que recebeu o número do pino em que os LEDs estão ligados no Arduino, nesse caso o pino dezessete. Este pino foi definido como saída. Após esse comando serão repetidas N vezes a sequência de comandos: mude o valor do pino para um, após 0,2 segundos, mude o valor do pino para zero e por fim espere mais 0,2 segundos. Esse conjunto de instruções faz com que os LEDs ligados ao pino dezessete ascendam e apaguem em intervalos de 0,2 segundos quantas vezes o usuário especificar (Figura 5.9).

Figura 5.9 – Tela com o *script* para funcionamento do novo bloco no LERO Extremo



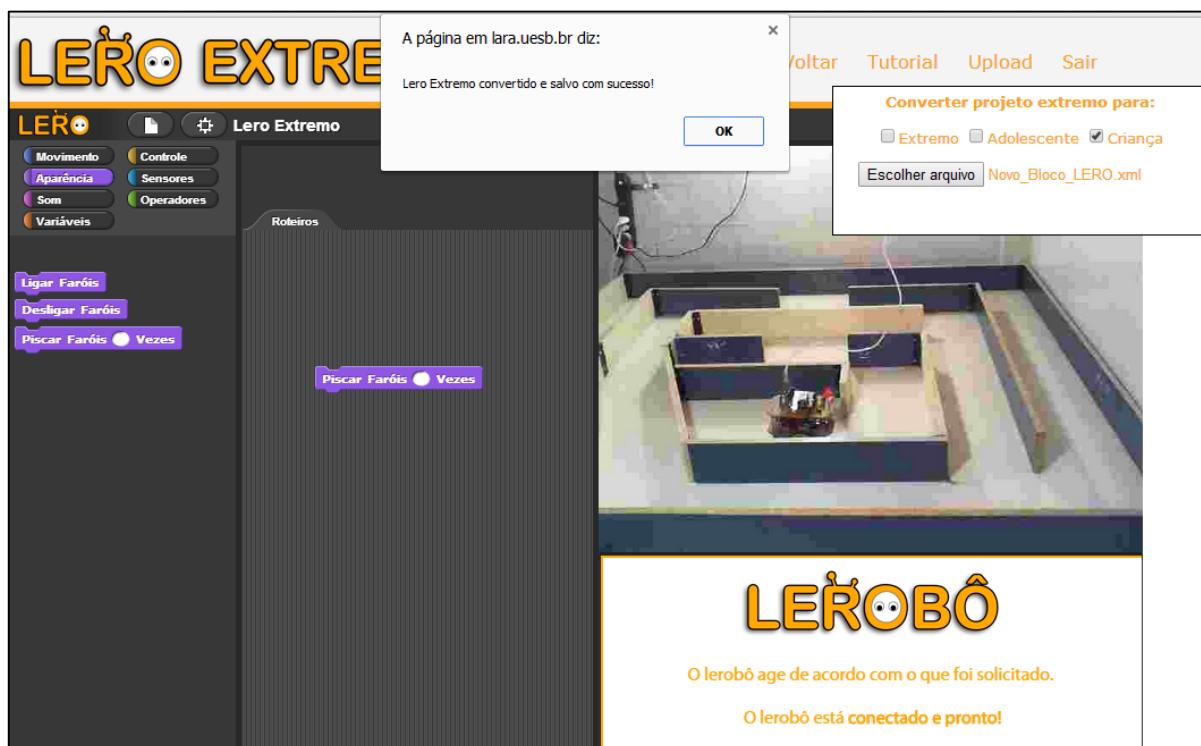
Fonte: Autoria própria (2015).

Após montar o *script*, o novo bloco estará pronto para ser testado no experimento físico. Após o teste e a verificação de que o novo bloco está funcionando corretamente, o usuário poderá fazer o upload do projeto para ser enviado ao ambiente escolhido para acomodar a nova funcionalidade. Nesse caso, o ambiente escolhido foi o LERO Criança.

Para fazer o upload do projeto criado no LERO Extremo com o novo bloco é preciso clicar no menu superior, na imagem de uma folha de caderno e escolher a opção: Exportar este projeto. Quando essa opção é escolhida irá abrir uma nova aba no seu navegador, logo ao lado da aba que está aberta o LERO Extremo. Clicando nessa nova aba, o usuário deverá clicar novamente em cima do código com o botão direito e escolher a opção Salvar como. O usuário deverá dar um nome ao novo projeto e armazenar o arquivo no computador como tipo XML. No exemplo foi escolhido o nome Novo_Bloco_LERO.

Quando o arquivo é salvo no computador, o usuário deve voltar à opção Upload do LERO Extremo e escolher o arquivo do novo projeto salvo localmente no seu computador. O usuário também poderá selecionar para qual perfil o projeto com o novo bloco será submetido. Ao enviar as informações uma mensagem será exibida ao usuário que irá perceber que a ação foi executada com sucesso (Figura 5.10).

Figura 5.10 – Tela com a mensagem de sucesso do Upload do novo projeto no LERO Extremo



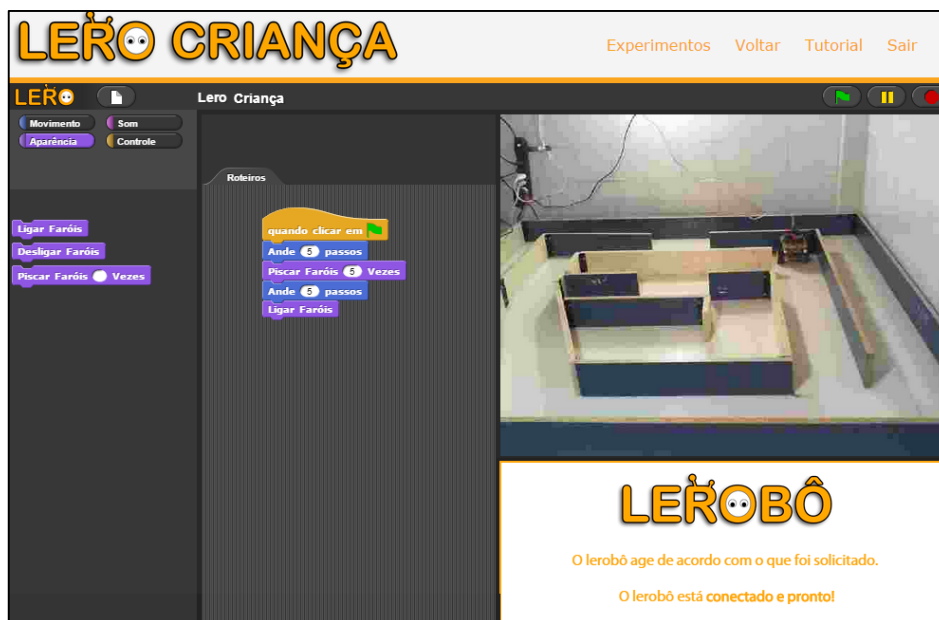
Fonte: Autoria própria (2015).

Quando o usuário retornar ao LERO Criança, o novo bloco já estará disponível para todos que acessarem esse perfil. Com essa característica o LERO demonstra ser um ambiente extensível, pois permite que os usuários contribuam interagindo, aprendendo e criando novas funcionalidades. A próxima seção mostrará o novo bloco criado através do próprio ambiente, Piscar Faróis N Vezes, sendo utilizado em um *script* construído no roteiro do LERO Criança para execução do Lerobô.

5.3 MAIS LERO EM AÇÃO

Para mostrar o novo bloco em ação um *script* foi criado no roteiro em que o Lerobô deve andar cinco passos, piscar os faróis cinco vezes, andar mais cinco passos e por fim ligar os faróis (Figura 5.11).

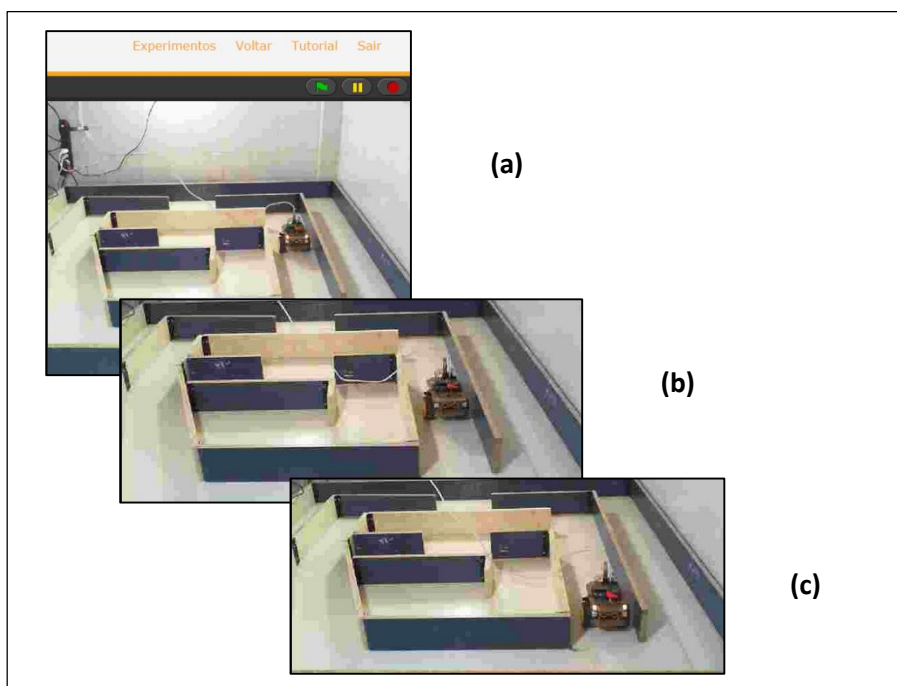
Figura 5.11 – Tela com *script* utilizando o novo bloco no roteiro do LERO Criança



Fonte: Autoria própria (2015).

As Figuras 5.12a e 5.12b mostram o Lerobô, após o clique na bandeira verde, na posição após a execução dos dois primeiros blocos, andar cinco vezes e piscar os faróis cinco vezes.

Figura 5.12 – Tela com ações iniciais do *script* utilizando o novo bloco do LERO Criança



Fonte: Autoria própria (2015).

A Figura 5.12c mostra o Lerobô após andar mais cinco vezes e mantendo os faróis ligados.

Um vídeo do experimento acima com o novo bloco foi disponibilizado no endereço lara.uesb.br/lero/video.mp4 para uma melhor visualização de como o Lerobô executa as ações descritas anteriormente.

5.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou algumas funcionalidades do ambiente de gerenciamento do laboratório remoto e possibilidades de uso do ambiente de desenvolvimento do LERO em seus diferentes perfis. Foram utilizados nos exemplos duas versões do ambiente cliente, enfatizando a característica de adaptabilidade do LERO. Foi mostrado também como criar um novo bloco no ambiente LERO Extremo e como disponibilizar este bloco para outros ambientes LERO. Esta demonstração foi feita para enfatizar o caráter extensível do LERO.

6 CONCLUSÕES E TRABALHOS FUTUROS

A robótica é uma potente ferramenta tecnológica de ensino e aprendizagem, mas sua aplicabilidade esbarra com algumas limitações como custo e disponibilidade. Ainda que as instituições de ensino estejam investindo nessa tecnologia, o custo dos kits de robótica no mercado varia substancialmente e as opções mais acessíveis muitas vezes necessitam de conhecimentos avançados na área de eletrônica e computação.

Os laboratórios remotos de robótica objetivam diminuir as limitações no acesso a essa tecnologia, disponibilizando experimentos físicos reais à distância para que usuários possam manipular exercitando seu aprendizado e também a criatividade. Alguns laboratórios remotos foram selecionados para análise nesse trabalho. Verificou-se que as iniciativas mais bem sucedidas utilizam algum ambiente de programação em blocos para configurar as ações dos robôs. Dentre as deficiências encontradas destaca-se o fato dos ambientes serem direcionados a um público-alvo restrito, com perfil bem definido e com a incapacidade de agregar novas funcionalidades às funcionalidades básicas ofertadas pelos sistemas.

Diante das deficiências encontradas nos trabalhos relacionados, O LERO foi desenvolvido baseado em duas características diferenciais: a adaptabilidade e a extensibilidade. Ser adaptável envolve adequar as funcionalidades do ambiente ao nível de conhecimento e habilidades do usuário. Para o LERO a idade e a habilidade em programação e robótica foram os critérios escolhidos para a construção do perfil do usuário. A extensibilidade permite que usuários experientes, através do próprio LERO, construam seus próprios blocos, teste no experimento físico, e façam o upload do seu projeto para que o seu novo bloco pertença à biblioteca do LERO. Esse processo de interação do sujeito com o ambiente produz e dissemina novos conhecimentos.

O desenvolvimento do LERO envolveu a criação e adaptação de soluções tecnológicas dentre elas destacam-se o Snap!, o s2a e Arduino. Os requisitos funcionais e não funcionais foram definidos, o modelo de dados construído e a partir disso foram implementadas as funcionalidades do LERO. Para avaliar o resultado, o ambiente foi submetido a um teste de usabilidade, através da aplicação das heurísticas de Nielsen. Com essa avaliação foram apontadas algumas

inconsistências que limitavam a usabilidade do ambiente LERO. Essas inconsistências foram avaliadas e apesar de algumas delas serem consideradas como erros gravíssimos, todas foram analisadas e a maioria corrigida.

A partir da pesquisa realizada e dos resultados obtidos, como trabalhos futuros pretende-se:

- a) Concluir um trabalho, já em andamento, de um estudo de caso do LERO utilizando o perfil LERO Extremo. Trabalho de conclusão de curso sendo desenvolvido por uma aluna de graduação que participou do projeto LERO;
- b) Refinar os movimentos de curva do Lerobô através da implantação de um encoder na estrutura física do robô;
- c) Implementar a funcionalidade de aprovação prévia de novos projetos pelo administrador, antes de disponibilizar esse novo projeto na biblioteca do LERO;
- d) Finalizar o desenvolvimento de alguns blocos do LERO Adolescente e disponibilizá-los no ambiente;
- e) Implementar a funcionalidade que permite ao robô voltar a uma posição inicial da arena quando um novo usuário se conectar ao ambiente de desenvolvimento;
- f) Disponibilizar outras modalidades de experimentos na plataforma robótica com novos projetos que comportem novos blocos da biblioteca LERO para controlar esses experimentos;
- g) Permitir o acesso simultâneo de vários usuários ao laboratório e disponibilizar um chat para comunicação entre os mesmos.

Acredita-se que o LERO contribua substancialmente na área da robótica educacional possibilitando o acesso de estudantes e simpatizantes da área a experimentos diversos com diferentes níveis de aprendizado. Nesse sentido, visando a disseminação e continuidade do projeto pretende-se disponibilizá-lo brevemente ao público através de um site criado especificamente para o LERO.

REFERÊNCIAS

- BAGNALL, B. *Maximum LEGO NXT: Building Robots with Java Brains*. [S.l.]: Variant Press, 2007.
- BARBOSA, S. D.; SILVA, B. S. *Interação humano-computador*. 9. ed. [S.l.]: Elsevier, 2010.
- BRITO, G. D. ; PURIFICAÇÃO, I. D. *Educação e novas tecnologias: um repensar*. 1. ed. [S.l.]: IBPEX, 2006.
- CITILAB. *Citilab*. Disponível em: <Citilab: <http://citilab.eu/>> Acesso em: 6 jun. 2015.
- CORTER, J. E. et al. Process and learning outcomes from remotely-operated, simulated, and hands-on student laboratories. *Computers & Education*, p. 2054–2067, 2011.
- DEBIAN. *Debian - O Sistema Operacional Universal*. Disponível em: <<https://www.debian.org/>> Acesso em: 10 ago. 2015.
- FABRÍCIO, P. R. de A. M.; COSTA NETO, O. E. da; DE SOUSA, E. L. de. Utilização da robótica na educação: uma realidade no Município de Solânea - PB. *Nuevas Ideas en Informática Educativa TISE*, v.10, p. 857 - 860, 2014.
- FIORIO, R. et al. Uma experiência prática da inserção da robótica e seus benefícios como ferramenta educativa em escolas públicas. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (CBIE 2014), 3., SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2014), 25., 2014. *Anais...* 2014.p.1223-1232.
- FIRMATA, A. *Arduino - Firmata*. 2015. Disponível em: <<https://www.arduino.cc/en/reference/firmata>> Acesso em: 10 ago. 2015.
- FRANÇA, R. S. ; AMARAL, H. J. Ensino de Computação na Educação Básica no Brasil: um mapeamento sistemático. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 33. 2013. *Anais...* 2013. p. 426 - 431.
- FRIEDRICH, R. V. et al. Proposta Metodológica para a Inserção ao Ensino de Lógica de Programação com Logo e Lego Mindstorms. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2012). 23., 2012. *Anais...* 2012
- GITHUB. *LLK/scratch-flash GitHub*. 2015. Disponível em: <<https://github.com/LLK/scratch-flash>> Acesso em: 6 jun. 2015.
- KHAMIS, A. M.; RODRÍGUEZ, F. J.; SALICHS, M. A. Remote Interaction with Mobile Robots. *Autonomous Robots*, p.267–281, 2003.
- LEGO. 2015. Disponível em: <<http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com>> Acesso em: 6 jun. 2015.
- MARTINS, A. *O que é Robótica?* São Paulo: Brasiliense, 2006.

MARTINS, A. R. *Usando o Scratch para potencializar o pensamento criativo em crianças do Ensino Fundamental*. 2012. Dissertação (Mestrado em Educação)- Universidade de Passo Fundo, Passo Fundo, 2012.

MÉLO, F. É. et al. Do scratch ao arduino: uma proposta para o ensino introdutório de programação para cursos superiores de tecnologia. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA - COBENGE 2011, 39., 2011. *Anais...* 2011.

MICROSOFT. *Microsoft WebCam: LifeCam HD 3000*. 2015.
<<https://www.microsoft.com/hardware/pt-br/p/lifecam-hd-3000#details>> Acesso em: 19 jun. 2015.

MIRANDA, L. C.; SAMPAIO, F. F.; BORGES, J. A. RoboFácil: especificação e implementação de um kit de robótica para a realidade educacional brasileira. *Revista Brasileira de Informática na Educação*, 18, p. 46- 58, 2010.

OLIVEIRA, G. A. et al. GrubiBots Educacional: jogo para o ensino de algoritmos na educação básica. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (CBIE 2014), 3., SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2014), 25., 2014. *Anais...* 2014.p.584 - 592.

OLIVEIRA, M. L. et al. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. n: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO - CSBC, 36., WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO - WEI, 22., 2014. *Anais...* 2014. p. 1525 - 1534.

PAPERT, S. *Logo: computadores e educação*. 2. ed. São Paulo: Brasiliense, 1986.

PAPERT, S. *A máquina das crianças: repensando a escola na era da informática*. Porto Alegre: Artes Médicas, 1994.

DE SOUZA PIO, J. L.; DE CASTRO, T. H. C.; DE CASTRO JÚNIOR, A. N. A robótica móvel como instrumento de apoio à aprendizagem de computação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 18., 2006. *Anais...* p.197-206, 2006.

POPESCU, D. et al. Tele-programming for a mobile robot via internet. In: PWSEAS INTERNATIONAL CONFERENCE ON APPLIED INFORMATICS AND COMMUNICATIONS, 7., 2007. *Proceedings...* 2007. p. 217-222. v.7.

PRESSMAN, R. S.; LOWE, D. *Engenharia WEB*. 1. ed. São Paulo: LTC, 2009.

QUINTANILHA, L. Irresistível Robô. *Revista ARede*, ano 3, p. 10-17, mar. 2008. Disponível em: <<http://www.revista.aredes.inf.br/site/edicao-n-34-marco-2008/3920-irresistivel-robo>> Acesso em: 1 jun. 2014.

s2a. 2015. Disponível em: <https://github.com/MrYsLab/s2a_fm> Acesso em: 21 maio 2015.

S4A. S4A. 2015. Disponível em: < <http://s4a.cat/>> Acesso em: 6 jun. 2015.

SCRATCH. 2015. *Scratch Project Editor*. Disponível em: <https://scratch.mit.edu/projects/editor/?tip_bar=getStarted> Acesso em: 5 jun. 2015.

SIEVERS JUNIOR, F. et al. WebLab um laboratório de acesso remoto controlado através da Internet. Um estudo de caso na Logística. In: CONGRESSO DE LOGÍSTICA DAS FACULDADES DE TECNOLOGIA DO CENTRO PAULA SOUZA, 4., 2010, Jundiaí. *Anais...* Jundiaí: Centro Paula Souza, 2010.

SILVA, A. F. et al. *LARA - Laboratório Remoto em AVA*. [Itapetinga - BA]: [UESB], 2014. (Projeto de Pesquisa. Departamento de Ciências Exatas e Tecnológicas - UESB).

SILVA, S. R.; BARRETO, L. P. Análise comparativa de kits de robótica educativa. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 39., 2011. *Anais...* 2011.

SMIDT, A. C. *Implementação de uma Plataforma Robótica controlada remotamente utilizando Arduino*. 2013. Trabalho de Conclusão de Curso (Curso de Engenharia de Computação) - USP, 2013.

SNAP. *Snap! (Build Your Own Blocks) 4.0*. 2015. Disponível em: <<http://snap.berkeley.edu/>> Acesso em: 21 maio 2015.

SNAP-BUILD_YOUR-OWN-BLOCKS, G. 2015. *Snap-Build_Your-Own-Blocks GitHub*. Disponível em: <<https://github.com/jmoenig/Snap--Build-Your-Own-Blocks>> Acesso em: 6 jun. 2015.

SOUZA, A. L.; OLIVEIRA, J. C. Laboratórios acessíveis via Internet: um recurso didático para o ensino/aprendizado de Engenharia Elétrica. In: *IEEE ENCONTRO DE EDUCAÇÃO EM ENGENHARIA*, 7., 2001. *Anais...* 2001.

SOUZA, C.; COSTA FILHO, J. T. Laboratório de acesso remoto para ensino orientado a experimentos aplicado em aprendizado a distância e presencial em Engenharia. In: *CONFERÊNCIA INTERNACIONAL DE EDUCAÇÃO EM ENGENHARIA E TECNOLOGIA (INTERTECH 2002)*, 7., 2002. *Anais...* 2002.

SOUZA, M. B. et al. Arcabouço de um ambiente telerobótico educacional baseado em sistemas multiagente. In: *SBIE - XVII WIE*, 22., 2011. *Anais...* 2011. p. 680-689.

TEIXEIRA, C. C. et al. Laboratório real remoto via internet aplicado a robótica móvel. In: *ENCONTRO LATINO AMERICANO DE INICIAÇÃO CIENTÍFICA*, 10., *ENCONTRO LATINO AMERICANO DE PÓS-GRADUAÇÃO – UNIVERSIDADE DO VALE DO PARAÍBA*, 6., 2006. *Anais...* 2006. p. 2430 - 2433.

TEIXEIRA, C. C. et al. Laboratório Real remoto via internet aplicado a robótica móvel. In: *ENCONTRO LATINO AMERICANO DE INICIAÇÃO CIENTÍFICA*, 10., *ENCONTRO LATINO AMERICANO DE PÓS-GRADUAÇÃO – UNIVERSIDADE DO VALE DO PARAÍBA*, 6., 2006. *Anais...* 2006. p. 2430 - 2433.

TZAFESTAS, C. S. ; PALAIOLOGOU, N.; ALIFRAGIS, M. Virtual and remote robotic laboratory: comparative experimental evaluation. *IEEE transactions on education*, p. 360 - 369, ago. 2006.

YORINKS, A. *A Scratch 2.0 and Snap! 4.0 Hardware Extension for Arduino Micro-Controllers*. [S.l.]: [s.n.], 2014.

ZILLI, G. M. ; LAMBERT, G. Desenvolvendo a educação através da robótica móvel: uma proposta pedagógica para o ensino de engenharia. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 38., 2010. *Anais...* 2010.

ANEXO A – AVALIAÇÃO DE USABILIDADE

Interface Avaliada: Projeto Lero – LERO Criança	
Avaliador: Kênia Xavier Teodoro de Oliveira	Data: 15/8/2015
Avaliação Heurística – Conjunto de 10 Heurísticas de Nielsen	

Escala de erros

Grau	Tipo	Descrição
Grau 0:	Sem importância	Não é encarado necessariamente como um problema de usabilidade.
Grau 1:	Cosmético	Problema estético; não necessita ser corrigido, a menos que haja tempo.
Grau 2:	Simples	Problema menor de usabilidade; O conserto deste problema é desejável, mas deve receber baixa prioridade – alguns usuários ficarão insatisfeitos em momentos específicos.
Grau 3:	Grave	Problema maior de usabilidade; alta prioridade para sua correção. Importante ser consertado - muitos usuários ficarão muito insatisfeitos.
Grau 4:	Catastrófico	Catástrofe de usabilidade, ou seja, o produto só será liberado se a correção for feita. É imperativo consertar este problema antes do lançamento do produto – muitos usuários não conseguirão atingir seus objetivos (para eles, o produto não funciona)

Tabela 1: Nível de severidade dos problemas de usabilidade

Tabela 2 - Heurísticas de Nielsen – Sessão de Avaliação da Autoria

<p>1. Visibilidade do status do sistema</p>
<p>Verificação: Os usuários são mantidos informados sobre o progresso do sistema com apropriado feedback em um tempo razoável?</p>
<p>Problema: No ambiente do laboratório remoto, ao concluir o bloco de comandos o usuário é informado da conclusão da ação através da mudança da cor do bloco, entretanto, entre o intervalo do envio do comando e execução da tarefa pelo robô, não há feedback para o usuário sobre o status do sistema. (Catastrófico – Grau 4)</p>

<p>Solução proposta: Implementar uma barra de progresso ou outro elemento que forneça ao usuário uma resposta enquanto ocorre a execução da tarefa.</p>
<p>2. Compatibilidade entre o sistema e o mundo real</p>
<p>Verificação: O sistema utiliza conceitos e linguagem familiar com o usuário em vez de termos orientados ao sistema? O sistema utiliza convenções do mundo real, exibindo informações com uma ordem lógica e natural?</p>
<p>Problema:</p>
<p>Solução proposta:</p>
<p>3. Liberdade e controle do usuário</p>
<p>Verificação: Os usuários podem fazer o que querem e quando querem?</p>
<p>Problema 1: Na tela de login, o sistema não oferece opção de lembrete, dica, pergunta secreta ou qualquer outro mecanismo para recuperação de senha. (Catastrófico – Grau 4)</p>
<p>Problema 2: Na tela de reserva do experimento, ao reservar um horário inferior ao atual, o sistema não oferece mensagem de erro e confirma a reserva, entretanto, a mesma não aparece na tabela de reservas realizadas, mas aparece juntamente com as demais reservas feitas, no histórico de reservas realizadas. (Grave – Grau 3)</p>
<p>Problema 3: Ao excluir uma reserva, antes da execução da tarefa, o sistema não oferece ao</p>

usuário uma mensagem de confirmação da exclusão (alerta), que permita desfazer/cancelar a ação. (Grave – Grau 3)

Solução proposta 1: Implementar a recuperação de senha na tela de login.

Solução proposta 2: O sistema não deve permitir a realização de reservas em horários/data inferiores aos atuais e deve emitir uma mensagem de alerta/erro ao usuário dando orientações sobre como proceder.

Solução proposta 3: Inserir uma mensagem de erro (alerta) permitindo confirmar ou cancelar a ação.

4. Consistência e Padronização


Verificação:

O projeto de elementos como objetos e ações tem o mesmo significado ou efeito em diferentes situações?

Problema 1: A opção “Meu perfil” não fica explícita como link, uma vez que as opções “Lero Criança” e “Acessar laboratório” foram desenhadas no mesmo padrão e não são links. Essa ação induz o usuário a passar para outra tela sem clicar na opção de escolha de perfil. (Grave – Grau 3)



Problema 2: A opção “Acessos ao laboratório” está “avulsa” na página. Não tem conexão com outros elementos, não é link e não está relacionada a texto ou tabela. (Simples – Grau 2)



LEÃO [Meu Perfil](#) [Minha Conta](#) [Reserva](#) [Histórico](#) [Ajuda](#) [Sair](#)

Histórico

Reservas realizadas

07/08/2015	17:00:00	Labirinto
10/08/2015	11:00:00	Labirinto
10/08/2015	21:00:00	Labirinto
11/08/2015	10:00:00	Labirinto
11/08/2015	15:00:00	Labirinto
12/08/2015	00:00:00	Labirinto
17/08/2015	20:00:00	Labirinto
26/08/2015	11:00:00	Labirinto

Acessos ao laboratório

Solução proposta 1: Deixar a aba da opção "Alterar perfil" expandida ou colocar uma imagem que sugira que há conteúdo oculto (+). No menu horizontal a palavra "Perfil" está escrita de forma incorreta (Pefil).

Solução proposta 2: Mesmo que o preenchimento do conteúdo esteja relacionado à realização de reserva pelo usuário, recomenda-se deixar uma mensagem do tipo "Nenhum acesso registrado" ou outra que indique que haverá conteúdo a ser registrado mediante execução de uma ação.

5. Prevenção contra erros

Verificação:

Os usuários podem cometer erros que poderiam ser prevenidos por um projeto melhor?

Problema: No formulário "Dados Gerais", a mensagem de erro retornada no preenchimento do campo "idade" não auxilia o usuário a corrigir o erro cometido, como por exemplo, digitar letras no lugar de números. (Grave – Grau 3)



LEÃO [Meu Perfil](#) [Minha Conta](#) [Reserva](#) [Histórico](#) [Ajuda](#) [Sair](#)

Minha Conta

Dados Gerais

Não deixe campos vazios!

Nome:

Idade:

Escolaridade:

Instituição:

Solução proposta: Melhorar a mensagem de erro de forma a auxiliar o usuário a resolver o problema.

6. Reconhecimento em lugar da lembrança
Verificação: Os elementos de projeto como objetos, ações e opções são possíveis? O usuário é forçado a relembrar informações de uma parte do sistema para outra?
Problema:
Solução proposta:
7. Flexibilidade e eficiência de uso
Verificação: As tarefas de usuário são eficientes e podem se adaptar ao gosto do usuário em suas ações mais frequentes ou ele utiliza atalhos?
Problema:
Solução proposta:
8. Estética e Design Minimalista
Verificação: Os diálogos contêm informações irrelevantes ou raramente necessárias?
Problema:
Solução proposta:
9. Auxiliar os usuários a reconhecer, diagnosticar e recuperar-se de erros
Verificação: As mensagens de erros são expressas em linguagem simples (sem códigos) descrevendo exatamente o problema e sugerindo uma solução?
Problema:
Solução proposta:
10. Ajuda e documentação
Verificação: São fornecidas apropriadas informações de ajuda e estas informações são fáceis de procurar e de focalizar nas tarefas do usuário?
Problema:
Solução proposta:

Referência:

CYBIS, Walter; BETIOL, Adriana Holtz; FAUST, Richard. **Ergonomia e usabilidade: conhecimentos, métodos e aplicações**. 2 ed. São Paulo: Novatec, 2010.

APENDICE A – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS DO LERO

Documento de Especificação de Requisitos

Ambiente LERO

Versão 5.0

2015

Histórico de Alterações

Data	Versão	Descrição	Autores
01/06/2015	1.0	Versão Inicial do Documento	Críjina Chagas
15/07/2015	2.0	Modificação nas Funcionalidades	Críjina Chagas
20/07/2015	3.0	Modificação nos Casos de Uso	Críjina Chagas
22/07/2015	4.0	Modificação nos Requisitos Não Funcionais	Críjina Chagas
27/08/2015	5.0	Revisão do Documento	Críjina Chagas

Conteúdo

1. INTRODUÇÃO.....	119
1.1 FINALIDADE	119
1.2 CONTEXTO	119
2. Descrição Geral	120
2.1 STAKEHOLDERS DO PROJETO	120
2.2 CLIENTES DO AMBIENTE LERO	120
3. Principais Funcionalidades	120
4. Requisitos Funcionais	123
4.1 DESCRIÇÃO DOS REQUISITOS FUNCIONAIS	123
[RF001] Acessar informações do projeto LERO	123
[RF002] Acessar o Ambiente LERO	124
[RF003] Inserir Usuário.....	124
[RF004] Alterar Usuário	124
[RF005] Visualizar perfil definido pelo sistema	124
[RF006] Alterar perfil.....	124
[RF007] Acessar módulo reserva	124
[RF008] Acessar módulo usuário	125
[RF009] Visualizar histórico de acessos.....	125
[RF010] Acessar módulo ambiente laboratório	125
[RF011] Reservar experimento	125
[RF012] Alterar reserva de experimento	125
[RF013] Excluir reserva de experimento	125
[RF014] Consultar reservas.....	126
[RF014] Acessar módulo ambiente laboratório	126
[RF016] Acessar a interface Snap! for LERO	126
[RF017] Acessar grupos de funções da interface	126
[RF018] Inserir blocos no roteiro da interface	126
[RF019] Alterar ordem dos blocos no roteiro da interface.....	126
[RF020] Excluir blocos no roteiro da interface.....	127
[RF021] Mandar executar os blocos do roteiro da interface	127
[RF022] Validar código gerado pelos blocos.....	127
[RF023] Visualizar ação do experimento.....	127
[RF024] Fazer upload de um projeto.....	127
[RF025] Acessar o módulo inicial do ambiente	127
[RF026] Acessar o tutorial do ambiente	128
[RF027] Acessar o módulo perfil do usuário	128
[RF028] Acessar o módulo reserva	128
[RF029] Permitir sair do ambiente LERO	128
[RF030] Manter usuário	128
[RF031] Manter perfil.....	128
[RF032] Manter reserva.....	129
[RF033] Manter upload	129
[RF034] Manter experimento	129
[RF035] Manter tutorial.....	129
5. Diagramas de Caso de Uso	130
5.1 DIAGRAMA DE CASOS DE USO.....	130
5.2 ESPECIFICAÇÃO DETALHADA DOS CASOS DE USO.....	131
6. Modelagem da Base de Dados	142
7. Requisitos Não Funcionais	143
7.1 USABILIDADE.....	143

<i>[RNF001] Interface Amigável</i>	<i>143</i>
<i>[RNF002] Componentes WEB.....</i>	<i>143</i>

Documento de Especificação de Requisitos

1. Introdução

O documento de requisitos é a especificação oficial dos requisitos do sistema para clientes, usuários finais e desenvolvedores de software.

Formalmente, podemos definir que o documento de requisitos contém: “*Os serviços e funcionalidades que o sistema deve prover; restrições; informações sobre o domínio da aplicação, bem como restrições no processo usado para desenvolver o sistema*”.

Além disso, tal documento pode ser visto como um contrato entre o cliente e o gerente de projeto, pois valida a conformidade segundo a especificação de requisitos do cliente para definição do escopo.

1.1 Finalidade

Este documento tem como finalidade definir o conjunto de requisitos a serem desenvolvidos para o Ambiente do Laboratório Remoto de Robótica - LERO.

1.2 Contexto

Destacada a possibilidade educativa na inserção de crianças e adolescentes no mundo da robótica, foi feita uma pesquisa para verificar soluções em ambiente WEB na estrutura de laboratório remoto de robótica destinada a esse público alvo. A autora do trabalho verificou que a escassez dessas soluções dificulta o acesso desse público alvo a esse mundo tão motivador e lúdico para estimular o conhecimento. Além disso, os kits de robótica disponíveis no mercado são caros e as peças para montagem de robôs, que têm os preços mais acessíveis, necessitam de conhecimentos avançados para manipulação. Mesmo escolas que já possuem esses kits de robótica têm restrições de dias e horários limitados para o acesso dos estudantes a esses materiais.

Considerando os problemas mencionados, este projeto tem como objetivo o desenvolvimento do aplicativo Laboratório Remoto de Robótica Educacional (LERO).

O LERO é uma aplicação em ambiente WEB que objetiva contribuir com o processo de ensino e aprendizagem de programação e robótica para diferentes faixas etárias, estimulando e diversificando a aquisição de conhecimentos de forma inovadora através de um sistema adaptável e extensível. São utilizados, nesse contexto, conceitos e metáforas de programação amplamente testadas e relatadas na literatura da área educacional.

2. Descrição Geral

2.1 Stakeholders do projeto

Nome	Responsabilidades no Projeto
Gerente de Projeto	Atribuições de caráter decisório e estratégico quanto aos rumos do projeto.
Analista de Sistemas	Definir e aprovar os requisitos e especificações do sistema e de interface do sistema.
Desenvolvedor	Definir arquitetura do sistema, programar e testar o sistema conforme as especificações.
Colaborador na área de IHC	Avaliar as interfaces do projeto de acordo com métricas de usabilidade definidas.

2.2 Clientes do Ambiente LERO

Nome	Descrição
Navegador	Acessam o ambiente e não são cadastrados no mesmo
Usuário	Cadastrados no ambiente e tem acesso a parte das suas funcionalidades
Administrador	Cadastrados no ambiente e tem acesso a todas as funcionalidades do mesmo

3. Principais Funcionalidades

Módulo Inicial			
Id	Funcionalidade	Descrição	Cliente
RF 001	Acessar informações do projeto LERO	Acessar informações disponibilizadas do ambiente LERO	Navegador
RF 002	Acessar o Ambiente LERO	Acessar o ambiente do laboratório remoto usando login e senha.	Navegador, Usuário e Administrador

Módulo Usuário			
Id	Funcionalidade	Descrição	Cliente
RF 003	Inserir Usuário	Cadastrar informações para acessar o ambiente LERO.	Usuário
RF 004	Alterar Usuário	Alterar informações cadastradas para acessar o ambiente LERO.	Usuário
Módulo Perfil			
Id	Funcionalidade	Descrição	Cliente
RF 005	Visualizar perfil definido pelo sistema	O usuário irá visualizar qual o perfil do ambiente LERO que foi definido pelo sistema para o usuário.	Usuário
RF 006	Alterar perfil	O usuário poderá modificar o perfil que irá acessar no ambiente LERO de acordo com sua habilidade.	Usuário
RF 007	Acessar módulo reserva	O usuário poderá acessar o módulo de reserva.	Usuário
RF 008	Acessar módulo usuário	O usuário poderá acessar o módulo usuário	Usuário
RF 009	Visualizar histórico de acessos	O usuário poderá visualizar o histórico de seus acessos ao ambiente LERO	Usuário
RF 010	Acessar módulo ambiente laboratório	O usuário poderá acessar a interface do laboratório remoto	Usuário
Módulo Reserva			
Id	Funcionalidade	Descrição	Cliente
RF 011	Reservar experimento	Inserir data e horário para utilizar um experimento remotamente.	Usuário
RF 012	Alterar reserva de experimento	Alterar a data e horário para utilizar um dispositivo remotamente.	Usuário
RF 013	Excluir reserva de experimento	Excluir a data e horário para utilizar um dispositivo remotamente.	Usuário
RF 014	Consultar reservas	Acessar o histórico de seus acessos ao ambiente LERO.	Usuário
RF 015	Acessar módulo ambiente laboratório	Acessar a interface do laboratório remoto.	Usuário

Módulo Ambiente Laboratório			
Id	Funcionalidade	Descrição	Cliente
RF 016	Acessar a interface Snap! for LERO	Programar usando a interface Snap! for LERO.	Usuário e Administrador
RF 017	Acessar grupos de funções da interface	Acessar os grupos de funções disponibilizados na interface do laboratório.	Usuário e Administrador
RF 018	Inserir blocos no roteiro da interface	Inserir blocos no roteiro disponibilizado na interface do laboratório.	Usuário e Administrador
RF 019	Alterar ordem dos blocos no roteiro da interface	Alterar a ordem dos blocos no roteiro disponibilizado na interface do laboratório.	Usuário e Administrador
RF 020	Excluir blocos no roteiro da interface	Excluir blocos no roteiro disponibilizado na interface do laboratório.	Usuário e Administrador
RF 021	Mandar executar os blocos do roteiro da interface	Mandar executar os blocos do roteiro da interface do laboratório	Usuário e Administrador
RF 022	Validar código gerado pelos blocos	Verificar se o código gerado pelos blocos pode ser executado pelo dispositivo (robô)	Sistema
RF 023	Visualizar ação do experimento	Assistir a execução online do experimento	Usuário e Administrador
RF 024	Fazer upload de um projeto	Fazer o upload de um projeto no ambiente.	Usuário e Administrador
RF 025	Acessar o módulo inicial do ambiente	Permitir o acesso ao módulo inicial do ambiente.	Usuário e Administrador
RF 026	Acessar o tutorial do ambiente	Permitir o acesso ao tutorial do ambiente.	Usuário e Administrador
RF 027	Acessar o módulo perfil do usuário	Permitir o acesso ao módulo perfil do ambiente.	Usuário e Administrador
RF 028	Acessar o módulo reserva do usuário	Permitir o acesso ao módulo reserva do ambiente.	Usuário e Administrador
RF	Permitir sair do	Permitir sair do ambiente.	Usuário e

029	ambiente LERO		Administrador
Módulo Administrativo			
Id	Funcionalidade	Descrição	Cliente
RF 030	Manter usuário	Inserir, atualizar e excluir usuários.	Administrador
RF 031	Manter perfil	Inserir e alterar perfis dos usuários.	Administrador
RF 032	Manter reserva	Inserir e alterar reservas dos usuários.	Administrador
RF 033	Manter upload	Permite autorizar ou não os uploads sugeridos pelos usuários para liberar na biblioteca LERO.	Administrador
RF 034	Manter experimento	Inserir, atualizar, bloquear e excluir dispositivos/experimento/equipamentos que podem ser acessados remotamente.	Administrador
RF 035	Manter tutorial	Inserir, atualizar e excluir arquivos de tutoriais disponibilizados aos perfis do ambiente LERO.	Administrador

4. Requisitos Funcionais

Nesta seção são detalhados os requisitos funcionais a serem contemplados pelo ambiente LERO, representados também através de Casos de Uso.

4.1 Descrição dos Requisitos Funcionais

Módulo: Módulo Inicial

[RF001] Acessar informações do projeto LERO

Descrição do requisito: Este requisito permite que o navegador acesse informações gerais sobre o projeto LERO.

Prioridade: Essencial Importante Desejável

[RF002] Acessar o Ambiente LERO

Descrição do requisito: Este requisito permite que o usuário acesse o ambiente LERO

Prioridade: Essencial Importante Desejável

Módulo: Módulo Usuário

[RF003] Inserir Usuário

Descrição do requisito: Este requisito permite inserir informações do usuário para acessar o ambiente LERO

Prioridade: Essencial Importante Desejável

[RF004] Alterar Usuário

Descrição do requisito: Este requisito permite alterar informações cadastradas do usuário.

Prioridade: Essencial Importante Desejável

Módulo: Módulo Perfil

[RF005] Visualizar perfil definido pelo sistema

Descrição do requisito: Este requisito permite inserir informações do usuário para acessar o ambiente LERO

Prioridade: Essencial Importante Desejável

[RF006] Alterar perfil

Descrição do requisito: Este requisito permite alterar informações do perfil definido para acessar o ambiente LERO de acordo com os dados do seu cadastro.

Prioridade: Essencial Importante Desejável

[RF007] Acessar módulo reserva

Descrição do requisito: Este requisito permite acessar informações do módulo que reserva períodos para acessar o ambiente do laboratório LERO.

Prioridade: Essencial Importante Desejável

[RF008] Acessar módulo usuário

Descrição do requisito: Este requisito permite acessar informações do módulo usuário para acessar dados cadastrados nesse módulo.

Prioridade: Essencial Importante Desejável

[RF009] Visualizar histórico de acessos

Descrição do requisito: Este requisito permite visualizar o histórico de acessos do usuário ao ambiente LERO.

Prioridade: Essencial Importante Desejável

[RF010] Acessar módulo ambiente laboratório

Descrição do requisito: Este requisito permite acessar o módulo do ambiente laboratório que disponibiliza a interface para execução de ações no experimento físico.

Prioridade: Essencial Importante Desejável

Módulo: Módulo Reserva

[RF011] Reservar experimento

Descrição do requisito: Este requisito permite reservar um experimento escolhido em um determinado período.

Prioridade: Essencial Importante Desejável

[RF012] Alterar reserva de experimento

Descrição do requisito: Este requisito permite alterar uma reserva de um experimento escolhido em um determinado período.

Prioridade: Essencial Importante Desejável

[RF013] Excluir reserva de experimento

Descrição do requisito: Este requisito permite excluir uma reserva de um experimento escolhido em um determinado período.

Prioridade: Essencial Importante Desejável

[RF014] Consultar reservas

Descrição do requisito: Este requisito permite consultar reservas em um determinado período.

Prioridade: Essencial Importante Desejável

[RF014] Acessar módulo ambiente laboratório

Descrição do requisito: Este requisito permite acessar o módulo do ambiente laboratório que disponibiliza a interface para execução de ações no experimento físico.

Prioridade: Essencial Importante Desejável

Módulo: Módulo Ambiente Laboratório

[RF016] Acessar a interface Snap! for LERO

Descrição do requisito: Este requisito permite acessar a interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF017] Acessar grupos de funções da interface

Descrição do requisito: Este requisito permite acessar os grupos de funções da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF018] Inserir blocos no roteiro da interface

Descrição do requisito: Este requisito permite inserir os blocos disponibilizados no roteiro da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF019] Alterar ordem dos blocos no roteiro da interface

Descrição do requisito: Este requisito permite alterar a ordem dos blocos no roteiro da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF020] Excluir blocos no roteiro da interface

Descrição do requisito: Este requisito permite excluir blocos no roteiro da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF021] Mandar executar os blocos do roteiro da interface

Descrição do requisito: Este requisito permite mandar executar os blocos no roteiro da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF022] Validar código gerado pelos blocos

Descrição do requisito: Este requisito permite validar código gerado pelos blocos no roteiro da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF023] Visualizar ação do experimento

Descrição do requisito: Este requisito permite visualizar as ações do experimento físico através da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF024] Fazer upload de um projeto

Descrição do requisito: Este requisito permite fazer um upload de um projeto através da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF025] Acessar o módulo inicial do ambiente

Descrição do requisito: Este requisito permite acessar o módulo inicial do ambiente LERO através da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF026] Acessar o tutorial do ambiente

Descrição do requisito: Este requisito permite acessar o tutorial do ambiente LERO através da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF027] Acessar o módulo perfil do usuário

Descrição do requisito: Este requisito permite acessar o módulo perfil do usuário através da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF028] Acessar o módulo reserva

Descrição do requisito: Este requisito permite acessar o módulo perfil do usuário através da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

[RF029] Permitir sair do ambiente LERO

Descrição do requisito: Este requisito permite sair do ambiente LERO através da interface do Snap! for LERO.

Prioridade: Essencial Importante Desejável

Módulo: Módulo Administrativo

RF030: Manter usuário

[RF030] Manter usuário

Descrição do requisito: Este requisito permite gerenciar (inserir, alterar, excluir e pesquisar) informações do usuário.

Prioridade: Essencial Importante Desejável

[RF031] Manter perfil

Descrição do requisito: Este requisito permite gerenciar (alterar e pesquisar) informações sobre o perfil do usuário.

Prioridade: Essencial Importante Desejável

[RF032] Manter reserva

Descrição do requisito: Este requisito permite gerenciar (inserir, alterar, excluir e pesquisar) reserva de experimentos do usuário.

Prioridade: Essencial Importante Desejável

[RF033] Manter upload

Descrição do requisito: Este requisito permite gerenciar (aceitar ou recusar) os uploads que foram enviados pelos usuários com o objetivo de serem disponibilizados na biblioteca do LERO.

Prioridade: Essencial Importante Desejável

[RF034] Manter experimento

Descrição do requisito: Este requisito permite gerenciar (inserir, alterar, bloquear e excluir) dispositivos/experimento/equipamentos que podem ser acessados remotamente na plataforma robótica.

Prioridade: Essencial Importante Desejável

[RF035] Manter tutorial

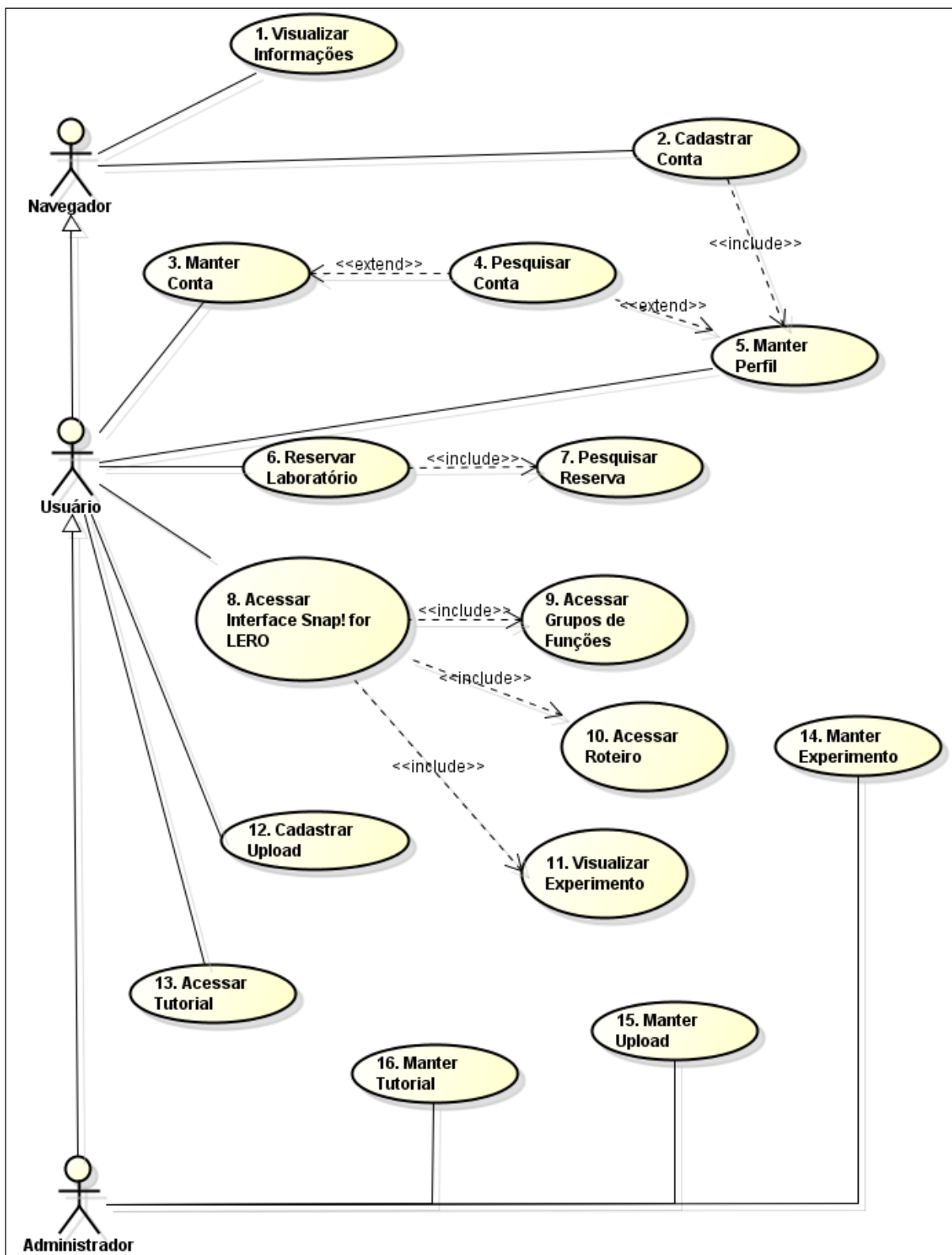
Descrição do requisito: Este requisito permite gerenciar (inserir, atualizar e excluir) arquivos de tutoriais disponibilizados aos perfis do ambiente LERO.

Prioridade: Essencial Importante Desejável

5. Diagramas de Caso de Uso

Nesta seção são detalhados os diagramas de casos de uso, composto pelos casos de uso e a especificação detalhada dos mesmos.

5.1 Diagrama de casos de uso



5.2 Especificação Detalhada dos Casos de Uso

A seguir são apresentadas as especificações detalhadas correspondentes a cada caso de uso.

UC1. Visualizar informações

<u>Finalidade:</u> o navegador acessa o site do projeto LERO e visualiza informações sobre o mesmo.
<u>Fluxo Principal</u> <ol style="list-style-type: none"> 1. O navegador acessa o site de projeto. 2. O sistema exibe as informações do site do projeto. 3. O caso de uso termina.
<u>Fluxos Alternativos</u> A1. <u>Cadastrar Conta</u> A1.1. O navegador seleciona cadastrar conta. A1.2. O sistema exibe UC2 A1.3. O caso de uso termina.
<u>Fluxos de Exceção</u> Não há.

UC2. Cadastrar Conta

<u>Finalidade:</u> o navegador cadastrar uma conta para acessar o ambiente.
<u>Fluxo Principal</u> <ol style="list-style-type: none"> 1. O sistema solicita os dados gerais (nome, idade, e-mail, escolarização e instituição) e dados de acesso (login e senha); 2. O navegador informa os dados correspondentes; 3. O sistema verifica se já existe se o e-mail cadastrado já existe na base de dados; 4. O sistema exibe a seguinte mensagem: "Sua conta foi criada com sucesso!"; 5. O sistema redireciona o usuário para o UC5; 6. O caso de uso termina.
<u>Fluxos Alternativos</u> Não há.
<u>Fluxos de Exceção</u> E1. <u>E-mail já cadastrado</u> E1.1. O sistema identifica que o e-mail informado já está cadastrado. E1.2. O sistema exibe a seguinte mensagem: "Esse e-mail já está cadastrado no ambiente!". E1.3. O caso de uso termina.

UC3. Manter Conta

Finalidade: gerenciar conta(s) cadastrada(s) no ambiente.

Fluxo Principal

Ator: Usuário

1. O sistema exibe os dados gerais (nome, idade, e-mail, escolarização e instituição) e dados de acesso (login e senha).
2. O navegador visualizar os dados correspondentes.
3. O caso de uso termina.

Ator: Administrador

1. O sistema exibe UC4.
2. O caso de uso termina.

Fluxos Alternativos

A1. Alterar Conta

A1.1. O navegador seleciona alterar conta.

A1.2. O sistema exibe os dados gerais (nome, idade, e-mail, escolarização e instituição) e dados de acesso (login e senha) cadastrados do usuário.

A1.3. O usuário informa os dados que serão alterados.

A1.4. O sistema verifica se já existe se o e-mail cadastrado já existe na base de dados.

A1.5. O sistema exibe a seguinte mensagem: “Sua conta foi alterada com sucesso!”.

A1.6. O caso de uso termina.

A1. Excluir Conta

A2.1. O usuário seleciona a opção excluir conta.

A2.2. O sistema exibe a seguinte mensagem: “Tem certeza que deseja excluir conta?”.

A2.3. O administrador clica em OK.

A2.4. O sistema exibe a seguinte mensagem: “Conta excluída com sucesso!”.

A2.5. O caso de uso termina.

A3. Visualizar Perfil

A3.1. O navegador seleciona visualizar perfil.

A3.2. O sistema exibe UC4

A3.3. O caso de uso termina.

Fluxos de Exceção

E1. E-mail já cadastrado

E1.1. O sistema identifica que o e-mail informado já está cadastrado.

E1.2. O sistema exibe a seguinte mensagem: “Esse e-mail já está cadastrado no

ambiente!”.

E1.3. O caso de uso termina.

UC4. Pesquisar Conta

Finalidade: pesquisar conta(s) cadastrada(s).

Fluxo Principal

1. O sistema disponibiliza filtros (nome, e-mail e login);
2. O administrador informa dados que serão buscados;
3. O sistema exibe dados encontrados;
4. O caso de uso termina.

Fluxos Alternativos

Não há.

Fluxos de Exceção

E1. Dados não encontrados

E1.1. O sistema identifica que os dados informados não foram encontrados.

E1.2. O sistema exibe a seguinte mensagem: “Dados não encontrados!”.

E1.3. O caso de uso termina.

UC5. Manter Perfil

Finalidade: gerenciar o perfil definido pelo ambiente.

Fluxo Principal

Ator: Usuário

1. O sistema exibe as informações sobre o seu perfil definido.
2. O usuário visualiza as informações.
3. O caso de uso termina.

Ator: Administrador

1. O sistema exibe UC4.
2. O caso de uso termina.

Fluxos Alternativos

A1. Alterar Perfil

A1.1. O sistema exibe as possíveis opções de perfil para ser selecionado.

A1.2. O ator seleciona o perfil desejado.

A1.3. O sistema exibe a seguinte mensagem: “O perfil foi alterado com sucesso!”.

A1.4. O caso de uso termina.

Fluxos de Exceção

Não há.

UC6. Reservar Laboratório

Finalidade: o usuário reservar período para acessar o laboratório.

Fluxo Principal

1. O sistema exibe UC7 e os experimentos disponíveis para reserva.
2. O usuário escolhe o experimento desejado.
3. O sistema exibe os dados (data e hora) disponíveis para o experimento escolhido.
4. O usuário seleciona dados desejados.
5. O sistema exibe a seguinte mensagem: “A reserva foi realizada com sucesso!”.
6. O usuário clica em OK.
7. O sistema exibe o UC7 atualizado.
8. O caso de uso termina.

Fluxos Alternativos

Não há.

Fluxos de Exceção

E1. Mais de duas reservas no dia

- E1.1. O sistema identifica que ator já tem duas reservas na data escolhida.
- E1.2. O sistema exibe a seguinte mensagem: “Não pode haver mais de duas reservas no dia escolhido!”.
- E1.3. O usuário clica em OK.
- E1.4. O caso de uso termina.

UC7. Pesquisar Reserva

Finalidade: o usuário consultar períodos já reservados para acessar o laboratório.

Fluxo Principal

1. O sistema exibe informações das reservas em aberto (data igual ou superior à data atual) do usuário;
2. O usuário visualizar informações;
3. O caso de uso termina.

Fluxos Alternativos

A1. Alterar Reserva

- A1.1. O navegador seleciona alterar reserva.
- A1.2. O sistema exibe dados da reserva.
- A1.3. O usuário altera dados da reserva.
- A1.4. O sistema exibe a seguinte mensagem: “A reserva foi alterada com sucesso!”
- A1.5. O caso de uso termina.

A2. Excluir Reserva

- A2.1. O navegador seleciona excluir reserva.

A2.2. O sistema exibe dados da reserva.
 A2.3. O usuário clica em excluir reserva.
 A2.4. O sistema exibe a seguinte mensagem: “Tem certeza que deseja excluir a reserva?”;
 A2.5. O usuário clica em OK;
 A2.6. O sistema exibe a seguinte mensagem: “A reserva foi excluída com sucesso!”;
 A1.7. O caso de uso termina.

Fluxos de Exceção

E1. Mais de duas reservas no dia

E1.1. O sistema identifica que ator já tem duas reservas na data escolhida.
 E1.2. O sistema exibe a seguinte mensagem: “Não pode haver mais de duas reservas no dia escolhido!”.
 E1.3. O usuário clica em OK.
 E1.4. O caso de uso termina.

UC8. Acessar Interface Snap! for LERO

Finalidade: o usuário acessar a interface de desenvolvimento Snap! for LERO.

Fluxo Principal

1. O sistema exibe UC9, UC10 e UC11;
2. O usuário visualiza informações;
3. O caso de uso termina.

Fluxos Alternativos

Não há.

Fluxos de Exceção

Não há.

UC9. Acessar Grupos de Funções

Finalidade: o usuário acessar os grupos de funções disponíveis no ambiente do seu perfil.

Fluxo Principal

1. O sistema exibe os grupos de funções disponíveis no ambiente do perfil do usuário;
2. O usuário visualiza informações;
3. O caso de uso termina.

Fluxos Alternativos

A1. Arrastar Blocos

A1.1. O usuário arrasta o bloco para o roteiro;
 A1.2. O sistema disponibiliza o bloco no roteiro.
 A1.3. O caso de uso termina.

A2. Alterar Blocos

- A2.1. O usuário do perfil LERO Extremo seleciona botão esquerdo em cima do bloco a ser alterado e clica em editar;
- A2.2. O sistema exibe dados do bloco.
- A2.3. O usuário altera dados do bloco e clica em OK.
- A2.4. O sistema altera os dados.
- A2.5. O caso de uso termina.

A3. Esconder Blocos

- A3.1. O usuário do perfil LERO Extremo seleciona botão esquerdo em cima do bloco a ser escondido e clica em esconder;
- A3.2. O sistema exibe dados do bloco.
- A3.3. O usuário exclui o bloco.
- A3.4. O sistema atualiza o grupo de funções.
- A3.5. O caso de uso termina.

Fluxos de Exceção

Não há.

UC10. Acessar Roteiro

Finalidade: o usuário poder acessar o roteiro disponibilizado pela interface de desenvolvimento Snap! for LERO.

Fluxo Principal

1. O sistema exibe a área de roteiro;
2. O usuário visualiza área de roteiro;
3. O caso de uso termina.

Fluxos AlternativosA1. Arrastar Blocos

- A1.1. O usuário arrasta o bloco para o roteiro;
- A1.2. O sistema disponibiliza o bloco no roteiro.
- A1.3. O caso de uso termina.

A2. Excluir Blocos

- A2.1. O usuário seleciona botão esquerdo em cima do(s) bloco(s) a ser (em) excluído(s) e clica em apague;
- A2.2. O sistema exclui o(s) bloco(s).
- A2.3. O caso de uso termina.

Fluxos de Exceção

Não há.

UC11. Visualizar Experimento

Finalidade: o usuário poder visualizar remotamente o experimento físico através dessa funcionalidade.

Fluxo Principal

1. O sistema exibe as imagens do experimento captadas pela câmera de vídeo;
2. O usuário visualiza as imagens do experimento;
3. O caso de uso termina.

Fluxos Alternativos

Não há.

Fluxos de Exceção

Não há.

UC12. Cadastrar Upload

Finalidade: o usuário poder cadastrar um upload do seu projeto no ambiente do perfil LERO Extremo para análise.

Fluxo Principal

1. O usuário clica em Salvar no disco;
2. O sistema salva o projeto no computador do usuário;
3. O usuário clica em upload;
4. O sistema permite escolher qual o perfil esse projeto novo deverá ser disponibilizado e escolher o arquivo que será enviado;
5. O usuário escolhe o perfil e o arquivo;
6. O sistema exibe a seguinte mensagem: “Projeto enviado para análise com sucesso!”;
7. O caso de uso termina.

Fluxos Alternativos

Não há.

Fluxos de ExceçãoE1. Perfil do novo projeto não selecionado

- E1.1. O sistema identifica que o perfil do novo projeto não foi selecionado.
- E1.2. O sistema exibe a seguinte mensagem: “Escolha pelo menos uma opção!”.
- E1.3. O caso de uso termina.

E2. Arquivo não selecionado

- E2.1. O sistema identifica que o arquivo não foi selecionado.
- E2.2. O sistema exibe a seguinte mensagem: “Escolha o arquivo!”.
- E2.3. O caso de uso termina.

UC13. Acessar Tutorial

<u>Finalidade</u> : poder acessar o tutorial disponível no ambiente do perfil.
<u>Fluxo Principal</u> <ol style="list-style-type: none"> 1. O usuário clica em Tutorial; 2. O sistema exibe o tutorial do ambiente do perfil ativo; 3. O caso de uso termina.
<u>Fluxos Alternativos</u> Não há.
<u>Fluxos de Exceção</u> Não há.

UC14. Manter Experimento

<u>Finalidade</u> : gerenciar experimento(s) disponibilizado(s) na plataforma robótica.
<u>Fluxo Principal</u> <ol style="list-style-type: none"> 1. O sistema exibe os experimentos já cadastrados. 2. O administrador visualizar os experimentos disponíveis. 3. O caso de uso termina.
<u>Fluxos Alternativos</u> <u>A1. Cadastrar Experimento</u> A1.1. O administrador seleciona inserir experimento. A1.2. O sistema solicita os dados (nome, descrição e ambiente do perfil a ser disponibilizado). A1.3. O administrador informa os dados solicitados. A1.4. O sistema verifica se já existe o nome do experimento na base de dados. A1.5. O sistema exibe a seguinte mensagem: "O experimento foi cadastrado com sucesso!". A1.6. O caso de uso termina. <u>A2. Alterar Experimento</u> A2.1. O navegador seleciona alterar. A2.2. O sistema exibe os dados (nome, descrição e ambiente do perfil a ser disponibilizado). A2.3. O administrador informa os dados que serão alterados. A2.4. O sistema verifica se já existe o nome do experimento na base de dados. A2.5. O sistema exibe a seguinte mensagem: "O experimento foi alterado com sucesso!". A2.6. O caso de uso termina. <u>A3. Excluir Experimento</u>

A3.1. O usuário seleciona a opção excluir.
 A3.2. O sistema exibe a seguinte mensagem: “Tem certeza que deseja excluir experimento?”.
 A3.3. O administrador clica em OK.
 A3.4. O sistema exibe a seguinte mensagem: “Experimento excluído com sucesso!”.
 A3.5. O caso de uso termina.

Fluxos de Exceção

E1. Experimento já cadastrado

E1.1. O sistema identifica que o experimento informado já está cadastrado.

E1.2. O sistema exibe a seguinte mensagem: “Esse experimento já está cadastrado!”.

E1.3. O caso de uso termina.

UC15. Manter Upload

Finalidade: gerenciar uploads enviados pelo usuário.

Fluxo Principal

1. O sistema exibe os uploads enviados.
2. O administrador visualizar os uploads disponíveis.
3. O caso de uso termina.

Fluxos Alternativos

A1. Aprovar Upload

A1.1. O administrador seleciona upload a ser analisado.

A1.2. O sistema faz download do upload no computador.

A1.3. O administrador clica em aprovar upload.

A1.4. O sistema exibe a seguinte mensagem: “O upload foi aprovado com sucesso!”.

A1.5. O caso de uso termina.

A2. Reprovar Upload

A2.1. O administrador seleciona upload a ser analisado.

A2.2. O sistema faz download do upload no computador.

A2.3. O administrador clica em reprovar upload.

A2.4. O sistema exibe a seguinte mensagem: “O upload foi reprovado com sucesso!”.

A2.5. O caso de uso termina.

A3. Excluir Upload

A3.1. O administrador seleciona upload a ser analisado.

A3.2. O sistema faz download do upload no computador.

- A3.3. O administrador clica em excluir upload.
- A3.4. O sistema exibe a seguinte mensagem: “Tem certeza que deseja excluir o upload?”.
- A3.5. O administrador clica em OK.
- A3.6. O sistema exibe a seguinte mensagem: “O upload foi excluído com sucesso!”.
- A3.7. O caso de uso termina.

Fluxos de Exceção

Não há.

UC16. Manter Tutorial

Finalidade: gerenciar arquivos dos tutoriais.

Fluxo Principal

1. O sistema exibe os perfis do ambiente LERO.
2. O administrador seleciona o perfil.
3. O sistema exibe os perfis do ambiente LERO.
4. O administrador seleciona o perfil desejado.
5. O sistema mostra dados sobre o perfil.
6. O caso de uso termina.

Fluxos Alternativos

A1. Cadastrar Tutorial

- A1.1. O administrador seleciona cadastrar tutorial.
- A1.2. O sistema fornece tela com botão para escolher arquivo.
- A1.3. O administrador clica no botão e escolhe arquivo.
- A1.4. O sistema exibe a seguinte mensagem: “O arquivo foi anexado com sucesso!”.
- A1.5. O caso de uso termina.

A2. Alterar Tutorial

- A2.1. O administrador seleciona alterar tutorial.
- A2.2. O sistema fornece tela com botão para escolher arquivo.
- A2.3. O administrador clica no botão e escolhe arquivo.
- A2.4. O sistema exibe a seguinte mensagem: “O arquivo foi anexado com sucesso!”.
- A2.5. O caso de uso termina.

A3. Excluir Tutorial

- A3.1. O administrador seleciona excluir tutorial.
- A3.2. O sistema exibe a seguinte mensagem: “Tem certeza que deseja excluir esse

arquivo do tutorial?”.

A3.3. O administrador clica em OK.

A3.4. O sistema exibe a seguinte mensagem: “Arquivo do tutorial excluído com sucesso!”.

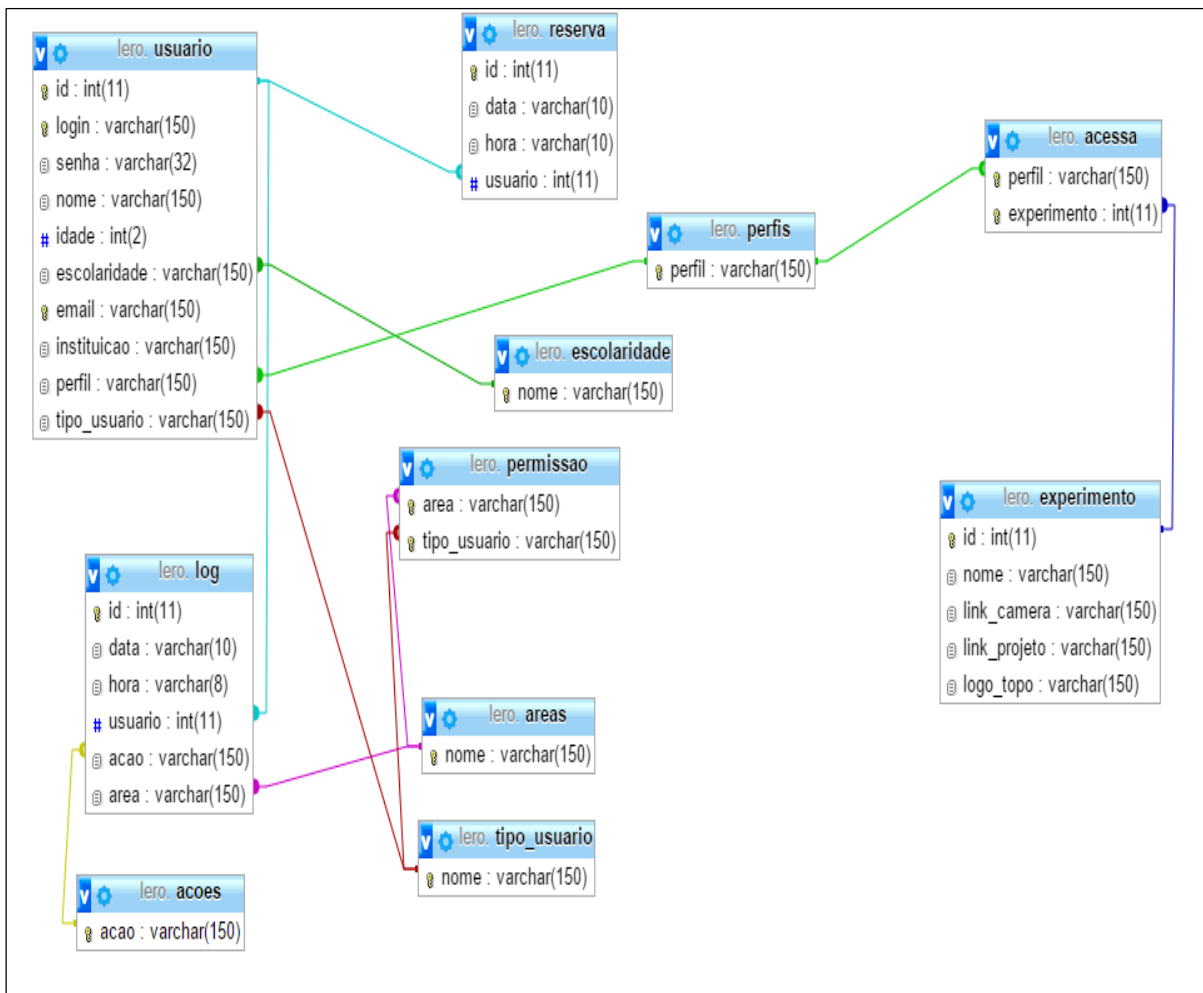
A3.5. O caso de uso termina.

Fluxos de Exceção

Não há.

6. Modelagem da Base de Dados

Nesta seção será apresentado o modelo do banco de dados do sistema.



7. Requisitos Não Funcionais

Nesta seção são detalhados os requisitos não funcionais a serem contemplados pelo ambiente LERO.

7.1 Usabilidade

Descreve os requisitos não funcionais associados à facilidade de uso da interface pelo cliente do sistema.

[RNF001] Interface Amigável

O sistema terá uma interface amigável ao usuário primário sem se tornar cansativa aos usuários mais experientes.

Prioridade: Essencial Importante Desejável

[RNF002] Componentes WEB

A interface deverá utilizar elementos comuns a usuários de sistemas web, como campos de texto, *combo-boxes*, *links* e botões, sem muito rebuscamento. A ideia é focar nos aspectos operacionais sem se preocupar tanto com a beleza da tela, de modo a facilitar o uso por usuários iniciantes.

Prioridade: Essencial Importante Desejável