



UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Thais Oliveira Almeida

Laboratório Remoto de Robótica como
Apoio ao Ensino de Programação

Manaus
Julho de 2016

Thais Oliveira Almeida

Laboratório Remoto de Robótica como
Apoio ao Ensino de Programação

Trabalho apresentado ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para obtenção do grau de Mestre em Informática.

Orientador: Prof. Dr. José Francisco de Magalhães Netto

Manaus
Julho de 2016

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

A4471 Almeida, Thais Oliveira
Laboratório Remoto de Robótica como Apoio ao Ensino de
Programação / Thais Oliveira Almeida. 2016
109 f.: il. color; 31 cm.

Orientador: José Francisco de Magalhães Netto
Dissertação (Mestrado em Informática) - Universidade Federal do
Amazonas.

1. Sistema Multiagente. 2. Aprendizagem de programação. 3.
Laboratório Remoto. 4. Robótica Educacional. I. Netto, José
Francisco de Magalhães II. Universidade Federal do Amazonas III.
Título



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



UFAM

FOLHA DE APROVAÇÃO

"Laboratório Remoto de Robótica como Apoio ao Ensino de Programação"

THAIS OLIVEIRA ALMEIDA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. José Francisco de Magalhães Netto - PRESIDENTE

Prof. Vicente Ferreira de Lucena Junior - MEMBRO INTERNO

Prof. José Luiz de Souza Pio - MEMBRO EXTERNO

Profa. Thais Helena Chaves de Castro - MEMBRO EXTERNO

Manaus, 04 de Julho de 2016

Agradecimentos

Agradeço em primeiro lugar a Deus por me permitir vivenciar este momento e me proporcionar saúde para a conclusão deste mestrado.

À minha mãe Nelci por sempre acreditar no meu potencial e me apoiar nos meus estudos.

À minha irmã Thalita e minha tia Jô por todo apoio e ajuda concebidos nesta caminhada.

Ao meu orientador Prof^o Dr. José Francisco de Magalhães Netto, por suas valiosas contribuições e exigência quanto à qualidade deste trabalho, e por todo companheirismo dedicado no período do mestrado.

Aos amigos Romero Gomes e Tiago Custódio pelas valiosas contribuições e ajuda no desenvolvimento desta dissertação.

À coordenadora pedagógica do Colégio de Aplicação da Universidade Federal de Roraima, Prof^a Leogete Joca da Costa, por ceder o espaço para realização dos testes e todo apoio fornecido.

Agradeço também aos amigos que conheci neste período e levarei para a vida toda. Em especial à Adria, Adriana, Amanda, Bernardo, Carlos, Michel, Pablo e Rayol. Obrigada por todos os momentos de companheirismo, ajuda e desespero. Sem vocês esse mestrado não teria sido o mesmo!

Para finalizar, agradeço à todos os professores do PPGI/UFAM que tive a oportunidade de ter contato, pelas trocas de experiência e conhecimentos adquiridos.

Resumo

Esta dissertação apresenta o desenvolvimento de um Laboratório de Ensino de Robótica e Programação com apoio de Sistemas Multiagente que auxilia usuários na aprendizagem de programação utilizando robôs. O laboratório desenvolvido está disponível em um Ambiente Virtual de Aprendizagem acessado via Web que permite que mesmo pessoas geograficamente distantes tenham experiência com experimentos desta natureza. O laboratório caracteriza-se por possibilitar a aprendizagem colaborativa, explorando a Robótica Educacional como elemento motivador de aprendizagem. Suas funcionalidades foram testadas com resultados promissores, onde os usuários sentiram-se bastante interessados e motivados a continuar aprendendo programação e robótica. O sistema sugere desafios diversos com base na pontuação e perfil dos usuários. O trabalho desenvolvido apresenta avanços sobre laboratórios remotos de robótica encontrados na literatura, uma vez que permite a programação remota dos robôs através da linguagem de programação NXT-Python, e não apenas a manipulação por meio de setas de comando.

Palavras-chave: Sistema Multiagente, Aprendizagem de programação, Laboratório Remoto, Robótica Educacional.

Abstract

This work presents the development of Laboratory for Teaching of Robotics and programming with support of Multi-Agent Systems that assists users in learning programming using robots. The laboratory developed is available in a Learning Management System accessed via Web that allows even geographically distant people have experience with experiments of this nature. The laboratory is characterized by enabling collaborative learning, exploring Educational Robotics as motivating learning element. His features were tested with promising results, where users felt quite interested and motivated to keep learning programming and robotics. The system suggests several challenges based on the score and user profiles. The work presents advances in remote laboratories of robotics in the literature, since it allows remote programming of robots by NXT-Python programming language, not just manipulation by command arrows.

Keywords: Multi-Agent System, Learning Programming, Remote Laboratory, Educational Robotic.

Lista de Figuras

2.1	Fases da Metodologia Prometheus	16
4.1	Arquitetura do LERP	27
4.2	Caso de Uso do ator Usuário	29
4.3	Diagrama de Sequência do Caso de Uso Cumprir Desafios . . .	31
4.4	Caso de Uso do Agente Administrador	32
4.5	Diagrama de Sequência do Caso de Uso Liberar Acesso	33
4.6	Caso de Uso do Agente Agendamento	33
4.7	Diagrama de Sequência do Caso de Uso Verificar Reservas . .	34
4.8	Caso de Uso do Agente Assistente	35
4.9	Diagrama de Sequência do Caso de Uso Sugerir Desafios . . .	36
4.10	Caso de Uso do Agente Robô	37
4.11	Diagrama de Sequência do Caso de Uso Executar Código . . .	38
4.12	Diagrama de Relacionamento dos Agentes	42
4.13	Diagrama de Revisão do Sistema	44
4.14	Tela Inicial do LERP	45
4.15	Instruções Iniciais	45
4.16	Formulário de Cadastro de Usuários	46

4.17	E-mail de Confirmação de Cadastro	46
4.18	Recuperação de Senha	47
4.19	Link de Agendar Experimento	47
4.20	Página de Agendamento de Experimentos	48
4.21	Confirmação de Reserva	48
4.22	E-mail de Confirmação de Agendamento	49
4.23	Link para Acessar Agendamentos Confirmados	49
4.24	Links para Acesso aos Desafios	50
4.25	Sugestão de Desafio	51
4.26	A Sala de Desafios	51
4.27	Processo de Manipulação Remota	52
4.28	Tabelas do Moodle	57
4.29	Interface Gráfica da Plataforma JADE	58
4.30	Troca de Mensagens entre Agentes	58
4.31	Peças do Kit Lego Mindstorms NXT	60
4.32	RCX do Kit Lego NXT. Fonte: [http://education.lego.com/en-us/]	60
5.1	Cadastramento de Usuário	63
5.2	Teste de Agendamento de Experimento	65
5.3	Datas Disponíveis para Agendamento	65
5.4	Teste de Agendamento com Indisponibilidade de Horário	65
5.5	Programando Robô na Sala de Desafios	66
5.6	Tentativa de Acesso à Sala de Desafios	67
5.7	Primeiro Contato com os Alunos	68

5.8	Caracterização dos Usuários	69
5.9	Conhecimento dos Usuários em Informática	70
5.10	Conhecimento dos Usuários em Navegar na Internet	70
5.11	Aprendizagem de Programação e Metodologia Utilizada	72
5.12	Aprendizado sobre os Conteúdos Apresentados	73
5.13	Experiência de Aprendizagem Positiva com Lego	73
5.14	Interesse e Motivação em Aprender Programação	74
B.1	Tabelas Adicionais Utilizadas no LERP	86
F.1	Primeiro Contato com os Alunos	94
F.2	Alunos Realizando Experimentos na Sala de Desafios (1)	94
F.3	Alunos Realizando Experimentos na Sala de Desafios (2)	95
F.4	Alunos Realizando Experimentos na Sala de Desafios (3)	95
F.5	Alunos Realizando Experimentos na Sala de Desafios (4)	95

Lista de Tabelas

3.1	Objetivo conforme paradigma GQM	18
3.2	Expressão de Busca	19
3.3	Critérios de Inclusão e Exclusão	19
3.4	Resultados obtidos com a string de busca	20
3.5	Resumo dos Artigos de Robótica no Contexto Educacional . .	21
3.6	Resumo dos Artigos de Aprendizagem de Programação com Robô	22
3.7	Resumo dos Artigos de Programação de Robôs a Distância . .	24
4.1	Funcionalidades dos Agentes Associadas aos Objetivos	40
4.2	Horários Disponíveis por Dia	47
4.3	Descrição dos Desafios	50
4.4	Níveis de Perfil do Usuário	51
4.5	Funções e Significados	54
5.1	Casos de Teste do CASU Solicitar Cadastro	64
5.2	Casos de Teste do CASU Agendar Experimento	66
5.3	Casos de Teste do CASU Programar Robô	67
5.4	Desafios Concluídos em Cada Experimento	71

Sumário

1	Introdução	1
1.1	Definição do Problema	2
1.2	Justificativa	3
1.3	Objetivos	5
1.3.1	Objetivo Geral	5
1.3.2	Objetivos Específicos	5
1.4	Metodologia	5
1.5	Histórico da Pesquisa	6
1.6	Organização do Documento	7
2	Fundamentos Teóricos	8
2.1	Robótica Educacional	8
2.2	Prática de Programação com Robôs	9
2.3	Agentes Inteligentes e Sistemas Multiagente	11
2.4	Laboratório Remoto de Robótica	14
2.5	Metodologia Prometheus	14
2.6	Considerações Finais	16

3	Trabalhos Correlatos	17
3.1	Revisão Sistemática da Literatura	17
3.1.1	Objetivo	17
3.1.2	Questões de Pesquisa	17
3.1.3	Estratégia de Busca	18
3.1.4	Idioma	18
3.1.5	Métodos de Busca das Publicações	18
3.1.6	Procedimentos de Seleção e Critérios	19
3.2	Robótica no Contexto Educacional	20
3.3	Aprendizagem de Programação com Robô	22
3.4	Programação de Robô a Distância	23
3.5	Considerações Finais	24
4	Laboratório Remoto de Robótica	26
4.1	Visão Geral do Sistema	26
4.2	Casos de Uso e Diagrama de Sequência	28
4.2.1	Ator envolvido: o Usuário	29
4.2.2	Ator envolvido: o agente Administrador	31
4.2.3	Ator envolvido: o agente Agendamento	32
4.2.4	Ator envolvido: o agente Assistente	35
4.2.5	Ator envolvido: o Agente Robô	36
4.2.6	Modelagem Utilizando Prometheus	37
4.3	Interface Gráfica e Funcionamento do LERP	43
4.3.1	Cadastramento de Usuário	43
4.3.2	Processo de Agendamento	46

4.3.3	Sala de Desafios	49
4.3.4	Manipulação Remota	52
4.4	Tecnologias Utilizadas	54
4.4.1	Moodle	55
4.4.2	PHP	55
4.4.3	MySQL	56
4.4.4	JADE	57
4.4.5	Lego Mindstorms	59
4.4.6	NXT-Python	60
5	Avaliação e Testes do Sistema	62
5.1	Testes dos Casos de Uso	63
5.1.1	Teste do Caso de Uso Solicitar Cadastro	63
5.1.2	Teste do Caso de Uso Agendar Experimento	64
5.1.3	Teste do Caso de Uso Programar Robô	65
5.2	Teste em uma Turma Real	67
5.2.1	Perfil dos Usuários	69
5.2.2	Experimentos Realizados	71
5.3	Considerações Finais	73
6	Conclusões e Trabalhos Futuros	75
	Referências Bibliográficas	78
A	Artigos Científicos Produzidos Durante o Mestrado	84
B	Tabelas Adicionais	85

C Código Fonte das Funções	87
D Questionário de Caracterização do Usuário	90
E Pesquisa de Opinião	92
F Fotos dos Testes Realizados	94

Capítulo 1

Introdução

As dificuldades na aprendizagem em disciplinas introdutórias de programação são constantes como tema para varias discussões e pesquisas, a exemplo de Ambrósio et al. (2011), Gomes et al. (2012), Santos, Gomes e Mendes (2013), Souza et al. (2013), Gomes e Mendes (2014), resultando também em muitas propostas de metodologias e ferramentas de suporte. Além dos estudantes dos cursos de Computação, estudantes de Engenharia e áreas afins costumam cursar disciplinas dessa natureza, e sentem dificuldades para assimilar os conteúdos e concluí-las com êxito.

Santos, Gomes e Mendes (2013) afirmam que para ser bem sucedido em cursos introdutórios de programação, o aluno deve aprender um conjunto de conceitos básicos, a sintaxe de uma linguagem de programação, e desenvolver habilidades para solucionar problemas.

Existe consenso público, acadêmico e governamental sobre a potencialização do aprendizado proporcionado pela interação com recursos tecnológicos, porém existe um paradoxo que reside sobre a séria carência na formação de professores para atuar com tecnologia educacional nas escolas (CRUZ et al., 2009). De acordo com Yu (2012), com o desenvolvimento da tecnologia educacional moderna, vários professores estão procurando métodos eficientes e ideias criativas, tanto para expor os conteúdos quanto para manter o interesse dos seus alunos.

De acordo com Resnick, Berg e Eisenberg (2000), quando uma pessoa está engajada em construir algo que seja significativo para si e para as pessoas ao seu redor, a construção do conhecimento se dá de forma mais efetiva. Por exemplo, quando o estudante se envolve na construção e na modelagem de robôs, este passa a ter um senso de cuidado muito maior na relação com este objeto do que se recebesse o robô pronto. Desta forma, os alunos estão propensos a explorar e a fazer profundas conexões com conceitos científicos baseados em suas atividades (RESNICK; BERG; EISENBERG, 2000).

Nesta dissertação apresentamos o Laboratório de Ensino de Robótica e Programação (LERP) baseado em Sistemas Multiagente (SMA). O LERP desenvolvido está disponível em um Ambiente Virtual de Aprendizagem (AVA) acessado via Web, que permite que mesmo pessoas geograficamente distantes tenham experiência com programação usando robôs. O laboratório caracteriza-se por possibilitar a aprendizagem colaborativa, para auxiliar no processo de aprendizagem de programação, explorando a Robótica Educacional (RE) como elemento motivador de aprendizagem.

A escolha da RE como objeto de experimentação científica atrelada à sistemas de ensino e aprendizagem, tem-se demonstrado possuir uma vantagem significativa para os alunos, por meio da melhoria do aprendizado. Os trabalhos publicados por Fernandes (2013), D'Abreu e Bastos (2013), Trentin, Pérez e Teixeira (2013), Friedrich et al. (2012), sustentam esta afirmação.

Quanto à escolha da arquitetura de SMA, o desenvolvimento de Agentes Inteligentes (AIs) associados à um AVA proporcionam habilidades de cooperação, coordenação e negociação entre os agentes, fazendo com que seus objetivos individuais e globais possam ser alcançados. A integração do LERP com o AVA permite que os alunos tenham tanto acesso aos materiais de ensino, quanto um espaço destinado para a prática de programação utilizando robôs.

1.1 Definição do Problema

A maioria dos estudantes universitários ingressam na universidade sem nunca ter tido qualquer experiência com programação. Dessa maneira, não é fácil para os alunos compreenderem os conceitos envolvidos em lógica de programação e ao mesmo tempo conhecer a sintaxe da linguagem. De acordo com Tan, Ting e Ling (2009) essas disciplinas têm sido consideradas um desafio aos alunos e os conteúdos são difíceis de aprender.

Santos, Gomes e Mendes (2013) enfatizam que as atividades de aprendizagem devem ser adaptadas ao ritmo de aprendizagem de cada aluno e necessidades específicas. Um estudo realizado por Tan, Ting e Ling (2009) investigou os fatores que levam à dificuldade de aprendizagem de alunos de graduação em cursos de programação, e também a percepção sobre qual metodologia poderia ser implementada para as aulas se tornarem mais interessantes. Como resultado da pesquisa, os alunos afirmaram que preferem aprender programação em um ambiente onde seja possível realizar a repetição sistemática de conceitos, exemplos e problemas práticos.

O ambiente de RE pressupõe a existência de professor, aluno e ferramentas que propiciam a montagem, automação e controle de dispositivos

mecânicos. Os kits de robótica (*Legó Mindstorms*) possuem alto custo, e os laboratórios didáticos ainda representam um custo elevado para serem instalados. Trentin, Pérez e Teixeira (2013) sustentam essa afirmação, e ainda atestam que esses kits não possuem muita adaptabilidade no que diz respeito à interação com outros dispositivos que não pertençam ao fabricante ou escolhas de outras linguagens de programação não presentes no kit.

Com o avanço da tecnologia, vários laboratórios remotos têm sido desenvolvidos a fim de permitir experimentos com robôs, a exemplo de Souza et al. (2011), Kulich et al. (2013), Chaos et al. (2013), Casini et al. (2014). Esses trabalhos utilizam a estrutura de um laboratório físico e o disponibilizam na Internet como forma de permitir que pessoas mesmo geograficamente distantes possam ter experiências com robôs.

Neste sentido, a Questão de Pesquisa desta investigação é: De que forma um laboratório remoto de robótica motiva os usuários a aprenderem programação, a partir de experiências que permitam à eles assimilar conteúdos teóricos na prática?

1.2 Justificativa

As ideias relacionadas ao construcionismo de Papert (1994) têm motivado o desenvolvimento de diferentes sistemas computadorizados voltados para a educação. Nesse contexto se destaca a RE, unindo o uso da linguagem Logo, proposta por Papert, à montagem e programação de robôs a partir de kits de robótica especialmente preparados para esse fim.

A RE, também chamada de robótica pedagógica, vem sendo utilizada como técnica de aprendizado que permite o desenvolvimento de atividades, e como ferramenta que estimula a criatividade dos alunos devido a sua natureza dinâmica, interativa e até mesmo lúdica, além de servir de motivador para estimular o interesse dos alunos no ensino tradicional (GOMES et al., 2008).

De acordo com Silva (2009), uma das metodologias adequadas para o desenvolvimento de aulas de RE em sala de aula é por meio de oficinas de robótica, nas quais os alunos seguem um conjunto de passos: análise do desafio, montagem do protótipo robótico e programação do robô. Dessa maneira, a robótica tem grande potencial como ferramenta interdisciplinar, visto que a construção de um novo mecanismo, ou a solução de um novo problema, frequentemente extrapola os exemplos abordados em sala de aula.

Na visão de Papert (1994) programar significa nada mais, nada menos, que comunicar-se com o computador, numa linguagem que tanto ele (computador) quanto o homem pode "entender". Nesse contexto de uso da ferramenta, o autor infere que os alunos tornam-se criadores de conhecimento.

O autor vê a máquina como uma ferramenta capaz de afetar a maneira das pessoas pensarem e aprenderem, uma vez que o aprender nesta situação se dá por meio da criação, reflexão e depuração das ideias.

Existem diversas formas de se programar um robô. A mais elementar e mais complexa é por meio da montagem de circuitos eletrônicos que correspondem ao programa. Também é possível programar robôs em linguagens de programação tradicionais como C/C++, Python e Java. Neste caso, deve-se buscar por uma biblioteca ou API (*Application Programming Interface*) correspondente àquele robô para a determinada linguagem (AZEVEDO; AGLAÉ; PITTA, 2010).

Alguns kits de robótica apresentam suas próprias linguagens e ambientes de programação, que podem ser por meio de ícones ou de textos. Para realizar estudos desta natureza, faz-se necessário adquirir materiais e componentes que possuem um custo bastante elevado.

Ramos et al. (2007) destacam que o custo desses ambientes constitui o principal obstáculo para o seu uso de forma difundida nas escolas, especialmente em países em desenvolvimento. O mesmo acontece com os laboratórios didáticos, pois ainda representam um custo alto para serem instalados. Sendo assim, a criação de laboratórios remotos apresenta-se como uma alternativa permitindo que a qualquer momento, várias pessoas compartilhem de experimentos, mesmo estando geograficamente distante.

Segundo Alencar e Netto (2012), a utilização de agentes com técnicas de Inteligência Artificial (IA) em Ambientes de Ensino e Aprendizagem (AEA) possibilitam uma maior flexibilidade e adaptabilidade, melhora a cooperação e favorece a sociedade de agentes. De acordo com Wooldridge (2009), um SMA caracteriza-se por ser um sistema onde vários agentes atuam em conjunto sobre um ambiente na busca da resolução de um problema. Dessa maneira, eles podem desenvolver atividades relacionadas à sua especialidade de acordo com o uso de seu conhecimento e habilidades.

Diante do exposto, compreendendo a importância do aprendizado de práticas de programação e observando-se a RE como uma ferramenta facilitadora para este fim, entende-se como mérito o desenvolvimento de um LERP, com apoio de SMAs, para auxiliar na aprendizagem de programação utilizando robôs. Vale ressaltar que esta proposta não pretende substituir os laboratórios físicos existentes, mas sim permitir acesso remoto aos equipamentos reais.

1.3 Objetivos

Para atingir o propósito desta pesquisa, descrevemos a seguir o objetivo geral e os objetivos específicos.

1.3.1 Objetivo Geral

Desenvolver e avaliar um Sistema Multiagente que auxilia na motivação de usuários na aprendizagem de programação usando robôs, em um laboratório remoto de robótica que disponibiliza equipamentos físicos para a realização de experimentos.

1.3.2 Objetivos Específicos

Para atingir este objetivo geral, o presente projeto enumera os seguintes objetivos específicos:

1. Estabelecer uma arquitetura de Sistema Multiagente que auxilie usuários na aprendizagem de programação utilizando robôs via Internet.
2. Implementar a arquitetura de Sistema Multiagente que atenda o item 1.
3. Integrar o Sistema Multiagente ao Ambiente Virtual de Aprendizagem Moodle, permitindo que os usuários manipulem os robôs através do laboratório remoto.
4. Analisar o interesse dos usuários na área de programação e robótica, a partir das experiências de programação utilizando robôs através do LERP.

1.4 Metodologia

Do ponto de vista da forma de abordagem do problema, a metodologia desta pesquisa é de natureza qualitativa, pois a interpretação dos fenômenos e a atribuição de significados são básicas no processo. Ela não requer o uso de métodos e técnicas estatísticas. O ambiente natural é a fonte direta para coleta de dados e o pesquisador é o instrumento-chave (GIL, 1991).

Do ponto de vista dos seus objetivos, é de natureza exploratória, pois não requer a formulação de hipóteses para serem testadas, e se restringe por definir objetivos e buscar mais informações sobre determinado assunto de estudo (GIL, 1991).

A metodologia de trabalho desta pesquisa consiste de uma série de procedimentos típicos de projetos desta natureza, sendo eles:

1. Levantamento bibliográfico sobre os laboratórios remotos que permitem manipular robôs a distância. Para tanto, foi necessária a realização de uma Revisão Sistemática da Literatura (RSL) a fim de identificar os trabalhos desenvolvidos nos últimos cinco anos, assim como verificar as tecnologias utilizadas e resultados alcançados.
2. Identificação de requisitos e planejamento da arquitetura.
3. Escolha e utilização de uma metodologia de Engenharia de Software voltada para a modelagem de Sistemas Multiagente.
4. Desenvolvimento do LERP e implementação dos agentes.
5. Identificação dos critérios dos usuários que irão participar dos testes.
6. Realização de testes com usuários do sistema. Esses testes envolveram usuários, com ou sem conhecimento de robótica a fim de testar e avaliar as funcionalidades do sistema.
7. Aplicação de questionário com os usuários para avaliar se o sistema proposto motivou-os a aprender programação.
8. Avaliação do sistema. Nessa fase foram analisados os resultados obtidos dos testes realizados com usuários reais. Os resultados serviram para validar a abordagem, e também foram fatores essenciais para melhoria do sistema.

1.5 Histórico da Pesquisa

Durante o período da pesquisa, o artigo "Robótica pedagógica aplicada ao ensino de programação: Uma revisão sistemática da literatura" Almeida e Netto (2015), foi publicado nos anais do XXVI Simpósio Brasileiro de Informática na Educação.

O artigo intitulado "Laboratório Remoto de Robótica como Apoio à Aprendizagem de Programação", foi submetido para o XXVII Simpósio Brasileiro de Informática na Educação, e encontra-se em fase de avaliação.

Com a sequência dos trabalhos e protótipos, é possível perceber o amadurecimento da ideia da arquitetura e desenvolvimento da solução proposta, e esse era o objetivo inicial deste trabalho.

1.6 Organização do Documento

Esta dissertação está dividida em cinco capítulos, a saber:

O presente capítulo aborda a introdução, apresentando a definição do problema, justificativa, objetivos e metodologia utilizada para o desenvolvimento desta pesquisa.

O Capítulo 2 demonstra uma breve fundamentação teórica, tendo como objetivo principal mostrar teorias a respeito dos temas abordados nesta pesquisa. Neste contexto, será apresentada uma explanação a respeito de Robótica Educacional, Prática de Programação com Robôs, Agentes Inteligentes e Sistemas Multiagente, Laboratório Remoto de Robótica. Ao final do capítulo é apresentada a metodologia utilizada para o desenvolvimento dos agentes.

O Capítulo 3 destina-se a apresentar os trabalhos correlatos, que são trabalhos que possuem foco semelhante com este projeto que foi desenvolvido. Neste sentido, detalha-se a RSL realizada, demonstrando os resultados que estes trabalhos obtiveram e suas contribuições para o meio acadêmico.

Uma visão geral do laboratório remoto desenvolvido, assim como as tecnologias que foram utilizadas são apresentadas no Capítulo 4. Neste capítulo é apresentada também a modelagem utilizada para a implementação e desenvolvimento dos agentes, os testes realizados e resultados obtidos.

Os testes realizados com a proposta e a avaliação, são apresentados no Capítulo 5.

O Capítulo 6 aborda as conclusões deste projeto, levando em consideração as contribuições, limitações e trabalhos futuros. Finalmente, são apresentadas as referências bibliográficas utilizadas na realização deste documento.

Capítulo 2

Fundamentos Teóricos

Este capítulo tem como objetivo contextualizar a proposta, apresentando teorias a respeito dos temas abordados na pesquisa que sustentam o desenvolvimento deste trabalho. Neste contexto, será realizada uma explanação a respeito de Robótica Educacional, Prática de Programação com Robôs, Sistemas Multiagente e Laboratório Remoto de Robótica. Ao final do capítulo é apresentada a metodologia de engenharia de software voltada para SMAs que é utilizada no desenvolvimento do trabalho.

2.1 Robótica Educacional

A RE é uma definição dada a AEA's que são caracterizados pela teoria e prática de conteúdos aplicados a montagem, operação e programação de robôs. Nesses ambientes estão inseridos o professor, o aluno, os kits de montagem de robótica e um espaço físico equipado e destinado para este fim (SOUZA et al., 2011).

De acordo com Azevedo, Aglaé e Pitta (2010), muitas pessoas conseguem identificar um dispositivo robótico, mas possuem alguma dificuldade em construir uma definição abrangente sobre o mesmo. Um robô é definido pela *Robotics Industries Association*¹ (RIA) como sendo um objeto reprogramável e multifuncional, pensado e projetado, visando auxiliar o humano na realização de determinadas tarefas.

Segundo D'Abreu, Mirisola e Ramos (2011) o desenvolvimento de atividades desta natureza envolve pelo menos a concepção, implementação, construção, automação e controle do dispositivo desenvolvido. Realizar atividades pedagógicas com robótica, em algumas escolas, vem se constituindo uma prática, na forma interdisciplinar de se promover o aprendizado de conceitos

¹<http://www.robotics.org/>

curriculares utilizando sistemas robóticos que integram o fazer como meio de aprender (D'ABREU; MIRISOLA; RAMOS, 2011).

Em computação Azevedo, Aglaé e Pitta (2010) afirmam que a robótica engloba diferentes tópicos que podem ser abordados para o desenvolvimento de um robô, com foco no *software* que este robô irá possuir - que determina em um nível mais alto na hierarquia em relação ao *hardware* como o robô irá funcionar. Desta maneira, a RE se constitui como um instrumento para promover a integração de atividades práticas com o conhecimento científico e trabalho em equipe, que pode ser viabilizado em países em desenvolvimento com alternativas de baixo custo.

Para facilitar o processo de construção de robôs, principalmente para principiantes, existem diversos kits de robótica que são usados no âmbito educacional. A inserção de recursos tecnológicos no processo de ensino/aprendizagem é um grande desafio para o Brasil, pois a maioria da população não tem acesso a recursos computacionais ou jogos educativos, sejam esses na escola ou em sua residência (SANTOS; NASCIMENTO; BEZERRA, 2010).

De acordo com Filho e Gonçalves (2008), apesar de muitas pesquisas indicarem a RE como sendo uma ferramenta que envolve questões multidisciplinares, portanto rica pedagogicamente, ela, infelizmente não faz parte do cotidiano das escolas brasileiras. A explicação para tal fato, passa pela dificuldade na aquisição do equipamento. Essa dificuldade reside, principalmente, no momento de sua compra, pois seu custo, ainda é proibitivo (FILHO; GONÇALVES, 2008).

Trentin, Pérez e Teixeira (2013) afirmam que neste contexto de emergência da tecnologia na escola, a robótica apresenta-se como dispositivo potencializador da aprendizagem sendo considerada uma das dez áreas mais importantes de pesquisa no mundo, ainda pouco difundida no Brasil, principalmente em termos educacionais.

2.2 Prática de Programação com Robôs

De acordo com o *Computer Science Teachers Association* CSTA (2012) a temática mais abordada nos diversos cursos de informática é a programação com 69%. Este fato revela a importância que a programação tem dentro do campo da informática. Antes dos alunos iniciarem a programar, eles precisam desenvolver o raciocínio lógico por meio de algoritmos.

Segundo Sousa (2013), o discente quando ingressa no curso superior, sente dificuldade em desenvolver o raciocínio lógico e em criar programas. A lógica é aplicada em diversas áreas do conhecimento, principalmente na computação, tanto no nível de *hardware* como de *software*. A lógica na educação

possibilita o uso do computador como um meio que permite ao aluno ter outra possibilidade no processo de construção da aprendizagem (SOUSA, 2013).

Tavares, Menezes e Nevado (2012) relatam que após tentativas frustradas de construir programas para resolver problemas, o aluno reinicia suas ações em busca de uma solução correta, tentando descobrir e corrigir os erros cometidos, inicialmente, em um processo de tentativa e erro, até que consiga ganhar experiência nesse processo que lhe permita tomar consciência sobre o mesmo e sobre a construção de soluções corretas.

Em uma tentativa de amenizar estas dificuldades, torna-se necessário pensar em novas estratégias para o ensino da programação, centradas nos interesses dos alunos. Alguns estudos Santos, Nascimento e Bezerra (2010), D'Abreu, Mirisola e Ramos (2011) e Trentin, Pérez e Teixeira (2013) apontam o uso da RE como uma tecnologia aliada na resolução de problemas de diversas áreas do conhecimento.

Os recursos necessários que devem conter em uma sala de aula, para o desenvolvimento de uma aula com apoio da RE são apresentados por Júnior (2011):

- Um kit de robótica, composto por controlador lógico programável, sensores, motores e peças para construção de uma estrutura mecânica para a carcaça do robô;
- Um *software* de desenvolvimento de programas;
- Um computador capaz de utilizar o *software* escolhido; e
- Um ambiente propício ao desenvolvimento de uma determinada atividade.

Para a escolha dos equipamentos citados acima é necessário levar em consideração a faixa etária dos alunos que participarão das aulas de RE, para que o kit de robótica e o *software* de programação sejam adequados a eles, ou seja, não sejam muito complexos. Além disso, é necessário levar em consideração também o preço desses equipamentos (FERNANDES, 2013).

Fernandes (2013) classifica os *softwares* utilizados em aulas de RE basicamente em duas categorias:

- *Softwares* de Programação: são os programas de computador que utilizam uma linguagem de programação para programar os movimentos de um robô. Este tipo de *software* pode ser utilizado para programar robôs reais, montados a partir de kits de robótica, ou robôs simulados. Por si só, o *software* não é suficiente para a realização de uma aula de robótica.

- *Softwares* de Simulação: são programas capazes de simular o comportamento de robôs em um ambiente virtual que simula o ambiente real no qual ocorreria uma aula de RE. Para a utilização de *softwares* de simulação também é necessário o uso de *softwares* de programação, que permitirão a programação do robô virtual. Em alguns casos, a programação é feita diretamente no *software* de simulação.

Uma pesquisa comparando os principais kits educacionais de robótica proprietários do mercado com kits de robótica livre é apresentada no trabalho de Santos, Nascimento e Bezerra (2010). A pesquisa aponta que embora os kits de robótica livre possuam um custo menor, eles não possuem facilidade de manuseio do equipamento.

Com relação aos kits proprietários o investimento dos kits básicos e autônomos representam, respectivamente, 2,9% e 13,9% do valor do kit proprietário mais barato, viabilizando a utilização também em escolas públicas (SANTOS; NASCIMENTO; BEZERRA, 2010). No entanto de acordo com Fernandes (2013), os kits não-comerciais, em sua maioria, possuem limitações quanto à programação e à proteção dos componentes elétricos, e são adequados a faixas etárias mais altas.

Neste trabalho, os alunos irão programar os movimentos do robô por meio da linguagem de programação NXT-Python (Python para Lego NXT). O software de programação estará associado ao AVA Moodle.

2.3 Agentes Inteligentes e Sistemas Multiagente

Um AI, é definido por Russell e Norvig (2002) como uma entidade autônoma, capaz de interagir com o ambiente, com outros AIs, cooperando ou, até mesmo, competindo entre si e de tomar decisões sem a interferência de um sistema ou de seres humanos.

Para Wooldridge (2009) agentes são sistemas computacionais capazes de ações autônomas em algum ambiente com a finalidade de alcançar seus objetivos de projeto. O autor ainda faz a distinção entre agente e agente inteligente, que tem a necessidade de ser reativo, proativo e social (WOOLDRIDGE, 2009).

A propriedade de autonomia, significa que os agentes são independentes e tomam suas próprias decisões. De acordo com Padgham e Winikoff (2005b) esta é uma das propriedades que distinguem os agentes de objetos. Os agentes tendem a ser usados onde o ambiente caracteriza-se como um desafio.

Segundo Padgham e Winikoff (2005b), esses ambientes são dinâmicos, imprevisíveis e pouco confiáveis. São dinâmicos pois mudam rapidamente (o agente não pode assumir que o ambiente vai permanecer estático enquanto ele

tenta alcançar um objetivo). Em particular, isto significa que um agente deve responder às mudanças significativas em seu ambiente. São imprevisíveis na medida em que não é possível prever os futuros estados do ambiente (muitas vezes isso ocorre pois um agente não possui informação completa sobre o ambiente, e porque o ambiente muda constantemente). E, não são confiáveis pois as ações de um agente podem falhar a medida que são executadas.

Quando empregados em um sistema, Wooldridge (2009) define que os AIs devem possuir as seguintes características:

- **Autonomia:** capacidade de executar ações sem intervenção direta do ser humano ou outro sistema;
- **Habilidade Social:** capacidade de interagir com outros agentes para satisfazer seus objetivos;
- **Reatividade:** capacidade do agente perceber mudanças no seu ambiente e determinar as ações a serem tomadas conforme essas mudanças;
- **Pró-atividade:** os agentes procuram tomar decisões, de forma a satisfazer seus objetivos.

Embora os objetos possam ser reativos, e possam ser vistos como tendo um objetivo implícito, eles não são pró-ativos no sentido de ter múltiplos objetivos explícitos e persistentes (PADGHAM; WINIKOFF, 2005b). Assim, a pró-atividade é outra propriedade que distingue agentes de objetos.

Os agentes devem ser robustos para que sejam capazes de se recuperar de possíveis falhas no ambiente. Uma abordagem natural para conseguir robustez é ser flexível, ou seja, ter várias formas para alcançar um determinado objetivo. Desta maneira, caso uma ação não seja concluída com êxito, o agente escolhe outra alternativa.

Como informado anteriormente, os agentes são sociais, ou seja quase sempre precisam interagir com outros agentes. Essa interação não está relacionada apenas à troca de mensagens, mas se enquadra em termos de performativas "informar", "pedir", "concordar", e assim sucessivamente. As performativas têm semântica padrão que são definidas em termos de seus efeitos sobre o estado mental de um agente (PADGHAM; WINIKOFF, 2005b).

Estas características são importantes para o desempenho de AVAs que utilizam AIs ou SMA, pois proporcionam as habilidades de cooperação, coordenação e negociação entre os agentes, fazendo com que seus objetivos individuais e globais dentro de um ambiente possam ser alcançados.

De acordo com Souza et al. (2011), o processo de implementação de um SMA consiste no desenvolvimento de padrões, envolvendo princípios, conceitos e modelos que permitam a criação de sociedades de AIs autônomos.

Quanto ao desenvolvimento de agentes em SMAs, os agentes são inseridos em ambientes, onde com o uso de seu conhecimento e habilidades, podem desenvolver atividades relacionadas à sua especialidade (BATISTA, 2008).

Segundo Netto (2006), a abordagem multiagente proporciona um desempenho melhor quando aplicado em um sistema. O autor relaciona isso à comunicação e cooperação que ocorrem entre os agentes. E para que exista a cooperação, além da comunicação, cada AI deve possuir auto-definição de suas funcionalidades, de suas características operacionais e do escalonamento de suas tarefas.

As principais aplicações de SMAs são listadas abaixo conforme (FERBER, 1999):

- Solução de problemas: Como uma alternativa para a resolução de problemas centralizados, seja porque os problemas são eles próprios distribuídos, ou porque a distribuição de resolução de problemas entre diferentes agentes revela-se forma mais eficiente de organizar a resolução de problemas - pode ser flexível e permitir falhas no sistema - ou porque, em alguns casos, é a única maneira de resolver o problema.
- Simulação Multiagente: Simulação é amplamente utilizada para melhorar o conhecimento em biologia ou em ciências sociais. A utilização da simulação nos dá a possibilidade de fazer universos artificiais que são pequenos laboratórios para o teste de teorias sobre comportamentos locais.
- Construção de mundos sintéticos: Esses universos artificiais podem ser usados para descrever os mecanismos de interação específicos, e analisar o seu impacto a nível global no sistema. O objetivo deste trabalho é fazer com que as sociedades de agentes que são muito flexíveis possam adaptar-se mesmo em casos de falha individual. Por exemplo, quando os robôs são enviados em uma expedição e eles são obrigados a serem muito independentes das instruções que eles poderiam receber.
- Robótica: Definindo os robôs como um SMA onde cada subsistema tem um objetivo específico e lida apenas com esse objetivo. Uma vez que todas as pequenas tarefas são realizadas a grande tarefa é demasiado. Abordagens SMA, também podem ser utilizadas na coordenação de diferentes robôs móveis em um espaço comum.

Com base no contexto apresentado nesta seção, pode-se observar que os agentes podem contribuir para aprendizagem quando inseridos em vários ambientes, inclusive em um AVA, usando suas habilidades para desenvolver atividades relacionadas à sua especialidade.

2.4 Laboratório Remoto de Robótica

As Instituições de Ensino Superior (IES) cujos cursos apresentam necessidades de experimentação laboratorial têm enfrentado muitas dificuldades quanto à qualidade de ensino. De acordo com Victorino et al. (2009), a realização de trabalhos experimentais em laboratórios devidamente equipados é essencial para se aprender ciência e adquirir cultura científica. Porém, nem todos os jovens têm acesso a estes equipamentos, especialmente aqueles que estudam em instituições públicas.

Outro fator de importância a ser analisado, é o alto custo da instalação e manutenção dos laboratórios nas escolas. Se por um lado a utilização de novas tecnologias aliada aos sistemas de comunicação proporcionam um avanço sem precedentes na área da educação e na rápida difusão do conhecimento, como ocorreu nos anos 80 com a informatização dos laboratórios de pesquisa científica, o fato é que a utilização das Tecnologias de Informação e Comunicação (TIC) em laboratórios didáticos ainda representa um custo elevado (VICTORINO et al., 2009).

Existem várias formas de interagir com os robôs criados em projetos de robótica. Vários laboratórios remotos têm sido desenvolvidos, a exemplo de Guimarães et al. (2003), Souza et al. (2011), Grandi, Falconi e Melchiorri (2011), Kulich et al. (2013), Chaos et al. (2013), Casini et al. (2014). Hoje em dia os laboratórios remotos podem ser integrados a um *Learning Management System* (LMS) ou *Content Management System* (CMS), onde ele pode ser visto como um recurso adicional disponível para os usuários.

Enquanto muitos projetos permitem que os usuários apenas manipulem o robô remotamente através de setas de comando, este trabalho possibilita que os usuários executem experimentos programando-o a distância. Assim, acredita-se que o desenvolvimento de um laboratório remoto com apoio de SMAs, sirva como elemento motivador de aprendizagem de programação por meio da RE, pois permitirá visualizar, explorar e manipular robôs reais, além de observar os efeitos destas alterações.

A próxima seção apresenta a metodologia que foi utilizada para o desenvolvimento dos AIs.

2.5 Metodologia Prometheus

Uma metodologia de desenvolvimento de SMAs é constituída por uma série de passos e procedimentos a serem seguidos durante o processo de concepção do sistema. Ela deve capturar a flexibilidade, autonomia dos agentes com variados graus de abstração, auxiliando o projetista nas tomadas de decisão

relativas à análise, projeto e implementação.

A Prometheus trata-se de uma metodologia para o desenvolvimento de sistemas baseados em AIs, para especificar, projetar e implementar sistemas de agentes. Essa metodologia foi desenvolvida por Padgham e Winikoff (2005b) ao longo de vários anos, em colaboração com o *AOS GROUP (Agent Oriented Software)*², uma empresa que comercializa plataforma para desenvolvimento de agentes.

Composta por um conjunto de conceitos, de um processo e anotações para obtenção dos requisitos e design do sistema, essa metodologia define um processo detalhado e iterativo para a especificação, projeto, implementação e teste/depuração de sistemas de software orientados a agentes (PADGHAM; WINIKOFF, 2005a).

De acordo com Dário (2005) a Prometheus proporciona mecanismos de estruturação hierárquica que permite ser executado em diferentes níveis de abstração. A metodologia utiliza o conceito de capacidade, a qual pode ser composta por planos, eventos, crenças e outras capacidades que dão habilidades específicas ao agente. O agente é composto por várias capacidades e cada uma delas tem uma função específica.

Essa metodologia define também o conjunto de artefatos (gráficos ou textuais) a serem produzidos durante o processo. A Figura 2.1 apresenta as fases dessa metodologia:

Conforme apresentado na Figura 2.1, a metodologia Prometheus possui três fases:

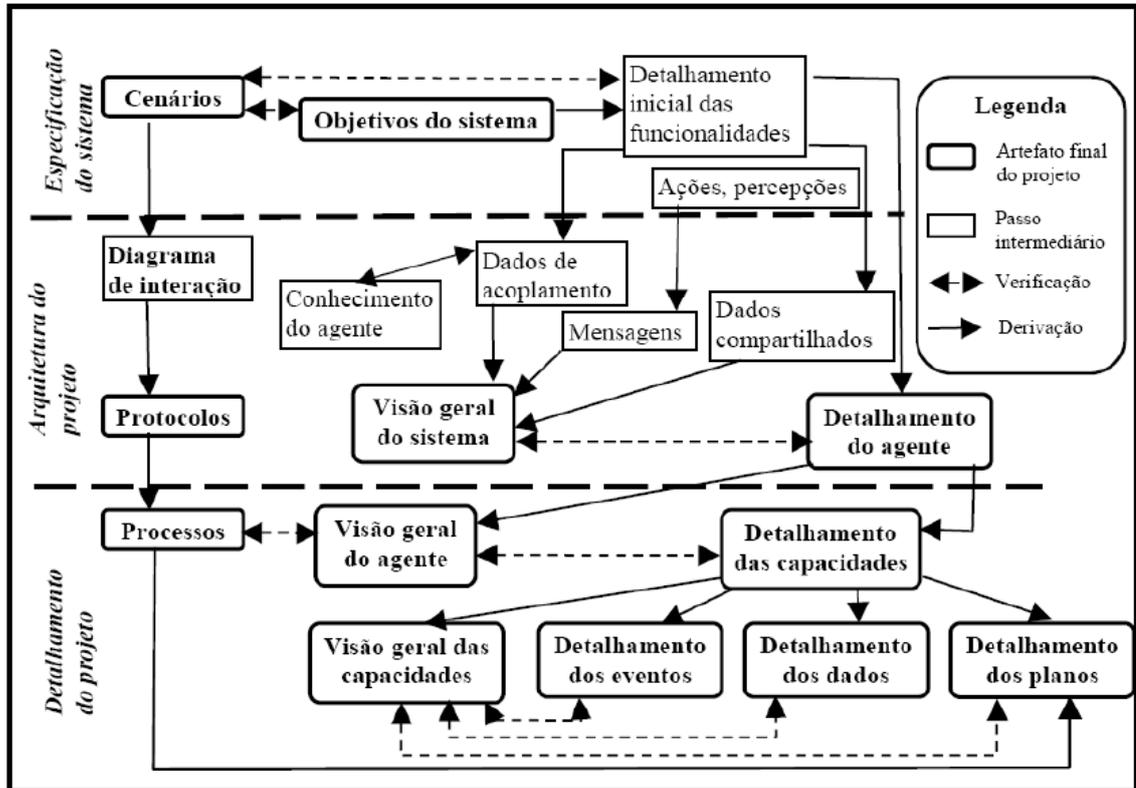
- Fase de Especificação do Sistema: Foco na identificação das funcionalidades básicas do sistema juntamente com os dados de entrada (percepções), de saída (ações) e qualquer informação importante de troca de dados.
- Fase de Design Arquitetural: Identifica os agentes e como eles interagem.
- Fase de Design Detalhado: Detalha a parte interna do agente e define como o agente irá cumprir suas tarefas.

De acordo com Padgham e Winikoff (2005b), a Prometheus tem sido utilizada tanto em oficinas industriais como em cursos universitários. Provando-se desde então ser eficaz ajudando estudantes e profissionais na documentação de projetos com SMAs, e encontra-se em um nível de maturidade que suporta algumas ferramentas desenvolvidas, tal como a *Prometheus Design Tool (PDT)*³.

²<http://www.agent-software.com/>

³<http://code.google.com/p/pdt-plugin/>

Figura 2.1: Fases da Metodologia Prometheus



Como a Prometheus foi utilizada para desenvolver o SMA deste trabalho, serão descritas na Subseção 4.2.6 a sua aplicação, abordando as três fases da metodologia explanadas anteriormente.

2.6 Considerações Finais

Neste capítulo foi apresentado um embasamento teórico sobre RE, prática de programação com robôs, AIs e SMAs, laboratório remoto de robótica e metodologia Prometheus. A abordagem desenvolvida fornece apoio à aprendizagem cooperativa, pois ela é um fator essencial em processos de aprendizagem. No próximo capítulo são apresentados os trabalhos correlatos que envolvem as áreas do conhecimento abordadas neste trabalho.

Capítulo 3

Trabalhos Correlatos

3.1 Revisão Sistemática da Literatura

Com intuito de avaliar o estado das pesquisas concernentes ao presente trabalho, inicialmente foi realizada uma RSL por Almeida e Netto (2015). Essa RSL utilizou a metodologia apresentada por Keele (2007) que descreve os passos necessários para a condução de uma RSL. Neste sentido, buscou-se encontrar pesquisas atuais e relevantes que têm sido realizadas com RE, aplicada ao ensino e aprendizagem de programação.

O mapeamento sistemático é um meio de categorizar e resumir informações existentes sobre uma questão de pesquisa de uma maneira imparcial. Essa RSL foi realizada em três etapas: protocolo da revisão sistemática, extração das informações e análise dos resultados. Além disso, foram analisadas publicações científicas, com o propósito de caracterizar e identificar técnicas utilizadas para desenvolver sistemas que auxiliem no ensino e aprendizagem de programação utilizando robôs, do ponto de vista dos pesquisadores no meio acadêmico.

3.1.1 Objetivo

O objetivo do protocolo da RSL foi definido conforme o paradigma GQM (*Goal-Question-Metric*) desenvolvido por Basili, Caldiera e Rombach (1994) que define métricas a serem coletadas de forma a atender determinados objetivos, e encontra-se descrito na Tabela 3.1.

3.1.2 Questões de Pesquisa

Q1. Quais pesquisas têm sido desenvolvidas para apoiar o ensino e aprendizagem de programação com robôs?

Tabela 3.1: Objetivo conforme paradigma GQM

ANALISAR	Publicações científicas
COM O PROPÓSITO DE	Caracterizar
EM RELAÇÃO A	Identificação de laboratórios virtuais e remotos de robótica como apoio ao ensino de programação.
DO PONTO DE VISTA DOS	Pesquisadores
NO CONTEXTO	Acadêmico

- Q2. Quais as ferramentas utilizadas para o desenvolvimento desses sistemas?
 Q3. Quais as técnicas utilizadas para avaliar esses sistemas?

3.1.3 Estratégia de Busca

As buscas avançadas foram realizadas em bibliotecas digitais (SCOPUS, IEEE Xplore e ACM Digital Library). Buscas manuais foram realizadas em eventos e conferências que possuem estudos relevantes na área de Informática na Educação, tais como: Conferência Internacional sobre Informática na Educação (TISE), Workshop de Informática na Escola (WIE) e Simpósio Brasileiro de Informática na Educação (SBIE).

a) Critérios adotados para seleção das fontes: Para seleção das fontes, levou-se em consideração eventos relevantes e bem qualificados com a área de pesquisa, assim como algumas bibliotecas digitais (ambos informados no tópico acima).

b) Restrições: A busca temporal foi realizada no período de 2010 a 2014. Não foi possível ter acesso a todos os artigos retornados na máquina de busca, tendo em vista que alguns eram pagos ou não estavam disponíveis em formato de artigo completo.

3.1.4 Idioma

O idioma escolhido para ser utilizado nas máquinas de busca foi o Inglês, tendo em vista que é o idioma adotado por grande parte das conferências e periódicos existentes relacionados com o tema de pesquisa. Para buscas manuais, o idioma foi de acordo com o utilizado no evento pesquisado.

3.1.5 Métodos de Busca das Publicações

a) Expressão de Busca: A fim de fazer a busca automática nas bibliotecas digitais escolhidas, foi utilizada uma string de busca que consiste de três partes, com o objetivo de cobrir os conceitos que representam programação de robôs a distância. A primeira parte está relacionada com os estudos

que são desenvolvidos para auxiliar no ensino de programação, a segunda parte está relacionada com os estudos que utilizam robôs para este fim, a terceira parte está relacionada com estudos que permitem esta manipulação a distância (seja através de um laboratório virtual ou remoto). A Tabela 3.2 apresenta a string de busca utilizada nesta pesquisa.

Tabela 3.2: Expressão de Busca

Conceito	Termos Alternativos e Sinônimos
Programação	("program"OR "programming") AND
Robô	("robot"OR "robots") AND
Distância	("telerobotic"OR "remote lab"OR "virtual lab" OR "e learning"OR "e-learning")

A pesquisa foi realizada pela aplicação da string de busca para os mesmos metadados (ou seja, título, resumo e palavras-chave) de cada artigo para todas as fontes. A sintaxe da string de busca foi adaptada para ser aplicada em cada biblioteca digital.

b) Busca Manual: Para buscas manuais, os termos de pesquisa apresentados na Tabela 3.1 também foram levados em consideração a fim de realizar uma pesquisa coerente. Neste sentido, algumas palavras-chave foram utilizadas, tais como: robô, robótica, telerobótica, programação, ensino de programação, aprendizagem de programação, laboratório virtual, laboratório remoto.

3.1.6 Procedimentos de Seleção e Critérios

Cada estudo encontrado a partir da busca avançada ou manual foi avaliado, a fim de decidir se deve ou não ser incluído, considerando seu título, resumo e palavras-chave. Para seleção dos artigos, foram utilizados os seguintes critérios de inclusão e exclusão que são apresentados na Tabela 3.3:

Tabela 3.3: Critérios de Inclusão e Exclusão

Critérios de Inclusão	Critérios de Exclusão
CI.1. Estar disponível em forma de artigo completo em uma biblioteca digital. CI.2. Artigos envolvendo ensino de programação utilizando robôs. CI.3. Permitir que os robôs sejam manipulados ou programados a distância.	CE.1. Artigos que não atendam aos critérios de inclusão. CE.2. Artigos que apresentem apenas recomendações ou orientações para desenvolvimento de laboratórios remotos e virtuais. CE.3. Artigos duplicados do mesmo estudo em diferentes fontes. CE.4. Artigos introdutórios para congressos e workshops, que não apresentem resultados de implementação.

A string de busca informada na Tabela 3.2 retornou os seguintes resultados:

Tabela 3.4: Resultados obtidos com a string de busca

Biblioteca Digital	String de Busca	Pago	1º Filtro	2º Filtro
SCOPUS	60	6	17	5
IEEE	21	-	9	4
ACM	1	-	1	1
TOTAL	82	6	27	10

Após a inclusão dos artigos em uma base de dados única restaram 82 artigos para serem avaliados. A primeira avaliação foi realizada através da leitura do título e do abstract dos artigos, sendo que, 27 artigos foram selecionados por fazerem parte da área de estudo. Na segunda etapa foi feita a leitura da introdução e conclusão dos demais artigos, sendo que, nesta etapa 17 artigos foram excluídos pois mesmo utilizando robôs para apoiar o ensino de programação, não permitem a programação dos mesmos através de uma plataforma remota. Restaram 10 artigos que foram lidos na íntegra para a sumarização das informações.

As buscas de novos artigos foram realizadas continuamente, durante todo o período de desenvolvimento desta pesquisa. Após a leitura dos artigos selecionados, os trabalhos foram organizados em três categorias: Robótica no Contexto Educacional, Aprendizagem de Programação com Robô e Programação de Robôs a Distância. A seguir são analisadas as contribuições desses trabalhos, assim como as ferramentas utilizadas, fazendo a verificação de suas vantagens e limitações no processo de ensino e aprendizagem.

3.2 Robótica no Contexto Educacional

D'Abreu, Mirisola e Ramos (2011) expõe um ambiente de robótica pedagógica de baixo custo (RPBC) chamado *Br GOGO*, construído com componentes eletrônicos e materiais alternativos de baixo custo para montar dispositivos robóticos que possam ser controlados com programas elaborados em computadores de baixo custo. Esse trabalho possui como objetivo apoiar o aprendizado do ensino de ciências de estudantes do ensino fundamental e médio.

Pinto, Elia e Sampaio (2012) apresentam a capacitação de docentes no campo da robótica educacional, aliado a tecnologias de baixo custo que possibilitem a inserção da robótica no cotidiano das escolas públicas. O curso possui dois eixos de orientação: pedagógico e tecnológico. Foi utilizada a placa GOGO-Board, e o software livre Arduíno para execução dos experimentos.

Fernandes (2013) apresenta um simulador de robôs do tipo Lego Minds-

torms NXT que permite a realização de uma aula de RE sem a utilização de kits de robótica. O robô virtual pode ser programado para seguir um conjunto de instruções definidas pelo programador, utilizando o software R-Educ. Os resultados mostram que a utilização de simuladores robóticos reduzem custos financeiros, facilitam testes e diminuem danos.

D'Abreu e Bastos (2013) relatam a implantação da prática da RE inserida ao currículo no Ensino Fundamental. Nesse sentido, os autores apresentam um projeto cujo objetivo é detalhar o processo de formação de professores para a apropriação dos conhecimentos de robótica integrados ao currículo. As atividades foram realizadas em sala de aula, de modo que professores e alunos se apropriassem do uso do laptop educacional Classmate, associando-o à construção de dispositivos robóticos controlados por esse computador. Os robôs são programados com o software Scratch.

A utilização da robótica pedagógica de baixo custo (RPBC) associada à conteúdos estudados nas aulas de Física é apresentada no trabalho de Trentin, Pérez e Teixeira (2013). Nesse trabalho não é esperado que os alunos interajam ou programem o robô por meio de comandos de linguagem de programação, já que eles cursam o Ensino Médio. Dessa maneira, foi criada uma interface em um navegador Web possibilitando o controle dos movimentos dos robôs e geração de gráficos.

Uma metodologia pedagógica para ambientação e inserção à lógica de programação é apresentada no trabalho de (FRIEDRICH et al., 2012). Voltado para crianças de sete a dez anos de idade, os autores utilizam como meio auxiliar de ensino o software Logo e o projeto Lego Mindstorms em atividades de montagem de blocos e programação.

A Tabela 3.5 apresenta um breve resumo dos trabalhos desta seção, de acordo com algumas características, tais como: tecnologias utilizadas, disciplinas, linguagem de programação e kit robótico utilizado.

Tabela 3.5: Resumo dos Artigos de Robótica no Contexto Educacional

Autores	Características			
	Tecnologias Utilizadas	Disciplinas	Linguagem de Programação	Kit Robótico
(D'ABREU; MIRISOLA; RAMOS, 2011)	Materiais alternativos de baixo custo.	Ciências	Logo	Placa Gogo board
(PINTO; ELIA; SAMPAIO, 2012)	Arduíno, placa Gogo board, computadores Classmate.	-	Arduíno	Placa Gogo board
(FERNANDES, 2013)	Simulador robótico implementado em Java. Ambientes gráficos implementados em JMonkey Engine.	Interdisciplinar	R-Educ	-
(D'ABREU; BASTOS, 2013)	Construção e programação de dispositivos robóticos via laptop.	Interdisciplinar	Scratch	Arduíno KTR-12
(TRENTIN; PÉREZ; TEIXEIRA, 2013)	Materiais de baixo custo.	Física	-	Arduíno
(FRIEDRICH et al., 2012)	Kit Lego Mindstorms.	Lógica de Programação	Logo	Lego Mindstorms.

3.3 Aprendizagem de Programação com Robô

Sell, Seiler e Ptasik (2012) apresentam o kit robótico HomeLab que funciona como um ambiente de laboratório portátil que pode ser ligado a um computador, e operado em casa ou no local de trabalho. Essa solução foi desenvolvida para o ensino e aprendizagem de robótica e sistemas embarcados. Os autores desenvolveram um laboratório que simula a criação de protótipos e experimentos robóticos. O laboratório desenvolvido é uma plataforma de Internet, onde diferentes laboratórios remotos e virtuais são integrados. Este projeto está voltado para apoiar o ensino de programação, assim como controlar os dispositivos conectados.

O RoboMind Halma (2015) é um programa que oferece uma linguagem de programação simples para um robô em um mundo bidimensional. O robô se movimenta através da programação do usuário, que indica a quantidade de casas que ele deve andar. O robô pode ser equipado com apenas um sensor ultrassônico que pode sensoriar se há objetos na frente ou nos lados do robô.

Buiu, Buga e Coman (2013) detalham o desenvolvimento de um curso que aborda conceitos de robótica e fundamentos de realidade virtual. Os autores fazem uso de pacotes de simulação para a construção de robôs virtuais, e trabalham também com robôs reais que podem ser programados nas linguagens Python e C. Os simuladores Webots e iCub são utilizados para análise de experimentos. A parte virtual do curso é realizada no ambiente Second Life, permitindo que os usuários construam e realizem testes sobre os robôs virtuais.

A Tabela 3.6 apresenta um breve resumo dos trabalhos desta seção, de acordo com algumas características, tais como: tecnologias utilizadas, disciplinas, linguagem de programação e integração com Ambiente Virtual de Aprendizagem (AVA).

Tabela 3.6: Resumo dos Artigos de Aprendizagem de Programação com Robô

Autores	Características			
	Tecnologias Utilizadas	Disciplinas	Linguagem de Programação	Integração com AVA
(SELL; SEILER; PTASIK, 2012)	Framework Avroa e linguagem de programação Java.	Robótica e Sistemas Embarcados	-	-
(HALMA, 2015)	Não informado.	Lógica de Programação	Específica do software Robomind	-
(BUIU; BUGA; COMAN, 2013)	Second Life. Simuladores Webots e iCub.	Robótica	Python e C	-

3.4 Programação de Robô a Distância

Um Laboratório de Acesso Remoto (LAR) para experimentos com robôs móveis é apresentado no trabalho de Guimarães et al. (2003). Neste projeto, são utilizados alguns frames para prover acesso remoto aos robôs móveis autônomos. Enquanto muitos projetos permitem que os usuários apenas manipulem o robô, este laboratório permite executar experimentos complexos programando-o a distância com a linguagem C.

Souza et al. (2011) apresentam o desenvolvimento de um Ambiente Telorobótico Educacional baseado em Sistemas Multiagente. O sistema possibilita a seleção de cenários associados a desafios distintos, de acordo com os níveis de habilidade do usuário, permitindo flexibilidade na ativação de cenários variados e uma interação usuário-sistema utilizando agentes.

Grandi, Falconi e Melchiorri (2011) apresentam um laboratório remoto de robótica que permite aos alunos definir um controlador completo para um robô móvel. Desta maneira, é possível interagir tanto com o ambiente, assim como com outros robôs presentes na arena (eventualmente conduzidos por diferentes usuários). Antes de trabalhar com os robôs reais, os usuários podem testar seus algoritmos em um ambiente virtual, onde os algoritmos de teste são reproduzidos.

Kulich et al. (2013) apresentam o SyRoTek, uma plataforma de e-learning para robótica móvel, Inteligência Artificial (IA) e engenharia de controle. Esse ambiente gerencia e disponibiliza um conjunto de robôs móveis autônomos em uma arena, para que os usuários possam executar os experimentos. Os robôs podem ser controlados remotamente via tecnologia ActiveX, ou por um programa escrito em C++, Delphi ou Java.

Um laboratório virtual e remoto de robótica é apresentado no trabalho de Chaos et al. (2013). Os autores utilizam as tecnologias EJS (Easy Java Simulation), Matlab, e LabView para o desenvolvimento do laboratório. A aplicação permite trabalhar com robôs simulados (laboratório virtual) e robôs reais (laboratório remoto). Os alunos podem desenvolver algoritmos para controle do robô, simulá-lo no laboratório virtual e implantá-lo no laboratório remoto para fins de testes.

Casini et al. (2014) apresentam um laboratório de acesso remoto (LAR) para experimentação com um time de robôs móveis. Os robôs são construídos com a tecnologia Lego Mindstorms, e os usuários exploram o ambiente Matlab para o desenvolvimento de estratégias de controle da equipe de robôs. O laboratório pode ser acessado por meio de navegadores Web padrão e uma interface gráfica que lhes permite selecionar o tipo de experimento que deseja realizar.

A Tabela 3.7 apresenta um breve resumo dos trabalhos desta seção, de

acordo com algumas características, tais como: tecnologias utilizadas, linguagem de programação, utilização de SMA como apoio, integração com AVA e utilização de telerobótica. Ainda nesta tabela, apresentamos na última linha as características do trabalho desenvolvido nesta dissertação.

Tabela 3.7: Resumo dos Artigos de Programação de Robôs a Distância

Autores	Características				
	Tecnologias Utilizadas	Linguagem de Programação	SMA como Apoio	Integração com AVA	Telerobótica
(GUIMARAES et al., 2003)	C++, Java	C	-	-	Sim
(SOUZA et al., 2011)	PHP, MySQL, J2SE	-	Sim	Moodle	Sim
(GRANDI; FALCONI; MELCHIORRI, 2011)	Java Unibot Simulation Environment (JUSE)	Matlab	Sim	-	Sim
(KULICH et al., 2013)	VRML, ActiveX	C++, Delphi ou Java	-	-	Sim
(CHAOS et al., 2013)	Easy Java Simulation (EJS), LabView	Matlab	-	Moodle	Sim
(CASINI et al., 2014)	Java Applet	Matlab	Sim	-	Sim
Proposta	PHP, MySQL, JADE, ActiveX, NXT-Python	Python	Sim	Moodle	Sim

3.5 Considerações Finais

Os projetos descritos nas seções anteriores possuem um reconhecimento significativo junto ao meio acadêmico. Desse modo, o estudo desses trabalhos sugere uma representação simplificada do estado em que se encontram as pesquisas, em termos tecnológicos, nas áreas de RE, aprendizagem de programação com robôs e programação de robôs a distância.

Com base no levantamento realizado, foi possível identificar que a robótica no contexto educacional (Seção 3.2) tem sido utilizada de forma interdisciplinar, associando a aplicação desenvolvida aos kits *Legó Mindstorms* ou *Arduíno*.

Quando analisados os trabalhos sobre aprendizagem de programação com robôs (Seção 3.3), é possível perceber que as tecnologias utilizadas para o desenvolvimento da aplicação são diversas. Dos trabalhos apresentados, apenas o trabalho desenvolvido por Halma (2015) auxilia no ensino de lógica de programação. O trabalho desenvolvido por Buiu, Buga e Coman (2013) permite que os robôs sejam programados nas linguagens Python e C, porém não permite que isso seja feito remotamente.

Algumas plataformas que permitem manipulação e programação de robôs a distância foram apresentadas na Seção 3.4 desta dissertação. Os trabalhos

de (GRANDI; FALCONI; MELCHIORRI, 2011; CHAOS et al., 2013; CASINI et al., 2014) permitem que os robôs sejam programados no Matlab.

Os trabalhos de Souza et al. (2011), Chaos et al. (2013) apresentam a integração de um laboratório remoto ao AVA Moodle. Os robôs podem ser programados na linguagem C no trabalho de Guimarães et al. (2003), e em C++, Delphi ou Java no trabalho de Kulich et al. (2013). Não ficou explícito nesses trabalhos as técnicas utilizadas para avaliar esses laboratórios.

Todos os trabalhos citados anteriormente contribuíram e motivaram o desenvolvimento desta dissertação. Como diferencial desta pesquisa, desenvolvemos um laboratório remoto integrado à um AVA, que permite programar robôs através da linguagem de programação Python com apoio de SMAs.

Capítulo 4

Laboratório Remoto de Robótica

Este capítulo destina-se a apresentar a modelagem do sistema, baseado no paradigma de agentes, a interface gráfica, o funcionamento e as ferramentas utilizadas no desenvolvimento do LERP. A Seção 4.1 apresenta uma visão geral do laboratório desenvolvido, explanando a arquitetura do sistema e uma breve introdução sobre os agentes. A Seção 4.2 apresenta os casos de uso e diagramas de sequência do sistema. Ainda nessa Seção é apresentada a modelagem dos AIs conforme a metodologia Prometheus. A Seção 4.3 aborda a interface gráfica e funcionamento do LERP. Neste sentido, é apresentada a interface de cadastramento de usuários, agendamento de experimentos, a sala de desafios e o processo de manipulação remota. A Seção 4.4 apresenta as tecnologias utilizadas para o desenvolvimento desta dissertação. Por fim, a Seção 5 apresenta os testes realizados com o sistema proposto e os resultados obtidos.

O LERP disponibiliza na Internet um espaço onde é possível realizar a prática de programação utilizando robôs *Lego Mindstorms* na linguagem de programação Python, mediado pela tecnologia SMA. A estratégia adotada para atingir o objetivo do trabalho (ver Seção 1.3), percorreu etapas como a especificação inicial dos AIs empregando a UML, a utilização da metodologia Prometheus na criação dos modelos gráficos e a implementação do LERP.

4.1 Visão Geral do Sistema

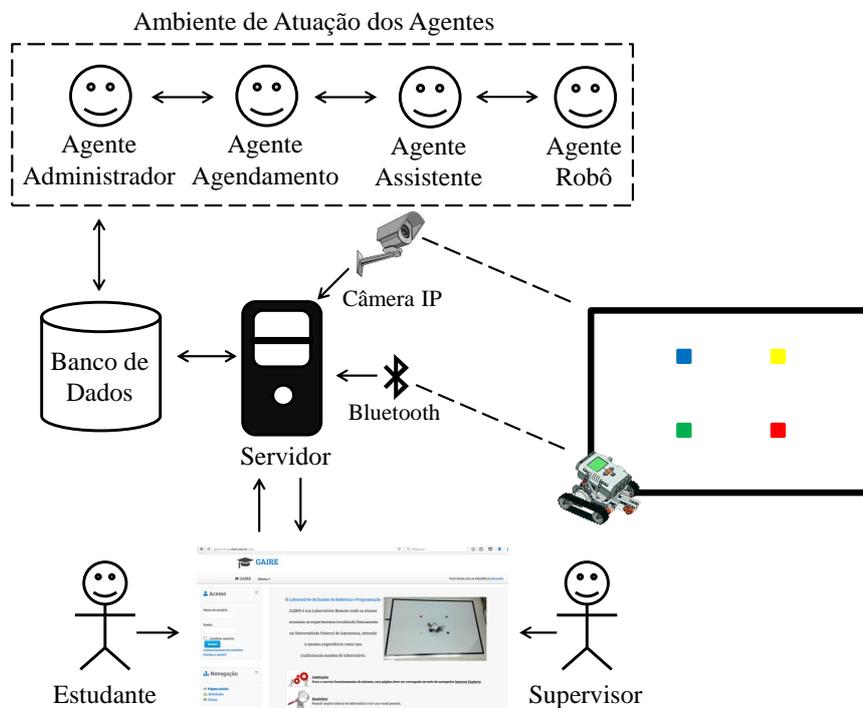
O LERP desenvolvido encontra-se disponível em um AVA acessado via Web, que permite que mesmo pessoas geograficamente distantes tenham experiência com programação utilizando robôs. O laboratório se caracteriza por possibilitar a aprendizagem colaborativa, com apoio de um SMA, para auxiliar no processo de aprendizagem de programação. Desta maneira, usuários

que possuem acesso à laboratórios de informática, mas que não contêm kits robóticos, podem usufruir de experiências desta natureza.

Para acesso aos experimentos robóticos do LERP, é necessário que o usuário esteja cadastrado no sistema. Para realização dos experimentos é necessário também um agendamento prévio tendo em vista a disponibilidade dos kits de robótica para uso. Neste ambiente os alunos têm desafios propostos a serem superados com o apoio de um SMA para auxiliar na realização dos mesmos. A programação dos robôs deve ser realizada na linguagem de programação NXT-Python.

O SMA desenvolvido oferece um serviço de ajuda no cadastramento dos usuários, confirmação de acesso à sala virtual, realização de reservas, agendamento de experimentos, liberação e controle da seção na sala de desafios, e programação remota do robô. A escolha deste paradigma de programação, justifica-se pela natureza distribuída do LERP, e por este servir de apoio à problemas semelhantes e de mesmo grau de complexidade. Os trabalhos desenvolvidos por Souza et al. (2011), Grandi, Falconi e Melchiorri (2011) sustentam essa afirmação. A Figura 4.1 apresenta uma visão geral do laboratório remoto de robótica desenvolvido.

Figura 4.1: Arquitetura do LERP



A arquitetura do sistema é composta por quatro agentes, cada qual com sua função, a saber:

- Agente Administrador: responsável por liberar o acesso ao AVA e armazenar informações dos usuários (maiores informações sobre a atuação do agente Administrador podem ser encontradas na Subseção 4.2.2).
- Agente Agendamento: responsável por gerenciar o agendamento para realização de experimentos no LERP (a Subseção 4.2.3 apresenta detalhes da atuação do agente Agendamento).
- Agente Assistente: auxilia os usuários do LERP no processo de programação do robô. Sugere desafios baseado no histórico de cada um, o que ajuda a estimar a aprendizagem do usuário do ambiente (maiores informações sobre a atuação do agente Assistente podem ser obtidas na Subseção 4.2.4).
- Agente Robô: responsável por executar os movimentos do robô conforme programado pelo usuário, e apresentar no AVA. Realiza também a análise sintática do código-fonte feito pelo usuário, antes de submeter ao servidor. (detalhes sobre a atuação do agente Robô podem ser encontrados na Subseção 4.2.5).

Os agentes atuarão no ambiente por meio de troca de mensagens enquanto o usuário estiver realizando os experimentos, além de oferecerem suporte com relação aos conteúdos de programação.

A Figura 4.1 ainda apresenta as ferramentas utilizadas no processo de comunicação entre o usuário e o kit robótico. Para tanto, será utilizado o AVA Moodle como interface gráfica do sistema, o banco de dados MySQL, o servidor Web, a câmera que irá capturar as imagens do laboratório físico, o kit robótico *Lego Mindstorms* e a tecnologia Bluetooth. As tecnologias utilizadas estão detalhadas na Seção 4.4 deste capítulo.

A próxima seção apresenta a modelagem do SMA, mostrado na Figura 4.1. Neste sentido, serão destacados os Casos de Uso (CASUs) e alguns Diagramas de Sequência (DSs) dos atores envolvidos na abordagem deste trabalho.

4.2 Casos de Uso e Diagrama de Sequência

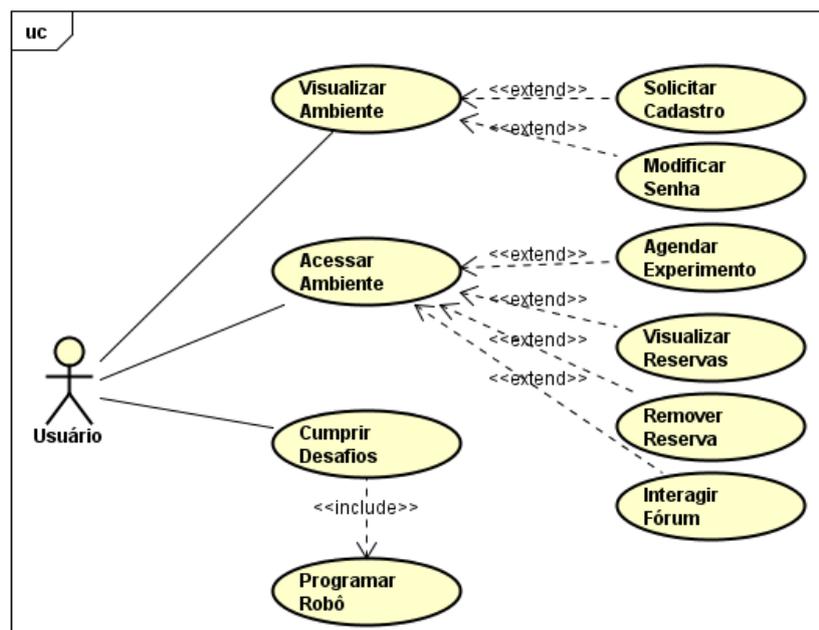
Um caso de uso é uma metodologia utilizada no desenvolvimento de sistemas para identificar, esclarecer e organizar os requisitos do sistema. Os CASUs são constituídos por um conjunto de possíveis sequências de interações entre sistemas e usuários em um ambiente particular e relacionado com um objetivo

particular. Além disso, os CASUs auxiliam na tarefa de análise, especificando funcionalidades e comportamentos do sistema. Esta seção apresenta os CASUs do LERP, além dos DSs de alguns deles e quais operações devem ser disparadas entre os objetos envolvidos. Os DSs são usados para mostrar as interações entre os objetos na ordem sequencial que ocorrem em um caso de uso.

4.2.1 Ator envolvido: o Usuário

No LERP, um usuário pode ser definido como um supervisor ou aluno. O supervisor é o responsável por acompanhar as ações dos alunos, a fim de analisar se os desafios foram concluídos ou não com êxito. A Figura 4.2 ilustra as ações que o usuário pode realizar no LERP, por meio da visualização de seus CASUs base e dos seus relacionamentos.

Figura 4.2: Caso de Uso do ator Usuário



Casos de Uso Base:

- **Visualizar Ambiente:** ao tentar visualizar a página inicial do LERP, o usuário só terá acesso caso tenha realizado seu cadastro anteriormente.

- **Acessar Ambiente:** após o usuário se cadastrar no sistema, o agente Administrador irá disparar um e-mail contendo um link para confirmar a inscrição no curso. Após a confirmação, o acesso ao LERP estará liberado.
- **Cumprir Desafios:** tendo acesso à sala do LERP, o usuário verá 5 (cinco) desafios disponíveis a serem concluídos. O usuário que nunca realizou nenhum desafio deverá começar pelo primeiro no horário reservado para ele. Caso o usuário já tenha concluído alguma desafio anteriormente, deverá recomeçar a partir do próximo. Não será possível cumprir desafios caso não tenha concluído o anterior.

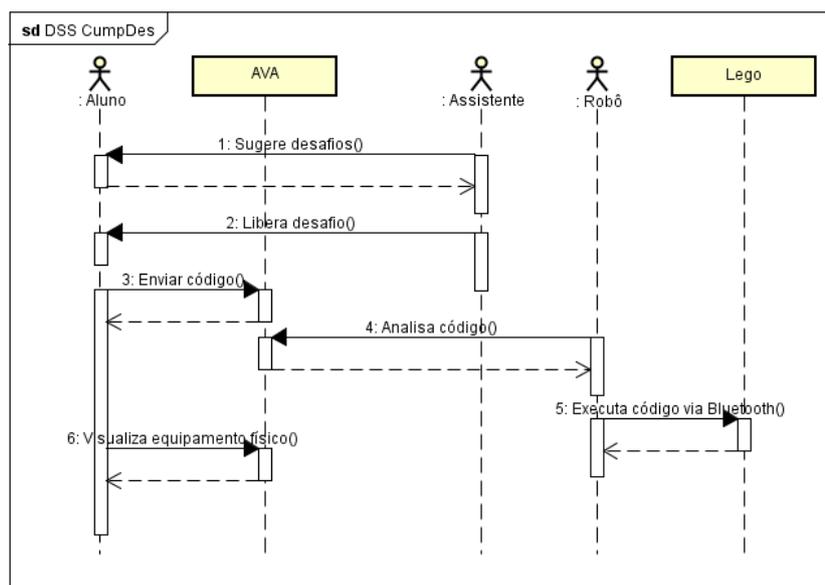
Relacionamentos dos Casos de Uso Base:

- **Solicitar Cadastro:** motivado pelo interesse em ter acesso à experimentos robóticos, o usuário precisa realizar seu cadastro no AVA para ter acesso à página do LERP e aos recursos disponíveis.
- **Modificar Senha:** caso o usuário esqueça sua senha de acesso, poderá modificá-la preenchendo o formulário disponível no ambiente. A alteração pode ser realizada informando o endereço de e-mail ou nome de usuário cadastrado no sistema.
- **Agendar Experimento:** após o usuário ter efetuado o cadastro e estar logado no sistema, ele tem a opção de agendar experimento para que possa programar o robô remotamente.
- **Visualizar Reserva:** tão logo o usuário agende um experimento, ele receberá um e-mail de confirmação informando a reserva conforme data e horário escolhido. A visualização da reserva efetuada pode ser realizada também através da sala do LERP.
- **Remover Reserva:** caso o usuário desista de uma reserva efetuada, ele poderá removê-la clicando no ícone destinado para este fim.
- **Interagir Fórum:** o usuário tem a opção de interagir com outros usuários utilizando o recurso Fórum, para tratar de assuntos relacionados à robótica.
- **Programar Robô:** uma vez que o usuário possua um horário agendado para realização de experimentos, ele poderá selecionar o desafio a ser cumprido. Desta maneira, ele poderá começar a realizar a programação do robô, de acordo com o desafio proposto.

Diagrama de Sequência do Caso de Uso Cumprir Desafios

O diagrama da Figura 4.3 representa a sequência de eventos que ocorrem no CASU Cumprir Desafios. O CASU é iniciado após o usuário ter acessado à sala de experimentos do LERP.

Figura 4.3: Diagrama de Sequência do Caso de Uso Cumprir Desafios



4.2.2 Ator envolvido: o agente Administrador

A Figura 4.4 apresenta o caso de uso base do agente Administrador, assim como os relacionamentos.

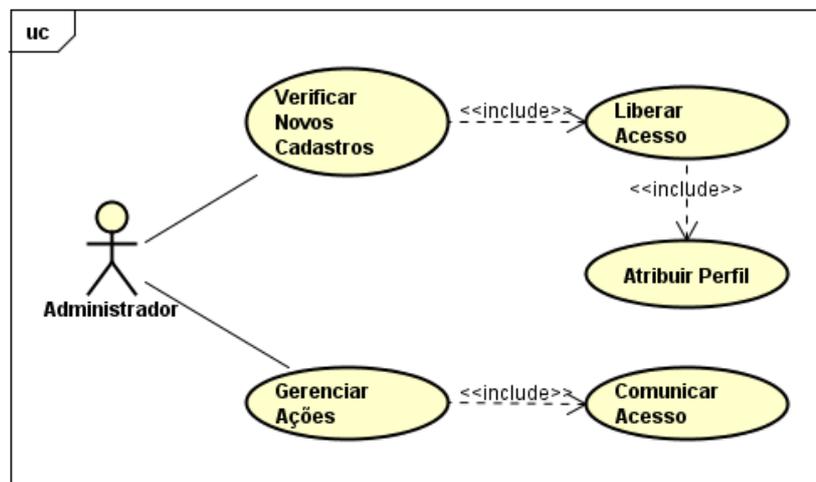
Casos de Uso Base:

- **Verificar novos cadastros:** o agente Administrador verifica no banco de dados se há novos cadastros. Caso exista, libera o acesso do usuário ao LERP.
- **Gerenciar Ações:** após o agente Administrador liberar o acesso ao LERP, ele armazena essa informação e repassa para o agente Assistente.

Relacionamentos dos Casos de Uso Base:

- **Liberar Acesso:** havendo a existência de novos cadastros, o agente Administrador realiza o processo de concessão de acesso ao LERP.

Figura 4.4: Caso de Uso do Agente Administrador



- **Atribuir Perfil:** o agente Administrador atribui o perfil de estudante ao usuário após liberar o acesso ao LERP.
- **Comunicar Acesso:** tão logo seja liberado o acesso à sala de experimentos, o agente Administrador informa ao agente Assistente que o usuário já possui acesso ao sistema, e solicita que o agente Assistente atribua desafios aos usuários.

Diagrama de Sequência do Caso de Uso Liberar Acesso:

A Figura 4.5 apresenta o diagrama de sequência do CASU Liberar Acesso, que é feito pelo agente Administrador após o usuário ter preenchido o formulário de cadastro no AVA.

4.2.3 Ator envolvido: o agente Agendamento

A Figura 4.6 demonstra como será a atuação do agente Agendamento no LERP, apresentando o caso de uso e as relações existentes.

Casos de Uso Base:

- **Verificar Reservas:** o agente Administrador verifica as reservas realizadas para um dia específico. As solicitações de reservas ficam armazenadas na tabela mdl_lerp_reserva do BD. Após uma reserva ter sido efetuada, ela é encaixada na tabela mdl_lerp_agendamento, pertencente ao mesmo BD.

Figura 4.5: Diagrama de Sequência do Caso de Uso Liberar Acesso

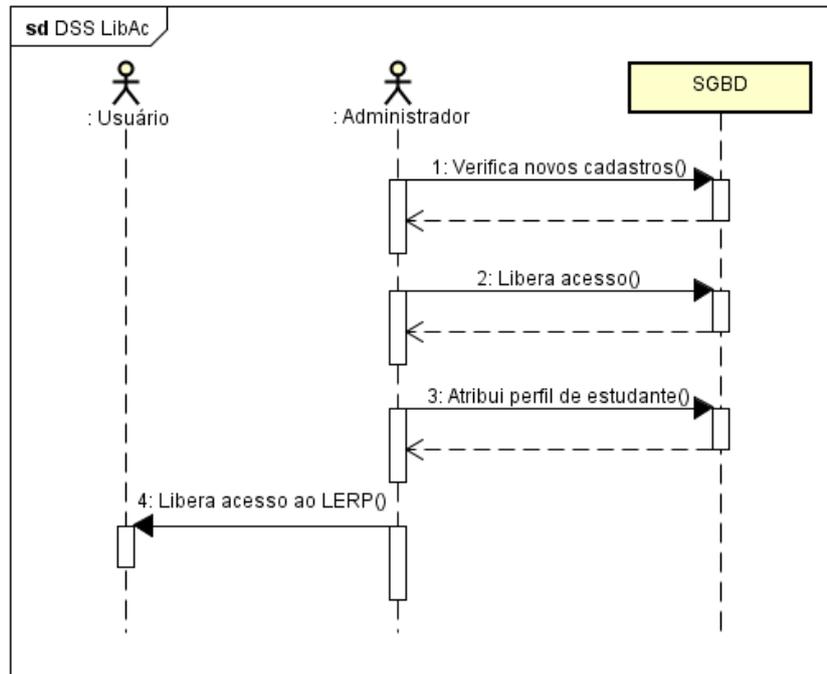
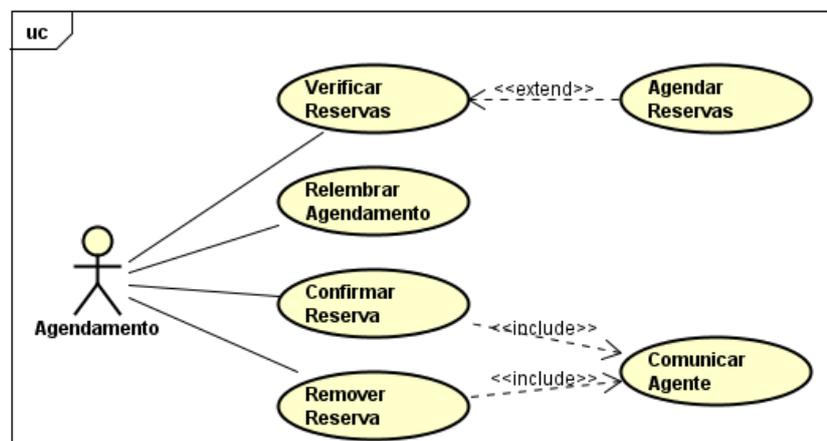


Figura 4.6: Caso de Uso do Agente Agendamento



- **Relembrar Agendamento:** o agente Administrador envia um e-mail para o usuário no dia do agendamento lembrando sobre o agendamento realizado. Este e-mail contém informações com data e hora da reserva, assim como a situação (confirmado ou cancelado).

- **Confirmar Reserva:** quando o usuário realiza a reserva de um experimento, fica a cargo do agente Agendamento confirmar esta reserva e verificar se é possível mudar para a situação de agendado no BD.
- **Remover Reserva:** quando o usuário deseja remover uma reserva, fica a cargo do agente Agendamento remover os registros nas tabelas mdl_lerp_reserva e mdl_lerp_agendamento do BD, e deixar o horário livre para que outro usuário possa agendar.

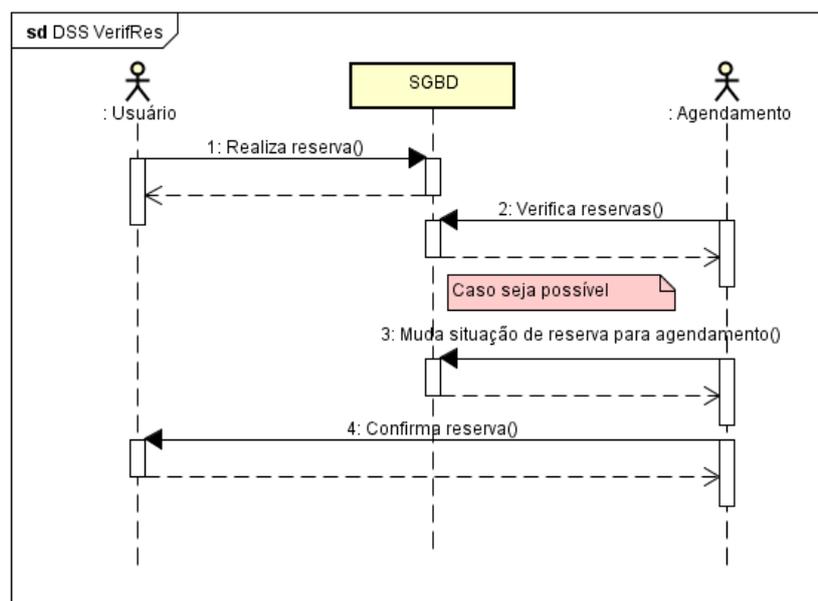
Relacionamentos dos Casos de Uso Base:

- **Agendar Reservas:** após verificar no BD se existem novas reservas, o agente Agendamento irá encaixar a reserva solicitada nos agendamentos.
- **Comunicar Agente:** após o usuário realizar ou remover uma reserva, o agente Agendamento comunica ao agente Assistente sobre a confirmação ou cancelamento da reserva.

Diagrama de Sequência do Caso de Uso Verificar Reservas:

O DSS Verificar Reservas está descrito na Figura 4.7 e inicia após o usuário agendar experimento, de acordo com a data e hora solicitada.

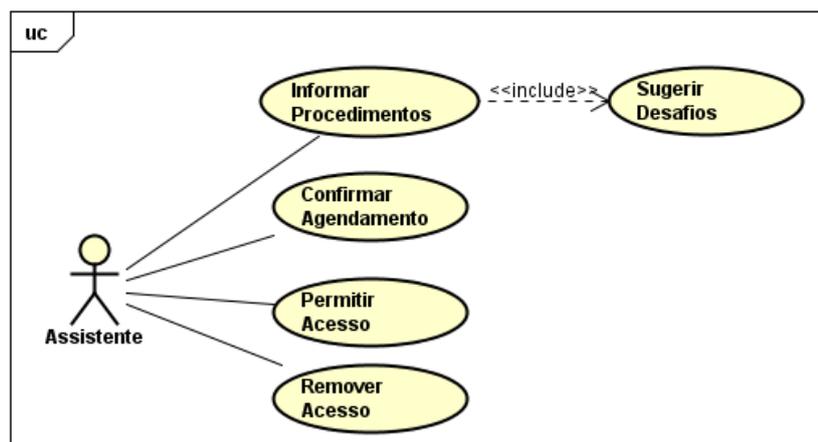
Figura 4.7: Diagrama de Sequência do Caso de Uso Verificar Reservas



4.2.4 Ator envolvido: o agente Assistente

A Figura 4.8 apresenta o caso de uso base do agente Assistente, assim como os relacionamentos.

Figura 4.8: Caso de Uso do Agente Assistente



Casos de Uso Base:

- **Informar Procedimentos:** após o usuário ter acesso ao LERP, o agente Assistente envia um e-mail ao usuário contendo as instruções de como realizar reservas, e informando que o mesmo já possui acesso ao sistema.
- **Confirmar Agendamento:** após o usuário ter realizado ou cancelado a reserva, o agente Assistente envia um e-mail ao usuário contendo informações sobre a confirmação ou cancelamento da reserva.
- **Permitir Acesso:** tão logo esteja no horário agendado para um usuário, o agente Assistente libera os desafios para que o usuário possa realizar, sendo permitido apenas um por vez.
- **Remover Acesso:** ao término do horário de um agendamento na sala de experimentos, o agente Assistente bloqueia o acesso do usuário aos desafios. Porém, o usuário continuará tendo acesso aos outros recursos disponíveis no LERP.

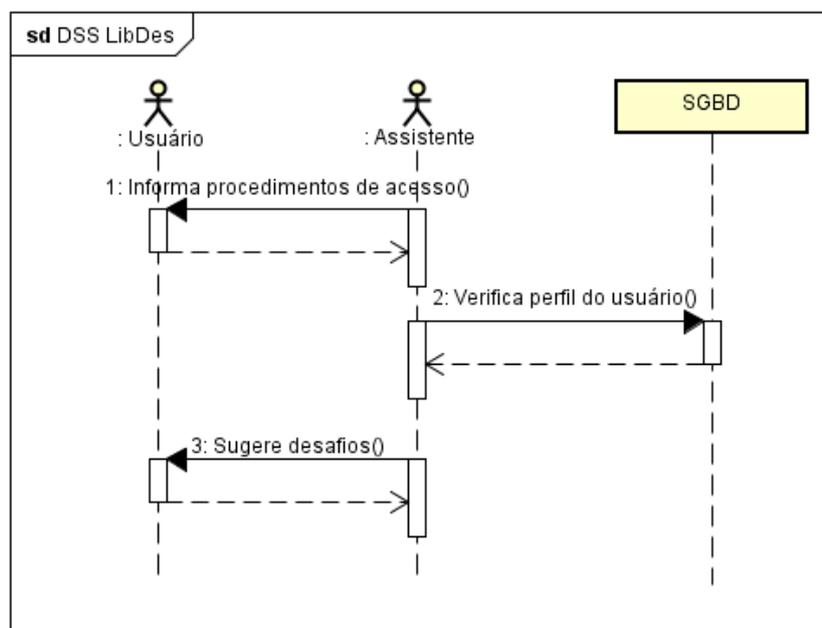
Relacionamentos dos Casos de Uso Base:

- **Sugerir Desafios:** baseado no histórico do usuário, o agente Assistente sugere desafios. Este agente faz uma consulta na tabela mdl_lerp_desafio_usuario do BD para consultar quais desafios o usuário já concluiu, e qual sugerir.

Diagrama de Sequência do Caso de Uso Sugerir Desafios:

O diagrama da Figura 4.9 retrata o CASU Sugerir Desafios, que ocorre após o usuário efetivar seu cadastro no LERP, e ter acesso ao mesmo.

Figura 4.9: Diagrama de Sequência do Caso de Uso Sugerir Desafios

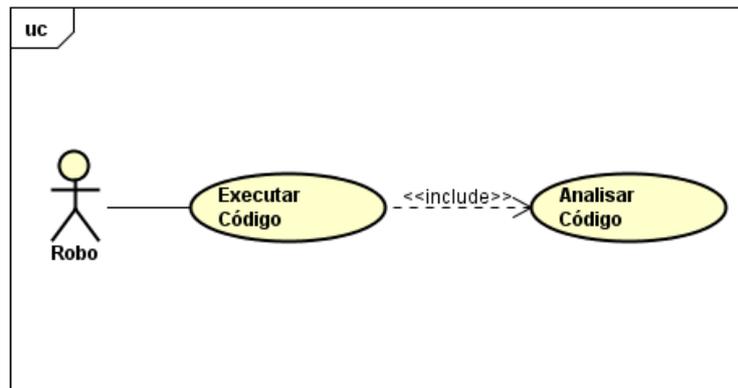
**4.2.5 Ator envolvido: o Agente Robô**

A Figura 4.10 demonstra como será a atuação do agente Robô no LERP, apresentando o caso de uso e as relações existentes.

Casos de Uso Base:

- **Executar Código:** o agente Robô cria um arquivo de execução com o código enviado pelo usuário no LERP, e envia os comandos via Bluetooth para o Lego.

Figura 4.10: Caso de Uso do Agente Robô



Relacionamentos dos Casos de Uso Base:

- **Analisar Código:** após o usuário enviar o código-fonte correspondente ao desafio que ele deve concluir, o agente Robô faz uma análise sintática do código e fornece um *feedback* pro usuário.

Diagrama de Sequência do Caso de Uso Executar Código

O DSS do CASU Executar Código é apresentado na Figura 4.11, e inicia após o usuário ter acesso à um desafio, e enviar o código-fonte para concluir o mesmo.

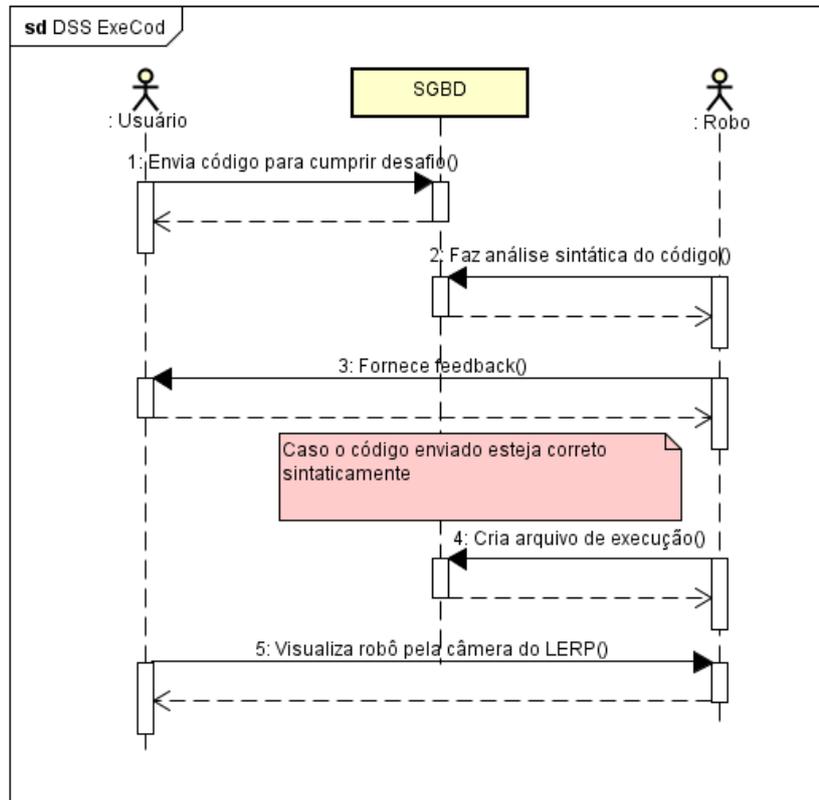
Baseado nos CASUs, nos DSs e nos AIs que atuam no LERP atuarão no LERP descritos nas subseções 4.2.1, 4.2.2, 4.2.3, 4.2.4, 4.2.5, a próxima subseção apresenta a modelagem do SMA desenvolvido utilizando a metodologia Prometheus.

4.2.6 Modelagem Utilizando Prometheus

Esta Subseção é uma extensão da Seção 2.5, onde foi realizada uma breve explanação a respeito da metodologia Prometheus. A escolha desta metodologia para modelar o SMA desenvolvido neste trabalho, justifica-se por englobar um conjunto de conceitos, além de possuir notações para modelar o software atendendo as fases de requisitos, design e implementação, e possui um processo bem elaborado a fim de produzir um software.

Além disso, a Prometheus possui um mecanismo de estruturação hierárquica que permite que o projeto seja realizado em vários níveis de abstração. Os diagramas permitem associar agentes a papéis, que executam tarefas específicas e trocam mensagens baseadas em protocolos de comunicação definidos

Figura 4.11: Diagrama de Sequência do Caso de Uso Executar Código



para o sistema. A seguir são descritas as fases de Especificação do Sistema, Projeto da Arquitetura e Projeto Detalhado.

4.2.6.1 Fase 1. Especificação do Sistema

Esta fase focaliza na identificação das funcionalidades básicas do sistema, suas entradas (percepções), suas saídas (ações) e as principais origens de dados compartilhados (DÁRIO, 2005). Neste momento, será explanado como o sistema de agentes interage com o ambiente.

De acordo com Padgham e Winikoff (2005a), a fase de especificação do sistema concentra-se em 4 (quatro) atividades:

- Identificar os objetivos do sistema.
- Identificar as funcionalidades básicas do sistema.
- Descrever os cenários de caso de uso que ilustram o funcionamento do sistema.

- Especificar a interface entre o sistema e o ambiente em termos de ações e percepções.

Não é necessário que essas quatro etapas sejam realizadas em sequência. Os objetivos precisam estar bem definidos, para que o desenvolvedor saiba o que o sistema precisa ser capaz de fazer. Quando comparados com as funcionalidades, os objetivos tendem a ser menos propensos à alteração a longo prazo.

Os cenários de caso de uso são úteis no momento em que os objetivos estão mais concretos, tendendo assim serem mais acessíveis e fáceis de visualizar. As funcionalidades devem ser específicas o suficiente para que possam ser descritas adequadamente em uma ou duas frases. O ambiente em que o agente irá atuar é definido em termos de ações (como os agentes irão agir sobre o ambiente), e percepções (informações recebidas a partir do ambiente).

1. Objetivos do Sistema

O SMA desenvolvido deverá ser capaz de:

1. Permitir cadastramento de usuários.
2. Confirmar cadastramentos realizados.
3. Notificar novos usuários por e-mail.
4. Atribuir perfis à novos usuários cadastrados.
5. Liberar o acesso à sala virtual.
6. Informar procedimentos para agendamento de um novo experimento.
7. Realizar reservas.
8. Confirmar reservas efetuadas.
9. Confirmar agendamentos efetuados.
10. Gerenciar ações dos usuários.
11. Relembrar aos usuários sobre experimentos agendados.
12. Sugerir desafios baseado no histórico de cada usuário.
13. Permitir acesso à sala de desafios.
14. Analisar o código fonte submetido pelos usuários.
15. Movimentar robô.

2. Funcionalidades do Sistema

As funcionalidades dos agentes, associadas aos objetivos descritos anteriormente encontram-se disponíveis na Tabela 4.1.

3. Descritor de Cenários Parciais

Cenário 1: Cadastramento de usuários.

Nome: Permitir que novos usuários se cadastrem.

Tabela 4.1: Funcionalidades dos Agentes Associadas aos Objetivos

Agentes	Funcionalidades	Objetivos Associados
Administrador.	Responsável por cadastrar usuários.	1, 2, 3, 4, 5, 10.
Agendamento.	Responsável por agendar experimentos.	7, 8, 11
Assistente.	Responsável por gerenciar as ações dos usuários.	6, 9, 12, 13.
Robô	Responsável por executar movimentos no robô.	14, 15.

Descrição: Após o usuário acessar o endereço do LERP através de um navegador Web, caso tenha interesse em ter acesso à sala virtual e experimentos robóticos, ele poderá efetuar seu cadastro.

Contexto: Assumir que o usuário possua um e-mail pessoal, conhecimentos básicos em informática e saiba navegar na Web.

Cenário 2: Agendamento de experimentos.

Nome: Permitir que novos usuários agendem experimentos.

Descrição: Após o usuário acessar o sistema, ele poderá reservar uma data e horário para que seja possível programar o robô remotamente e ter acesso à sala de desafios.

Contexto: Assumir que o usuário já possui cadastro no sistema, e tenha lido as instruções iniciais disponíveis na sala virtual.

Cenário 3: Gerenciamento de ações dos usuários.

Nome: Armazenar informações dos usuários.

Descrição: Após o usuário ter acesso liberado ao LERP, serão sugeridos desafios que o usuário pode realizar, levando em consideração o histórico dele na sala virtual.

Contexto: Assumir que o usuário já esteja cadastrado no sistema.

Cenário 4: Execução dos movimentos no robô.

Nome: Programar robôs remotamente.

Descrição: Deverá ser criado um arquivo de execução com o código enviado pelo usuário, e os comandos serão enviados via Bluetooth para o robô.

Contexto: Assumir que o usuário agendou um experimento, acessou a sala virtual no dia e horário agendado e submeteu o código para o servidor sem erros sintáticos.

4.2.6.2 Fase 2. Projeto da Arquitetura

Uma decisão importante a ser feita durante a fase do projeto arquitetônico, é definir quais tipos de agentes deverão existir. A Prometheus sugere agrupar os tipos dos agentes conforme as funcionalidades. Tão logo os agentes sejam definidos, deve-se identificar a quais percepções os agentes reagem, e quais ações eles irão executar no ambiente externo.

Segundo Padgham e Winikoff (2005a), a fase do projeto da arquitetura concentra-se em 3 (três) etapas:

- Decidir quais agentes serão desenvolvidos, e a descrição de cada um deles.
- Desenvolver a estrutura geral de funcionamento do sistema.
- Descrever o comportamento dinâmico do sistema usando diagramas e protocolos de interação.

1. Descritor de Funcionalidades dos Agentes

Nome: Administrador.

Descrição: Responsável por liberar acesso ao LERP e armazenar informações dos usuários.

Percepções: Verifica a existência de novos cadastros no BD.

Ações: Concede acesso ao LERP, atribui perfil de estudante e gerencia ações dos usuários.

Interações: Informa ao agente Assistente que o usuário possui acesso ao LERP, e solicita que o mesmo atribua desafios aos usuários.

Nome: Agendamento.

Descrição: Responsável por gerenciar reservas e agendamentos.

Percepções: Verifica as reservas realizadas para um dia específico no BD, que fica armazenada na tabela mdl_lerp_reserva.

Ações: Fica a cargo deste agente, verificar se é possível alterar reservas para a situação de agendado no BD, e confirmar ou remover reservas efetuadas.

Interações: Informa ao agente Assistente que o usuário efetuou ou cancelou alguma reserva.

Nome: Assistente.

Descrição: Auxilia usuários no processo de programação do robô.

Percepções: Sugere desafios baseados no histórico de cada usuário.

Ações: Informa ao usuário que o mesmo já possui acesso ao LERP e envia instruções de como realizar agendamentos. Responsável também por controlar o acesso na sala de desafios.

Interações: Envia um e-mail ao usuário com informações sobre confirmação ou cancelamento de reservas.

Nome: Robô.

Descrição: Responsável por executar os movimentos no robô conforme programado pelo usuário.

Percepções: Realiza a análise sintática do código submetido pelo usuário.

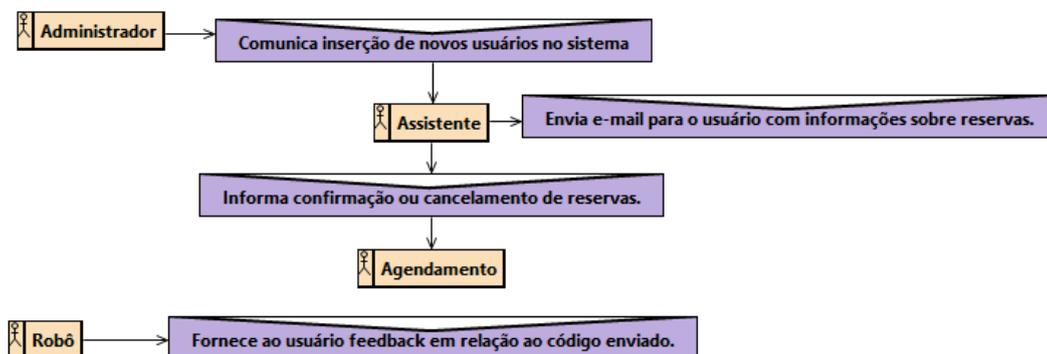
Ações: Fornece um feedback para o usuário em relação ao código que foi submetido e cria um arquivo de execução.

Interações: Envia os comandos via Bluetooth para o robô.

2. Diagrama de Relacionamentos de Agentes

O diagrama de relacionamentos mostra a comunicação entre os agentes, e tem como objetivo representar uma possível sequência de trocas de mensagens entre eles. A Figura 4.12 apresenta o diagrama de relacionamentos de agentes do LERP.

Figura 4.12: Diagrama de Relacionamento dos Agentes



Por meio desta figura, temos os 4 (quatro) agentes que serão desenvolvidos, e as trocas de mensagens entre eles. Os agentes Assistente e Robô se comunicam diretamente com usuários do sistema. A próxima subseção apresenta a última fase da metodologia Prometheus.

4.2.6.3 Fase 3. Projeto Detalhado

A fase do projeto detalhado, concentra-se no desenvolvimento da estrutura interna de cada agente e como ele irá cumprir suas tarefas dentro do sistema (PADGHAM; WINIKOFF, 2005a). O processo de projeto detalhado é dividido em 2 (duas) etapas:

- Refinamento dos agentes em termos de capacidade, eventos, planos e dados.
- Desenvolvimento das especificações do processo.

Essa fase do projeto, independe de qualquer plataforma de implementação particular ou arquitetura. As especificações do processo fornecem uma visão detalhada de uma parte individual de um agente em um processo específico. Como não foi necessário realizar refinamento nas capacidades e planos dos agentes, a Figura 4.13 apresenta o diagrama de revisão SMA.

A Figura 4.13 apresenta os agentes desenvolvidos, as trocas de mensagens realizadas, percepções de cada um e acesso ao banco de dados. A próxima seção apresenta a interface gráfica e funcionamento do LERP.

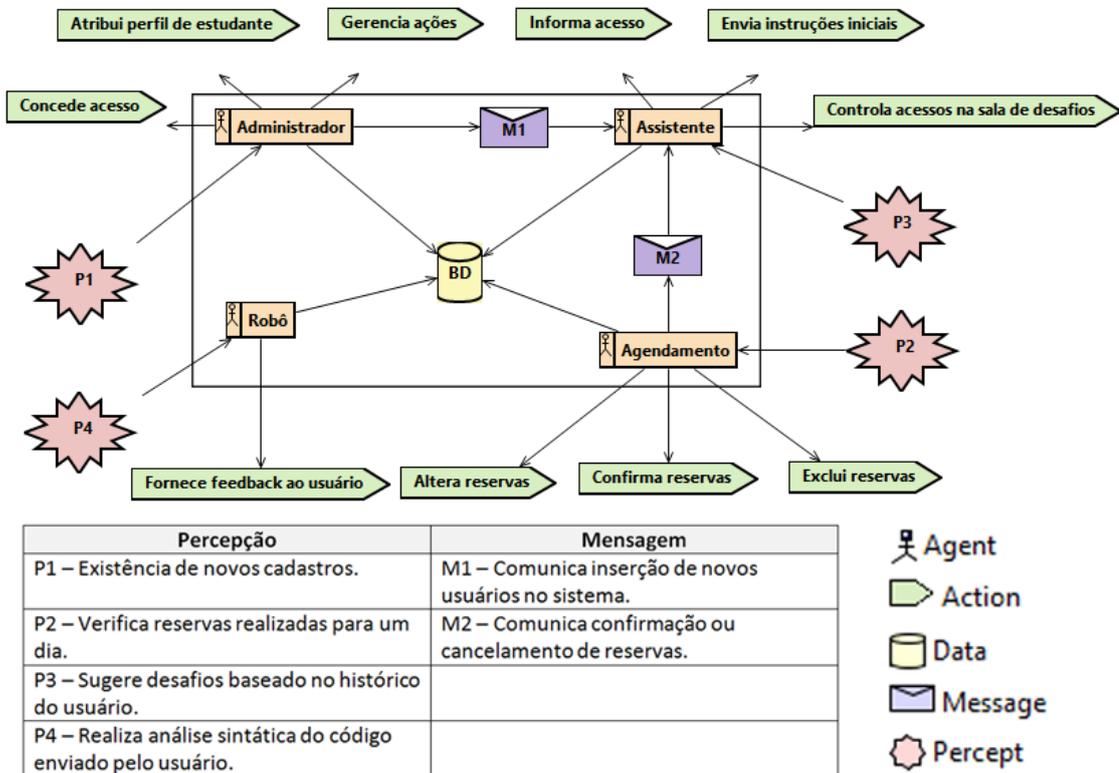
4.3 Interface Gráfica e Funcionamento do LERP

Na Seção 4.1 foi apresentada uma visão do SMA desenvolvido para apoiar nas atividades do LERP. A Seção 4.2 apresentou a documentação dos agentes desenvolvidos. Esta seção destina-se a apresentar a interface gráfica e funcionamento do sistema que está integrado com o AVA Moodle. Neste sentido, a Subseção 4.3.1 apresenta o processo de cadastramento de usuários, a Subseção 4.3.2 apresenta a política de agendamento, a Subseção 4.3.3 apresenta a sala de desafios, e o processo de manipulação remota é apresentado na Subseção 4.3.4.

4.3.1 Cadastramento de Usuário

Para acesso à sala virtual do LERP e experimentos robóticos, é necessário que o usuário efetue seu cadastro. As informações dos usuários são armazenadas no banco de dados do LERP, que utiliza a estrutura das tabelas do Moodle,

Figura 4.13: Diagrama de Revisão do Sistema



além de algumas tabelas adicionais que são mostradas no Apêndice B desta dissertação.

A Figura 4.14 apresenta a página inicial do LERP que está disponível na Internet e pode ser acessada por qualquer usuário pelo endereço <<http://gaire.icomp.ufam.edu.br/ava>>.

Além disso, na página inicial do LERP são apresentadas algumas informações adicionais aos usuários tais como: limitações, requisitos, informações de acesso e recuperação de senha, conforme podemos visualizar na Figura 4.15.

Ainda na Figura 4.14, podemos verificar o link de cadastro de usuários no sistema. Ao clicar neste link, o sistema disponibiliza para o usuário o formulário de cadastramento no LERP, detalhado na Figura 4.16. Os campos marcados com * são campos obrigatórios, portanto não podem ser deixados em branco.

A liberação para acesso à sala virtual é realizada pelo agente Administrador, tão logo o usuário tenha preenchido o formulário acima e os dados sejam

Figura 4.14: Tela Inicial do LERP

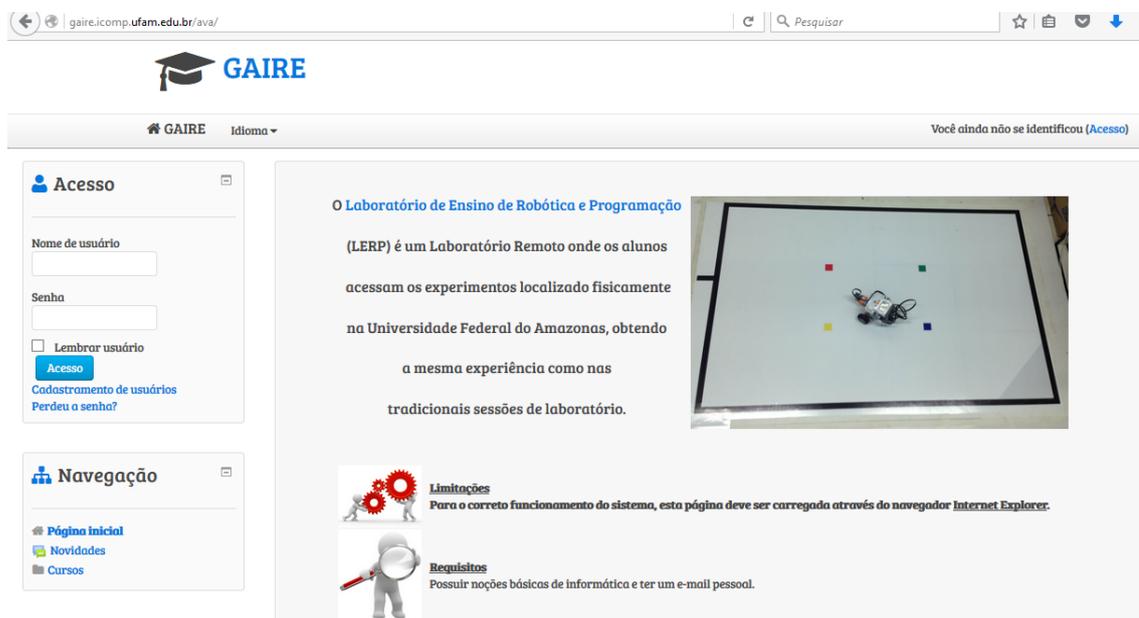
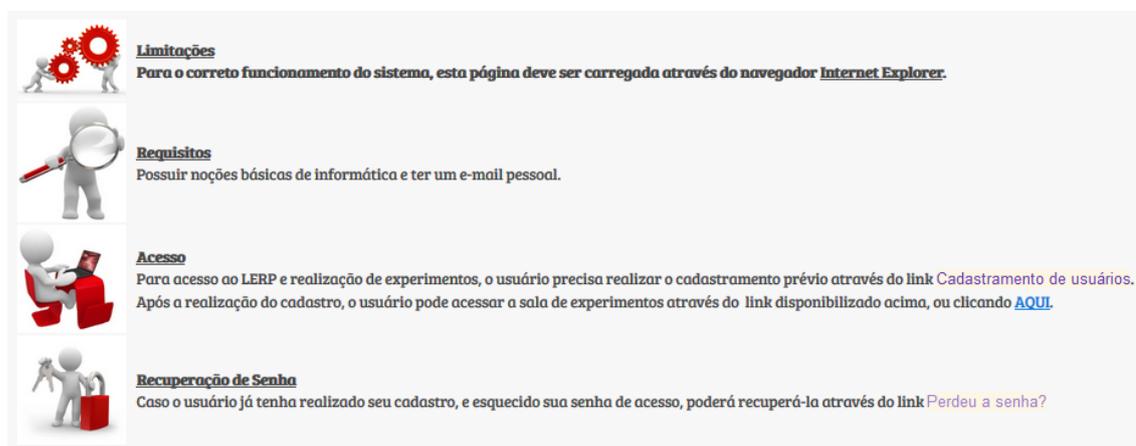


Figura 4.15: Instruções Iniciais



inseridos no BD. É responsabilidade do agente Administrador o envio de um e-mail para o usuário informando sobre a realização do cadastro (conforme Figura 4.17).

Após realizado o cadastro, o usuário pode solicitar também a recuperação de senha informando tanto o nome de usuário ou o endereço de e-mail, como

Figura 4.16: Formulário de Cadastro de Usuários

The screenshot shows a web browser window with the URL `gaire.icomp.ufam.edu.br/ava/login/signup.php`. The page title is "Página inicial > Acesso > Cadastro de novo usuário". The form is titled "Escolha seu usuário e senha" and is divided into two sections: "Escolha seu usuário e senha" and "Mais detalhes".

Escolha seu usuário e senha:

- Nome de usuário* (text input)
- Senha* (password input with a "Mostrar" checkbox)

Mais detalhes:

- Endereço de email* (text input)
- Email (outra vez)* (text input)
- Nome* (text input)
- Sobrenome* (text input)
- Cidade/Município (text input)
- País (dropdown menu with "Selecione um país")

At the bottom of the form, there are two buttons: "Cadastrar este novo usuário" (green) and "Cancelar" (white).

Figura 4.17: E-mail de Confirmação de Cadastro



podemos visualizar na Figura 4.18.

Após o usuário ter acesso ao LERP, ele poderá acessar a sala virtual, visualizar os materiais disponíveis e agendar os experimentos. O processo de agendamento e realização de reservas encontram-se detalhados na próxima subseção.

4.3.2 Processo de Agendamento

Foi necessário adotar uma política de agendamentos no LERP tendo em vista o quantitativo de kits de robótica disponíveis para uso. Para o caso

Figura 4.18: Recuperação de Senha

[Página inicial](#) > [Acesso](#) > [Senha esquecida](#)

Para redefinir sua senha, preencha seu usuário ou seu email abaixo. Se sua conta for encontrada no banco de dados, um email será enviado para seu endereço de email, com as instruções sobre como restabelecer seu acesso.

Buscar por usuário

Nome de usuário

Buscar pelo endereço de email

Endereço de email

específico desta dissertação, um usuário pode ter acesso aos experimentos robóticos conforme horário agendado para um dia específico, desde que sejam observados os horários disponíveis por dia (detalhado na Tabela 4.2).

Tabela 4.2: Horários Disponíveis por Dia

Horários	
1	09:00h às 09:40h
2	10:00h às 10:40h
3	11:00h às 11:40h
4	14:00h às 14:40h
5	15:00h às 15:40h
6	16:00h às 16:40h
7	17:00h às 17:40h

O agendamento pode ser realizado através do link "Agendar Experimento" disponível na sala virtual do LERP, e apresentado na Figura 4.19.

Figura 4.19: Link de Agendar Experimento



Ao clicar neste link, é disponibilizado para o usuário um formulário (Fi-

gura 4.20) contendo os campos de data e hora para realização da reserva . Não é possível realizar agendamento para datas e horários passados, assim como para horários já reservados por outros usuários (o agente Agendamento é o responsável por resolver esses conflitos).

Figura 4.20: Página de Agendamento de Experimentos

[Página inicial](#) > [Meus cursos](#) > [Miscelânea](#) > [LERP](#) > [Geral](#) > [Agende seu Experimento](#)



Página de Agendamentos!

Solicitar Agendamento

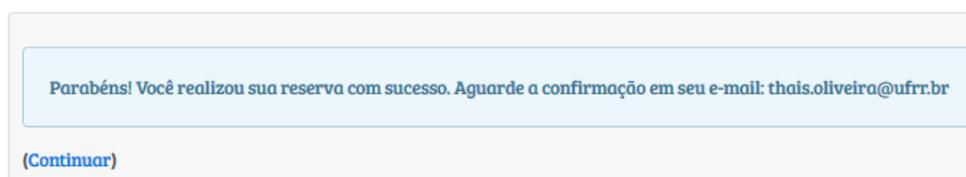
Data:

Horário:

Caso o usuário consiga efetuar a reserva, aparecerá uma mensagem de confirmação (Figura 4.21). Internamente cada reserva fica armazenada na tabela mdl_lerp_reserva do BD. Vale ressaltar que solicitar a reserva não garante a confirmação do agendamento.

Figura 4.21: Confirmação de Reserva

[Página inicial](#) > [Meus cursos](#) > [Miscelânea](#) > [LERP](#) > [Geral](#) > [Agende seu Experimento](#)



Parabéns! Você realizou sua reserva com sucesso. Aguarde a confirmação em seu e-mail: thais.oliveira@ufr.br

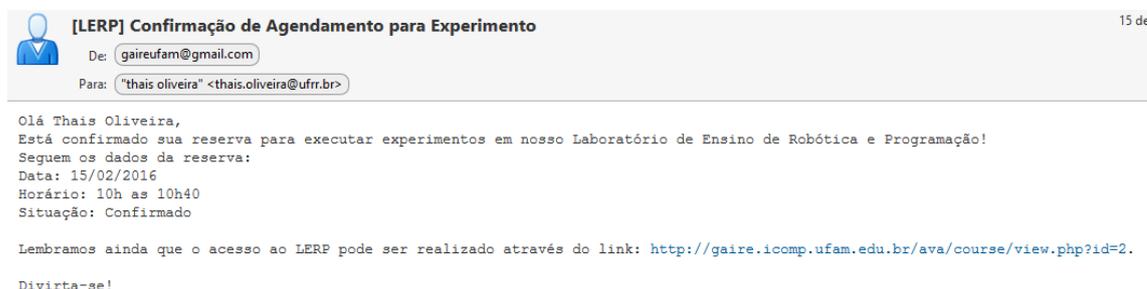
[\(Continuar\)](#)

É de responsabilidade do agente Agendamento tentar realizar todas as reservas solicitadas pelos usuários. O agente Agendamento atua no ambiente fazendo consultas às tabelas mdl_lerp_reserva e mdl_lerp_agendamento, a fim de verificar se existem novas solicitações de reservas, e se há vagas disponíveis para agendamento.

O agente Assistente compara os dados dessas duas tabelas e realiza o agendamento caso seja possível. Após a confirmação do agendamento, um e-mail é enviado para o usuário conforme podemos visualizar na Figura 4.22.

O usuário pode acompanhar as reservas solicitadas tanto pela página de agendamento de experimentos (apresentada na Figura 4.20) ou pelo link

Figura 4.22: E-mail de Confirmação de Agendamento



"Agendamentos Confirmados" (Figura 4.23) disponível na sala virtual do LERP. Neste último caso, o usuário precisa informar o intervalo de datas que deseja consultar.

Figura 4.23: Link para Acessar Agendamentos Confirmados



4.3.3 Sala de Desafios

O laboratório desenvolvido nesta dissertação possibilita aos usuários a realização de experimentos com o kit robótico *Lego Mindstorms*. Neste sentido, é permitido que os alunos programem os robôs por meio da linguagem de programação NXT-Python, e não apenas manipulem-o através de setas de controle. Os trabalhos apresentados no Capítulo 3 desta dissertação colaboraram e motivaram seu desenvolvimento.

O LERP possui cinco desafios a serem realizados pelos usuários, e podem ser acessados após a realização de um agendamento através dos links disponíveis na sala virtual (Figura 4.24).

O tempo de permanência do usuário em cada desafio está de acordo com o que foi estipulado na Tabela 4.2 (máximo de quarenta minutos). É importante destacar que ao carregar um desafio e concluí-lo com êxito, é atribuída uma pontuação ao usuário, conforme apresentado na Tabela 4.3.

A atribuição do perfil do usuário está relacionada ao seu desempenho na sala dos desafios ao programar o robô Lego NXT. Vale ressaltar que todas as ações dos usuários no LERP são armazenadas em tabelas específicas do BD. É função do Supervisor acompanhar os usuários durante a realização dos desafios, e após a conclusão do mesmo, "aprovar" ou "reprovar".

Figura 4.24: Links para Acesso aos Desafios

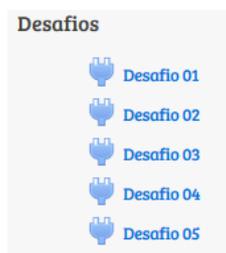


Tabela 4.3: Descrição dos Desafios

Desafios		
ID	Descrição	Pontuação
1	Neste desafio você deve acender a luz verde do sensor de luz, mantê-la acesa por 5 segundos e desligá-la.	10
2	Neste segundo desafio você deve acender a luz vermelha do sensor de luz, mantê-la acesa por 5 segundos e desligá-la. Após isso você deve fazer o mesmo para a luz azul.	15
3	Neste terceiro desafio você deve acender a luz azul do robô, fazer ele andar em linha reta por 10 (dez) segundos, e apagar a luz azul após esse tempo.	25
4	Neste quarto desafio você deve acender a luz verde para indicar que o robô irá partir. Faça ele caminhar por 10 segundos em linha reta e gire o robô 180°. Desligue a luz verde e mantenha apenas a luz vermelha acesa. Programe o robô para caminhar por mais 10 segundos de maneira que ele volte ao ponto de partida. Ao final do trajeto, a luz vermelha deverá estar apagada.	30
5	Neste último desafio você deve fazer o robô percorrer o retângulo utilizado na nossa sala de experimentos. Quando ele percorrer o comprimento deverá acender a luz azul. Quando percorrer a largura deverá acender a luz verde.	40

O agente Assistente é o responsável por atribuir o perfil aos usuários automaticamente, baseando-se na pontuação dele de acordo com os desafios que já foram realizados, e encontra-se detalhada abaixo na Tabela 4.4.

Baseado na consulta de informações da Tabela 4.4, o agente Assistente sugere desafios conforme histórico do usuário, conforme pode ser visto na Figura 4.5. O e-mail contém informações como a pontuação do usuário e nível em que se encontra.

A subseção seguinte explana como ocorre o processo de manipulação remota, quando o usuário envia um código-fonte para o servidor.

Tabela 4.4: Níveis de Perfil do Usuário

Perfil do Usuário	
Níveis	Pontuação
Iniciante	10
Básico	15
Intermediário	25
Avançado	30
Expert	40

Figura 4.25: Sugestão de Desafio

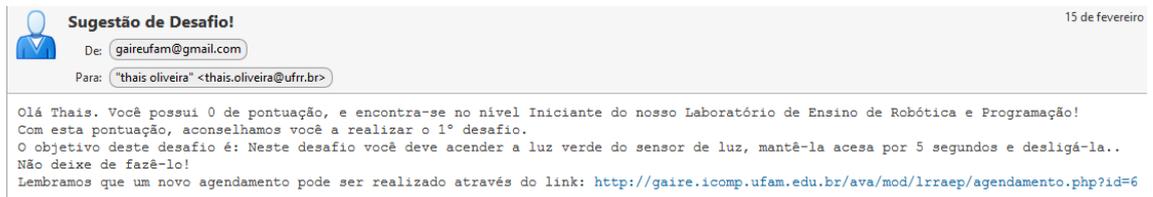
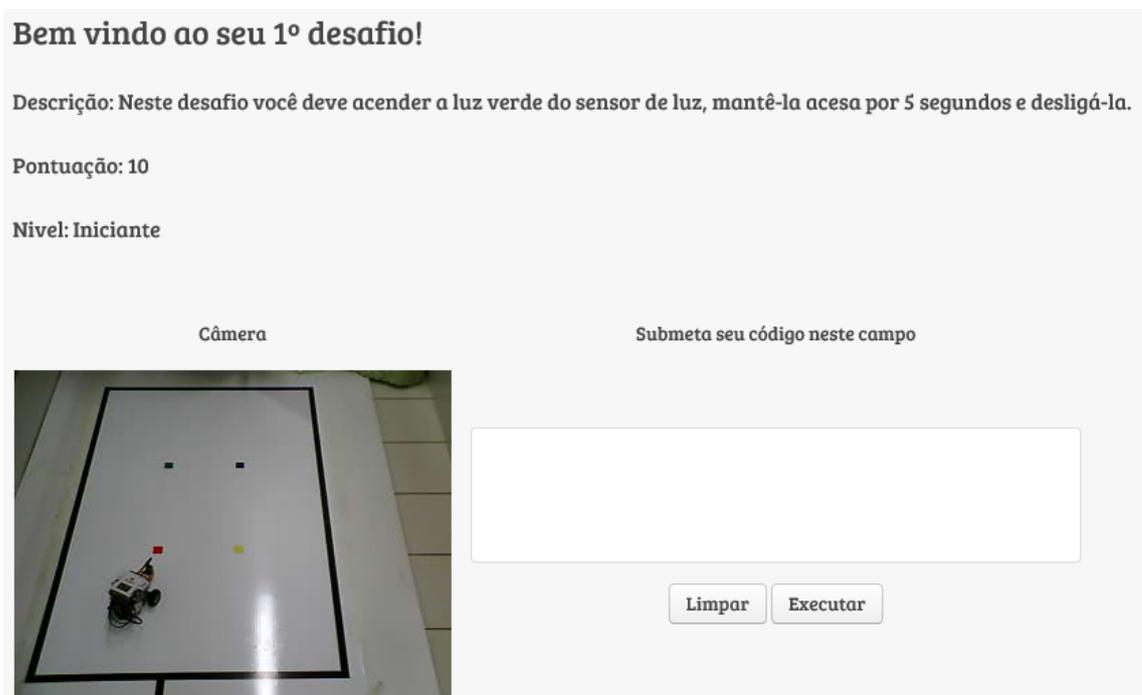


Figura 4.26: A Sala de Desafios

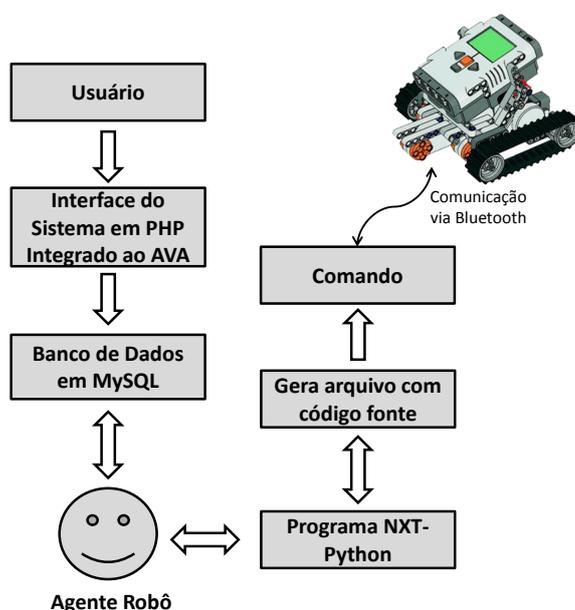


4.3.4 Manipulação Remota

No LERP um programa é escrito na linguagem de programação NXT-Python (detalhado na Subseção 4.4.6). O servidor e o robô estão ligados por meio da tecnologia Bluetooth. Após o programa ser enviado pelo usuário para o servidor, o agente Robô busca o código na base de dados, gera o arquivo executável e faz a análise sintática do código, verificando se há erros ou não. Caso não haja nenhum erro sintático, o programa em execução enviado pelo LERP controla o robô remotamente.

A Figura 4.27 apresenta a estrutura de camadas que foi desenvolvida para permitir a integração das tecnologias.

Figura 4.27: Processo de Manipulação Remota



Por padrão o Moodle utiliza a estrutura de camadas para armazenar todas as ocorrências no BD. O usuário entra com o programa escrito na linguagem NXT-Python a partir da interface gráfica, que por meio de scripts desenvolvidos na linguagem PHP armazenam esses dados no BD. Os códigos enviados pelos usuários ficam armazenados na tabela mdl_lerp_execucao.

O SMA que roda no servidor, foi desenvolvido em JADE e necessita da linguagem Java para ser executado. Os agentes desenvolvidos são os responsáveis por auxiliar no cadastramento de usuários, agendamento de experimentos e manipulação do robô no LERP.

As três linhas abaixo devem ser incluídas no início de um programa escrito em NXT-Python, a ser enviado pelo LERP, para que o processo de

manipulação remota funcione corretamente. A variável ID é utilizada para controlar o robô. O objeto é iniciado a partir da classe definida BlueSock, como segue:

```
1.     from nxt.bluesock import BlueSock
2.     from funcoes import *
3.     ID = '00:16:53:11:25:73'
```

A primeira linha importa a classe "BlueSock" do módulo "NXT BlueSock". Esta classe deve ser utilizada obrigatoriamente para que os comandos sejam enviados corretamente através da tecnologia Bluetooth, que emprega um endereço MAC de 48 bits. A segunda linha importa as funções que irão auxiliar os usuários na realização dos desafios, e são definidas na Tabela 4.5. Para utilizá-las, basta apenas que o usuário invoque-as em seu programa informando o nome da mesma seguido dos argumentos.

A terceira linha gera o objeto "ID", em que o parâmetro '00:16:53:11:25:73' é um exemplo de endereço MAC do robô. Como a classe "nxt.bluesock" oculta como a comunicação Bluetooth é realizada e atualização das variáveis, os usuários podem desenvolver os seus programas sem conhecer os detalhes internos.

A Tabela 4.5 apresenta as funções pré-definidas que os usuários podem utilizar em seus programas para a realização dos desafios (detalhes da implementação dessas funções constam no Apêndice C desta dissertação).

O robô pode utilizar o sensor de luz para acender uma cor (vermelho, azul ou verde). Pode ainda ser movido para frente, direita ou esquerda de acordo com as funções disponíveis para os usuários. A velocidade em que o robô irá andar, assim como o ângulo que ele deverá seguir para dobrar, são controlados também pelo programa.

Abaixo temos um trecho de código exemplificando os itens explanados anteriormente. Esse código mantém acesa a luz azul do sensor de luz por 5 segundos e após esse tempo apaga. Esse mesmo exemplo está disponível na sala virtual do LERP para que os usuários sigam como base para construção dos demais programas.

```
from nxt.bluesock import BlueSock
from funcoes import *

ID = '00:16:53:11:25:73'
sock = BlueSock(ID)

brick = sock.connect()
```

Tabela 4.5: Funções e Significados

Funções	Significado	Descrição
luz(brick, string_cor)	Acende sensor de luz.	O sensor de luz verifica a cor passada como parâmetro e acende a mesma. As cores podem ser: "vermelho", "azul", ou "verde".
apaga_luz (brick)	Desliga sensor de luz.	É possível desligar o sensor de luz utilizando a função apaga_luz e passando como argumento apenas o nome do brick que está sendo utilizado.
tempo_de_espera(tempo)	Tempo de espera.	A função de tempo de espera, tem como parâmetro apenas o argumento de tempo que deve ser informado em segundos. Essa função faz com que o robô espere por um tempo em segundos, até executar uma nova ação.
correr_por_tempo(brick, velocidade, tempo)	Percorre distância.	A função correr_por_tempo percorre uma distância com base no tempo velocidade determinados. Também é necessário passar o nome do brick como argumento.
rotaciona(brick, velocidade, numero_rotacao)	Rotaciona o robô.	A função rotaciona recebe três argumentos como parâmetro: o nome do brick, velocidade e número de rotação. Através desta função é possível rotacionar o robô.

```

luz ( brick , " azul " )
tempo_de_espera ( 5 )
apaga_luz ( brick )

sock . close ( )

```

A próxima seção apresenta as tecnologias utilizadas para o desenvolvimento do LERP.

4.4 Tecnologias Utilizadas

Nesta seção são apresentadas as tecnologias que foram utilizadas para o desenvolvimento desta dissertação. Neste contexto, será realizada uma breve explanação sobre Moodle (Subseção 4.4.1), PHP (Subseção 4.4.2), MySQL (Subseção 4.4.3), JADE (Subseção 4.4.4), *Legó Mindstorms* (Subseção 4.4.5) e NXT-Python (Subseção 4.4.6).

4.4.1 Moodle

Moodle¹ é uma abreviação para *Modular Object Oriented Dynamic Learning Environment*. É um dos Ambientes Virtuais de Aprendizagem (AVA) mais populares e possui um impacto significativo no processo de ensino e aprendizagem (RODRIGUES et al., 2013). Ele prevê muitas ferramentas colaborativas de aprendizagem, que permite aos professores a interação com seus alunos, acréscimo de atividades e gerenciamento do curso.

De acordo com Walker, Prytherch e Turner (2013), o Moodle combina uma visão construtivista de ensino e aprendizagem com o poder da Web para integrar sua comunidade de usuários. Para os tutores, é um lugar onde eles podem ter experiências de aprendizagem envolventes. Para os alunos, é um lugar onde eles podem acessar seus materiais a qualquer hora do dia e conversar com seus tutores e colegas. Com base na abordagem construtivista, os alunos são responsáveis por construir sua própria aprendizagem, por meio da colaboração com outros estudantes (WALKER; PRYTHERCH; TURNER, 2013).

Por tratar-se de um software de código aberto organizado em módulos, sua interface permite aumentar os recursos que vêm com o Moodle padrão. Desta maneira, o laboratório desenvolvido nesta dissertação está integrado com o Moodle de maneira a aproveitar a interface gráfica e as funcionalidades que ele fornece. Sendo assim, além de disponibilizar uma área onde é possível programar os robôs remotamente, podemos acrescentar outros materiais com recursos do Moodle padrão, como chat, fórum e inserção de documentos.

O Moodle é escrito na linguagem de script PHP (detalhado na próxima subseção) e armazena seus dados em um banco de dados (recomenda-se o uso do MySQL). Embora se reconheça que o Moodle é extremamente poderoso e oferece uma gama de ferramentas e potencialidades, é possível a criação de novas funcionalidades.

4.4.2 PHP

PHP² é uma abreviação para *Hypertext Preprocessor*. O poder e facilidade de uso do PHP fez com que ela se tornasse a linguagem de script do lado do servidor mais popular do mundo (HILLS; KLINT; VINJU, 2013). Desenvolvido em 1994 por Rasmus Lerdof, inicialmente foi utilizada para o desenvolvimento e manutenção de sites dinâmicos (WRENCH; IRWIN, 2014).

De acordo com Wrench e Irwin (2014), com o PHP é possível transformar páginas Web estáticas com conteúdo pré-definido, em páginas dinâmicas

¹<http://moodle.org>

²<http://php.net>

baseado em um conjunto de parâmetros. Embora tenha sido desenvolvido como uma linguagem puramente interpretada, vários compiladores já foram desenvolvidos para o PHP, permitindo-lhe funcionar como uma plataforma para aplicações standalone (WRENCH; IRWIN, 2014).

O Moodle é desenvolvido na linguagem de script PHP, que está integrado com seu servidor Web. O servidor Web detecta as páginas PHP (pela sua extensão) e os envia para execução. Para que o Moodle trabalhe de forma eficaz, é necessário que o PHP esteja instalado e configurado corretamente.

Para o desenvolvimento das telas dos desafios e agendamento de experimentos, foram criadas páginas em PHP para o correto funcionamento do sistema, fazendo a comunicação com o banco de dados MySQL que será explanado na próxima subseção.

4.4.3 MySQL

O MySQL³ é um sistema gerenciador de banco de dados relacional (SGBDR), desenvolvido e fornecido pela Oracle⁴, que pode ser utilizado sob a licença GNU⁵ (*General Public License*) ou sob a licença comercial (VICKNAIR; WILKINS; CHEN, 2012). O MySQL utiliza a SQL (*Structure Query Language*) para inserir, acessar e gerenciar o conteúdo armazenado no banco de dados.

MySQL é altamente escalável e pode ser distribuído em vários servidores. De acordo com Miyamoto, Higuchi e Tsuji (2014), um dos principais recursos do MySQL é que os usuários podem escolher diferentes mecanismos de armazenamento dependendo do seu objetivo. Por exemplo, um usuário pode escolher um mecanismo de armazenamento em memória, caso deseje acessar o banco de dados rapidamente.

Durante a instalação do Moodle, recomenda-se a utilização do MySQL para criação e gerenciamento das tabelas. A Figura 4.28 apresenta algumas das tabelas do banco de dados do Moodle, que é gerenciada pelo phpMyAdmin.

No Apêndice B desta dissertação são apresentadas detalhadamente as tabelas adicionais que foram criadas no Banco de Dados do Moodle para o correto funcionamento do sistema. Neste sentido, além das tabelas padrão do Moodle, foi necessário acrescentar oito tabelas no BD para que fosse possível armazenar informações de agendamentos, reservas, desafios, execução e feedback aos usuários. A subseção seguinte irá explorar a ferramenta JADE que foi utilizada para desenvolver o SMA integrado ao LERP.

³<http://www.mysql.com>

⁴<http://www.oracle.com>

⁵<http://www.gnu.org/licenses/gpl.html>

Figura 4.28: Tabelas do Moodle

The screenshot shows the phpMyAdmin interface for a Moodle database. The main area displays a table of database tables with the following columns: Tabela, Ação, Linhas, Tipo, Colaço, Tamanho, and Sobrecarga. The table lists various Moodle tables such as mdl_assign, mdl_assignfeedback_comments, and mdl_assignment_submissions.

Tabela	Ação	Linhas	Tipo	Colaço	Tamanho	Sobrecarga
mdl_assign	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assignfeedback_comments	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assignfeedback_editpdf_annot	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assignfeedback_editpdf_cmnt	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assignfeedback_editpdf_quick	Visualizar	~0	InnoDB	utf8_general_ci	32 KB	-
mdl_assignfeedback_file	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assignment	Visualizar	~0	InnoDB	utf8_general_ci	32 KB	-
mdl_assignment_submissions	Visualizar	~0	InnoDB	utf8_general_ci	80 KB	-
mdl_assignment_upgrade	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assignmentsubmission_file	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assignmentsubmission_onlinetext	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_assign_grades	Visualizar	~0	InnoDB	utf8_general_ci	80 KB	-
mdl_assign_plugin_config	Visualizar	~0	InnoDB	utf8_general_ci	80 KB	-
mdl_assign_submission	Visualizar	~0	InnoDB	utf8_general_ci	36 KB	-
mdl_assign_user_flags	Visualizar	~0	InnoDB	utf8_general_ci	64 KB	-
mdl_assign_user_mapping	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_backup_courses	Visualizar	~0	InnoDB	utf8_general_ci	64 KB	-
mdl_backup_logs	Visualizar	~0	InnoDB	utf8_general_ci	48 KB	-
mdl_badge	Visualizar	~0	InnoDB	utf8_general_ci	80 KB	-

4.4.4 JADE

A tecnologia JADE⁶ (*Java Agent Development*) é um framework eficiente no suporte, gerenciamento e monitoração de SMA, principalmente na parte de comunicação entre cada instância de agente (MARCHI et al., 2010). Toda a documentação está disponível no endereço <http://jade.tilab.com>. A comunicação dos agentes segue os protocolos FIPA (*The Foundation for Intelligent Physical Agents*), uma organização aceita pela *IEEE Computer Society* como reguladora de padrões para interação de agentes e sistemas de agentes.

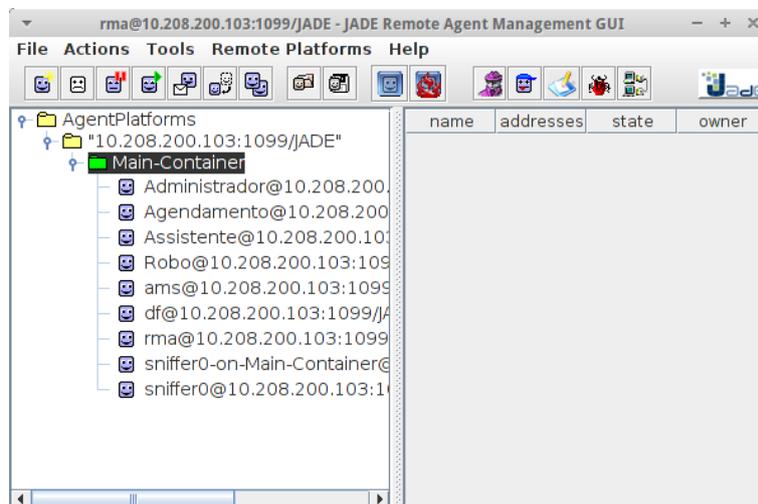
Cicirelli e Nigro (2014) definem o JADE como um middleware *open source* baseado em Java para o desenvolvimento de SMA distribuídos. Os agentes em execução são chamados de *containers*, e ficam organizados nas plataformas. Uma plataforma quando inicializada, cria um *main-container* que irá hospedar alguns agentes para prestar os serviços fundamentais, tais como inicialização do serviço e compartilhamento de informações (CICIRELLI; NIGRO, 2014).

Os agentes também podem ser criados por meio da interface gráfica da plataforma JADE, por meio do agente RMA (*Remote Management Agent*), como podemos visualizar na Figura 4.29.

Por meio dessa plataforma, é possível visualizar e gerenciar os agentes criados no sistema. O sistema possibilita ainda a execução dos agentes *Dummy*, *Sniffer* e *Introspector*. Os agentes recebem as mensagens por meio de uma

⁶<http://jade.tilab.com>

Figura 4.29: Interface Gráfica da Plataforma JADE



caixa de correio local, e as processa individualmente através de uma estrutura de comportamento. Esses comportamentos podem ser adicionados a um agente dinamicamente.

O modelo de comunicação do JADE é baseado em mensagens assíncronas expressas utilizando o padrão FIPA ACL (*Agent Communication Language*). A Figura 4.30 apresenta a troca de mensagens entre agentes, utilizando o agente Sniffer para realizar esse acompanhamento.

Figura 4.30: Troca de Mensagens entre Agentes



Além dos agentes criados nesta dissertação, por padrão o JADE possui

outros agentes que podem ser observados na Figura 4.30, sendo eles:

- *Agent Management System* (AMS): é o agente responsável por controlar e supervisionar o acesso e uso da plataforma. Ele mantém um diretório com informações dos identificadores e estados dos agentes.
- *Directory Facilitator* (DF): é o agente que fornece o serviço de página amarela na plataforma.
- *Remote Management Agent* (RMA): interface gráfica para gerenciamento e controle da plataforma.
- *Sniffer Agent*: agente responsável por interceptar as mensagens ACL enquanto elas estão sendo enviadas, e exibe esta troca graficamente.

A próxima subseção apresenta o kit de robótica *Lego Mindstorms*, que foi integrado ao LERP e utilizado na realização dos experimentos.

4.4.5 Lego Mindstorms

O kit *Lego Mindstorms*⁷ é voltado para ensinar programação de robôs, e foi lançado comercialmente em 1998 (LIANG et al., 2013). O projeto foi originalmente inspirado por Seymour Papert (1994) e desenvolvido em uma parceria entre o Media Lab do *Massachusetts Institute of Technology* (MIT) e o Lego Group.

Trata-se de uma plataforma educacional que representa a evolução de seis décadas de técnicas de construção modular do Lego e métodos de ensino baseado em computador (proposta por Papert (1994)). Utilizando essa plataforma robótica, diferentes tipos de cenários podem ser facilmente montados.

Liang et al. (2013) definem o kit como um conjunto composto por uma variedade de peças físicas, que quando combinadas podem produzir uma variedade de robôs. Basicamente as peças (Figura 4.31) que compõem o kit são: quatro tipos de sensores, três motores e um controlador central, além de um conjunto de peças da linha tradicional (tijolos cheios, placas, rodas) e da linha Lego Technic (tijolos vazados, motores, eixos, engrenagens, polias e correntes).

Cada componente tem suas funcionalidades específicas: os motores são os responsáveis por movimentar a estrutura da montagem; os sensores são os responsáveis pela coleta das informações junto ao meio externo. O controlador central (Figura 4.32), também conhecido por módulo RCX (*Robotic*

⁷<http://www.lego.com/en-us/mindstorms>

Figura 4.31: Peças do Kit Lego Mindstorms NXT



Command Explorer), é o responsável pela parte inteligente, contendo o software que gerencia o sistema.

Figura 4.32: RCX do Kit Lego NXT.

Fonte: [<http://education.lego.com/en-us/>]



A utilização deste kit justifica-se por ser o disponível para uso no Laboratório de Sistemas Inteligentes. Como trabalho futuro, pretende-se fazer a substituição do *Lego Mindstorms* por Arduino⁸, devido ao baixo custo deste. A próxima subseção apresenta a linguagem de programação NXT-Python, que é utilizada no LERP e permite programar os robôs remotamente.

4.4.6 NXT-Python

O NXT-Python⁹ é um pacote utilizado para controlar um robô Lego NXT utilizando a linguagem de programação Python¹⁰. A comunicação com o robô pode ser realizada via USB ou Bluetooth. Como no LERP utilizamos

⁸<http://www.arduino.cc/>

⁹<http://pypi.python.org/pypi/nxt-python>

¹⁰<http://www.python.org/>

a tecnologia Bluetooth, para o correto funcionamento do sistema foi necessário realizar também a instalação do Pybluez¹¹ no servidor, um módulo em Python que permite acesso aos recursos Bluetooth.

O Python é uma linguagem de script orientada a objetos, disponível sem nenhum custo e suportada por várias plataformas, incluindo Windows, Linux e Mac. A adoção do Python integrado ao LERP justifica-se por ser uma linguagem utilizada globalmente, bem definida e fácil para usuários iniciantes em programação. O trabalho desenvolvido por Shikama (2014) sustenta essa afirmação.

De maneira geral, um programa em Python pode ser lido facilmente. Como trata-se de uma linguagem de script, o desempenho de um programa em Python torna-se lento. Além disso, o Python possui um problema de compatibilidade entre as versões 2 e 3. Para o correto funcionamento do NXT-Python, deve-se instalar a versão 2.6 do Python ou superior. O pacote utilizado nesta dissertação não funciona com a versão 3 da linguagem.

Ao contrário de outras linguagens de programação como C e C++, o Python utiliza a indentação (reco) como meio de identificar os blocos do programa. Caso um programa não esteja indentado corretamente, ele não funcionará como esperado. Acredita-se que este não é um problema sério para iniciantes, tendo em vista que as definições de funções e variáveis não são complexas.

O próximo capítulo apresenta a avaliação, testes realizados com o sistema e resultados obtidos. Detalha-se ainda as versões das tecnologias utilizadas e explanadas anteriormente, testes dos principais casos de uso do sistema, perfil dos usuários e resultados obtidos.

¹¹<https://github.com/karulis/pybluez>

Capítulo 5

Avaliação e Testes do Sistema

Este capítulo apresenta os testes realizados com o sistema e os resultados obtidos. Para o desenvolvimento do SMA proposto, foi utilizado um computador Desktop que funcionou como servidor Web, um kit robótico *Legó Mindstorms*, uma bateria recarregável para o RCX do Legó, um conector Bluetooth USB 2.0 e uma câmera IP.

O servidor Web possui o sistema operacional Ubuntu, com servidor Apache¹ versão 2.4.7, Banco de dados MySQL versão 5.5.46 e PHPMyAdmin² versão 4.0.10. O AVA Moodle versão 2.8 foi utilizado como interface gráfica do LERP. O framework JADE versão 4.2.2 e a linguagem de programação Java foram utilizados no desenvolvimento dos agentes.

A abordagem desenvolvida é proposta para ser utilizada por pessoas com pouco ou nenhum conhecimento em programação de computadores, bastando apenas possuir um computador pessoal com acesso à internet e conhecimentos básicos de informática. O laboratório proposto fornece suporte ao cadastramento de usuários, agendamento de experimentos e realização de desafios.

Foram realizados diversos testes para detectar a presença de falhas no sistema. A análise dos registros armazenados no BD e o *feedback* dos usuários envolvidos na pesquisa, colaboraram para que estas falhas fossem corrigidas. A abordagem da implementação utilizada foi a prototipação. A cada nova funcionalidade adicionada ao sistema, testes específicos eram realizados para melhorar a qualidade do software.

A Subseção 5.1 apresenta a descrição dos testes de caso de uso realizados com base nas principais funcionalidades do sistema. A Subseção 5.2 apresenta o perfil dos usuários que realizaram os testes no LERP, e resultado dos experimentos realizados. Para finalizar, a Seção 5.3 apresenta as considerações finais dos testes realizados.

¹<http://www.apache.org>

²<http://www.phpmyadmin.net/>

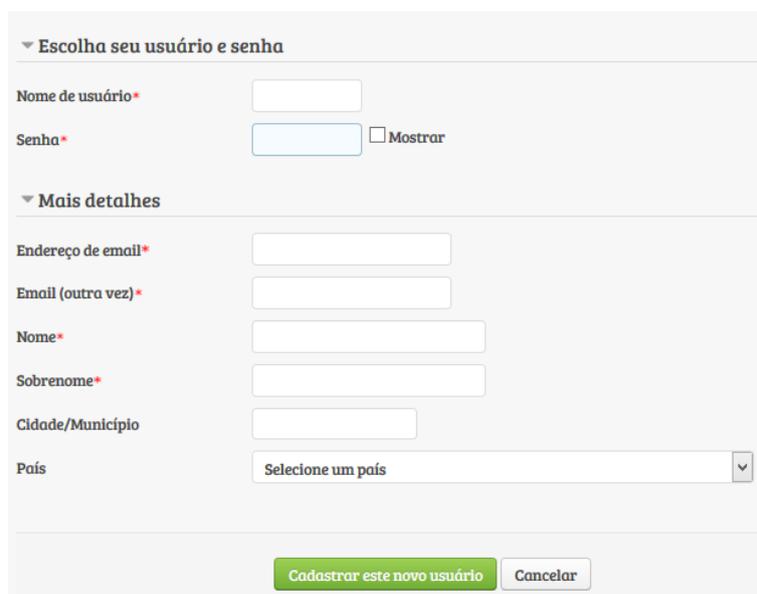
5.1 Testes dos Casos de Uso

Um caso de teste (CT) é um conjunto de condições em que um testador irá determinar se um sistema funciona corretamente ou não. Desta maneira, possui componentes que descrevem uma entrada, a ação e uma resposta esperada, para determinar se parte do sistema funciona conforme o esperado. Esta subseção apresenta os testes realizados com base nas principais funcionalidades do LERP, considerando os CASUs do sistema apresentados na Seção 4.2. Sendo assim, abaixo podemos visualizar os testes dos CASUs Solicitar Cadastro, Agendar Experimento e Programar Robô, respectivamente.

5.1.1 Teste do Caso de Uso Solicitar Cadastro

O Caso de Uso Solicitar Cadastro está associado à tela apresentada na Figura 5.1, que deve ser preenchida pelo usuário para que ele possa ter acesso à sala do LERP.

Figura 5.1: Cadastramento de Usuário



O formulário de cadastro de usuário é dividido em duas seções principais: "Escolha seu usuário e senha" e "Mais detalhes".

- Escolha seu usuário e senha:**
 - Nome de usuário* (campo de texto)
 - Senha* (campo de texto com ícone de olho para alternar visibilidade) e opção "Mostrar" (checkbox)
- Mais detalhes:**
 - Endereço de email* (campo de texto)
 - Email (outra vez)* (campo de texto)
 - Nome* (campo de texto)
 - Sobrenome* (campo de texto)
 - Cidade/Município (campo de texto)
 - País (menu suspenso com o texto "Selecione um país")

Na base do formulário, há dois botões: "Cadastrar este novo usuário" (em verde) e "Cancelar" (em cinza).

Os CTs que foram empregados utilizando o CASU Solicitar Cadastro estão descritos na Tabela 5.1.

Na coluna dos resultados, foram definidos os termos "Correto" para registrar que o sistema apresentou na tela um resultado esperado pelo seu desenvolvedor, e "Não Detectado" para apontar que existe a necessidade de validar o campo senha com no mínimo seis caracteres.

Tabela 5.1: Casos de Teste do CASU Solicitar Cadastro

Caso de Teste	Entrada de Dados	Resultados
CT01	<(" ", " ", " ", " ", " ", " ", " ", " ", " ", " ") , dados inválidos>	Correto.
CT02	<("thais", "12345", "thais.oliveira@ufr.br", " ", "Thais", "Oliveira", "Boa Vista", "Brasil"), está faltando o endereço de e-mail>	Correto.
CT03	<("thais", "12345", "thais.oliveira@ufr.br", "thais.oliveira@ufr.br", "Thais", " ", "Boa Vista", "Brasil"), está faltando o sobrenome>	Correto.
CT04	<("thais", "12345", "thais.oliveira@ufr.br", "thais.oliveira@ufr.br", "Thais", "Oliveira", "Boa Vista",), cadastro efetuado>	Correto.
CT05	<("thais", "12345", "thais.oliveira", "thais.oliveira", "Thais", "Oliveira", "Boa Vista", "Brasil"), endereço de e-mail inválido>	Correto.
CT06	<("thais", "12345", "thais.oliveira@ufr.br", "thais.oliveira@ufr.br", "Thais", "Oliveira", " ", " ") , cadastro efetuado>	Correto.
CT07	<("thais", "1", "thais.oliveira@ufr.br", "thais.oliveira@ufr.br", "Thais", "Oliveira", "Boa Vista", "Brasil"), cadastro efetuado>	Não detectado.

O teste para o CASU mostrou-se satisfatório. A incorreção detectada será corrigida nas próximas versões do sistema, por meio de alteração de código do AVA Moodle que realiza parte do cadastramento.

5.1.2 Teste do Caso de Uso Agendar Experimento

A Figura 5.2 mostra uma solicitação de reserva, apresentando os dados preenchidos pelo testador, que associa ao CASU Agendar Experimento. Após o usuário ter sua reserva confirmada, a solicitação feita aparecerá na relação de "Agendamentos Confirmados". O sistema permite também que o usuário cancele Agendamentos ou Reservas, a partir do ícone "Cancelar", conforme podemos visualizar na figura abaixo.

O LERP não permite que sejam realizados agendamentos para datas passadas, assim como para horários já ocupados por outros usuários (Figura 5.3).

Neste sentido, aparece para o usuário apenas os horários disponíveis para o dia selecionado. No caso de todos os horários de um dia estarem ocupados, aparecerá a mensagem de indisponibilidade de horário (Figura 5.4). Caso o usuário tente realizar o agendamento mesmo assim, aparecerá a mensagem informando que ele deve selecionar outra data com horários disponíveis.

Os CTs do CASU Agendar Experimentos estão descritos na Tabela 5.2.

Na coluna dos resultados, foi definido o termo "Correto" para registrar que o sistema apresentou na tela um resultado esperado pelo seu desenvolvedor. O teste para o CASU Agendar Experimento mostrou-se satisfatório, tendo em vista que nenhuma incorreção foi detectada.

Figura 5.2: Teste de Agendamento de Experimento

Página de Agendamentos!

Solicitar Agendamento

Data:

Horário:

Reserva(s) de Agendamento(s)

Usuário	Data	Horario	Situação	Cancelar
thais	17/02/2016	15h as 15h40	Pendente	✘

Agendamento(s) Confirmado(s)

Usuário	Data	Horario	Situação	Cancelar
thais	17/02/2016	9h as 9h40	Confirmado	✘

Figura 5.3: Datas Disponíveis para Agendamento

Página de Agendamentos!

Solicitar Agendamento

Data:

Horário:

February 2016

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29					

Figura 5.4: Teste de Agendamento com Indisponibilidade de Horário

Página de Agendamentos!

Solicitar Agendamento

Data:

Horário:

5.1.3 Teste do Caso de Uso Programar Robô

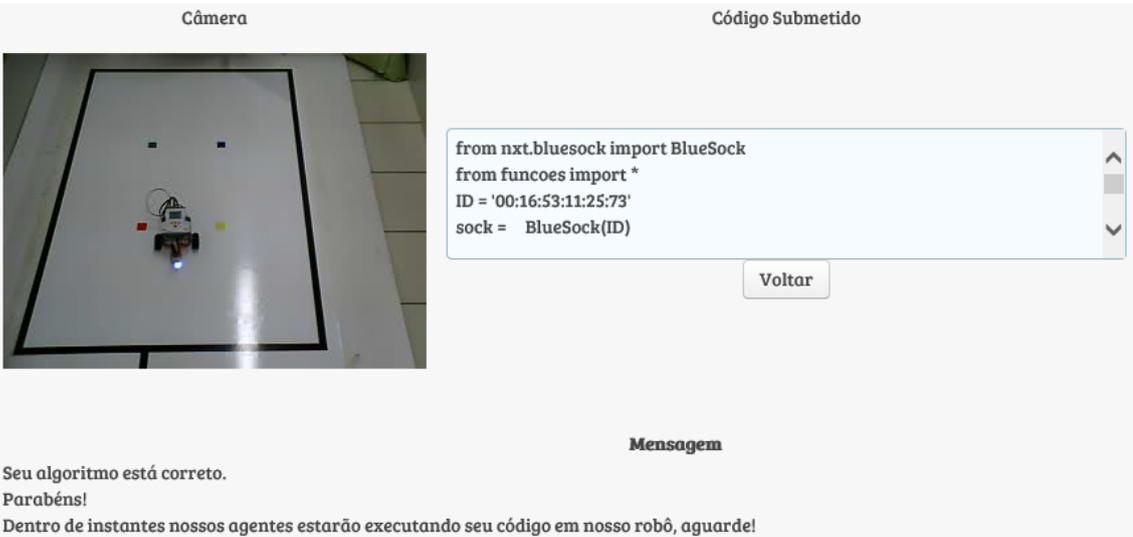
O CASU Programar Robô está associado à tela apresentada na Figura 5.5. O usuário tem acesso à essa tela após ter agendado um experimento e acessar o desafio no horário reservado.

Caso o usuário tente acessar o desafio sem agendar um horário ou fora do horário agendado, aparecerá uma mensagem informando que ele não pode ter acesso à sala de desafios (Figura 5.6).

Tabela 5.2: Casos de Teste do CASU Agendar Experimento

Caso de Teste	Entrada de Dados	Resultados
CT01	<(" ", " "), usuário deve selecionar uma data.>	Correto.
CT02	<("17/02/2016", "Indisponível"), usuário deve selecionar outra data com horários disponíveis.>	Correto.
CT03	<("15/02/2016", "9h as 9h40"), reserva efetuada com sucesso.>	Correto.
CT04	<("16/02/2016", "11h as 11h40"), reserva efetuada com sucesso.>	Correto.
CT05	<("17/02/2016", "13h as 13h40"), reserva efetuada com sucesso.>	Correto.
CT06	<("17/02/2016", "14h as 14h40"), reserva efetuada com sucesso.>	Correto.
CT07	<("17/02/2016", "15h as 15h40"), reserva efetuada com sucesso.>	Correto.

Figura 5.5: Programando Robô na Sala de Desafios



Câmera

Código Submetido

```
from nxt.bluesock import BlueSock
from funcoes import *
ID = '00:16:53:11:25:73'
sock = BlueSock(ID)
```

Voltar

Mensagem

Seu algoritmo está correto.
Parabéns!
Dentro de instantes nossos agentes estarão executando seu código em nosso robô, aguarde!

Os CTs do CASU Programar Robô envolvendo a sala de desafios, estão descritos na Tabela 5.3.

Na coluna dos resultados, foram definidos os termos "Correto" para registrar que o sistema apresentou na tela um resultado esperado pelo seu desenvolvedor, e "Não Detectado" para apontar que existe a necessidade de

Figura 5.6: Tentativa de Acesso à Sala de Desafios

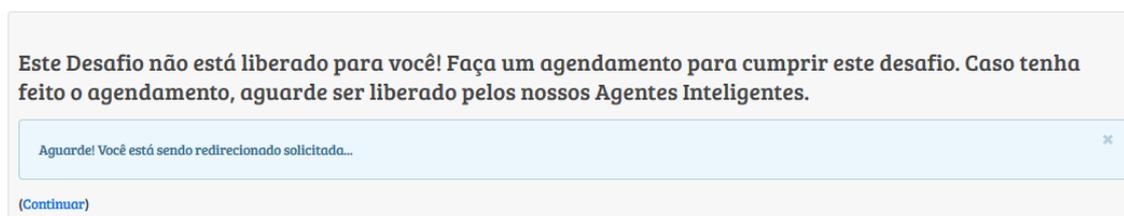


Tabela 5.3: Casos de Teste do CASU Programar Robô

Caso de Teste	Entrada de Dados	Resultados
CT01	<("desafio 1", " "), algoritmo correto.>	Não detectado.
CT02	<("desafio 2", ". "), unexpected exception.>	Correto.
CT03	<("desafio 3", "oi"), unexpected exception.>	Correto.
CT04	<("desafio 4", "from nxt.bluesock import BlueSock from funcoes import * ID = '00:16:53:11:25:73' sock = BlueSock(ID) brick = sock.connect() luz(brick, "azul") tempo_de_espera(5) apaga_luz(brick) sock.close()"), código executado com sucesso.>	Correto.
CT05	<("desafio 5", "from nxt.bluesock import BlueSock from funcoes import * ID = '00:16:53:11:25:73' sock = BlueSock(ID) brick = sock.connect() luz(brick, "azul") tempo_de_espera(5) apaga_luz(brick) sock.close()"), unexpected exception before line 4.>	Correto.

informar ao usuário que ele deve submeter um código no espaço destinado para tal, não o deixando em branco.

O teste para o CASU mostrou-se satisfatório. A incorreção detectada será corrigida nas próximas versões do sistema, por meio de alteração de código do AVA Moodle que realiza a verificação do conteúdo enviado no campo "código submetido".

5.2 Teste em uma Turma Real

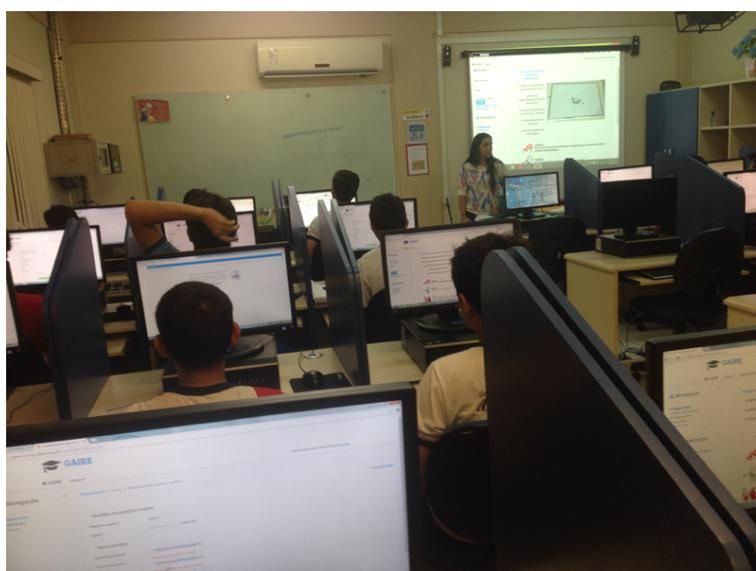
Foram realizados testes com alunos do Colégio de Aplicação da Universidade Federal de Roraima (CAP/UFRR), que apresentou-se promissor quanto à

utilização do LERP para motivar usuários a aprenderem programação e robótica. Os testes ocorreram no mês de Março do ano de 2016, na cidade de Boa Vista/RR, e teve duração de 2 (duas) semanas. O Apêndice F desta dissertação apresenta algumas fotos do contato inicial com os alunos e experimentos.

Inicialmente foi necessário contactar a coordenadora pedagógica do colégio escolhido para explanar sobre o projeto e seus objetivos. Em seguida, a coordenadora enviou uma autorização para os pais assinarem permitindo que seus filhos comparecessem no horário oposto das aulas para participar dos testes.

Vários alunos se mostraram interessados em participar dos testes. Tendo em vista a limitação de espaço e tempo, foram selecionados 20 (vinte) alunos do Ensino Fundamental, de turmas do 7^o e 8^o ano. No primeiro contato (Figura 5.7), foi realizada uma breve explanação sobre o LERP apresentando o ambiente utilizado e como ocorre o processo de manipulação remota. Neste dia foi solicitado também que os eles efetuassem o cadastro para ter acesso ao LERP.

Figura 5.7: Primeiro Contato com os Alunos



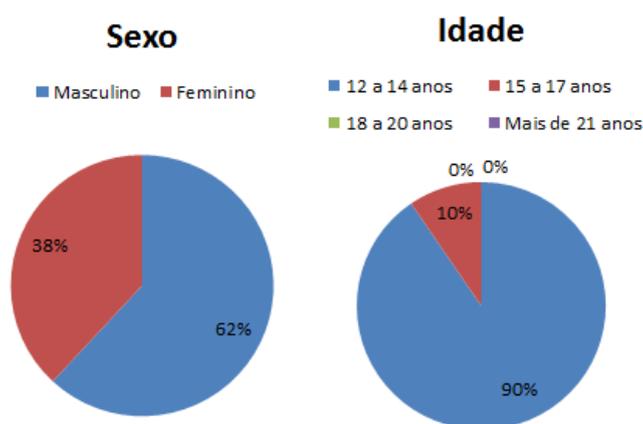
Todos os alunos conseguiram se cadastrar no sistema sem muitas dificuldades. Após terem realizado o cadastro, os alunos foram orientados a agendar os experimentos (cada aluno agendou um dia e um horário). O perfil dos usuários participantes dos testes, assim como os resultados obtidos são detalhados a seguir.

5.2.1 Perfil dos Usuários

Foi disponibilizado um questionário na sala virtual (Apêndice D), com objetivo de caracterizar os usuários que participaram dos testes no LERP. A importância de caracterizar o usuário está no fato de se estabelecer um olhar mais crítico sobre o resultado do questionário utilizado pelo usuário para analisar o trabalho desenvolvido. O critério escolhido para sintetizar o questionário foi o de aplicabilidade geral, ou seja, não foram incluídas neste questionário questões de caráter muito específico.

Primeiramente buscamos saber o sexo das pessoas que iriam realizar os testes. Dos entrevistados, 62% eram homens e 38% mulheres. Com relação à faixa etária 90% possuíam entre 12 a 14 anos, e 10% tinham de 15 a 17 anos. Quanto ao grau de escolaridade dos entrevistados, todos possuíam o Ensino Fundamental Incompleto. A Figura 5.8 apresenta os gráficos com os dados de caracterização dos usuários.

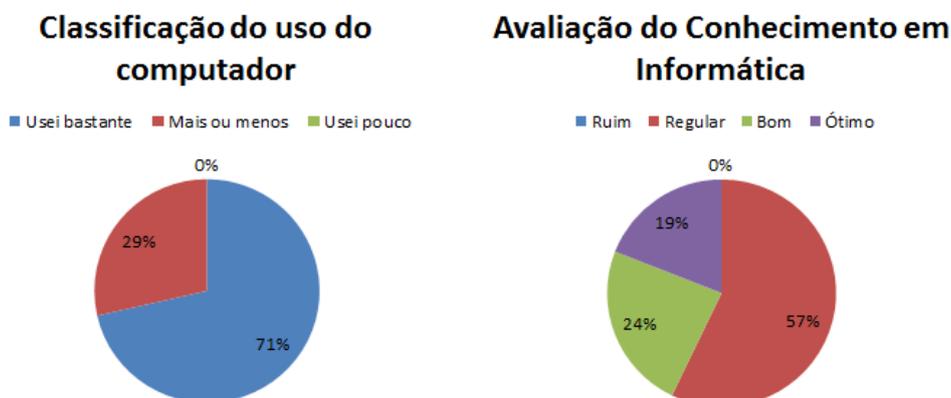
Figura 5.8: Caracterização dos Usuários



Os entrevistados foram questionados sobre o uso do computador. 100% afirmaram já ter utilizado. Sobre a classificação deste uso, 71% usaram bastante e 29% mais ou menos. Nenhum usuário relatou ter usado pouco. Quando questionados sobre o conhecimento que eles possuíam em informática, 57% avaliaram como regular, 24% afirmaram possuir conhecimento bom, e 19% avaliaram como ótimo. A Figura 5.9 apresenta os gráficos com a classificação em relação ao uso do computador e conhecimento dos usuários em informática.

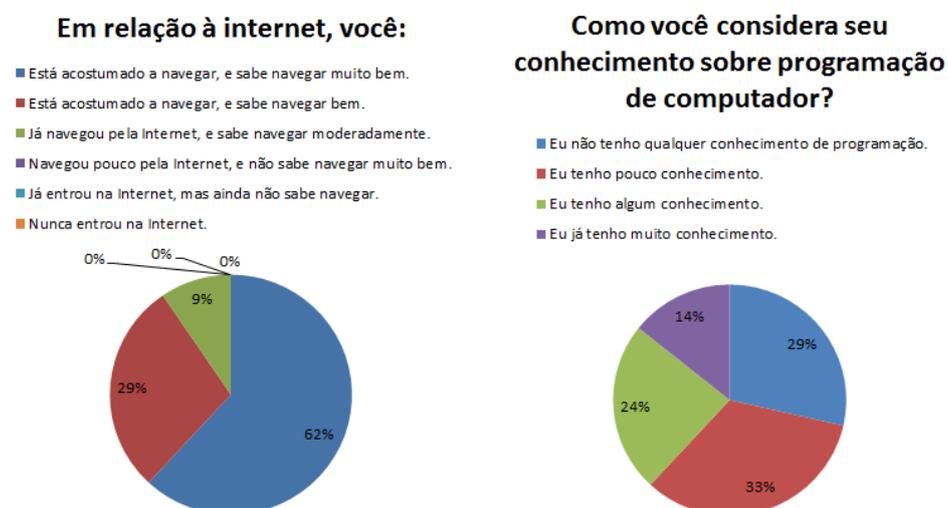
Como o LERP está disponível através de uma página Web, foi necessário saber o quanto de conhecimento os usuários possuíam em relação à navegabilidade na Internet. 62% dos entrevistados afirmaram estar acostumados

Figura 5.9: Conhecimento dos Usuários em Informática



a navegar, e sabe navegar muito bem. 29% afirmaram estar acostumados a navegar, e sabe navegar bem. 9% já navegou pela Internet, e sabe navegar moderadamente. Não houveram relatos de não usuários que não sabiam navegar. A Figura 5.10 apresenta os gráficos com o conhecimento dos usuários em relação à navegabilidade na Internet.

Figura 5.10: Conhecimento dos Usuários em Navegar na Internet



De maneira geral, como todos os alunos possuíam conhecimentos básicos em informática e sabiam navegar na Internet, acreditou-se que eles não teriam tanta dificuldade para acessar a sala virtual do LERP e realizar experimentos.

Os experimentos realizados e resultados obtidos nos testes são detalhados a seguir.

5.2.2 Experimentos Realizados

Embora não seja desejável, alguns alunos que se comprometeram a participar dos testes não compareceram no dia e horário agendado. Em conversa com a coordenadora pedagógica, ela informou que eles estavam em semana de prova, e em decorrência disso alguns optaram por desistir, já que a participação nos testes não agregaria nota nas disciplinas.

Dos 20 (vinte) alunos que haviam se cadastrado no sistema, 14 (quatorze) compareceram para realizar os testes. A Tabela 5.4 apresenta a relação do quantitativo de alunos participantes, com a porcentagem dos desafios que eles concluíram no prazo agendado para o experimento.

Tabela 5.4: Desafios Concluídos em Cada Experimento

Quantidade de Alunos	Desafios Concluídos	Porcentagem
1	2	40%
2	3	60%
7	4	80%
4	5	100%

Cada experimento teve a duração máxima de 40 (quarenta) minutos. Conforme exposto na Tabela 5.4, 4 (quatro) alunos concluíram os 5 (cinco) desafios disponíveis na sala virtual, no tempo estipulado. 7 (sete) alunos concluíram 4 (quatro) desafios a tempo. 2 (dois) alunos conseguiram concluir apenas 3 (três) desafios no período de 40 (quarenta) minutos, pois a Internet estava muito lenta e prejudicou o resultado. Somente 1 (um) aluno conseguiu concluir 2 (dois) desafios no período de 40 (quarenta) minutos, pois ele chegou 25 minutos atrasado para realizar os testes.

De maneira geral os alunos não tiveram dificuldades para concluir os desafios, tendo em vista a explanação que foi realizada sobre o sistema e o funcionamento no primeiro contato com eles, assim como foram apresentadas as funções que estavam disponíveis para uso na solução dos desafios.

Ao término dos experimentos, uma pesquisa de opinião (Apêndice E) foi disponibilizada na sala virtual, para que os usuários pudessem avaliar o sistema desenvolvido após terem realizado os testes. Este questionário teve como objetivo analisar a impressão dos usuários após terem experiências de programar robôs no LERP. Neste sentido, buscou-se saber se a metodologia

proposta auxiliou como elemento motivador na aprendizagem de programação. Os resultados obtidos foram promissores e encontram-se detalhados abaixo.

As duas primeiras perguntas do questionário buscaram avaliar a eficácia do kit *Legó Mindstorms* para apoiar na aprendizagem de programação. Quando questionados sobre o aprendizado em relação à programação no LERP 43% dos usuários afirmaram ter sido muito facilitado, e 57% julgaram como facilitado. Na oportunidade, buscou-se avaliar a metodologia utilizada para aprendizagem, onde 100% dos usuários avaliaram como muito importante. Os gráficos exibindo o resultado dessas duas questões encontram-se abaixo na Figura 5.11.

Figura 5.11: Aprendizagem de Programação e Metodologia Utilizada



Quando questionados sobre o aprendizado em relação aos conteúdos apresentados, 71% dos usuários classificaram como ótimos, e 29% como excelentes. Em relação ao conhecimento adquirido em programação de computador, 43% afirmaram já ter algum conhecimento antes, e 57% obtiveram muitos conhecimentos. A Figura 5.12 apresenta os gráficos com as respostas dos usuários sobre o nível de conhecimento adquirido e aprendizagem dos conteúdos apresentados.

Todos os usuários concordaram plenamente quando foram questionados se achavam que os robôs *Legó* proporcionaram uma experiência de aprendizagem positiva. Quando questionados se a utilização do kit robótico tornou fácil a aprendizagem de programação, 71% dos usuários concordaram plenamente, e 29% concordaram parcialmente. A Figura 5.13 apresenta os gráficos com os resultados dessas duas questões.

Quando questionados se a utilização de robôs tornou a aprendizagem de programação divertida, 86% dos usuários concordaram plenamente, e somente 14% concordaram parcialmente. Todos os usuários se sentiram motivados e interessados em aprender mais sobre programação usando robôs. A

Figura 5.12: Aprendizado sobre os Conteúdos Apresentados

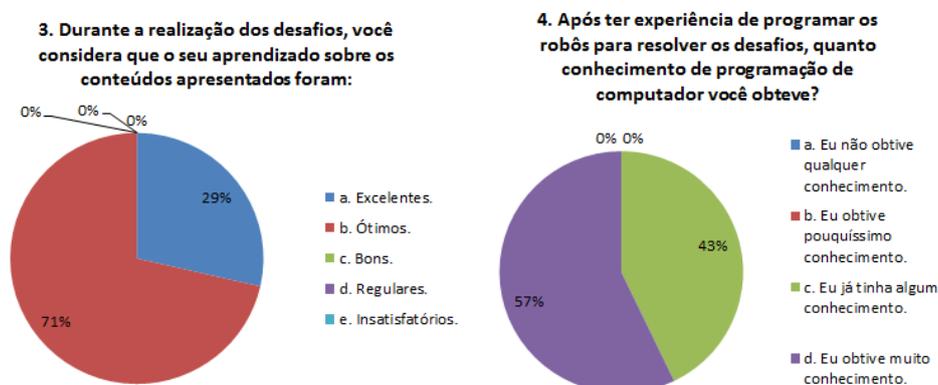


Figura 5.13: Experiência de Aprendizagem Positiva com Lego

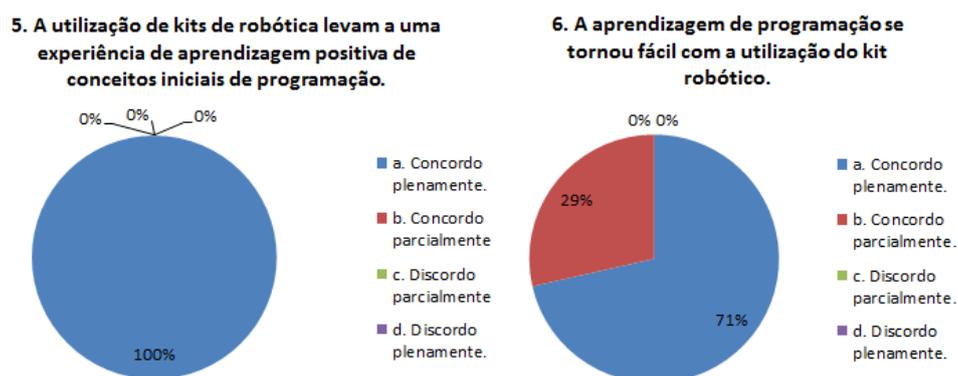


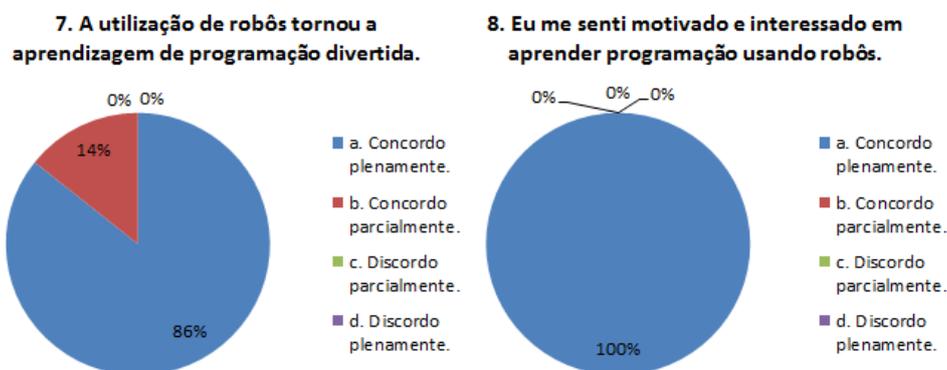
Figura 5.14 apresenta os gráficos com os resultados obtidos nas das últimas perguntas da pesquisa de opinião.

Os resultados obtidos com os testes realizados foram satisfatórios e mostraram-se promissores, pois no final foi possível perceber que o sistema proposto facilitou a aprendizagem de programação. Além disso, a utilização de robôs serviu como elemento motivador e tornou a aprendizagem divertida. A próxima subseção apresenta as conclusões dos testes.

5.3 Considerações Finais

Os resultados dos testes realizados nesta dissertação evidenciam que o sistema é viável de ser utilizado por qualquer pessoa que tenha interesse em aprender programação e robótica, possua conhecimentos básicos em infor-

Figura 5.14: Interesse e Motivação em Aprender Programação



mática e saiba navegar na Web. A única limitação para acessar a sala de desafios corretamente, é que a página do LERP deve ser acessada através do navegador *Internet Explorer*, para a correta exibição da imagem da câmera que utiliza o plugin *ActiveX*.

De maneira geral, o usuários sentiram-se interessados e motivados em aprender mais sobre programação utilizando robôs e realizar mais desafios. Eles questionaram também se poderiam agendar outros experimentos e praticar o que foi aprendido em outras datas, e se teriam mais desafios disponíveis para realizar.

A Seção 4.1 apresentou uma visão geral do laboratório desenvolvido, explanando a arquitetura do sistema e uma breve introdução sobre os agentes. A Seção 4.2 apresentou os casos de uso e diagramas de sequência do sistema. A Seção 4.3 demonstrou a interface de cadastramento de usuários, agendamento de experimentos, a sala de desafios e o processo de manipulação remota. A Seção 4.4 expôs as tecnologias utilizadas para o desenvolvimento desta dissertação. Por fim, a Seção 5 apresentou os testes que ocorreram com uma turma real, em que foi possível verificar que o sistema proposto contribuiu para a aprendizagem de programação, e despertou o interesse de continuar estudando robótica.

Logo, a estratégia apresentada neste trabalho atende aos objetivos expostos no Capítulo 1 e colabora para que os usuários sintam-se motivados a aprender programação utilizando robôs. O próximo Capítulo, faz uma conclusão a respeito deste projeto de Mestrado, citando também as contribuições e os trabalhos futuros que poderão surgir a partir do desenvolvimento deste.

Capítulo 6

Conclusões e Trabalhos Futuros

Nesta dissertação foi apresentada uma abordagem de ensino de programação com apoio de Sistemas Multiagente que serviu como elemento motivador para a aprendizagem utilizando robôs, proporcionando aos usuários a oportunidade de ter contato com experimentos desta natureza.

A abordagem proposta se demonstrou adequada para motivar usuários com conhecimentos básicos em programação, e até mesmo aqueles que não possuíam conhecimento. Nosso propósito por intermédio dessa pesquisa, foi investigar se os alunos se sentiriam motivados a aprender programação e robótica.

O trabalho apresentado utilizou uma metodologia de Engenharia de Software, voltada para Agentes Inteligentes, denominada Prometheus. A finalidade foi modelar os agentes implementados no sistema e assim ter a documentação necessária para uma melhor compreensão da organização dos agentes desenvolvidos. Neste sentido, serviu como apoio no levantamento de requisitos, modelagem do sistema e pode ser utilizado em caso de falhas.

Tendo em vista a complexidade do problema apresentado no trabalho, é possível afirmar que a metodologia Prometheus foi adequada ao processo de criação do Sistema Multiagente, pois ela possibilitou uma visão mais abrangente e detalhada da disposição dos agentes e de suas interações. A arquitetura do sistema foi proposta pensando em possíveis inclusões de novos agentes, sendo em tese uma arquitetura extensível.

Após a modelagem, o trabalho passou por diversos refinamentos para que pudesse ser testado. A avaliação consistiu de duas etapas: realização dos testes de caso de uso em cima das principais funcionalidades do sistema, e teste com usuários reais. A primeira etapa permitiu verificar o comportamento do sistema para diferentes entradas de dados. Assim, foi possível realizar alguns ajustes no funcionamento do sistema e comportamento dos agentes. A segunda etapa foi realizada na cidade de Boa Vista/RR, com estudantes

do Colégio de Aplicação da Universidade Federal de Roraima. Desta maneira, foi possível ter um contato inicial com os alunos para explanar sobre o sistema desenvolvido e seu funcionamento.

Mesmo com a redução no número de alunos que compareceram para realizar os testes, os resultados foram satisfatórios, pois o sistema foi capaz de despertar o interesse e motivá-los a aprender programação e robótica. As atividades foram disponibilizadas e desenvolvidas por níveis de dificuldades. Desta maneira os alunos sentiram-se motivados à superar os novos desafios que foram controlados e monitorados pelo SMA.

Foi necessário familiarizá-los com as tecnologias voltadas a computadores e robôs, dar suporte sobre o funcionamento do sistema, explanar os conteúdos e a metodologia a ser utilizada, dentre outras particularidades de pessoas leigas no assunto. Nos testes realizados, foi possível perceber melhorias e ajustes que podem ser realizados a fim de potencializar o LERP, visando melhorar o aprendizado.

Como no desenvolvimento do protótipo a interface do LERP utilizou o AVA Moodle, as suas ferramentas não foram incluídas nos testes, visto que já existe na literatura uma vasta exposição sobre elas. Os testes exploraram o cadastro de usuários no ambiente, confirmação de acesso à sala virtual, realização de reservas, agendamento de experimentos, liberação e controle da seção na sala de desafios, e programação remota do robô.

Ao final do projeto, foi realizada uma avaliação com alunos para identificar as contribuições em relação ao uso da metodologia proposta. Os resultados foram promissores, e o projeto continuará sendo fomentado de forma continuada, de maneira a abranger e disponibilizar ao maior número possível de usuários, a oportunidade de ter contato com programação utilizando robôs através de um laboratório remoto.

A abordagem apresenta avanços sobre os laboratórios remotos de robótica encontrados na literatura, pois permite selecionar desafios distintos, programando os robôs em uma linguagem de programação e utiliza uma interação usuário/sistema com o apoio dos Agentes Inteligentes. O sistema desenvolvido está funcional, e com pequenas adaptações pode ser extensível para aceitar outras linguagens de programação. Da mesma maneira, é possível modificá-lo e disponibilizá-lo em uma página Web, tornando-o independente da estrutura do AVA Moodle.

Devido à natureza distribuída do problema, os SMAs acompanham as ações dos usuários no ambiente, além de sugerir desafios com base no histórico de cada um. A implementação do sistema é uma consequência desse estudo. Ainda durante o desenvolvimento deste trabalho, foi realizado um levantamento bibliográfico sobre o estado da arte que abordam os temas robótica educacional, prática de programação com robôs, laboratório remoto

de robótica e sistemas multiagente.

Dois artigos referentes à pesquisa foram produzidos e encontram-se detalhados no Apêndice A. Como principal contribuição desta dissertação destaca-se o desenvolvimento de um Laboratório de Ensino de Robótica e Programação baseado em SMAs, para atender ao objetivo proposto no trabalho.

Para trabalhos futuros pretende-se substituir os kits robóticos *Lego Mindstorms* pelo Arduino, devido esse possuir menor custo e ser mais flexível para aceitar outras linguagens de programação. Pretende-se ainda especializar o trabalho dos agentes explorando o cognitivo, sugerindo os desafios adequados com base no perfil de aprendizagem de cada usuário.

Referências Bibliográficas

ALENCAR, M.; NETTO, J. d. M. Sistema multiagente para apoiar a percepção e o acompanhamento de atividades em ambientes virtuais de aprendizagem. In: *CBIE 2012-Congresso de Informática na Educação*. [S.l.: s.n.], 2012. v. 1, p. 1000–1006.

ALMEIDA, T.; NETTO, J. F. Robótica pedagógica aplicada ao ensino de programação: Uma revisão sistemática da literatura. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2015. v. 26, n. 1, p. 597–606.

AMBRÓSIO, A. P. L. et al. Programação de computadores: compreender as dificuldades de aprendizagem dos alunos. Universidade da Coruña/Universidade do Minho, 2011.

AZEVEDO, S.; AGLAÉ, A.; PITTA, R. Minicurso: Introdução à robótica educacional. *62^a Reunião Anual da SBPC*. Disponível em: < <http://www.sbpnet.org.br/livro/62ra/minicursos/MC%20Samuel%20Azevedo.pdf>, 2010.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. Goal question metric paradigm. *Encyclopedia of software engineering*, John Wiley e Sons, Inc, 1994.

BATISTA, A. F. d. M. Desenvolvendo sistemas multiagentes na plataforma jade. *Santo André: Universidade Federal do Abc*, p. 32, 2008.

BUIU, C.; BUGA, A.; COMAN, A. M. Teaching robotics and virtual reality in a synergistic approach. In: IEEE. *e-Learning in Industrial Electronics (ICELIE), 2013 7th IEEE International Conference on*. [S.l.], 2013. p. 71–75.

CASINI, M. et al. A remote lab for experiments with a team of mobile robots. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 14, n. 9, p. 16486–16507, 2014.

CHAOS, D. et al. Virtual and remote robotic laboratory using ejs, matlab and labview. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 13, n. 2, p. 2595–2612, 2013.

CICIRELLI, F.; NIGRO, L. A control framework for model continuity in jade. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications*. [S.l.], 2014. p. 97–104.

CRUZ, M. K. da et al. Controle de kit de robótica através de laboratório remoto pela internet: uma aplicação para a formação docente e para a educação básica. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2009. v. 1, n. 1.

CSTA. *CSTA National Secondary Computer Science Survey: Comparison of Results from 2005, 2007, 2009, and 2011*. 2012.

D'ABREU, J. V. V.; BASTOS, B. L. Robótica pedagógica: Uma reflexão sobre a apropriação de professores da escola elza maria pellegrini de aguiar. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2013. v. 1, n. 1, p. 280–289.

D'ABREU, J. V. V.; MIRISOLA, L. G.; RAMOS, J. J. Ambiente de robótica pedagógica com br_gogo e computadores de baixo custo: Uma contribuição para o ensino médio. In: *Anais do XXII Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2011. v. 1, n. 1, p. 100–109.

DÁRIO, C. F. B. *Uma Metodologia Unificada para o Desenvolvimento de Sistemas Orientados a Agentes*. Tese (Doutorado) — Dissertação-Faculdade de Engenharia Elétrica e de Computação-UNICAMP.-Campinas, São Paulo:[sn], 2005.

FERBER, J. *Multi-agent systems: an introduction to distributed artificial intelligence*. [S.l.]: Addison-Wesley Reading, 1999. v. 1.

FERNANDES, C. d. C. *S-educ: Um simulador de ambiente de robótica educacional em plataforma virtual*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2013.

FILHO, D. A. M.; GONÇALVES, P. C. Robótica educacional de baixo custo: Uma realidade para as escolas brasileiras. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2008. v. 1, n. 1.

- FRIEDRICH, R. V. et al. Proposta metodológica para a inserção ao ensino de lógica de programação com logo e lego mindstorms. In: *Anais do XXIII Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2012. v. 23, n. 1, p. 26–30.
- GIL, A. C. Como elaborar projetos de pesquisa. são paulo: Atlas, 2002. *Métodos e técnicas de pesquisa social*, v. 5, p. 64–73, 1991.
- GOMES, A. et al. Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. *Revista Portuguesa de Pedagogia*, n. 42-2, p. 161–179, 2008.
- GOMES, A.; MENDES, A. A teacher’s view about introductory programming teaching and learning: Difficulties, strategies and motivations. In: IEEE. *Frontiers in Education Conference (FIE), 2014 IEEE*. [S.l.], 2014. p. 1–8.
- GOMES, G. et al. Dificuldades na aprendizagem da programação no ensino profissional–perspetiva dos alunos. In: *Lisboa: Actas do II Congresso Internacional TIC e Educação*. [S.l.: s.n.], 2012. p. 438–448.
- GRANDI, R.; FALCONI, R.; MELCHIORRI, C. Unibot remote laboratory: A scalable web-based set-up for education and experimental activities in robotics. *Proc. 18th IFAC WC, Milan, Italy*, p. 8521–8526, 2011.
- GUIMARÃES, E. et al. Real: A virtual laboratory for mobile robot experiments. *Education, IEEE Transactions on*, IEEE, v. 46, n. 1, p. 37–42, 2003.
- HALMA, A. *Robomind.net - Welcome to Robomind.net, the new way to learn programming*. 2015. <http://www.robomind.net>. Disponível em: <<http://www.robomind.net>>.
- HILLS, M.; KLINT, P.; VINJU, J. An empirical study of php feature usage: a static analysis perspective. In: ACM. *Proceedings of the 2013 International Symposium on Software Testing and Analysis*. [S.l.], 2013. p. 325–335.
- JÚNIOR, O. d. R. N. Desenvolvimento da fluência tecnológica em programa educacional de robótica pedagógica. Florianópolis, SC, 2011.
- KEELE, S. Guidelines for performing systematic literature reviews in software engineering. In: *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. [S.l.: s.n.], 2007.
- KULICH, M. et al. Syrotek-distance teaching of mobile robotics. *Education, IEEE Transactions on*, IEEE, v. 56, n. 1, p. 18–23, 2013.

LIANG, H.-N. et al. A first introduction to programming for first-year students at a chinese university using lego mindstorms. In: IEEE. *Teaching, Assessment and Learning for Engineering (TALE), 2013 IEEE International Conference on*. [S.l.], 2013. p. 233–238.

MARCHI, A. C. B. D. et al. Monitorando a comunicação na cv-muzar com o uso de agentes inteligentes. *Revista Brasileira de Computação Aplicada*, v. 2, n. 1, p. 57–68, 2010.

MIYAMOTO, N.; HIGUCHI, K.; TSUJI, T. Incremental data migration for multi-database systems based on mysql with spider storage engine. In: IEEE. *Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on*. [S.l.], 2014. p. 745–750.

NETTO, J. F. d. M. *Uma Arquitetura para Ambientes Virtuais de Convivência: uma Proposta Baseada em Sistemas Multiagente*. Tese (Doutorado) — Tese de Doutorado, Universidade Federal do Espírito Santo. Espírito Santo, Vitória, 2006.

PADGHAM, L.; WINIKOFF, M. *Developing intelligent agent systems: A practical guide*. [S.l.]: John Wiley & Sons, 2005. v. 13.

PADGHAM, L.; WINIKOFF, M. Prometheus: A practical agent-oriented methodology. *Agent-oriented methodologies*, Idea Group, Hershey, PA, p. 107–135, 2005.

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

PINTO, M. d. C.; ELIA, M. da F.; SAMPAIO, F. F. Formação de professores em robótica educacional com hardware livre arduino no contexto um computador por aluno. In: *Anais do XVIII Workshop de Informática na Escola*. [S.l.: s.n.], 2012. v. 1, n. 1.

RAMOS, J. J. et al. Iniciativa para robótica pedagógica aberta e de baixo custo para inclusão social e digital no brasil. *Anais do VIII Simpósio Brasileiro de Automação Inteligente (SBAI 2007), Florianópolis, SC*, 2007.

RESNICK, M.; BERG, R.; EISENBERG, M. Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *The Journal of the Learning Sciences*, Taylor & Francis, v. 9, n. 1, p. 7–30, 2000.

RODRIGUES, J. R. et al. iquiz: integrated assessment environment to improve moodle quiz. In: IEEE. *Frontiers in Education Conference, 2013 IEEE*. [S.l.], 2013. p. 293–295.

RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 2002.

SANTOS, A.; GOMES, A.; MENDES, A. A taxonomy of exercises to support individual learning paths in initial programming learning. In: IEEE. *Frontiers in Education Conference, 2013 IEEE*. [S.l.], 2013. p. 87–93.

SANTOS, F. L.; NASCIMENTO, F. M. S.; BEZERRA, R. M. Reduc: A robótica educacional como abordagem de baixo custo para o ensino de computação em cursos técnicos e tecnológicos. 2010.

SELL, R.; SEILER, S.; PTASIK, D. Embedded system and robotic education in a blended learning environment utilizing remote and virtual labs in the cloud, accompanied by robotic homelab kit. *International Journal of Emerging Technologies in Learning (iJET)*, v. 7, n. 4, p. 25–33, 2012.

SHIKAMA, T. Development of teaching materials for computer programming using a robot remotely controlled by a pc through wireless communication. *Informatics Society*, p. 29–36, 2014.

SILVA, A. F. D. *RoboEduc: Uma metodologia de aprendizado com Robótica Educacional*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2009.

SOUSA, D. F. d. Desenvolvendo a lógica e algoritmos no ensino médio. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2013. v. 1, n. 1.

SOUZA, M. B. et al. Arcabouço de um ambiente telerobótico educacional baseado em sistemas multiagente. In: *Anais do XXII Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2011. v. 1, n. 1, p. 680–689.

SOUZA, M. B. d. et al. Uma abordagem metodológica voltada para o ensino-aprendizagem de algoritmos. *RENOTE*, v. 11, n. 1, p. 1–10, 2013.

TAN, P.-H.; TING, C.-Y.; LING, S.-W. Learning difficulties in programming courses: undergraduates' perspective and perception. In: IEEE. *Computer Technology and Development, 2009. ICCTD'09. International Conference on*. [S.l.], 2009. v. 1, p. 42–46.

TAVARES, O. d. L.; MENEZES, C. S. de; NEVADO, R. A. de. Pedagogical architectures to support the process of teaching and learning of computer programming. In: IEEE. *Frontiers in Education Conference (FIE), 2012*. [S.l.], 2012. p. 1–6.

TRENTIN, M. A.; PÉREZ, C. A. S.; TEIXEIRA, A. C. A robótica livre no auxílio da aprendizagem do movimento retilíneo. In: *Anais do XIX Workshop de Informática na Escola*. [S.l.: s.n.], 2013. v. 1, n. 1, p. 51–59.

VICKNAIR, C.; WILKINS, D.; CHEN, Y. Mysql and the trouble with temporal data. In: ACM. *Proceedings of the 50th Annual Southeast Regional Conference*. [S.l.], 2012. p. 176–181.

VICTORINO, L. et al. Laboratório virtual de atividades didáticas-labvad. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2009. v. 1, n. 1, p. 1723–1732.

WALKER, S.; PRYTHERCH, D.; TURNER, J. The pivotal role of staff user experiences in moodle and the potential impact on student learning. In: IEEE. *e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on*. [S.l.], 2013. p. 192–197.

WOOLDRIDGE, M. *An introduction to multiagent systems*. [S.l.]: John Wiley & Sons, 2009.

WRENCH, P. M.; IRWIN, B. V. Towards a sandbox for the deobfuscation and dissection of php malware. In: IEEE. *Information Security for South Africa (ISSA), 2014*. [S.l.], 2014. p. 1–8.

YU, X. Using lego mindstorms in the undergraduate curriculum of it. In: IEEE. *Information Technology in Medicine and Education (ITME), 2012 International Symposium on*. [S.l.], 2012. v. 1, p. 270–273.

Apêndice A

Artigos Científicos Produzidos Durante o Mestrado

Neste apêndice listamos os artigos científicos completos produzidos no decorrer do Mestrado. O primeiro artigo foi aceito para publicação, e o segundo encontra-se em fase de avaliação.

Almeida, T., Netto, J. F. (2015). Robótica pedagógica aplicada ao ensino de programação: Uma revisão sistemática da literatura. In *Anais do XXVI Simpósio Brasileiro de Informática na Educação*, volume 26, páginas 597606.

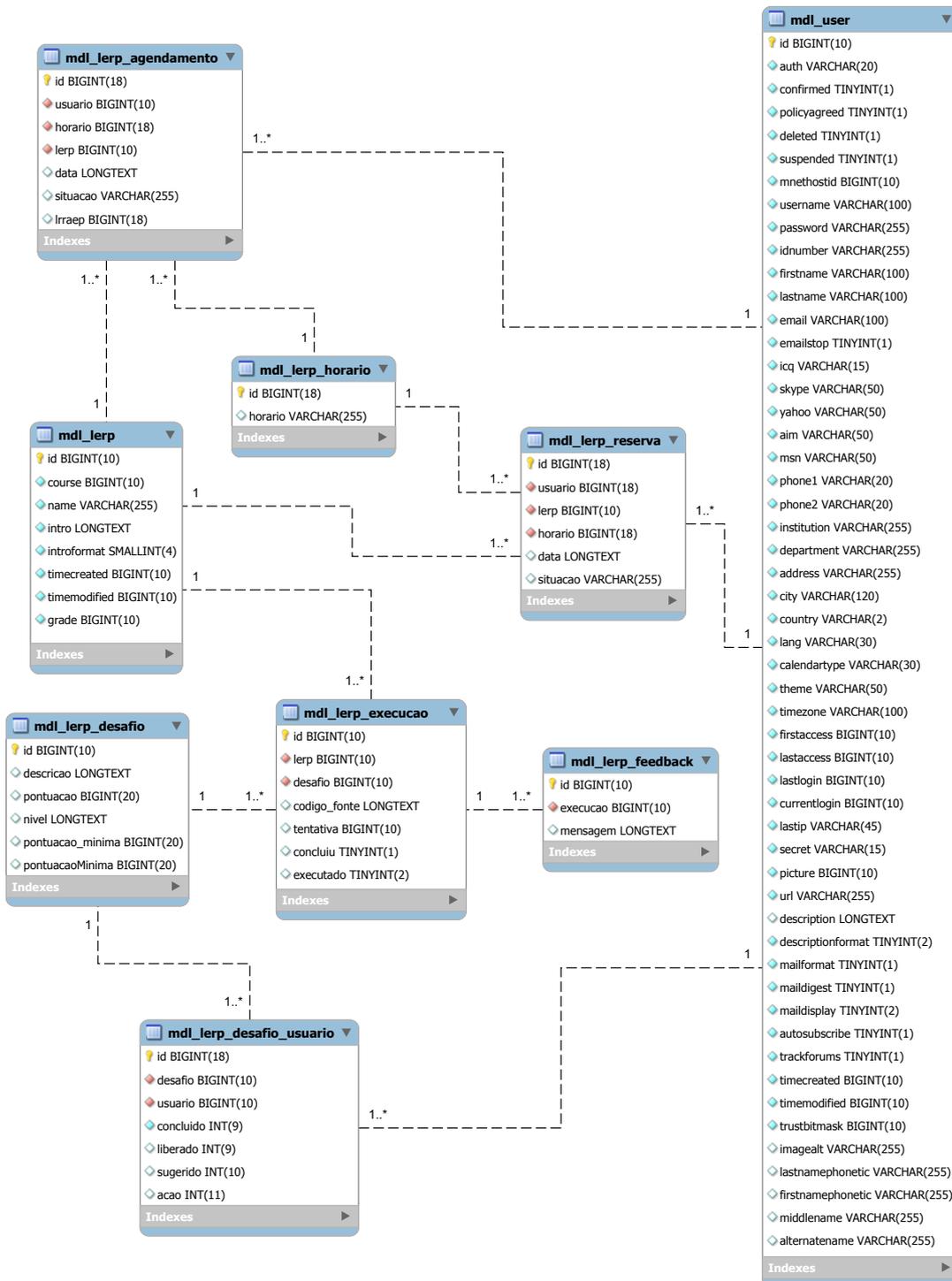
Almeida, T., Netto, J. F., Silva, R. G., Custódio, T. P. Laboratório Remoto de Robótica como Apoio à Aprendizagem de Programação. (Submetido para o *XXVII Simpósio Brasileiro de Informática na Educação*).

Apêndice B

Tabelas Adicionais

Neste apêndice são apresentadas as tabelas adicionais (Figura B.1) que foram criadas no Banco de Dados do Moodle para o correto funcionamento do sistema apresentado nesta dissertação. Neste sentido, foi necessário acrescentar oito tabelas no BD, para que fosse possível armazenar informações de agendamentos, reservas, desafios, execução e feedback aos usuários.

Figura B.1: Tabelas Adicionais Utilizadas no LERP



Apêndice C

Código Fonte das Funções

Neste apêndice são apresentados os códigos fonte das funções definidas na Tabela 4.5. Essas funções foram criadas para auxiliar os usuários do LERP na realização dos desafios.

```
from nxt.bluesock import BlueSock
from nxt.motor import Motor

from sensor.generic import Light, Color20
from sensor.common import Type

from datetime import datetime, timedelta
import time

#Portas dos motores
PORTA_A = 0x00
PORTA_B = 0x01
PORTA_C = 0x02

#Porta dos Sensores
PORTA_1 = 0x00
PORTA_2 = 0x01
PORTA_3 = 0x02
PORTA_4 = 0x03

#Variaveis Globais
ID = '00:16:53:11:25:73'

def luz(brick, string_cor):
```

```
cor = Color20(brick , PORTA_3)

if (string_cor == "vermelho"):
print "vermelho"
cor.set_light_color(Type.COLORRED)
elif (string_cor == "azul"):
cor.set_light_color(Type.COLORBLUE)
print "azul"
elif(string_cor == "verde"):
print "Verde"
cor.set_light_color(Type.COLORGREEN)

```

```
def apaga_luz(brick):

cor = Color20(brick , PORTA_3)
cor.set_light_color(Type.COLORNONE)

```

```
def tempo_de_espera(tempo):

last = time.time()
now = time.time()

while ((now - last) <= tempo):
now = time.time()

```

```
def correr_por_tempo(brick , velocidade , tempo):

Motor_direito = Motor(brick , PORTA_B)
Motor_esquerdo = Motor(brick , PORTA_C)

Motor_esquerdo.run(velocidade , True)
Motor_direito.run(velocidade , True)

tempo_de_espera(tempo)

Motor_esquerdo.run(0 , True)
Motor_direito.run(0 , True)

```

```
def rotaciona(brick , velocidade , numero_rotacao):
```

```
Motor_direito = Motor(brick , PORTA_B)
Motor_esquerdo = Motor(brick , PORTA_C)

tacho = Motor_esquerdo.get_tacho().tacho_count
numero_rotacao = tacho + numero_rotacao

while (tacho <= numero_rotacao):
Motor_esquerdo.run(velocidade , True)
Motor_direito.run(-velocidade , True)
tacho = Motor_esquerdo.get_tacho().tacho_count
#print "numero_rotacao" , numero_rotacao
#print "tacho: " , tacho

Motor_esquerdo.run(0 , True)
Motor_direito.run(0 , True)
```

Apêndice D

Questionário de Caracterização do Usuário

O objetivo deste questionário é caracterizar o usuário do LERP. A importância de caracterizar o usuário está no fato de se estabelecer um olhar mais crítico sobre o resultado do questionário utilizado pelo usuário para analisar o trabalho desenvolvido. O critério escolhido para sintetizar os questionários foi o de aplicabilidade geral, ou seja, não foram incluídas neste questionário questões de caráter muito específico.

1. Indique seu sexo:
 - (a) Masculino.
 - (b) Feminino.

2. Indique sua idade:
 - (a) De 12 a 14 anos.
 - (b) De 15 a 17 anos.
 - (c) De 18 a 20 anos.
 - (d) Mais de 21 anos.

3. Indique seu grau de escolaridade:
 - (a) Ensino Fundamental Incompleto.
 - (b) Ensino Fundamental Completo.
 - (c) Ensino Médio Incompleto.
 - (d) Ensino Médio Completo.

APÊNDICE D. QUESTIONÁRIO DE CARACTERIZAÇÃO DO USUÁRIO91

- (e) Superior Incompleto.
 - (f) Superior Completo.
4. Você já usou computador alguma vez?
- (a) Sim.
 - (b) Não.
5. Se você já usou, como classifica este uso?
- (a) Usei bastante.
 - (b) Mais ou menos.
 - (c) Usei pouco.
6. Como você avalia seu conhecimento em informática?
- (a) Ruim.
 - (b) Regular.
 - (c) Bom.
 - (d) Ótimo.
7. Em relação à Internet, você:
- (a) Está acostumado a navegar, e sabe navegar muito bem.
 - (b) Está acostumado a navegar, e sabe navegar bem.
 - (c) Já navegou pela Internet, e sabe navegar moderadamente.
 - (d) Navegou pouco pela Internet, e não sabe navegar muito bem.
 - (e) Já entrou na Internet, mas ainda não sabe navegar.
 - (f) Nunca entrou na Internet.
8. Como você considera seu conhecimento sobre programação de computador antes de ter experiência com kits robóticos?
- (a) Eu não tenho qualquer conhecimento de programação.
 - (b) Eu tenho pouco conhecimento.
 - (c) Eu tenho algum conhecimento.
 - (d) Eu já tenho muito conhecimento.

Apêndice E

Pesquisa de Opinião

O objetivo deste questionário é analisar a impressão dos usuários após terem experiências de programar robôs no LERP. Neste sentido, buscou-se saber se o a metodologia proposta auxiliou como elemento motivador na aprendizagem de programação.

1. Quanto às atividades realizadas no LERP, o seu aprendizado em relação à programação foi:
 - (a) Muito facilitado.
 - (b) Facilitado.
 - (c) Razoavelmente facilitado.
 - (d) Pouco facilitado.
 - (e) Não facilitou.

2. Você considera que a metodologia proposta, utilizando recursos tecnológicos é importante para aprendizagem?
 - (a) Muito importante.
 - (b) Pouco importante.
 - (c) Não faz diferença.

3. Durante a realização dos desafios, você considera que o seu aprendizado sobre os conteúdos apresentados foram:
 - (a) Excelentes.
 - (b) Ótimos.
 - (c) Bons.

- (d) Regulares.
 - (e) Insatisfatórios.
4. Após ter experiência de programar os robôs para resolver os desafios, quanto conhecimento de programação de computador você ^a obteve?
- (a) Eu não obtive qualquer conhecimento.
 - (b) Eu obtive pouquíssimo conhecimento.
 - (c) Eu já tinha algum conhecimento.
 - (d) Eu obtive muito conhecimento.
5. Na sua opinião, a utilização de kits de robótica levam a uma experiência de aprendizagem positiva de conceitos iniciais de programação:
- (a) Concordo plenamente.
 - (b) Concordo parcialmente.
 - (c) Discordo parcialmente.
 - (d) Discordo plenamente.
6. A aprendizagem de programação se tornou fácil com a utilização do kit robótico:
- (a) Concordo plenamente.
 - (b) Concordo parcialmente.
 - (c) Discordo parcialmente.
 - (d) Discordo plenamente.
7. A utilização de robôs tornou a aprendizagem de programação divertida:
- (a) Concordo plenamente.
 - (b) Concordo parcialmente.
 - (c) Discordo parcialmente.
 - (d) Discordo plenamente.
8. Eu me senti motivado e interessado em aprender programação usando robôs:
- (a) Concordo plenamente.
 - (b) Concordo parcialmente.
 - (c) Discordo parcialmente.
 - (d) Discordo plenamente.

Apêndice F

Fotos dos Testes Realizados

A seguir são apresentadas algumas fotos dos testes que foram realizados no mês de Março/2016, na cidade de Boa Vista/RR, com alunos do Colégio de Aplicação da Universidade Federal de Roraima (CAP/UFRR), e tiveram duração de 2 (duas) semanas.

Figura F.1: Primeiro Contato com os Alunos

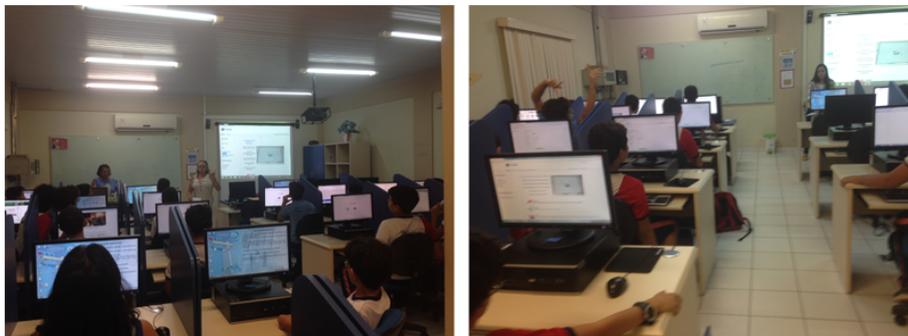


Figura F.2: Alunos Realizando Experimentos na Sala de Desafios (1)

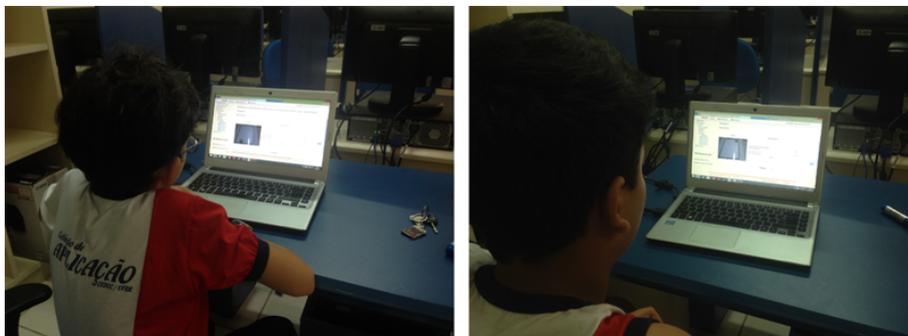


Figura F.3: Alunos Realizando Experimentos na Sala de Desafios (2)

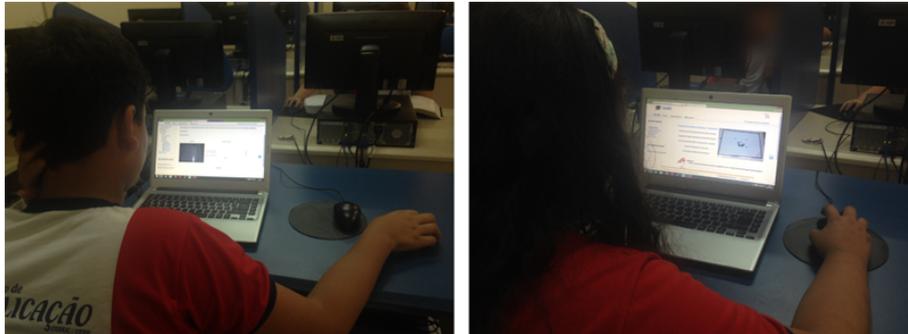


Figura F.4: Alunos Realizando Experimentos na Sala de Desafios (3)

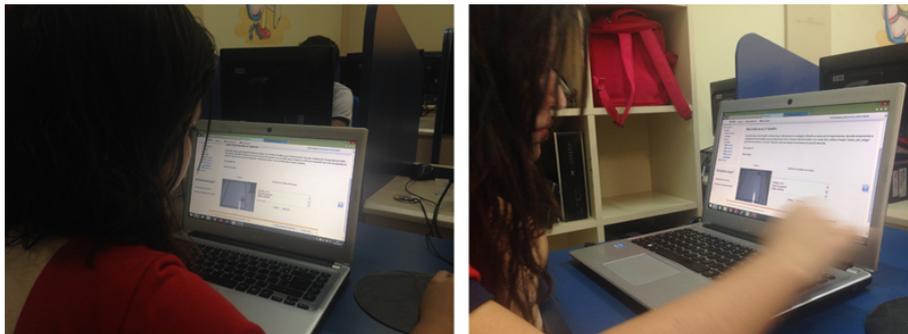


Figura F.5: Alunos Realizando Experimentos na Sala de Desafios (4)

