



Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica e de
Computação



Uma Metodologia de Avaliação Automática para Aulas de Robótica Educativa

Carla da Costa Fernandes

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Número de ordem PPgEEC: D202
Natal, RN, 14 de julho de 2017

Universidade Federal do Rio Grande do Norte - UFRN Sistema de Bibliotecas -
SISBI

Catálogo de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Fernandes, Carla da Costa.

Uma metodologia de avaliação automática para aulas de robótica educacional / Carla da Costa Fernandes. - 2017.

106 f.: il.

Tese (doutorado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação. Natal, RN, 2017.

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves.

1. Robótica educacional - Tese. 2. Metodologia de avaliação - Tese. 3. Avaliação automática - Tese. 4. Simulador robótico - Tese. I. Gonçalves, Luiz Marcos Garcia. II. Título.

RN/UF/BCZM

CDU 004.896:37

*Ao meu filho, Daniel, que me
inspirou a finalizar este trabalho.*

Agradecimentos

Ao meu orientador, Professor Luiz Marcos Garcia Gonçalves, sou grata pela orientação e por toda a ajuda durante a minha vida acadêmica.

Aos meus pais, Flávio e Silvia, e à minha irmã Flávia, pelo apoio durante esta jornada.

Ao meu marido Danilo, pelo apoio e amor incondicional, e por me dar o melhor presente da minha vida.

À equipe RoboEduc, sem a qual eu não teria me interessado na área de robótica educacional.

Aos Top&Amigos.

À CAPES, pelo apoio financeiro.

Resumo

O processo de ensino da robótica educacional envolve não apenas a criação de um robô, mas também a utilização do mesmo para a realização de uma atividade interdisciplinar. O uso de robótica em sala de aula levanta uma questão: como os alunos serão avaliados nos conceitos que envolvem a aula de robótica educacional? Desenvolvemos neste trabalho uma metodologia de avaliação dos alunos em aulas de robótica educacional. Baseado em pesquisas com professores e no conhecimento da equipe sobre a área, levantamos critérios de avaliação relevantes para esta área de ensino, considerando os tópicos de uma aula de robótica educacional que devem ser avaliados: montagem do robô, programação do robô e resolução da atividade interdisciplinar proposta. Além destes critérios, existem pontos sobre relação social que também são relevantes em aulas de robótica educacional, como interação social, trabalho em equipe, concentração, criatividade, participação, entre outros. Com este objetivo, definimos seis critérios avaliativos que podem ser aplicados na avaliação a critério do professor. No entanto, reconhecemos o desafio da aplicação de uma metodologia avaliativa em aulas de robótica educacional, por demandar muito tempo e esforço do professor, que deve ficar atento a todos os passos da aula de robótica, e aplicar os critérios avaliativos a todos os alunos. Com o objetivo de auxiliar o docente nesta tarefa, inserimos esta metodologia de avaliação em um simulador de robótica educacional, permitindo que o simulador realize a avaliação de forma automática, fornecendo ao professor apenas os resultados da mesma. O professor ainda é o responsável por definir todos os critérios avaliativos, mas o simulador é o responsável por analisar o aluno à medida que ele executa uma atividade e atribuir a ele notas considerando os critérios definidos pelo professor. Por fim, para validar a metodologia de avaliação, foram realizados testes com professores de robótica, nos quais os professores deveriam cadastrar os parâmetros avaliativos de algumas atividades e analisar os resultados fornecidos pelo simulador à medida que os alunos executavam estas atividades. Os resultados mostram que os professores tem interesse em utilizar esta metodologia e que acreditam que os critérios avaliativos propostos são de grande importância para uma aula de robótica educacional, permitindo que o professor realize modificações no processo de ensino caso identifique dificuldades por parte dos alunos.

Palavras-chave: Robótica educacional, Metodologia de avaliação, Avaliação automática, Simulador robótico.

Abstract

The process of teaching educational robotics involves not only the creation of a robot, but also the use of it for conducting an interdisciplinary activity. The use of robotics in the classroom raises an issue: how will students be evaluated in the concepts that involves educational robotics topics? We developed a methodology used to evaluate students on educational robotics classes. Based on researches with robotics teachers and the knowledge of this research team about the area, we raised evaluation criterias relevant to this area of education, considering the topics of an educational robotics class that should be evaluated: robot assembly, robot programming and the resolution of the interdisciplinary proposed activity. In addition to these criteria, there are points about social relationships that are also relevant in educational robotics classes, such as social interaction, teamwork, concentration, creativity, participation, among others. With this objective, we defined six evaluation criteria that can be applied in the evaluation at the teacher's discretion. However, we recognize the challenge of applying an evaluative methodology in educational robotics classes, because it requires a lot of time and effort from the teacher, who must be attentive to all the steps of the robotics class, and apply the evaluation criteria to all students. With the objective of assisting the teacher in this task, we inserted this evaluation methodology in an educational robotic simulator, allowing the simulator to perform the evaluation automatically, providing the teacher with the results. The teacher is still responsible for defining all the evaluative criteria, but the simulator is responsible for analyzing the student as he performs an activity and assigning grades to the student considering the criteria defined by the teacher. Finally, to validate the evaluative methodology, tests were performed with robotic teachers, in which teachers should register the evaluative parameters of some activities and analyze the results provided by the simulator as the students performed these activities. The results show that teachers are interested in using this methodology and believe that the proposed evaluation criteria are of great importance for an educational robotics class, allowing the teacher to carry out changes in the teaching process if it identifies difficulties by students.

Keywords: Educational Robotics, Evaluation Methodology, Automatic Evaluation, Robotic Simulator.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
1.1 Escopo do trabalho	3
1.2 Motivação	4
1.3 Tema, problema e hipótese	5
1.4 Metodologia	6
1.5 Aplicações	6
1.6 Estrutura do trabalho	7
2 Fundamentação Teórica	9
2.1 A Robótica Educacional	9
2.1.1 Ferramentas utilizadas em aulas de robótica educacional . . .	10
2.2 Simuladores robóticos	12
2.3 Processos de avaliação	14
2.3.1 Modalidades do processo avaliativo	15
2.3.2 Avaliação automática do ensino	16
3 Trabalhos relacionados	19
3.1 Sistemas de gestão acadêmica	19
3.2 Sistemas automáticos de avaliação	20
3.2.1 Avaliação de questões de múltipla escolha ou discursivas . . .	21
3.2.2 Avaliação de códigos de programação	22
3.3 Sistemas de avaliação para robótica educacional	24
4 Solução para Avaliação em Robótica Educacional	27
4.1 Avaliação de aulas de robótica educacional	27
4.1.1 Parâmetros de avaliação	28
4.1.2 Avaliação automática	30
4.2 Metodologia de avaliação proposta	31
4.2.1 Detecção da posição do robô	32
4.2.2 Atribuição de notas subjetivas	33
4.2.3 Detecção de elementos no robô	34

4.2.4	Movimentação e detecção de objetos no ambiente virtual . . .	34
4.2.5	Análise do código de programação	35
4.2.6	Quantidade de tentativas de execução	41
5	Implementação	45
5.1	Simulador S-Educ	45
5.2	Mudanças no simulador S-Educ	47
5.3	Modelagem do banco de dados	48
5.4	Diagramas UML	51
5.5	Cadastro de critérios de avaliação	54
5.5.1	Detecção da posição do robô	55
5.5.2	Atribuição de notas subjetivas	57
5.5.3	Detecção de elementos no robô	57
5.5.4	Movimentação e detecção de objetos no ambiente virtual . . .	57
5.5.5	Análise do código de programação	58
5.5.6	Quantidade de tentativas de execução	60
5.6	Avaliação automática	60
5.7	Visualização dos resultados	61
6	Experimentos e Resultados	67
6.1	Plataforma de testes	67
6.1.1	Descrição das atividades	68
6.1.2	Grupo de testes	69
6.2	Análise dos resultados	69
7	Conclusão	73
7.1	Trabalhos Futuros	75
	Referências bibliográficas	76
A	Descrição dos testes: definição dos critérios avaliativos	83
B	Descrição dos testes: análise dos resultados dos alunos	87
C	Questionário aos professores	89

Lista de Figuras

5.1	Exemplos de robôs do simulador S-Educ.	46
5.2	Tela inicial do simulador S-Educ.	48
5.3	Tela de simulação do simulador S-Educ.	49
5.4	Criação de novas turmas.	50
5.5	Adicionar alunos a turmas.	50
5.6	Diagrama de fluxo de dados do S-Educ.	50
5.7	Diagrama entidade-relacionamento.	52
5.8	Diagrama de atividades do caso de uso Criação de Atividade.	53
5.9	Diagrama de atividades do caso de uso Execução de Atividade.	54
5.10	Diagrama de atividades do caso de uso Finalizar Atividade.	55
5.11	Avaliações cadastradas na atividade.	56
5.12	Cadastro do critério 1.	56
5.13	Cadastro do critério 2.	57
5.14	Cadastro do critério 3.	58
5.15	Cadastro do critério 4.	58
5.16	Cadastro do critério 4: definição dos parâmetros.	59
5.17	Cadastro do critério 4: visualização dos parâmetros avaliativos.	59
5.18	Cadastro do critério 5.	60
5.19	Cadastro do critério 6.	61
5.20	Tela de visualização dos resultados.	62
5.21	Visualização de atividades por nível.	63
5.22	Visualização de resultados de um aluno em uma atividade.	64
5.23	Visualização de resultados de um aluno.	65
5.24	Visualização de resultados de uma atividade.	66
6.1	Gráfico 1: importância dos critérios avaliativos.	71
6.2	Gráfico 2: facilidade da definição dos parâmetros avaliativos.	71

Lista de Tabelas

4.1	Exemplo dos caractere utilizados na definição das <i>strings</i> dos códigos.	36
4.2	Transformação de códigos em <i>strings</i>	37
4.3	Comparação da aplicação da Distância de Levenshtein e Hamming.	40
4.4	Exemplo da quantidade de tentativas em uma atividade.	42
4.5	Análise da nota do aluno seguindo a equação 4.6.	43
5.1	Exemplos de comandos em linguagem R-Educ.	46

Capítulo 1

Introdução

A robótica é uma área tecnológica relativamente recente, capaz de relacionar as áreas de mecânica, eletrônica e computação. Na última década a robótica tem sido fortemente citada como ferramenta educacional estratégica para apoiar o processo de ensino-aprendizagem nos diversos níveis educacionais. Por ser uma forma atrativa e lúdica de trabalhar os conceitos vistos em sala de aula, a robótica educacional motiva os alunos a refletirem sobre os temas apresentados e a resolverem os problemas expostos. Na robótica educacional é possível estimular a solução de problemas com conceitos multidisciplinares, como física, matemática, geografia, entre outros (Alimisis & Boulougaris 2014).

O uso de robótica no ambiente escolar se destaca por possibilitar aos alunos a aplicação de seu conhecimento para a resolução de problemas através do raciocínio lógico e da investigação no desenvolvimento de um método científico, mostrando, na prática, conceitos teóricos (Miranda et al. 2011). De acordo com Prado (2008), o uso de tecnologias como a robótica permite transformar a vida escolar em um ambiente mais desafiador, criativo e dinâmico, que viabiliza a construção de um conhecimento crescente baseado em experimentações.

Em uma aula de robótica educacional, ao se deparar com uma atividade ou um desafio proposto pelo professor, os alunos devem utilizar a criatividade para contextualizar o problema e resolvê-lo utilizando robôs. Seja trabalho em grupo ou individual, o aluno tem que testar hipóteses, construir e reconstruir protótipos e programas, socializando e contribuindo com ideias (Fornaza 2016).

Para o desenvolvimento de aulas de robótica educacional são utilizados *kits* de robótica, que são conjuntos de peças específicas, que envolvem motores e sensores, utilizados para a montagem de robôs. Existem vários tipos de *kits* de robótica, com características e funcionalidades diferentes, sendo os mais utilizados os *kits* importados devido a sua superioridade quando se trata de qualidade e aplicabilidade. No entanto, estes *kits* possuem um alto custo, o que os tornam inacessíveis para muitas escolas brasileiras (Miranda et al. 2011).

Com o objetivo de disseminar a robótica educacional nas escolas brasileiras sem a necessidade da aquisição de *kits* de robótica, preservando a qualidade da aula e considerando um custo de inserção baixo, desenvolvemos um simulador de robótica, chamado S-Educ (Fernandes 2013). Este simulador permite que o aluno monte robôs virtualmente e os utilize para desenvolver atividades interdisciplinares que envolvem

o controle e a programação do robô. As aulas podem ser desenvolvidas também utilizando *kits* de robótica e simuladores, sendo estes utilizados para a realização dos testes ou para dar continuidade à atividade fora de sala de aula. Embora o simulador reproduza o comportamento do robô, ele não é capaz de substituir a presença do robô físico em uma aula de robótica educacional, pois a montagem do robô utilizando um *kit* de robótica estimula a criatividade e a coordenação motora do aluno. Além disso, a presença de um robô em sala de aula é um fator muito estimulante para os alunos.

Mesmo considerando os benefícios oriundos da presença do robô real em sala de aula, o simulador é uma ferramenta que apresenta vantagens quando utilizado em uma aula de robótica educacional. Ele permite a realização de testes com grande facilidade, permitindo que o aluno pratique também em casa, além de ser uma alternativa mais viável economicamente para a inserção da robótica educacional.

Um sistema simulado para a realização de aulas de robótica educacional foi desenvolvido por Seymour Papert, precursor da robótica educacional, na década de 70, ao utilizar a linguagem LOGO com o *software* LOGO (Matte 2011). Este sistema era capaz de simular os movimentos executados pela Tartaruga de Chão, robô utilizado inicialmente para a realização das atividades de robótica.

Não encontramos, no entanto, metodologias que auxiliem o professor de robótica educacional na avaliação dos seus alunos. Alguns estudos citam avaliações específicas do conhecimento interdisciplinar, avaliações subjetivas, feitas pelos professores, ou avaliações de interesse, feitas pelos alunos (Silva et al. 2008, Santos & de Menezes 2005, Soares & Borges 2011). Surge então a dúvida: como avaliar os alunos em aulas de robótica educacional?

Benitti (2012) realizou uma pesquisa sobre artigos científicos que descrevem o uso de robótica em escolas, com o objetivo de identificar benefícios da robótica e definir temas de pesquisa ainda em aberto. Em sua pesquisa, ele focou em dez artigos que julgou mais relevantes. Um dos tópicos de sua análise foi: como o aprendizado dos alunos é avaliado.

Em sua pesquisa, Benitti (2012) relatou que em apenas um dos trabalhos foi feita a avaliação dos alunos, enquanto que nos outros artigos não foi feita uma análise quantitativa do aprendizado do aluno.

Através de pesquisas realizadas com professores de robótica educacional, vimos que é possível fornecer subsídios ao professor para avaliação dos alunos usando informações fornecidas de forma automática pelo ambiente computacional utilizado nas aulas. Os critérios de avaliação podem ser diferentes dependendo da aula que será aplicada e do nível de conhecimento dos alunos. No entanto, é de extrema importância a avaliação subjetiva do professor, capaz de atribuir notas ao aluno dependendo do seu desempenho.

É importante que a avaliação de aulas de robótica educacional permeie as três modalidades de avaliação, começando com uma avaliação diagnóstica, que guie o professor para a preparação do curso, seguindo com uma avaliação continuada, formativa, que auxilie-o a realizar mudanças que julgue necessárias no processo de ensino, finalizando com uma avaliação somativa, classificatória.

Neste trabalho desenvolvemos uma metodologia de avaliação e acompanhamento do aluno inserida em um sistema computacional, que apresenta ao professor uma análise da evolução da turma e de cada aluno, mostrando em que pontos eles estão com mais dificuldades, permitindo que as aulas sejam adaptadas para atender às necessidades de cada turma. Esta análise é feita através de dados obtidos por uma avaliação automática, uma avaliação realizada a partir de um sistema computacional que utiliza dados fornecidos pelo professor ao criar as atividades para definir os critérios avaliativos, realizando a avaliação a partir da execução das atividades pelos alunos neste sistema, que neste caso será um simulador robótico. O sistema pode ser utilizado como auxílio para o professor, que obtém dados empíricos sobre o desempenho dos alunos.

O aluno realizará as atividades em um simulador de robótica educacional, que analisará o desenvolvimento da mesma para avaliar o progresso do aluno. É importante frisar que este sistema permite que os alunos exercitem os conhecimentos aprendidos em aulas de robótica educacional através da execução de atividades, mas que não é ferramenta suficiente para uma aula de robótica educacional. Sua utilização como ferramenta de ensino depende inteiramente dos professores de robótica educacional, responsáveis por ensinar os tópicos das aulas, cadastrar as atividades no sistema, definir os critérios avaliativos e acompanhar o desenvolvimento dos alunos.

O sistema permite também uma avaliação subjetiva do professor, que leva em consideração tópicos como trabalho em equipe, criatividade, elaboração de hipóteses, trabalho colaborativo, entre outros, tópicos estes importantes quando analisamos uma aula de robótica educacional.

Esta metodologia de avaliação pode ser aplicada também em aulas de robótica educacional que não utilizam simuladores, apenas *kits* de robótica. Desta forma, a avaliação seria realizada pelo professor, responsável não apenas por definir os critérios avaliativos, mas também por analisá-los durante a execução das aulas pelos alunos. Neste caso, esta pesquisa oferece uma base sobre os pontos de avaliação a serem utilizados pelo professor durante o processo avaliativo.

1.1 Escopo do trabalho

Visando aumentar a inserção tecnológica nas escolas públicas do Rio Grande do Norte, uma equipe de pesquisadores da Universidade Federal do Rio Grande do Norte vem desenvolvendo, desde 2006, projetos relacionados a robótica educacional. Inicialmente, a equipe participou de um projeto no qual foram ministradas aulas de robótica educacional na Escola Municipal Prof. Ascendino de Almeida, localizada na periferia de Natal/RN, nos anos de 2006 e 2007. Para o desenvolvimento destas aulas foram utilizados *kits* de robótica Lego RCX e um *software* de programação, na época ainda em desenvolvimento, chamado RoboEduc. Silva (2009) descreve os resultados dos testes realizados neste projeto, frisando que foram obtidos resultados satisfatórios com relação ao aprendizado dos alunos, mas que o projeto não teve continuidade devido ao desinteresse dos professores envolvidos.

O desenvolvimento do *software* RoboEduc foi continuado por Pitta (2008), que

além de implementar cinco diferentes níveis de programação no RoboEduc, realizou testes de usabilidade com esta ferramenta (Pitta 2011).

A equipe envolvida no desenvolvimento do *software* RoboEduc e na realização dos testes na Escola Ascendino de Almeida também desenvolveu outras ferramentas para serem utilizadas em aulas de robótica educacional.

Foi desenvolvido um sistema online que permite que os alunos programem diferentes tipos de robôs utilizando a mesma linguagem de programação (Sá 2013, Sá 2016), linguagem esta baseada nos trabalhos anteriores da equipe (Pitta 2008). Este sistema permite a difusão da robótica educacional, possibilitando a aplicação de aulas de robótica utilizando diferentes tipos de *hardware*.

Levando em consideração a dificuldade de aquisição de *kits* de robótica como, por exemplo, os *kits* Lego, esta equipe desenvolveu ferramentas capazes de aumentar a adesão das escolas às aulas de robótica educacional. Os trabalhos precursores deste incluem o desenvolvimento de um simulador robótico bidimensional, e em seguida o aprimoramento do mesmo, com a criação do simulador S-Educ, com a inserção de novas funcionalidades (Fernandes 2013). A metodologia descrita neste trabalho foi aplicada utilizando este simulador como base para a execução das aulas de robótica educacional, realizando modificações no mesmo que serão discutidas na Seção 5.2, e utilizando o sistema desenvolvido por Sá (2016) para a programação do robô.

1.2 Motivação

Ao desenvolver uma atividade de robótica educacional, o aluno realiza diversos testes, com o objetivo de analisar se a solução por ele abordada está correta. Possíveis erros podem incluir erros na montagem do robô, na programação, ou até mesmo no entendimento da atividade. Nesse caso, o aluno deve refazer sua estratégia.

Baseado em nossas experiências na área de robótica educacional, percebemos que cada aluno evolui a um determinado passo. Embora os alunos tenham aula juntos, realizando as mesmas atividades, é importante que eles tenham um acompanhamento diferenciado, baseado em seu avanço em cada ponto que constitui uma aula de robótica educacional. Por exemplo, alguns alunos tem mais facilidade do que outros para programar robôs, enquanto que outros se interessam mais no tema interdisciplinar abordado na aula.

No entanto, os professores de robótica educacional não possuem ferramentas que os auxiliem a identificar as dificuldades de cada aluno, permitindo a aplicação de atividades específicas, ou até mesmo a realização de atividades extra-classe.

Por esse motivo, vimos a necessidade de um sistema que forneça dados ao professor acerca do desempenho dos alunos na realização das atividades propostas, permitindo que ele possa analisar a evolução do aluno e sua média em relação à turma e em relação ao valor esperado. Os dados obtidos na avaliação podem ser utilizados para realizar modificações no processo de ensino ou para classificar os alunos seguindo os critérios adotados na instituição de ensino. Com este objetivo, associamos ao uso do simulador S-Educ uma metodologia de avaliação, capaz de gerar dados sobre a execução das aulas de robótica educacional, indicando informações sobre a média da

turma e de cada aluno separadamente.

1.3 Tema, problema e hipótese

Em atividades envolvendo programação, não apenas em robótica educacional, mas de uma forma geral um problema enfrentado é como quantificar ou valorar adequadamente cada atividade proposta em uma dada oficina. Algumas são mais complexas, outras mais simples. Assim, focando ainda mais o escopo desta tese, definimos o tema específico de pesquisa como sendo o estudo e desenvolvimento de uma metodologia de avaliação educacional com o uso da tecnologia de robótica educacional.

Ainda mais, queremos uma metodologia automatizada, que auxilie o professor, mas sem tirar o seu papel principal de mediador do processo de ensino-aprendizagem. Com o papel de mediador, o professor é responsável por definir os parâmetros de avaliação e de indicar como eles serão utilizados para que o sistema realize a avaliação automática dos alunos. Sendo assim, o sistema visionado é capaz de quantificar a evolução do aluno nas aulas de robótica educacional. A avaliação pode ser feita quando as atividades forem executadas em sala de aula ou como atividades extra-classe. Além disso, também é possível que o professor adicione uma nota subjetiva sobre o desempenho do aluno em sala de aula, atribuindo a ele a nota que julgar merecida.

Com este tema em mente, torna-se simples definir o problema a ser estudado nesta tese, que se relaciona também com avaliação de aulas usando a tecnologia de robótica educacional, que pode ser anunciado em forma de três perguntas simples como: que modalidades de avaliação podem ser aplicadas, o que elas devem avaliar, e como os dados obtidos podem auxiliar o professor a ajustar a forma de ensino? Novamente, note que o papel de mediador do professor é levado em conta. A ferramenta vem no sentido de ajudá-lo a fazer a avaliação de forma simplificada.

Uma vez enunciado o tema e problema, a hipótese de pesquisa a ser verificada neste trabalho é de que é possível criar uma metodologia de avaliação automática voltada para aulas de robótica educacional, considerando as particularidades deste tipo de aula, que envolvem desde a análise do robô utilizado até o desenvolvimento do programa utilizado para controlá-lo, desde que isto seja realizado de forma guiada pelo professor que, em última instância é o sujeito mediador do processo ensino-aprendizagem.

O professor é responsável por designar as atividades para as suas turmas e indicar o nível de conhecimento necessário para a realização de cada uma delas. Caso o professor perceba que os alunos não atingiram o conhecimento necessário, ele pode optar por mantê-los neste nível, e criar novas atividades para que eles exercitem as habilidades nas quais tem mais dificuldades.

1.4 Metodologia

Para verificar a validade da hipótese apresentada, na primeira etapa deste trabalho fizemos uma pesquisa a respeito dos critérios a serem avaliados em uma aula de robótica educacional, com o objetivo de estabelecer um referencial teórico para o projeto. Levando em consideração o conhecimento do nosso grupo de pesquisa nesta área e os requisitos funcionais citados por professores de robótica educacional, definimos os critérios de avaliação que seriam implementados no sistema.

Na próxima etapa inserimos estes critérios avaliativos no simulador S-Educ, levando em consideração os requisitos identificados na primeira etapa. O aluno é avaliado toda vez que uma atividade é realizada, analisando os critérios avaliativos escolhidos pelo professor na criação da atividade.

Na terceira etapa inserimos exemplos de aulas de robótica no simulador S-Educ, aulas essas que foram utilizadas na fase de testes. As aulas envolvem os requisitos levantados na primeira etapa, de forma a testar todos os critérios avaliativos propostos nestes trabalho.

Por fim, na última etapa, fizemos testes com professores, analisando o comportamento do sistema e o interesse dos educadores em utilizá-lo para avaliar alunos em aulas de robótica educacional. Os resultados desses testes são apresentados adiante, neste texto.

1.5 Aplicações

Em aulas de robótica educacional é possível utilizar simuladores para simular as atividades que seriam realizadas com os robôs reais. O uso de simuladores, embora não substitua por completo o uso de *kits* de robótica, permite a realização de diversas atividades e a facilidade da implantação da robótica educacional nas escolas brasileiras, dado que esta é uma ferramenta mais acessível economicamente, e que permite atingir um maior número de alunos. Além disso, os alunos podem praticar as atividades de robótica fora da sala de aula mesmo que não tenham acesso a *kits* de montagem.

O uso de sistemas computacionais que simulem aulas de robótica educacional permitem também uma análise automática a respeito da execução das atividades propostas pelo professor. O professor é capaz de cadastrar atividades e indicar que tópicos da aula devem ser avaliados, como a movimentação do robô no ambiente virtual, a leitura de sensores, a análise de códigos desenvolvidos, entre outros, habilitando o sistema a realizar esta análise sempre que uma atividade for executada.

A realização de análises automáticas fornece ao professor informações úteis para o decorrer das aulas, como os pontos em que os alunos estão tendo mais dificuldades, para que o professor foque mais nestes itens. Este tipo de análise permite identificar também quais os alunos que estão tendo mais dificuldades e que devem ter uma atenção reforçada, e também os alunos que estão com notas muitas altas, que podem precisar de atividades mais desafiadoras.

1.6 Estrutura do trabalho

Neste trabalho foi desenvolvida uma metodologia para avaliação automática de aulas de robótica educacional, utilizada em um simulador de robótica. No Capítulo 1 mostramos o escopo do trabalho, destacando as aplicações de tal sistema e a metodologia que foi utilizada. Expomos uma visão geral da motivação deste trabalho, que é a criação de uma solução para o problema de avaliação de aulas de robótica educacional.

O Capítulo 2 apresenta os referenciais teóricos relacionados com este trabalho, como explicações sobre robótica educacional, simulação de robôs, e processos de avaliação, sejam eles automáticos ou não. No Capítulo 3 são apresentados alguns sistemas de avaliação, focando nos sistemas automáticos.

O Capítulo 4 descreve detalhadamente o problema de avaliação de aulas de robótica educacional, descrevendo os parâmetros de avaliação de uma aula de robótica e indicando como esta avaliação pode ser feita de forma automática. Ainda neste Capítulo, são descritos os critérios avaliativos inseridos no simulador, suas particularidades e como eles podem ser utilizados para atribuição da nota final do aluno.

No Capítulo 5 é descrito o simulador utilizado neste trabalho, o S-Educ, são mostradas as modificações que foram feitas neste simulador para inserir a metodologia de avaliação desenvolvida, e é descrita a modelagem do banco de dados utilizada para realizar a avaliação. Neste Capítulo, são mostradas a nova interface gráfica do sistema.

No Capítulo 6 são mostrados exemplos de cadastro de atividades e de critérios avaliativos, mostrando experimentos realizados com o sistema desenvolvido. Por fim, no Capítulo 7 são apresentadas as conclusões deste trabalho, formuladas a partir dos experimentos realizados.

Capítulo 2

Fundamentação Teórica

Como visto no Capítulo 1, neste trabalho propomos uma metodologia que permite acompanhar o desenvolvimento do aluno em uma aula de robótica educacional. Esta metodologia foi inserida no simulador de robótica educacional S-Educ.

Neste Capítulo, colocamos todo o embasamento teórico necessário ao entendimento dos conceitos utilizados no trabalho, que incluem as definições e aspectos metodológicos e práticos da robótica educacional, o uso de simuladores de robôs e os processos de avaliação existentes.

2.1 A Robótica Educacional

Pode-se definir a robótica educacional como um ambiente de aprendizagem composto por artefatos manipuláveis providos de sensores, motores, processadores e um *software* de computador. Denomina-se esses artefatos como robôs pedagógicos, que têm a função de interagir com o meio externo executando ações programadas (Zanatta 2013, Silva 2009).

O uso de robôs possibilita aos alunos demonstrar e provar suas próprias ideias, criando uma participação ativa entre alunos e professores (Okada & Santos 2014).

Segundo Zilli & others (2004), as principais vantagens da utilização da robótica educacional são:

- Desenvolver o raciocínio lógico e as habilidades manuais;
- Favorecer a utilização dos conceitos aprendidos na elaboração e execução de projetos;
- Estimular a investigação e a compreensão;
- Preparar o aluno para o trabalho em grupo;
- Promover a criatividade;
- Estimular o hábito do trabalho organizado;
- Reelaborar hipóteses a partir do erro; e
- Aplicar a teoria formulada em atividades práticas.

As aulas de robótica educacional podem envolver diversos temas, desde temas sobre ética e responsabilidade social até temas dos tradicionais componentes curriculares, sempre trabalhando de forma lúdica e atrativa aos alunos. Estas aulas

possibilitam que os problemas colocados pelo professor sejam discutidos em grupo, incentivando assim o trabalho colaborativo para a obtenção das devidas soluções. Tudo isso a partir de uma interação entre alunos, professores e recursos tecnológicos de *hardware* e *software*.

Uma aula de robótica educacional pode ser separada em quatro momentos distintos (Miranda 2006):

1. **Problematização:** é o momento de apresentação da atividade, apresentando uma situação-problema instigante para os alunos. A situação-problema pode envolver temas interdisciplinares ou sociais acessíveis aos alunos, dependendo de sua faixa etária. Podem ser introduzidos novos conceitos aos estudantes, desde que não seja muito discrepante ao seu nível de conhecimento;
2. **Exploração de potenciais soluções:** o professor incentiva um diálogo e interação entre os alunos, criando discussões que auxiliem o grupo na elaboração de estratégias para resolução do problema. O grupo deve utilizar as explicações do professor na aula, assim como o seu conhecimento prévio sobre robótica e programação;
3. **Desenvolvimento das soluções:** a partir das soluções encontradas no segundo momento, o grupo deve montar um robô capaz de resolver a situação-problema. Em seguida, eles devem criar um programa seguindo os passos escolhidos no segundo momento. A criação do programa pode ser desenvolvida em qualquer *software* de programação de robôs; e
4. **Análise dos resultados:** o grupo realiza testes para verificar se o robô montado e programado consegue resolver a situação-problema. Caso a montagem ou a programação do robô necessitem de ajustes, o grupo deve voltar para o terceiro momento. Caso a solução escolhida no segundo momento não apresente bons resultados, o grupo deve retornar ao segundo momento e iniciar uma nova discussão sobre possíveis soluções.

2.1.1 Ferramentas utilizadas em aulas de robótica educacional

Em uma aula de robótica educacional são utilizadas ferramentas de *hardware* e *software*. O *hardware* refere-se a um *kit* de robótica para construção do robô, enquanto que o *software* envolve uma ferramenta para a programação.

Kits de robótica são conjuntos de peças específicas que, juntas, são capazes de formar as partes físicas dos robôs. Estes *kits* podem possuir três funções diferentes:

- Apenas para montagem: são *kits* que permitem a montagem de diferentes robôs, mas que não podem ser programados.
- Apenas para programação: são *kits* utilizados para a criação de um modelo específico de robô, mas que permite a programação do mesmo.
- Para montagem e programação: são *kits* que permitem a montagem de diversos modelos de robôs, e que podem ser programados utilizando alguma linguagem de programação.

Com relação ao tipo de material utilizado para a criação dos robôs, os *kits* de robótica podem ser separados em três categorias distintas, a saber, os *kits* comerciais, os *kits* de sucata e os *kits* com componentes comerciais.

Os *kits* comerciais, como o Lego NXT e VEX ¹, apresentam, em sua maioria, uma melhor qualidade quando comparados aos *kits* das outras categorias (Miranda et al. 2011). No entanto, isso vem atrelado a um alto custo do produto. Este fator pode tornar esses equipamentos inacessíveis a determinados públicos.

Os *kits* de sucata são *kits* de robótica nos quais os materiais que montam a carcaça do robô são feitos de sucata. Na maioria dos casos, é utilizada sucata tecnológica. No entanto, grande parte destes projetos possuem limitações de *hardware*, oriundos da pequena quantidade de testes, ou da utilização de materiais de baixa qualidade. Por esse motivo, nem todos os *kits* de robótica com sucata possuem os requisitos necessários para serem utilizados em aulas de robótica educacional (Miranda et al. 2011, Gonçalves 2007). *Kits* de robótica desta categoria vem sendo menos utilizados, devido à facilidade na utilização de *kits* de melhor qualidade.

Por fim, existem os *kits* que utilizam apenas componentes eletrônicos e peças comercialmente disponíveis, como por exemplo o N-Bot, desenvolvido por Aroca (2012), no qual são usados componentes eletrônicos para fazer a parte de controle de baixo nível do robô (para ler sensores e mover atuadores) e peças (novas) que são geralmente utilizadas na fabricação de brinquedos (polias, rodas, engrenagens, eixos) e de móveis (rodas bobas, parafusos, suportes de metal), que podem ser facilmente adquiridas em lojas comerciais. Embora possuam semelhanças com os *kits* de sucata, estes são mais eficazes por não utilizarem material de sucata, o que aumenta a qualidade do robô, e atrelado a isso o custo. No entanto, eles apresentam uma maior complexidade, pois o usuário necessita de um conhecimento sobre eletrônica para o uso de tal ferramenta.

Convém ressaltar que, embora existam várias alternativas para os *kits* de robótica, poucos apresentam uma boa relação entre custo e facilidade de uso (Miranda et al. 2011). Este problema dificulta a adesão da robótica educacional nas escolas brasileiras.

Os *softwares* utilizados em aulas de robótica educacional são geralmente ambientes para o desenvolvimento de programas capazes de lidar com os atuadores (motores) e sensores de um robô, permitindo criar diferentes tipos de programação para diferentes tipos de robôs.

Por si só, o *software* não é suficiente para a realização de uma aula de robótica educacional. Dentre eles, podemos destacar o NXT-G, BricxCC e R-Educ (Pitta 2011, Sá 2013), utilizados para programar robôs do tipo Lego.

Embora muitos autores citem apenas os *softwares* de programação (Neves Júnior & others 2011, Santos & de Menezes 2005, Miranda et al. 2011), existem também os *softwares* de simulação, que podem ser utilizados em aulas de robótica educacional para simular a movimentação dos robôs e/ou até mesmo a sua montagem. Estes simuladores são capazes de simular o comportamento de robôs em um ambiente vir-

¹*Kits* de robótica comercialmente distribuídos e amplamente utilizados em aulas de robótica educacional.

tual que simula o ambiente real no qual ocorreria uma aula de robótica educacional. 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.

2.2 Simuladores robóticos

Como descrevemos na Seção 2.1, durante a realização de aulas de robótica educacional são efetuados testes que envolvem a construção e a programação do robô. Para diminuir o custo e o tempo de pesquisa no desenvolvimento de robôs, pesquisas desenvolvem e usam ferramentas próprias para prover uma maneira fácil e simples de testar ideias, teorias e programas com robôs sem depender fisicamente da máquina. Uma das maneiras de fazer isso é a partir de simuladores computacionais (Becker 2010, Obst & Rollmann 2004, Pedrosa 2010). Simuladores são ambientes computacionais que emulam o acontecimento de algum fenômeno real que os usuários conseguem manipular, explorar e experimentar.

Em algumas experiências, os testes se tornam extensos, sendo necessária a constante repetição dos testes com os robôs. Em outros casos, são necessárias várias modificações no modelo do robô, ou seja, o uso de diferentes tipos de carcaças, bases e sensores. A adaptação do robô e a contínua repetição dos testes podem aumentar o tempo para a resolução da atividade proposta, sem mencionar a troca de baterias da máquina. Simuladores podem ser utilizados nestes casos para facilitar e agilizar a realização dos testes necessários.

Os simuladores permitem que o usuário visualize e interaja em tempo real com o ambiente virtual, de forma que possa executar atividades assim como elas seriam executadas no ambiente real.

O ambiente virtual de simulação robótica é uma ferramenta poderosa que apresenta uma série de vantagens (Wolf et al. 2009, Michel 2004, Pedrosa 2010):

- Facilidade de criação do ambiente no qual o robô será inserido;
- Utilização de robôs de maior qualidade, visto que não há danos no robô virtual devido a uso extensivo, eliminando também erros devido ao uso de robôs avariados;
- Redução de danos ao robô real, em consequência da realização de parte dos testes com o robô virtual, garantindo que quando o mesmo seja utilizado apresentará menos falhas;
- Economia de recursos financeiros, dado que o custo de um simulador é menor do que o custo de *kits* de robótica, além da economia de baterias;
- Economia de tempo, considerando que a realização de testes no ambiente virtual é mais rápida do que a realização de testes no ambiente real;
- Facilidade nos testes de novos algoritmos e modelos de robôs; e
- Facilidade nos testes com vários robôs.

Simuladores podem ser utilizados de duas formas em aulas de robótica educacional:

1. Aulas de robótica utilizando apenas simuladores: neste caso todos os testes são realizados no robô virtual. Embora permita concluir a atividade em menor tempo, e apresentar um menor custo, exclui os benefícios que o uso de *kits* de robótica traz aos alunos.
2. Aulas de robótica utilizando simuladores e *kits* de robótica: neste cenário, o simulador pode ser utilizado na realização dos testes iniciais, auxiliando na escolha do robô, dos sensores, dos atuadores, e na criação do código de programação capaz de realizar a atividade proposta. Quando estes pontos forem definidos, o aluno pode utilizar o robô real, dado que a maior parte dos testes já foi realizada, sendo necessário apenas pequenas modificações no modelo do robô e no programa, previamente definidos.

Além disso, o uso de simuladores permite que o aluno exercite os conhecimentos adquiridos em sala de aula com a realização de atividade extra-classe, ou que possa continuar a atividade proposta em sala de aula em casa. Sem o uso de simulador, o aluno deve possuir um *kit* de robótica para poder estudar e praticar em casa, o que não representa a realidade de todos os alunos de robótica educacional.

Este tipo de tecnologia já está sendo utilizada em algumas competições de robótica, como a RoboCup (*RoboCup 2016* n.d.). Nesta competição existem quatro categoriais que competem em ambientes simulados: *Robocup Soccer 2D Simulation League*, *Robocup Soccer 3D Simulation League*, *Rescue Simulation League* e *RoboCupJunior - CoSpace Demo Challenge*.

Encontram-se na literatura algumas iniciativas de desenvolvimento de ambientes de programação específicos para a área educacional e utilizados em aulas de robótica educacional. Alguns trabalhos, como os de Benitti et al. (2009) e Gomes et al. (2008), citam o uso de simuladores robóticos em aulas de matemática e geografia. Os autores citam o estímulo dos alunos no desenvolvimento do conhecimento e um ganho em conhecimento sob o aspecto tecnológico, de inclusão social e digital, além do estímulo dos professores com relação ao desenvolvimento das aulas utilizando simuladores robóticos para participar de seu cotidiano.

Em testes realizados com o simulador desenvolvido por esta equipe, o S-Educ (Fernandes 2013), alunos e professores indicaram que o simulador representa muito bem a movimentação do robô real, e apresentaram interesse no uso de simuladores em aulas de robótica educacional, indicando que teriam maior interesse em aulas de robótica que utilizassem o simulador para a realização de testes, e robôs reais para executar as atividades testadas no simulador.

Embora apresentem diversas vantagens quando comparados aos robôs reais, os simuladores não conseguem substituir por completo a utilização de *kits* de robótica. A montagem de robôs reais é uma forma lúdica de aperfeiçoar a coordenação motora e a criatividade, além de ser mais atrativa aos alunos do que a montagem de um robô virtual. Além disso, nem todos os simuladores são capazes de simular todos os movimentos desenvolvidos por um robô. Ou seja, mesmo apresentando algumas desvantagens, os simuladores podem ser utilizados com sucesso em aulas de robótica educacional.

2.3 Processos de avaliação

Um dos maiores desafios na educação é estabelecer metodologias de avaliação que envolvam todo o processo de ensino e aprendizagem de forma sistemática, contínua e abrangente. A avaliação não acontece em momentos distintos do processo de ensino, mas permeia todo o trabalho, sendo finalizada ao final do mesmo (Chueiri 2008). Avaliar é uma tarefa didática essencial, que consiste em coletar dados, qualitativos e/ou quantitativos de critérios baseado em uma escala de valores (Santos 2006, Buriasco et al. 2009).

O ato de avaliar é benéfico e inevitável durante o processo de ensino. Benéfico por permitir uma análise do nível de aprendizagem dos alunos, e inevitável porque é essencial ao professor, ao estar em sala de aula, reconhecer os conhecimentos dos alunos, com base em determinados critérios.

De acordo com Boggino (2009), através da avaliação é possível que os educadores identifiquem as teorias e hipóteses propostas pelos alunos na resolução dos problemas apresentados a eles e, a partir daí, identifiquem o entendimento dos alunos acerca dos saberes a eles apresentados. Sistemas avaliativos permitem comparar resultados obtidos com uma representação ideal. A construção desta representação ideal consiste na identificação dos pontos que representam a aprendizagem dos alunos (Pacheco 1998, Santos 2003).

No entanto, avaliar só tem sentido se tiver como principal objetivo buscar novos caminhos para a aprendizagem. Possuindo informações sobre o conhecimento adquirido pelos alunos, é possível que os professores realizem intervenções pedagógicas, ajustando suas técnicas de ensino com o objetivo de aumentar progressivamente a compreensão dos alunos acerca dos temas tratados em sala de aula, diminuindo a distância entre os alunos e os conteúdos curriculares.

Devido às mudanças ocorridas no sistema educacional nos últimos tempos, os sistemas de avaliação tradicionais devem ser adaptados, de forma a atender as propostas de prática educativa modernas. As práticas avaliativas devem responder às exigências do sistema educativo como um todo. A adequação do sistema avaliativo aos novos parâmetros do sistema educacional permitem um *feedback* maior com relação à evolução dos alunos em sala de aula (Barbosa & Neves 2006).

Boggino (2009) cita a importância do professor ir além dos resultados, caracterizando os tipos de erros, de forma a criar novas metodologias de ensino que permitam apoio pedagógico nas áreas de maior deficiência dos alunos. No entanto, a análise destes pontos pode atrasar o sistema avaliativo, o que pode se tornar um obstáculo ao estudo dos dados obtidos.

Em qualquer processo de avaliação em ambientes de ensino, o professor é a figura responsável por identificar os pontos a serem avaliados, elaborar um sistema de avaliação, atribuir um sistema de contagem, levando em consideração o rendimento dos alunos, e realizar as modificações que julgar necessárias em sala de aula.

De acordo com Queiroz (2010), a avaliação da aprendizagem tem duas funções: social e pedagógica. Com a função de caráter social, a avaliação informa os estudantes sobre o seu rendimento, podendo ser utilizada para selecionar e classificar

os alunos em um ambiente de ensino. Através de certificados de desempenho os alunos são categorizados, sem nenhuma finalidade pedagógica. Com a função de caráter pedagógico, a avaliação gera informações úteis ao educador para adaptação das atividades às necessidades dos alunos, com o objetivo de melhorar a qualidade do ensino.

2.3.1 Modalidades do processo avaliativo

A avaliação do ensino-aprendizagem deve ocorrer antes, durante e depois do processo de aprendizagem, podendo se encaixar em três modalidades específicas: diagnóstica, formativa e somativa (Santos 2006, Kraemer 2005, Queiroz 2010).

A avaliação diagnóstica, também conhecida como avaliação preditiva ou inicial, refere-se àquela normalmente realizada no início de um curso com o objetivo de identificar as capacidades dos alunos em uma determinada área, permitindo traçar um perfil dos alunos. Esta avaliação permite constatar possíveis problemas de aprendizado e auxiliar o professor na criação de estratégias de ensino que melhor o auxiliem em sala de aula.

Este tipo de avaliação permite verificar a posição do aluno face às novas aprendizagens que serão propostas e a aprendizagens anteriores que servem de base aos novos saberes.

Esta modalidade de avaliação pode ser aplicada em qualquer momento do curso no qual será iniciada uma nova unidade com novos conteúdos, embora seja geralmente aplicada no início do curso.

A avaliação formativa refere-se à análise que ocorre durante todo o processo de ensino-aprendizagem, verificando o rendimento do aluno, identificando possíveis deficiências de aprendizagem e buscando as mudanças necessárias para atingir os objetivos previamente definidos. Representa a forma a partir da qual o estudante conhece seus erros e acertos.

Este tipo de avaliação é um veículo de informação tanto para o aluno quanto para o professor. Para o aluno serve como um estímulo de estudo, apresentando informações sobre o andamento do curso, servindo como orientação para identificar deficiências e reformular trabalhos. Para o professor possibilita a identificação de deficiências na forma de ensinar, permitindo reformulações no trabalho didático. A frequência com a qual estas análises são realizadas possibilitam um melhor ajuste do processo de ensino-aprendizagem.

Trata-se de uma avaliação que tem a função de ajustar o processo de ensino-aprendizagem, buscando agir sobre os pontos frágeis da aprendizagem.

Por fim, a avaliação somativa refere-se àquela realizada, em geral, no fim do curso, com o objetivo de quantificar o aproveitamento do aluno, seguindo critérios previamente imposto. Este tipo de avaliação permite determinar o grau de domínio do aluno, aferindo resultados à aprendizagem.

Esta avaliação pode ser usada também para classificar os alunos, proporcionando uma evolução em níveis de conhecimento, apresentando uma função social de comparar o desempenho dos alunos com as exigências do sistema educacional. Neste

ponto, os alunos devem se adequar ao ensino, e não o oposto.

Este tipo de avaliação corresponde a um balanço final, obtendo indicadores que permitam uma melhor análise do processo de ensino-aprendizagem, apontando uma função formativa.

De acordo com Fernandes (2011), o entendimento da funcionalidade de cada tipo de avaliação permite extinguir ambiguidades e concepções errôneas existentes acerca do assunto, como, por exemplo, a concepção de que a avaliação somativa é objetiva e sempre quantitativa, enquanto que a formativa é subjetiva e qualitativa. Podemos citar também a visão de que a avaliação somativa é rigorosa e a avaliação formativa não é. O melhor conhecimento dos tipos de avaliação permite a melhor prática de avaliação em sala de aula.

Santos (2006) e Romeiro (2000) citam a importância de aplicar todas as modalidades do processo de avaliação, não de forma independente, mas de forma complementar, garantindo que os resultados obtidos em uma fase sirvam como auxílio para a aplicação da fase seguinte. Desta forma, começaríamos pela aplicação da avaliação diagnóstica, que garante dados que podem ser utilizados para projetar os passos da avaliação formativa, sendo ela executada durante todo o processo de aprendizagem.

Ao fim do curso ou de uma unidade, os dados da avaliação formativa servem de base para a aplicação da avaliação somativa, analisando os critérios de rendimento estabelecidos. No entanto, o autor relata que esta forma completa de avaliar raramente é aplicada, passando direto para uma avaliação somativa classificatória.

2.3.2 Avaliação automática do ensino

O sistema educacional passou por transformações pedagógicas importantes, e sendo assim os sistemas tradicionais de avaliação não atendiam às metodologias de ensino e às exigências curriculares (Queiroz 2010).

À medida que a quantidade de alunos em sala de aula cresce, a quantidade de trabalho relacionado à avaliação deve ser minimizado de alguma forma, gerando um prejuízo de aprendizado e de retorno (*feedback*) por parte do aluno (Ihantola et al. 2010).

Neste contexto, surgiu a necessidade da criação de sistemas computacionais que permitem realizar a avaliação dos alunos de forma automática, permitindo a avaliação de um grande número de alunos ou de um grande número de questões, fornecendo um *feedback* em curto prazo para que o estudante possa aperfeiçoar sua solução. A avaliação automática tem como objetivo o auxílio ao docente no processo de verificação do conhecimento adquirido pelo estudante (Moreira & Favero 2009, Caldas & Favero 2009, Montero et al. 2014).

A aplicação de avaliações automáticas acarreta benefícios ao processo de ensino-aprendizagem como menor esforço do docente, melhor administração dos estudantes, através da análise de relatórios avaliativos, maior tempo de prática e mais tempo para contato com os estudantes.

Vale ressaltar que o uso de avaliações automáticas feitas a partir de um sistema computacional não exclui o professor do processo avaliativo. Os sistemas automáti-

cos de avaliação funcionam a partir de dados inseridos pelo professor responsável, e fornecem resultados que podem ser interpretados pelo professor para então fornecer a avaliação conclusiva.

A avaliação automática pode permear todas as modalidades de avaliação vistas neste Capítulo, desde a avaliação diagnóstica até a avaliação somativa. A sua função vai depender de como o professor irá aplicá-la e como irá analisar seus resultados.

Podemos encontrar na literatura diversos trabalhos a respeito de sistemas utilizados para avaliação automática, sendo eles aplicados a diversas áreas de ensino. Na seção 3.2 mostraremos alguns sistemas e suas aplicações, comparando-os com o sistema desenvolvido neste trabalho.

Capítulo 3

Trabalhos relacionados

Os sistemas de avaliação são utilizados na educação com o propósito de auxiliar o professor a quantificar o conhecimento dos seus alunos nos temas abordados em sala de aula. Encontramos na literatura diversas formas diferentes de avaliar alunos, sendo algumas bastante subjetivas, enquanto que outras utilizam apenas valores quantitativos para efetuar a avaliação.

Neste Capítulo apresentamos alguns sistemas de avaliação utilizados na educação, mostrando os benefícios oriundos da utilização de sistemas automáticos de avaliação, buscando realizar uma comparação com a metodologia avaliativa desenvolvida neste trabalho, enfatizando nesta comparação as vantagens e desvantagens de tais sistemas.

3.1 Sistemas de gestão acadêmica

Os sistemas de gestão acadêmica são aqueles comumente utilizados em instituições de ensino, sejam elas escolas ou universidades. Estes sistemas são responsáveis pelo compartilhamento de dados, visualização de notas, entre outras funcionalidades. Estes sistemas possuem, em geral, uma área na qual o professor é capaz de cadastrar as notas dos alunos para definir suas notas ao fim do curso.

Dentre estes sistemas podemos destacar o SIGAA, um sistema de informação *web* corporativo que informatiza os procedimentos da área acadêmica através de módulos que envolvem desde o ensino infantil até o ensino da pós-graduação (Barroca et al. 2013). Através de turmas virtuais, o sistema permite que professores compartilhem informações com os alunos de uma turma, disponibilizando material, realizando atividades *online*, entre outras funcionalidades. Dentre as instituições de ensino que utilizam este sistema, podemos destacar a Universidade Federal do Rio Grande do Norte, instituição na qual o sistema foi desenvolvido.

Para critério de avaliação, o SIGAA permite que os professores insiram no sistema as notas dos alunos que serão utilizadas para calcular a nota final. O sistema realiza uma média das notas do aluno dependendo dos parâmetros estabelecidos pela instituição de ensino. No entanto, as notas dos alunos devem ser atribuídas manualmente pelo professor, e o sistema é capaz apenas de utilizá-las para definir a nota final.

O ActiveSoft é outro exemplo de sistema de gestão acadêmica, utilizado em diversas escolas (*ActiveSoft* n.d.). O sistema permite a criação de turmas, cadastramento de dados dos alunos, fornecimento de relatórios para os professores, que envolvem diário de classe e lista de chamadas, entre outras funcionalidades.

O sistema permite também uma avaliação por nota ou conceito, na qual o professor indica as notas do aluno e o sistema calcula a nota final do mesmo, seja semestral, bimestral, etc. Além disso, o professor pode visualizar um *ranking* dos alunos e um gráfico de rendimento por turma, série ou disciplina.

O Moodle é uma plataforma de aprendizagem utilizada em todo o mundo, comumente aplicada a cursos de ensino à distância (*Moodle* n.d.). O Moodle possui uma série de funcionalidades, dentre as quais podemos destacar o acesso a aulas inseridas pelos docentes, conversas em chats e fóruns, e o controle do acesso dos utilizadores.

Com relação à avaliação dos alunos, ela pode ser constituída por trabalhos enviados, comentários em atividades e respostas aos questionários inseridos na plataforma.

Estes exemplos de sistemas de gestão acadêmica demonstram uma realidade nos sistemas tradicionalmente utilizados em instituições de ensino: o professor é responsável por atribuir as notas aos alunos e inseri-las nos sistemas computacionais, enquanto que os sistemas apenas a utilizam para definir as notas finais dos alunos. Este tipo de ferramenta, embora bastante útil, ainda requer muito do professor, e em turmas grandes o processo de avaliação pode demandar muito tempo. Muitos outros similares existem no mercado, geralmente desenvolvido para facilitar apenas o processo de sintetização das notas bem como de outras atividades necessárias no dia a dia acadêmico.

Buscamos na literatura sistemas computacionais que facilitam o processo de avaliação através da realização de uma avaliação automática, realizada através de parâmetros pré-definidos, auxiliando o professor em sala de aula e fornecendo um *feedback* mais rápido aos alunos.

3.2 Sistemas automáticos de avaliação

Como descrevemos na Seção 2.3.2, os sistemas automáticos de avaliação surgiram a partir da necessidade de agilizar o processo de avaliação com o auxílio de um sistema computacional. Estes sistemas avaliam o desempenho do aluno a partir de dados previamente adicionados no sistema pelo professor responsável, fornecendo dados que podem ser utilizados pelo professor para analisar o desempenho dos alunos.

Na literatura é possível identificar vários tipos de sistemas automáticos de avaliação, cada um voltado para uma área específica. Seguindo o propósito da utilização destes sistemas na área de ensino, separamo-os em duas subcategorias: sistemas para avaliação de questões de múltipla escolha ou discursivas e sistemas para avaliação de códigos computacionais.

3.2.1 Avaliação de questões de múltipla escolha ou discursivas

Com o objetivo de amenizar os problemas de avaliação, principalmente no sistema de educação à distância, surgem na literatura diversos sistemas automatizados de correção de questões de múltipla escolha e questões discursivas. No entanto, as pesquisas sobre avaliação automática obtêm melhores resultados ao se tratar de questões fechadas, objetivas (Rocha 2007, Santos 2006).

Com relação à aplicação de avaliação automática em questões objetivas, destacamos o trabalho de Zampiroli et al. (2016), no qual foi desenvolvido um sistema de geração de diferentes versões de testes de múltipla escolha, seguido de uma avaliação automática das respostas dos alunos.

O sistema utiliza imagens de celular para escanear formulários de resposta. Neste projeto, o professor escaneia um formulário com as respostas corretas e em seguida escaneia as respostas dos alunos. Através de um processamento de imagens, o sistema consegue identificar o aluno ao qual aquele teste pertence, a classificar suas respostas em correta, incorreta ou inválida. Após a avaliação automática os resultados são exportados para um arquivo CSV.

Nguyen et al. (2011) desenvolveu um sistema similar, obtendo bons resultados com relação à classificação das respostas dos alunos.

Nesta mesma área encontramos sistemas distribuídos comercialmente, como o *Multiple Choice Scanner* (*Multiple Choice Scanner* n.d.), um aplicativo de celular que utiliza fotos retiradas dos formulários de resposta para avaliar os alunos. O *QCM Direct* (*QCM direct* n.d.) segue o mesmo método de avaliação baseado na análise das imagens, mas neste caso é um sistema online, que fornece também um conjunto de estatísticas gerais e por indivíduo a respeito dos resultados obtidos.

Ainda a respeito de sistemas voltados para questões de múltipla escolha, destacamos os sistemas para criação de testes online, nos quais o professor cadastra questões de múltipla-escolha que serão respondidas pelos alunos. O sistema é responsável por avaliar os alunos fornecendo os resultados finais para o professor.

Paiva (2016) desenvolveu um sistema de competições no qual o professor cria torneios com questões objetivas, e os alunos podem resolver questões em equipe ou individualmente. O sistema, chamado de Gamimpíada, adiciona características de jogos para criar um ambiente mais atrativo aos alunos. Ao fim das competições, o aluno recebe acesso às questões respondidas, podendo estudar através das questões aplicadas, e o professor passa a visualizar relatórios de rendimento dos alunos através de gráficos de pontuação. Foram realizados testes com professores e alunos do ensino superior, e os resultados mostraram que ambos apresentam interesse no uso deste sistema em sala de aula.

O *ClassMarker* (*Class Marker* n.d.) é um serviço de testes online, focado para empresas e escolas, no qual é criado um quiz com perguntas de múltipla escolha, e que fornece informações sobre os resultados dos participantes incluindo pontuação, duração total do quiz, além de estatísticas sobre as questões aplicadas, como percentagem de acertos. O sistema é online e gratuito, facilitando a criação de testes

de múltipla escolha e sua aplicação para grandes turmas.

Embora tenhamos encontrado diversos sistemas focados à avaliação automática de questões objetivas, sabemos que tais questões não demonstram como o aluno construiu o conhecimento e o quanto ele entende sobre tal assunto. Sendo assim, a aplicação de questões abertas, ou discursivas, demonstram de maneira mais clara o raciocínio do aluno. No entanto, criar tal sistema é mais desafiador, pois requer compreender a linha de raciocínio do estudante, e não somente o resultado final obtido (Caldas & Favero 2009).

Lin & Hovy (2003) sugere a análise de similaridade entre conceitos utilizando a técnica de N-gramas, na qual é medida a proximidade entre a resposta do aluno e a resposta de referência. Os resultados obtidos nos testes classificam a ferramenta desenvolvida como um bom sistema de avaliação automática.

Santos Jr et al. (2004) também desenvolveu um sistema para computar a similaridade entre dois textos. A técnica utilizada por ele foi a da medição do cosseno de similaridade. No entanto, o autor descreve possíveis problemas nos resultados da avaliação quando o aluno utiliza um vocabulário não comumente utilizado, e portanto não presente nos documentos originais passados como parâmetro.

Concluimos então que existem diversos sistemas para avaliação automática de testes com questões objetivas, todos obtendo bons resultados. Visto que este tipo de avaliação não é suficientemente completa para avaliar o conhecimento dos alunos, surgiu a necessidade de criação de sistemas para avaliação de questões discursivas. Nesta área, as pesquisas ainda apontam para a necessidade de criação de técnicas mais completas.

3.2.2 Avaliação de códigos de programação

É muito importante dar aos alunos a oportunidade de praticar e de resolver exercícios de programação por si mesmos. Porém a eficácia máxima desta aproximação exige a capacidade do docente para avaliar e comentar cada resolução. O *feedback* imediato possibilita aos alunos aprender com seus erros, e construir novas teorias para o desenvolvimento das questões aplicadas. Contudo, em turmas grandes e com poucas horas esta atitude é impraticável, pois demanda do professor um tempo demasiado, correndo o risco dele não conseguir atender a toda a turma (Tavares et al. 2015).

Para atenuar este problema, existe um grande número de sistemas que permitem realizar a avaliação automática de questões de programação, auxiliando o aluno e o professor em aulas básicas de programação.

Ihantola et al. (2010) realizou uma pesquisa sobre sistemas com esta funcionalidade. De acordo com seus resultados, a maioria dos sistemas tem suporte para Java, C/C++, Python e Pascal, embora existam alguns que analisem códigos desenvolvidos em uma linguagem de programação independente.

Para analisar o código dos alunos, podem ser utilizadas algumas técnicas de comparação. Podemos separá-las em duas categorias: técnicas dinâmicas e técnicas estáticas.

As técnicas dinâmicas se baseiam em testes, separando um conjunto de entradas para um algoritmo e fazendo a correspondência com um determinado conjunto de saídas. Então, são executados os testes que verificam se para dada uma entrada a sua saída esperada é obtida. Com base nisso, o sistema é capaz de avaliar e atribuir notas para os exercícios. A maioria dos sistemas de ensino-aprendizagem de programação aplicam esta técnica (Moreira & Favero 2009).

As técnicas dinâmicas de avaliação são utilizada em competições de programação (*Maratona de Programação* n.d.), assim como em outros sistemas, como o ProgTest (Souza et al. 2012).

Na aplicação deste tipo de técnica, as entradas e saídas utilizadas como teste devem ser cuidadosamente escolhidas, para não comprometer a qualidade da correção automática.

Quando aplicada a avaliação por testes, assim que o resultado desenvolvido pelo aluno é obtido, o estudante abandona a solução e vai para o próximo problema, mesmo que esta seja uma solução com alta complexidade ou que não utilize os elementos de programação que o professor julgue necessário.

De acordo com Mota et al. (2009), os resultados obtidos utilizando apenas esta técnica são insuficientes para realizar inferências sobre o aprendizado de habilidade de programação.

Neste contexto, surgem as técnicas estáticas, capazes de avaliar o código desenvolvido, realizando comparações acerca dos elementos de programação aplicados, comparando o código com uma resposta-modelo cadastrada pelo professor, supostamente ideal.

Moreira & Favero (2009) sugere a aplicação de duas técnicas de análise de códigos: utilizando métricas de engenharia de software e utilizando N-gramas. No primeiro caso, são feitas comparações entre os códigos analisando a quantidade de uso de funções da linguagem, o número de palavras reservadas, o número de declarações, o número de linhas do programa, entre outras métricas.

No segundo caso, são extraídos pedaços de caracteres do código, e são feitas comparações entre os objetos identificados e os objetos extraídos da solução ideal. Dependendo da quantidade de N-gramas que os textos compartilham, podemos dizer que eles são similares. Os nomes das variáveis e das classes são igualadas antes de se aplicar o método dos N-gramas.

Com o objetivo de criar uma avaliação mais abrangente, Pelz et al. (2012) criou um mecanismo de avaliação automática composto por quatro etapas: verificação sintática, verificação de comandos de programação considerados obrigatórios, verificação da similaridade estrutural entre o programa do aluno e a solução proposta pelo professor, e comparação entre as saídas obtidas nos dois códigos quando apresentado um conjunto de entradas.

Para verificação da similaridade estrutural, Pelz et al. (2012) utilizou o método de serialização das estruturas dos programas, em que o programa é representado como uma árvore sintática abstrata, sendo cada nó da árvore uma estrutura de programação diferente. As árvores geradas pelo código do aluno e do professor são em seguida comparadas utilizando um método de distância, para determinar um

coeficiente de similaridade.

O autor sugere que os problemas tenham várias soluções gabaritos, para que a solução do aluno seja comparada com todas estas soluções. O maior coeficiente de similaridade obtido representa a nota do aluno.

O trabalho de Jesus (2010) segue o mesmo princípio de Pelz et al. (2012), separando a avaliação em quatro etapas, de forma a garantir melhores resultados.

Ainda no contexto de avaliação de códigos, existem jogos computacionais que são utilizados como ferramentas didáticas no ensino de programação introdutória. Podemos destacar o *Game2Learn* (Barnes et al. 2007), no qual o aluno deve criar códigos para resolver as questões propostas, que incluem resgatar uma princesa, lutar contra um inseto, entre outros desafios.

O *RoboCode* (*RoboCode* n.d.) consiste em uma arena em que os tanques devem destruir outros tanques, todos eles programados por alunos. O fato do jogo estabelecer uma competição entre os alunos motiva-os a querer jogar cada vez mais, sempre melhorando seus algoritmos, e portanto praticando lógica de programação.

Vale destacar também o *LightBot* (*LightBot* n.d.), um site em que são utilizados comandos de programação para mover um robô em uma arena. As diferentes fases do jogo requerem a utilização de diferentes estruturas de programação, como laços de repetição e funções.

3.3 Sistemas de avaliação para robótica educacional

Como descrito na Seção 2.1, a robótica educacional é um ambiente de aprendizagem onde o aluno utiliza *kits* de robótica em aulas interdisciplinares, estimulando a criatividade e o pensamento investigativo.

Os *kits* de robótica são utilizados para a montagem de robôs que podem ser controlados ou programados, com o propósito de resolver uma situação problema proposta pelo professor. A temática envolvida nesta situação problema pode incluir temas como história, matemática, biologia, ou até mesmo temas não relacionados a disciplinas curriculares, como inclusão social, meio ambiente, entre outros.

Algumas escolas já oferecem aulas de robótica educacional, sendo na grade curricular ou extracurricular, como a Escola Municipal São Miguel (Curitiba/PR), Escola Estadual São Luis (Santa Maria de Jetibá/ES), Escola Estadual Escritor José de Alencar (Paulista/PE), entre outras. Existem também cursos de robótica independentes, focados ao ensino de robótica e programação, como o RoboEduc ¹ e o SuperGeeks ².

Ao ministrar aulas de robótica, deve-se ter cuidado com relação à avaliação do desempenho dos alunos. Por se tratar de tecnologias consideravelmente modernas,

¹Curso de robótica educacional, voltado para crianças e adolescentes, focado para o ensino multidisciplinar.

²Cursos voltado para o ensino da programação para crianças e adolescentes, que utiliza também a robótica educacional como ambiente de aprendizagem.

o uso de técnicas de avaliação tradicionais podem não se aplicar corretamente neste caso.

Ao buscar sistemas utilizados em aulas de robótica educacional, não encontramos técnicas de avaliação focadas para a robótica educacional, levando em consideração suas particularidades.

No trabalho de Frangou et al. (2008), o autor descreve uma metodologia para aplicação de aulas de robótica educacional dividida em cinco estágios: estágio inicial, onde é feito o estudo de um problema, estágio de exploração, com a introdução de novas tecnologias ou conhecimentos, estágio de investigação, para análise do problema proposto, estágio de criação, onde os alunos propõem soluções para o problema apresentado, e por fim o estágio de avaliação.

Neste último estágio, os alunos são avaliados a partir de uma apresentação do seu trabalho seguida de *feedbacks* dados pelo professor e pelos outros alunos da sala.

Em seus resultados, Frangou et al. (2008) relata que este tipo de avaliação é atrativa aos alunos, podendo resultar em novas ideias e perspectivas de desenvolvimento.

Chiou (2012) utiliza a robótica educacional para motivar alunos em aulas de matemática e ciências. Ele utiliza métodos tradicionais de avaliação, em que os alunos resolvem a situação problema e a pontuação é definida através da avaliação subjetiva do professor.

Embora não seja voltado para a robótica educacional, Jesus (2010) desenvolveu uma espécie de jogo, no qual os alunos devem criar programas para resolver desafios propostos. Em sua interface foi utilizado um modelo robótico, mais atrativo para os alunos. Os desafios propostos foram elaborados para abranger os tópicos de programação introdutória, como estruturas de condição, estruturas de repetição e matrizes.

A linguagem de programação utilizada é uma linguagem não-textual, com componentes gráficos que podem ser arrastados e encaixados uns nos outros.

A cada desafio completado, o aluno tem acesso ao próximo. O desafio é considerado completado quando é executado e nenhum erro é encontrado pelo mecanismo de correção automática. Para a correção automática dos desafios são verificados quatro aspectos do programa: sintaxe, verificação das saídas, comparação da estrutura do programa do aluno com um conjunto de estruturas consideradas como solução para o problema, e verificação de estruturas de programação obrigatórias.

Para estas análises foram utilizadas técnicas semelhantes às utilizadas por Pelz et al. (2012), descritas na Seção 3.2.2, em que são utilizadas árvores sintáticas para a comparação das estruturas do programa.

Os resultados obtidos por Jesus (2010) mostram que os alunos que utilizaram o jogo para praticar exercícios de programação apresentaram um desempenho ligeiramente melhor que os alunos que não utilizaram o jogo. O autor descreve que o uso de ferramentas que se assemelham a jogos trazem benefícios aos alunos, incluindo melhores resultados na aprendizagem e na motivação.

Neste trabalho desenvolvemos uma metodologia de avaliação automática para aulas de robótica educacional. O professor é responsável por criar atividades em

um simulador robótico e cadastrar métricas para avaliação que serão utilizadas para determinar as pontuações dos alunos.

Embora não tenhamos encontrado ferramentas que apresentem o mesmo propósito, conseguimos identificar importantes parâmetros de avaliação utilizadas nos sistemas vistos neste Capítulo.

Frangou et al. (2008) destaca a importância da avaliação subjetiva do professor, que surge a partir da apresentação do trabalho feita pelos alunos e do debate em sala de aula. Jesus (2010) e Pelz et al. (2012) sugerem o uso de diferentes métricas de avaliação de códigos de programação, permitindo que a avaliação do código seja avaliada não somente pela execução de testes, mas também pela forma com a qual o código foi desenvolvido.

A partir do entendimento destes pontos, realizamos uma análise dos principais tópicos de avaliação de uma aula de robótica educacional, que serão mais discutidos no Capítulo 4.

Capítulo 4

Solução para Avaliação em Robótica Educacional

A prática da robótica educacional em sala de aula tem se tornado cada vez mais frequente, e seu uso como ferramenta de auxílio ao ensino de temas interdisciplinares vem sendo comprovada como eficaz. No entanto, percebemos que a aplicação de aulas de robótica educacional é normalmente associada a aulas extra-curriculares ou optativas.

Em nossas pesquisas, identificamos a falta de recursos avaliativos que auxiliem o professor na avaliação de tais aulas. Constatamos então a necessidade de metodologias avaliativas específicas para a robótica educacional, que levem em consideração as particularidades deste tipo de aula.

Desenvolvemos uma metodologia de avaliação de aulas de robótica educacional, aplicada com o auxílio de um simulador robótico, que permite que os professores definam os critérios de avaliação que serão utilizados e analisem os dados sobre a evolução dos alunos e da turma.

Neste Capítulo, explanaremos sobre as técnicas de avaliação que compõem esta metodologia avaliativa, o motivo pelo qual foram escolhidas e como a avaliação é realizada.

4.1 Avaliação de aulas de robótica educacional

A robótica educacional está cada vez mais sendo inserida em escolas e em cursos no Brasil. Seus benefícios incluem não apenas o hábito do uso de tecnologias em ambiente escolar, mas também estimula o raciocínio lógico, o pensamento investigativo, e a criatividade dos alunos.

No entanto, ao utilizar a robótica em sala de aula o professor se depara com um problema: como será feita a avaliação dos alunos nas aulas de robótica educacional?

Na Seção 2.3 apresentamos metodologias de avaliação utilizadas no processo de ensino-aprendizagem. Destacamos as modalidades diagnóstica, formativa e somativa.

A modalidade diagnóstica refere-se à análise do conhecimento dos alunos, geralmente realizada no início de um curso ou de uma unidade. A modalidade for-

mativa refere-se à avaliação contínua realizada durante todo o processo de ensino-aprendizagem, gerando um *feedback* continuado ao aluno, permitindo intervenções do professor e do aluno durante o curso a partir dos resultados obtidos. Por fim, a modalidade somativa refere-se à avaliação realizada normalmente ao fim do curso, quantificando o conhecimento do aluno, permitindo uma evolução do mesmo em níveis.

A aplicação dos três tipos de avaliação apresenta muitos benefícios ao processo de ensino-aprendizagem, permitindo uma análise geral do aluno, ao invés de análises pontuais, possibilitando também modificações do ensino no decorrer do curso.

A prática de tais modalidade de avaliação deve se moldar ao tipo de curso, através do uso de metodologias diferentes dependendo do conteúdo que está sendo ministrado. Contudo, não encontramos na literatura metodologias de aplicação das modalidades de avaliação em aulas de robótica educacional.

Assim, o problema de avaliação em aulas de robótica educacional persiste: que modalidades de avaliação podem ser aplicadas, o que elas devem avaliar, e como os dados obtidos podem auxiliar o professor a ajustar a forma de ensino?

A proposta deste trabalho é criar uma metodologia de avaliação focada para aulas de robótica educacional, considerando as particularidades deste tipo de aula, que envolvem desde a análise do robô utilizado até o desenvolvimento do programa utilizado para controlá-lo.

4.1.1 Parâmetros de avaliação

Como descrito na Seção 1.1, nossa equipe trabalha com robótica educacional desde 2006, ministrando aulas e desenvolvendo tecnologias na área focadas para alunos desde o nível fundamental até o nível médio. Considerando o conhecimento da equipe na área, e a partir de pesquisas com professores de robótica educacional, definimos alguns parâmetros que devem ser avaliados em uma aula de robótica educacional:

1. Montagem do robô
2. Programação do robô
3. Execução da atividade

O primeiro tópico a ser levado em consideração é o robô utilizado pelo aluno. Ao se deparar com uma situação problema, o aluno deve definir que tipo de robô irá utilizar para executar a atividade proposta, estabelecendo o tipo de base, os sensores e os atuadores que o robô vai possuir.

Diferentes tipos de bases permitem que os robôs se movimentem de formas distintas, com alteração em sua velocidade, no atrito presente ao realizar curvas, e na forma como se comporta em terrenos irregulares.

Os sensores podem ser de vários tipos, sendo cada um utilizado para uma função diferente. Focaremos apenas nos sensores utilizados pelo robô do tipo Lego, o mais utilizado comercialmente. Sensores de toque permitem identificar objetos desde que

aconteça o contato entre o robô e o objeto. Os sensores ultrassônicos também permitem identificar objetos, mas esta identificação é feita à distância, sem a necessidade de contato. Por fim, sensores de cor permitem identificar cores no ambiente, desde que o sensor esteja a uma pequena distância do objeto que será reconhecido.

Dependendo da atividade, o aluno deve ser capaz de identificar que tipo de sensor precisará utilizar. Por exemplo, vamos analisar uma atividade na qual o robô deve caminhar seguindo uma linha preta no chão e desviar de obstáculos leves que possam existir em cima da linha.

O aluno deve perceber que para seguir uma linha no chão o robô deve possuir um sensor de cor virado para baixo. Sendo assim, ele pode optar por utilizar um ou dois sensores de cor, sabendo que o uso de dois sensores facilitaria a programação do robô. Com relação à detecção do obstáculo, o aluno deve notar que ao se tratar de um obstáculo leve, o sensor de toque não deve ser utilizado, pois pode ser que o obstáculo não ofereça atrito suficiente e por isso o sensor de toque não o detecte. Além disso, o robô pode acidentalmente mover o obstáculo, criando um ambiente diferente do esperado. Desta forma, o robô deve possuir um sensor ultrassônico, capaz de identificar os obstáculos de uma longa distância.

Por fim, devem ser analisados os atuadores escolhidos para o robô. Atuadores são elementos que realizam movimentos, modificando o ambiente. Exemplos de atuadores são garras, guindastes, atiradores, canetas, entre outros. Dependendo da atividade o aluno deve identificar os tipos de atuadores capazes de executá-la.

O segundo tópico a ser avaliado é a programação do robô. Dependendo da idade dos alunos, as aulas de robótica educacional já podem incluir tópicos de introdução a lógica de programação, incluindo conceitos como variáveis, execução sequencial de movimentos, estruturas de condição, estruturas de repetição, entre outros.

O fato dos programas serem utilizados para controlar um robô torna estas aulas bastante atrativa aos alunos, permitindo criar uma geração que tenha conhecimento sobre algoritmos, conhecimento este aplicável a qualquer área de atuação.

Como vimos na Seção 3.2.2, existem técnicas dinâmicas e estáticas para avaliação de códigos. As técnicas dinâmicas analisam os resultados dos códigos, ou seja, para saber se o código do aluno está correto, compara-se a execução deste com a execução do código do professor. Se ambos obtiverem os mesmos resultados, o código é considerado correto. As técnicas estáticas avaliam como o código foi desenvolvido, ou seja, que estruturas de programação foram utilizadas, que condições foram testadas e que comandos devem ser executados.

Sabemos que um mesmo problema pode ser resolvido de diferentes formas, utilizando diferentes estruturas, com vários tipos de condições. Por este motivo, as técnicas dinâmicas de análise de códigos computacionais são tão amplamente utilizadas, por permitirem a criação de diferentes algoritmos, partindo da criatividade do aluno. No entanto, estas técnicas não garantem que os códigos criados possuam baixa complexidade, ou que utilizem as estruturas de programação esperadas pelo professor.

Visando a solução deste problema, há a utilização das técnicas estáticas. No entanto, para uma melhor avaliação do código do aluno, os professores devem cadastrar

várias opções de códigos considerados corretos.

A análise de códigos é uma importante ferramenta de auxílio ao professor, pois em turmas com muitos alunos, não é uma tarefa simples analisar o código de cada um deles, considerando ainda que em alguns casos os alunos criam diversos códigos, o modificam durante a aula, e isto pode demandar muito tempo do professor.

O terceiro tópico a ser avaliado é a execução da atividade proposta pelo professor. As atividades aplicadas em uma aula de robótica educacional podem envolver temas interdisciplinares, propondo que o robô execute uma tarefa com o propósito de testar seus conhecimentos no assunto abordado. Dependendo da execução das atividades, o professor é capaz de identificar se o aluno adquiriu o conhecimento esperado na aula.

Por exemplo, em uma aula que envolve geografia e história com o tema de Segunda Guerra Mundial, o robô deve percorrer o mapa mundi, seguindo a ordem dos países que sofreram grandes ataques durante a guerra, e levando para cada um destes países um identificador de quando eles foram atacados. Para cada grupo, o professor deve identificar se o robô passou por todos os países, na ordem correta, e se carregou os identificadores corretos para cada país em que o robô passar.

A análise da execução da atividade pode ser uma tarefa custosa para o professor, levando em conta que o aluno pode realizar vários testes até conseguir realizar a atividade como esperado.

Além destes pontos específicos de aulas de robótica educacional, é possível analisar também pontos como colaboração entre os alunos, trabalho em equipe, criatividade, trabalho organizado, habilidades manuais, entre outras competências. É importante que a evolução do aluno nestes pontos seja levada em consideração durante a avaliação do aluno, já que estes são parâmetros que também fazem parte deste tipo de aula, e se encaixam em uma categoria de competências sociais que podem ser promovidas em aulas de robótica educacional.

Constatamos que a avaliação de uma aula de robótica educacional não é uma tarefa simples para o professor, uma vez que exige sua atenção em todos os passos da aula, desde a construção do robô, à programação do mesmo, finalizando com a execução da tarefa proposta.

A construção de uma metodologia de avaliação deve levar em consideração todos estes passos. O professor deve conseguir identificar também que pontos devem ser analisados em sua aula, e quais deles tem maior importância.

4.1.2 Avaliação automática

A aplicação de metodologias de avaliação em aulas de robótica educacional é de responsabilidade do professor, que deve definir os critérios de avaliação e analisar cada aluno ou grupo individualmente. Para auxiliar o professor, agilizando o processo avaliativo, propomos a utilização de avaliações automáticas, realizadas com o apoio de um sistema computacional.

Na Seção 3.2 descrevemos alguns sistemas automáticos de avaliação, como os sistemas de avaliações de questões objetivas e discursivas, e os sistemas para análise

de códigos, utilizados também em competições de programação. No entanto, não encontramos na literatura sistemas automáticos voltados para robótica educacional.

Para realizar uma avaliação automática em uma aula de robótica educacional propomos a utilização de um simulador robótico. As atividades de robótica seriam realizadas no simulador, e o sistema de avaliação analisaria os critérios de avaliação escolhidos.

Para utilizar o robô real, o professor deveria adicionar ao sistema de avaliação todos os dados sobre a execução da atividade (que robô foi utilizado, como ele foi programado, e como ele executou a atividade), tornando o processo avaliativo mais complicado e custoso. Com o uso de simuladores, o próprio sistema é responsável por adquirir estas informações, analisando-as e então avaliando o aluno.

O simulador utilizado neste projeto foi o simulador S-Educ, desenvolvido por esta equipe. Mais detalhes sobre o simulador e as funcionalidades dele serão discutidas no Capítulo 5.

Ao utilizar este sistema é possível resolver o problema de avaliação em aulas de robótica educacional. A aplicação da metodologia desenvolvida juntamente com o simulador permite a aplicação das modalidades de avaliação diagnóstica, formativa e somativa, dependendo dos critérios que forem escolhidos pelo professor.

Além disso, os dados obtidos pelo sistema permitem ao professor analisar o progresso não apenas dos alunos individualmente, mas também da turma como um todo. Com estes dados, é possível realizar as modificações que julgar necessárias em seu processo de ensino, podendo também propor mais atividades aos alunos, com o objetivo de aumentar a prática deles em determinados tópicos.

O objetivo desta proposta não é remover o professor do processo avaliativo, e sim fornecer uma ferramenta de assistência. Ele continua sendo responsável por definir os pontos de avaliação e por analisá-los, utilizando o sistema apenas como auxílio para obtenção de dados utilizados na avaliação dos alunos.

A agilização do processo avaliativo traz benefícios não apenas para o professor, mas também para o aluno, que recebe um *feedback* imediato da realização da atividade, podendo analisar os pontos em que precisa melhorar. O uso de simuladores por si só apresenta diversas vantagens, permitindo realizar aulas de robótica educacional sem a presença do robô, realizar as atividades em um menor tempo, facilidade na realização de testes, entre outras. Adicionando a este sistema uma ferramenta avaliativa, o aluno pode até realizar atividades extra-classe, ampliando ainda mais as fronteiras do ensino da robótica educacional.

4.2 Metodologia de avaliação proposta

Levando em consideração os parâmetros de avaliação discutidos neste Capítulo, criamos uma metodologia de avaliação para aulas de robótica educacional composta por seis critérios, buscando compreender todos os passos da aula, fornecendo ao professor os parâmetros necessários para analisar o desempenho dos alunos.

Os critérios de avaliação escolhidos foram implementados e inseridos no simulador S-Educ, e o resultado da avaliação pode ser visualizado pelo professor também

através do simulador.

Os critérios escolhidos foram:

1. Detecção da posição do robô
2. Atribuição de notas subjetivas
3. Detecção de elementos no robô
4. Movimentação e detecção de objetos no ambiente virtual
5. Análise do código de programação
6. Quantidade de tentativas de execução

Para a execução de uma atividade, é possível utilizar mais de um critério de avaliação, escolha esta feita pelo professor na criação da atividade. Ao escolher os critérios, o professor define os pesos de cada um deles, e a nota final do aluno é calculada através de uma média ponderada, utilizando os pesos definidos (Equação 4.1). As notas individuais de cada critério são calculadas utilizando métodos distintos que serão descritos adiante. Nesta equação, N_c representa a nota em cada um dos critérios e P_c representa o peso de cada um destes critérios na nota final do aluno.

Utilizando esta estratégia para a definição da nota final, o professor pode escolher que elementos farão parte da avaliação do aluno, permitindo que o professor defina quais deles possuem maior importância, através da definição de maiores pesos, possibilitando uma avaliação específica para cada atividade.

$$Nota_Final = \frac{N_{c1} * P_{c1} + N_{c2} * P_{c2} + N_{c3} * P_{c3} + N_{c4} * P_{c4} + N_{c5} * P_{c5} + N_{c6} * P_{c6}}{P_{c1} + P_{c2} + P_{c3} + P_{c4} + P_{c5} + P_{c6}} \quad (4.1)$$

As notas finais de cada critério são recalculadas toda vez que o aluno executa a atividade. Caso ele modifique algo na sua resolução, os novos parâmetros são analisados e a sua nota pode modificar. A partir do momento que o professor finaliza a atividade, as notas dos alunos não sofrem mais modificações.

O professor pode visualizar as notas por aluno e por turma, sendo a nota dividida nos critérios de avaliação cadastrados. Todos os critérios de avaliação implementados podem ser utilizados em qualquer modalidade de avaliação, seja diagnóstica, formativa ou somativa, dependendo dos dados que o professor deseja obter a partir dela.

4.2.1 Detecção da posição do robô

O critério de detecção da posição do robô permite analisar sua movimentação no ambiente virtual, identificando os pontos em que ele passou, com o objetivo de comparar esta movimentação com o caminho esperado pelo professor.

O professor, ao cadastrar este critério de avaliação, marca no ambiente virtual os pontos nos quais o robô deve passar para executar a atividade com sucesso. Ao executar um programa o sistema compara os pontos no qual o robô passa com os pontos esperados.

Caso a atividade proposta exija do aluno a criação de um programa, este critério de avaliação representa uma técnica dinâmica para avaliação de códigos. O professor, ao cadastrar a atividade, sabe o que o código desenvolvido pelo aluno deve realizar, podendo identificar como o robô se movimentaria no ambiente virtual. Embora o professor não precise cadastrar nenhum código, é feita a comparação da execução do código do aluno com a execução esperada pelo professor.

Além disso, este tipo de avaliação permite que o professor avalie a compreensão do aluno acerca do tema da aula, caso a movimentação do robô represente diretamente o conhecimento adquirido por ele.

Sendo assim, este critério de avaliação pode ser utilizado para avaliar tanto a programação do robô quanto a execução da atividade.

Para implementar a detecção da posição do robô, são cadastradas as posições escolhidas pelo professor, e o robô do aluno deve se movimentar seguindo esta mesma ordem. A movimentação do robô é comparada com a movimentação esperada enquanto o programa estiver executando. Adicionamos uma margem de erro para detecção da posição, considerando que o robô não precisa passar exatamente na posição em que o professor cadastrou, e sim em uma posição aproximada.

Toda vez que o sistema analisa a movimentação do robô, ele atualiza a pontuação do aluno naquele critério. A pontuação segue a Equação 4.2, na qual PV representa o número de pontos visitados pelo robô e QPC representa a quantidade de pontos cadastrados. Com o objetivo de garantir a movimentação do robô como esperado pelo professor, o robô deve visitar os pontos exatamente na ordem em que eles foram cadastrados pelo professor; caso contrário eles não serão computados.

$$Nota_criterio1 = 10 * \frac{PV}{QPC} \quad (4.2)$$

4.2.2 Atribuição de notas subjetivas

Na Seção 3.3 descrevemos alguns trabalhos nos quais o professor aplica robótica educacional em sala de aula, e nos quais a avaliação é feita através da interpretação subjetiva do professor.

A avaliação subjetiva apresenta problemas com relação a imparcialidade do professor, sem poder garantir que os mesmos critérios serão aplicados a todos os alunos. No entanto, os autores relatam a importância da avaliação subjetiva, feita levando em consideração não apenas a execução da atividade, mas também o comportamento dos alunos e a criatividade na criação do robô e dos programas.

Este tipo de avaliação permite também uma auto-avaliação do aluno, ou uma avaliação baseada em conversas com a turma.

Utilizando uma avaliação subjetiva é possível que o professor avalie parâmetros como trabalho em equipe, pensamento investigativo, raciocínio lógico, habilidades manuais, entre outros pontos que podem ser analisados em aulas de robótica educacional.

Por este motivo, adicionamos um critério de avaliação no qual a nota é atribuída subjetivamente pelo professor. Esta nota pode considerar todos os parâmetros aqui

discutidos, possibilitando uma avaliação que considere também a evolução do aluno em sala de aula, e não apenas em pontos específicos de aulas de robótica educacional. Vale considerar que esta nota subjetiva pode compor toda a nota da atividade, ou pode ser apenas uma parcela da nota final, a partir dos pesos de cada critério definidos pelo professor.

4.2.3 Detecção de elementos no robô

Inserimos no sistema um critério de avaliação baseado na detecção de elementos no robô. Como descrevemos neste Capítulo, o conhecimento das partes do robô e o entendimento de suas funcionalidades é um dos parâmetros importantes de serem avaliados em aulas de robótica educacional.

O aluno deve ser capaz de identificar que tipo de robô deve criar, levando em consideração que mudanças no robô produzem diferenças na execução da atividade, nos programas que podem ser criados e na complexidade da resolução desenvolvida.

Com o objetivo de avaliar o robô criado pelo aluno, o professor cadastra que elementos o robô deve possuir para executar aquela atividade, escolhendo entre bases, sensores e atuadores. É importante salientar que o professor está limitado às ferramentas incluídas no simulador S-Educ, visto que as atividades serão executadas utilizando este simulador. Fernandes (2013) descreve que tipo de elementos podem ser adicionados no robô virtual neste simulador.

Sendo assim, este critério de avaliação pode ser utilizado para avaliar a montagem do robô.

Toda vez que o robô executa um programa, o sistema analisa que elementos este robô possui, comparando-os com os elementos cadastrados pelo professor. Para calcular a nota atribuída a este critério de avaliação, é utilizada a Equação 4.3, na qual ECA representa a quantidade de elementos corretos adicionados, EIA representa a quantidade de elementos incorretos adicionados, e QEC representa a quantidade total de elementos cadastrados.

$$Nota_criterio3 = \begin{cases} 10 * \frac{ECA-EIA}{QEC}, & \text{se } ECA > EIA \\ 0, & \text{caso contrário} \end{cases} \quad (4.3)$$

Como podemos ver a partir desta equação, elementos corretos produzem uma pontuação positiva, enquanto que elementos incorretos produzem uma pontuação negativa. Este tipo de análise diminui a possibilidade de que o aluno crie soluções trabalhosas e custosas, que envolvem não apenas o uso de muitos elementos no robô, como também uma programação mais complicada.

4.2.4 Movimentação e detecção de objetos no ambiente virtual

Um dos critérios de avaliação adicionados ao sistema foi a análise dos objetos virtuais utilizados para montar o ambiente virtual. Como descreveremos mais na

Seção 5.1, o simulador utilizado neste trabalho, o S-Educ, permite a adição de alguns tipos de elementos virtuais: imagens, linhas, obstáculos leves e obstáculos pesados.

As imagens não são reconhecidas pelo robô, e são utilizadas apenas para facilitar a criação da atividade no ambiente virtual. Entretanto, as linhas e obstáculos são reconhecidos por sensores de cor, toque e ultrassônico presentes no robô.

Durante a execução de uma atividade o robô pode identificar tais elementos e, no caso dos obstáculos leves, o robô pode também movê-los no ambiente virtual. A análise de como foi feita a detecção e manipulação destes elementos pelo aluno fornece ao professor informações sobre o entendimento do mesmo acerca da atividade a ele apresentada, e do programa criado pelo aluno.

O professor pode cadastrar que tipos de elementos virtuais o robô do aluno deve reconhecer e como eles devem ser reconhecidos, ou seja, que tipo de sensor deve detectá-lo. No caso de manipulação de obstáculos leves, o professor indica para que posição do ambiente virtual este obstáculo deve ser levado ao final da execução da atividade.

A análise da detecção de objetos é feita durante a execução do programa, enquanto que a análise da movimentação de objetos virtuais é feita ao final da execução de um programa. A nota do aluno neste critério segue a Equação 4.4, onde QEI representa a quantidade de elementos corretamente identificados pelo aluno, e QEC representa a quantidade de elementos cadastrados. Caso uma nova execução resulte em melhores resultados do que uma execução anterior, a nota do aluno é atualizada.

$$Nota_critério4 = 10 * \frac{QEI}{QEC} \quad (4.4)$$

Com este critério de avaliação é possível avaliar a montagem do robô, que deve possuir os sensores capazes de identificar os elementos virtuais; a programação do robô, funcionando como uma técnica dinâmica, em que o reconhecimento dos objetos virtuais está diretamente relacionado ao programa criado, visto que o professor, sabendo como deve ser o programa correto, é capaz de indicar que elementos o robô deve reconhecer ou movimentar; e também a execução da atividade, visto que o aluno deve possuir o conhecimento sobre a aula para discernir que elementos devem ser identificados.

4.2.5 Análise do código de programação

Para analisar os códigos de programação criados pelos alunos, inserimos no sistema um critério de avaliação que realiza uma análise do código utilizando técnicas estáticas, nas quais analisamos as estruturas que foram utilizadas, as condições testadas, e os comandos executados. Com a detecção da posição do robô e a movimentação de objetos no ambiente virtual já conseguimos aplicar técnicas dinâmicas de análise de código, então nesta etapa focamos apenas nas técnicas estáticas.

Como descrevemos na Seção 3.2.2, várias técnicas podem ser utilizadas para comparação de códigos de programação. Baseado nas estratégias encontradas na literatura, implementamos o cálculo da Distância de Levenshtein para verificação de

similaridade estrutural entre o programa do aluno e a solução proposta pelo professor (Yujian & Bo 2007).

O foco desta técnica é a avaliação da similaridade entre duas *strings* com base no número de operações necessárias para transformar uma *string* em outra, sendo que as operações possíveis são a inserção, a exclusão e a substituição. A distância zero indica que as *strings* são idênticas.

A partir do tamanho de cada *string* é montada uma matriz, onde serão setados os custos de cada operação; geralmente, cada uma delas possui custo 1. Ao final das comparações, a distância é dada pela última posição da matriz. Apesar de ser um algoritmo relativamente antigo, o algoritmo da Distância de Levenshtein ainda é bastante utilizado.

Para melhor utilização do critério de avaliação, permitimos ao professor cadastrar mais de um código para ser comparado com o código do aluno. Dessa forma, é feita a comparação com todos os códigos inseridos pelo professor, e então o que obtiver melhor resultado de comparação será utilizado para atribuição da nota final.

A verificação da similaridade dos códigos é composta basicamente por quatro etapas:

1. Definição de *strings* a partir do código do aluno e dos códigos do professor
2. Análise das variações dos códigos do aluno e do professor
3. Comparação dos códigos
4. Definição da nota final do aluno

Definição de *strings*

O primeiro passo deste critério de avaliação é a criação de uma *string* que represente os códigos do aluno e do professor, *strings* estas que serão comparadas utilizando o cálculo da Distância de Levenshtein. Atribuímos a cada estrutura do código um caractere que o represente. A Tabela 4.1 mostra alguns exemplos dos caracteres utilizados.

Estrutura	Caractere	Estrutura	Caractere
se	'T'	frente	'F'
enquanto	'W'	tras	'B'
toque	'T'	preto	'1'
cor	'C'	verde	'3'

Tabela 4.1: Exemplo dos caracteres utilizados na definição das *strings* dos códigos.

A necessidade da conversão do código em uma única *string* na qual cada estrutura é representada por um caractere surgiu devido ao uso da Distância de Levenshtein. Consideremos por exemplo os códigos representados pelos Algoritmos 1 e 2, no qual podemos identificar um erro na linha 2, em que o aluno utilizou a estrutura *SE* quando deveria ter utilizado a estrutura *ENQUANTO*.

Se o cálculo da Distância de Levenshtein fosse aplicado diretamente aos algoritmos, o valor da distância seria 8, o suficiente para transformar a palavra *SE* em *ENQUANTO*, sugerindo que o código do aluno possui 8 erros quando comparado com o código do professor, quando na verdade há apenas um erro: o uso de uma estrutura de condição ao invés de uma estrutura de repetição.

Algoritmo 1: Algoritmo do aluno.

```

1 inicio
2 se (cor(1)=preto ou cor(2)=preto) entao {
3   frente(bc, 100, 100)
4 }
5 fim
```

Algoritmo 2: Algoritmo do professor.

```

1 inicio
2 enquanto (cor(1)=preto ou cor(2)=preto) farei {
3   frente(bc, 100, 100)
4 }
5 fim
```

Sendo assim, transformamos ambos os códigos em conjuntos de caracteres, como mostrado na Tabela 4.2. Comparando as *strings* geradas, é possível identificar apenas um erro entre elas, indicando que há apenas um erro quando comparando o código do aluno com o código do professor.

Algoritmos	<i>Strings</i> geradas
Aluno	"IC1=1OC2=1TF"
Professor	"WC1=1OC2=1TF"

Tabela 4.2: Transformação de códigos em *strings*.

Variações dos códigos

Como mencionamos anteriormente, um mesmo problema pode ser resolvido de diversas formas, utilizando estruturas de programação diferentes, e por esse motivo sugerimos que o professor cadastre várias opções de códigos considerados corretos, para que o código do aluno seja comparado com todos eles. A comparação que gerar melhores resultados será a utilizada para definir a nota final do aluno.

No entanto, em um código que contenha condições utilizadas em estruturas como *SE* ou *ENQUANTO*, o número de variações cresce ainda mais. Por exemplo, considere a porção de código descrita no Algoritmo 3. A propriedade comutativa dos

operadores lógicos E e OU garantem diversas combinações, todas gerando o mesmo resultado lógico.

Algoritmo 3: Diferentes combinações de condições lógicas.

```

1 ...
2 se (cor(1)=preto ou cor(2)=preto e toque(1)=1 e toque(2)=1) entao {
3 se (cor(1)=preto ou toque(1)=1 e cor(2)=preto e toque(2)=1) entao {
4 se (cor(1)=preto ou cor(2)=preto e toque(2)=1 e toque(1)=1) entao {
5 se (cor(1)=preto ou toque(1)=1 e toque(2)=1 e cor(2)=preto) entao {
6 se (cor(1)=preto ou toque(2)=1 e cor(2)=preto e toque(1)=1) entao {
7 se (cor(1)=preto ou toque(2)=1 e toque(1)=1 e cor(2)=preto) entao {
8 se (cor(2)=preto e toque(1)=1 e toque(2)=1 ou cor(1)=preto) entao {
9 se (toque(1)=1 e cor(2)=preto e toque(2)=1 ou cor(1)=preto) entao {
10 se (cor(2)=preto e toque(2)=1 e toque(1)=1 ou cor(1)=preto) entao {
11 se (toque(1)=1 e toque(2)=1 e cor(2)=preto ou cor(1)=preto) entao {
12 se (toque(2)=1 e cor(2)=preto e toque(1)=1 ou cor(1)=preto) entao {
13 se (toque(2)=1 e toque(1)=1 e cor(2)=preto ou cor(1)=preto) entao {
14 ...

```

Para que o professor não precise adicionar todas estas diferentes combinações, e para que o aluno não tenha que prever qual ordem o professor utilizou ao cadastrar a resposta correta, desenvolvemos um algoritmo capaz de identificar condições em estruturas de SE ou $ENQUANTO$ e gerar as diferentes combinações possíveis, dependendo dos operadores lógicos utilizados.

O Algoritmo 4 mostra como são encontradas todas as combinações das condições lógicas, analisando os operadores lógicos E ou OU . O operador lógico $NÃO$ não foi analisado. A entrada do algoritmo é a condição identificada no código e o tipo de operador que se deseja analisar, e a saída é uma lista com todas as variações desta condição.

Para definir a lista de possíveis condições, utilizamos esta função buscando analisar o operador lógico OU . Como podemos ver no Algoritmo 4, o primeiro passo é identificar as variações devido ao operador lógico E , visto que ele tem precedência em operações lógicas. O algoritmo para encontrar estas combinações é o mesmo aqui representado. Por exemplo, considerando que apliquemos este algoritmo de identificação de variações devido ao operador lógico E à condição mostrada no Algoritmo 3, linha 2, o resultado desta operação seria uma lista contendo as condições das linhas 2 a 7, nas quais foram feitos apenas permutações entre os operandos dos operadores lógicos E . Estas condições seriam utilizadas para fazer em seguida a permutação com os operadores OU .

Em seguida definimos a quantidade de operadores, que será utilizada para definir a quantidade de permutações realizadas. Realizamos as permutações para cada condição encontrada na lista *condicoes*, e para cada operador desta condição. Para realizar a permuta, identificamos o índice do operador, separamos os operandos, realizamos a permuta, e em seguida adicionamos esta nova condição à lista *condicoes*.

Caso esta operação gere uma condição já existente na lista, ela não será novamente adicionada.

Esta operação é repetida enquanto estivermos adicionado novos elementos à lista *condicoes*, analisando sempre os operadores separadamente.

Ao fim da execução deste algoritmo, possuiremos uma lista contendo diferentes combinações das condições lógicas passadas como parâmetro. Este algoritmo é aplicado aos códigos do professor e ao código do aluno, garantindo um melhor resultado para comparação dos códigos.

Algoritmo 4: Algoritmo *encontrar_condicoes* para definição de diferentes combinações de condições lógicas.

Entrada: *condicao*: uma condição identificada no código em uma estrutura de *SE* ou *ENQUANTO*; *tipo*: indica se encontraremos as condições para operadores *E* ou *OU*

Saída: uma lista com todas as variações lógicas desta condição

```

1 se tipo = OU então
2   | condicoes ← encontrar_condicoes(condicao, E);
3 senão
4   | condicoes ← condicao;
5 fim
6 n ← comprimento(condicoes);
7 qnt_operadores ← contar_operadores(condicao);
8 para i = 0 até n faça
9   | para j = 0 até qnt_operadores faça
10    | indice_operadores ← identificar_operadores(condicoes(i));
11    | operandos ← identificar_operandos(tipo, j);
12    | resultado ← operandos(1) + tipo;
13    | resultado ← resultado + operandos(0);
14    | se !condicoes.contem(resultado) então
15    |   | condicoes ← condicoes + resultado;
16    |   | n ← comprimento(condicoes);
17    | fim
18  | fim
19 fim

```

Comparação dos códigos

Como vimos nesta Seção, para realizar a comparação do código do aluno com os códigos dos professores, representamos todos eles utilizando uma *string* na qual cada caractere representa uma estrutura de programação. Em seguida, analisamos este código para reconhecer as variações de cada uma das condições lógicas presentes.

Sendo assim, possuímos uma lista de *strings* que representa todas as variações do código do aluno, e uma lista de *strings* representando cada um dos códigos do

professor e todas as variações presentes neles.

O próximo passo é comparar tais *strings* para avaliar o desempenho do aluno. Para isso, utilizamos o cálculo da Distância de Levenshtein, que fornece um indicador da quantidade de operações necessárias para transformar uma *string* em outra. Quanto menor for esta distância, maior o grau de similaridade das *strings* passadas como parâmetro. Assim, um resultado igual a 0 nos indica que as *strings* são iguais (Yujian & Bo 2007).

Para a aplicação em análise de códigos, podemos inferir que quanto menor for a Distância de Levenshtein, mais similares estão os códigos, pois são necessárias menos inserções, alterações ou exclusões de caracteres, que representam elementos do código de programação.

Existem outras medidas de distâncias para comparação de *strings*, como a Distância de Hamming, que analisa apenas cadeias de caracteres com o mesmo tamanho, e permite a realização apenas de alteração de caracteres. No entanto, estas duas restrições são os motivos pelos quais não aplicamos tal métrica. O tamanho do código é um limitante para a aplicação em análise de códigos, considerando que há grande chance dos códigos gerarem *strings* com tamanhos diferentes.

Além disso, o fato de realizar apenas alteração de caracteres pode gerar um resultado menos satisfatório. Considere por exemplo os dois códigos mostrados na Tabela 4.3, no qual o aluno inverteu a ordem de execução de uma ação (B), que está sendo executada no início do programa ao invés de ser colocada no final. Ao aplicar o algoritmo da Distância de Levenshtein, identificamos que com apenas duas operações, uma remoção e uma adição, é possível transformar a *string* do aluno na *string* do professor, enquanto que aplicando o algoritmo da Distância de Hamming, temos que realizar 12 operações de alteração.

Algoritmos	<i>Strings</i> geradas	Levenshtein	Hamming
Aluno	"BIC1=1OC2=1TF"	2	12
Professor	"IC1=1OC2=1TBF"		

Tabela 4.3: Comparação da aplicação da Distância de Levenshtein e Hamming.

Definição da nota final do aluno

Depois que definirmos uma lista de *strings* representando o código do aluno e uma lista de *strings* que representam o código do professor e suas variações, aplicamos a Distância de Levenshtein comparando o código do aluno com todos os códigos do professor. Aquele que apresentar uma menor distância será utilizado para calcular a nota final do aluno.

A Equação 4.5 mostra como esta nota é calculada, considerando que DL representa a menor Distância de Levenshtein encontrada, e QCP representa a quantidade de caracteres da *string* do professor utilizada para calcular a Distância de Levenshtein.

$$Nota_criterio5 = 10 - 10 * \frac{DL}{QCP} \quad (4.5)$$

4.2.6 Quantidade de tentativas de execução

Outro critério de avaliação inserido no sistema foi a contagem das tentativas de execução. Uma tentativa de execução é contada toda vez que um programa é executado.

Ao cadastrar os critérios de avaliação, o professor deve definir o número máximo de tentativas na execução da atividade. Caso o aluno ultrapasse esta quantidade, sua nota irá diminuindo, à medida que ele for executando mais vezes o programa. Caso o aluno finalize a tarefa realizando menos tentativas do que previsto pelo professor, sua nota receberá um acréscimo.

Para definir a nota final neste critério, consideramos duas possibilidades: nenhum aluno atingiu a meta definida, ou algum aluno conseguiu finalizar a atividade com menos tentativas do que foi previsto pelo professor. A Equação 4.6 descreve como as notas finais foram atribuídas. Para isso, definimos alguns parâmetros importantes:

- **QMaxT**: quantidade máxima de tentativas, estabelecida previamente pelo professor ao cadastrar a atividade.
- **TMin**: quantidade mínima de tentativas, observando todos os alunos. Se nenhum aluno finalizou a atividade com menos tentativas do que estabelecido pelo professor, esse valor foi definido como QMaxT.
- **QT**: quantidade de tentativas do aluno.
- **D**: desvio de cada aluno com relação à quantidade mínima de tentativas.
 $D = QT - TMin$
- **DP**: desvio padrão das tentativas. Este parâmetro representa o grau de oscilação das tentativas dos alunos em relação à quantidade mínima de tentativas (TMin).

Com estes parâmetros, conseguimos definir as seguintes medições:

- **RD (razão de desvio)**: representa a razão entre o desvio do aluno e o desvio padrão da turma.
- **RT (razão de tentativas)**: representa a razão entre quantidade de tentativas do aluno e a quantidade mínima de tentativas, indicando quão distante da quantidade mínima o aluno está.
- **RTMax (razão da tentativa máxima)**: representa a razão entre quantidade de tentativas do aluno e a quantidade máxima estabelecida pelo professor, indicando quão distante da quantidade máxima de tentativas o aluno está.

$$\begin{aligned}
RD &= D/DP \\
RT &= QT/TMin \\
RTMax &= QT/QMaxT \\
Nota_criterio6 &= \begin{cases} 10 - RT * RD, \text{ se } TMin \geq QMaxT \\ 10 - \frac{RT}{RT-RTMax} * RD, \text{ caso contrário} \end{cases}
\end{aligned} \tag{4.6}$$

Quando nenhum aluno finaliza a atividade com menos tentativas do que o estabelecido, temos $TMin \geq QMaxT$. Neste caso, para definir a nota final do aluno, atribuímos a ele a nota máxima (10,0) e subtraímos uma fração desta nota, definida a partir do desvio do aluno.

A razão do desvio reflete o desvio do aluno com relação ao desvio padrão da turma. No entanto, este parâmetro por si só não é suficiente para definir a nota final do aluno. Considerando a Tabela 4.4, podemos analisar duas situações, em que o desvio dos alunos é o mesmo (10), e que o desvio padrão é aproximadamente o mesmo (6,47 e 6,50), resultando em uma razão de desvio praticamente igual (1,54 e 1,53). Entretanto, este desvio representa valores diferentes nos dois casos.

QMaxT	QT	D	DP	RD	RT
10	20	10	6,47	1,54	2,00
5	15	10	6,50	1,53	3,00

Tabela 4.4: Exemplo da quantidade de tentativas em uma atividade.

Um desvio de 10 é mais representativo quando temos uma quantidade máxima de 5 tentativas do que quando temos uma quantidade máxima de 10 tentativas. No primeiro caso, o desvio de 10 representa uma RT de 2,00, indicando que a quantidade de tentativas foi o dobro do esperado. No segundo caso, temos uma RT de 3,00, indicando que a quantidade de tentativas foi três vezes o esperado. Com estes dados, podemos concluir que a RD não representa uma medição suficiente para definir a nota do aluno, deve-se levar em consideração também a razão das tentativas.

Sendo assim, utilizamos a RT para estabelecer um peso à fração que será removida da nota do aluno, como podemos ver na Equação 4.6.

Quando algum aluno finaliza a atividade com menos tentativas do que o estabelecido, o cálculo da nota final é feito de forma diferente. Estabelecemos que o aluno que executar a atividade com um menor número de tentativas receberá nota máxima (10,0), e que as notas dos outros alunos será calculada comparando a sua quantidade de tentativas com a quantidade de tentativas deste aluno.

Como pode-se ver na Equação 4.6, para calcular a nota final do aluno nesta situação, levamos em consideração também a razão da tentativa máxima (RTMax), medição que representa a razão entre a quantidade de tentativas do aluno e a quantidade máxima de tentativas cadastrada pelo professor.

Embora utilizemos a RT para definir a nota do aluno, ou seja, relações entre a quantidade de tentativas do aluno e a quantidade de tentativas mínima encontrada, devemos considerar também quão distante o aluno está da meta definida pelo professor. Essa razão é utilizada para reduzir a fração que será diminuída da nota do aluno, ou seja, embora não tenha o melhor resultado da turma, quanto mais perto do resultado esperado o aluno estiver, menor será a diminuição da sua nota.

A partir da Tabela 4.5, podemos ver um exemplo da execução desta equação. Em ambos os casos o aluno fez 25 tentativas, e a tentativa mínima foi de 15 tentativas. Como podemos ver na primeira linha desta tabela, a quantidade de tentativas estipulada pelo professor era de 16 tentativas, muito próximo da quantidade mínima de tentativas, e longe da quantidade de tentativas do aluno. Sendo assim, a fração negativa da nota do aluno será de 13.83. Na segunda linha desta tabela, a quantidade máxima de tentativas foi definida para 24, se aproximando da quantidade de tentativas do aluno e se distanciando da quantidade mínima. Sendo assim, a fração negativa da nota do aluno passa a ser de 2.67.

Com isso podemos concluir que esta métrica compara o desempenho do aluno com o melhor desempenho da turma, mas o compara também com o desempenho esperado pelo professor.

QT	QMaxT	TMin	RT	RTMax	Fração negativa da nota
25	16	15	1.66	1.56	13.83
	24		1.66	1.04	2.67

Tabela 4.5: Análise da nota do aluno seguindo a equação 4.6.

Em ambos os casos descritos na Equação 4.6, definimos a nota mínima para 5,0, garantindo que mesmo que o aluno não tenha um desempenho muito bom, ele obtenha pelo menos uma nota média.

A aplicação destas equações permite definir as notas dos alunos comparando seu desempenho não apenas com a meta definida pelo professor, mas também com o desempenho dos outros alunos da turma. Dessa forma, caso grande parte da turma obtenha um desempenho baixo, pior do que o esperado pelo professor, a nota do aluno não será muito baixa, pois a RD será pequena. Da mesma forma, se grande parte da turma obtiver ótimo desempenho, os alunos com desempenho baixo serão mais penalizados.

O fato de ponderar as notas dos alunos considerando a turma toda apresenta também um retorno para o professor, que pode utilizar os dados desta análise para interpretar se houve muita divergência entre a performance dos alunos e a performance esperada, além de mostrar se algum aluno ficou muito fora da curva. Esses dados podem ser utilizados para que o professor realize qualquer modificação que julgar necessária ao processo de ensino.

Este critério avaliativo não deve ser interpretado de forma punitiva, e sim como uma estratégia para a análise de erros, estimulando os alunos a analisar bem seus resultados para formular novas respostas à atividade proposta. O objetivo aqui é

estimular soluções baseadas nos erros, mas garantindo que os alunos não estarão apenas realizando testes de forma não pensada até encontrar a solução correta.

Este tipo de critério avalia o processo de ensino-aprendizagem como um todo, visto que o aluno deve possuir bom entendimento sobre a montagem do robô, a programação e a execução da atividade. O melhor conhecimento sobre estas áreas garante que o aluno precisará de menos tentativas de execução para realizar a atividade com sucesso, obtendo então melhores notas.

Capítulo 5

Implementação

Como descrevemos no Capítulo 4, desenvolvemos uma metodologia para auxiliar os professores de robótica educacional a avaliar o desempenho dos alunos baseando-se em parâmetros específicos para este tipo de aula. Para auxiliar o professor na avaliação dos alunos, e visando validar a nossa proposta, utilizamos o simulador de robótica educacional S-Educ.

Adicionamos ao simulador algumas funcionalidades para auxiliar na realização das avaliações, modificando também a modelagem do banco de dados utilizado, para permitir inserir tais alterações. Não obstante, note que esta mesma metodologia pode ser adaptada para ser utilizada em outros ambientes.

Neste Capítulo descreveremos o simulador S-Educ, apontando as modificações realizadas no mesmo para este trabalho, mostrando como os critérios de avaliação são adicionados pelo professor e como seus resultados são posteriormente analisados.

5.1 Simulador S-Educ

O simulador S-Educ foi desenvolvido por pesquisadores do laboratório Natal-Net, voltado para aulas de robótica educacional (Fernandes 2013). Este simulador permite a execução de atividades de robótica educacional nas quais o aluno pode controlar o robô ou programá-lo, utilizando o sistema online W-Educ (Sá 2016).

O W-Educ permite que o aluno programe robôs utilizando a linguagem R-Educ, linguagem esta estruturada utilizando comandos em português, nos quais o aluno pode controlar motores e receber dados de sensores inseridos no robô. A Tabela 5.1 mostra exemplos de comandos da linguagem R-Educ. O programa desenvolvido no W-Educ pode ser utilizado para controlar robôs reais ou robôs virtuais, através do simulador S-Educ.

O simulador S-Educ foi desenvolvido inicialmente para simular robôs do tipo Lego NXT, pois inicialmente robôs Lego eram os únicos que poderiam ser programados utilizando a linguagem R-Educ. Sendo assim, os robôs virtuais são parecidos com os robôs Lego, como podemos ver na Figura 5.1, e possuem os sensores presentes neste *kit* de robótica.

Com o trabalho de Sá (2016), tornou-se possível programar outros tipos de robô utilizando a linguagem R-Educ, no entanto mantivemos o estilo dos robôs Lego

Tipos de comandos	Exemplos
Motores	<i>frente, tras, girar</i>
Sensores	<i>cor, toque, ultra</i>
Estruturas de programação	<i>enquanto, se, para</i>
Outros	<i>texto, luz</i>

Tabela 5.1: Exemplos de comandos em linguagem R-Educ.

NXT nos robôs virtuais por possuir uma aparência agradável aos alunos. Além da aparência, mantivemos os sensores deste *kit* de robótica: sensor de cor, sensor de toque e sensor ultrassônico.



Figura 5.1: Exemplos de robôs do simulador S-Educ.

O simulador S-Educ permite adicionar ao ambiente virtual robôs com diferentes bases, sensores e atuadores, possibilitando ainda o uso de múltiplos robôs na execução de uma mesma atividade. O aluno ou o professor pode montar um robô, escolhendo os elementos que vão constituir-lo, indicando também sua posição no robô. Na Figura 5.1, as representações vermelhas não fazem parte do robô, e estão presentes apenas para visualizar o alcance dos sensores de toque, cor e ultrassônico, respectivamente.

O simulador permite também que o aluno e o professor modifiquem o ambiente virtual, adicionando figuras, linhas ou obstáculos. Figuras são elementos meramente ilustrativos, utilizados para auxiliar na criação de um ambiente virtual que será utilizado na aula de robótica, embora não sejam reconhecidos por nenhum sensor. Estes elementos se assemelham a cartolinas utilizadas para criar aulas de robótica educacional.

As linhas são objetos adicionados no chão, assim como as figuras, mas que permitem o reconhecimento através de sensores de cor. Por fim, os obstáculos são elementos que permitem a criação de paredes, além de representarem objetos que podem ser manipulados pelo robô e identificados por todos os tipos de sensores.

O S-Educ possui ainda outras funcionalidades, como modificação de câmera e gravação de vídeo. Todas as funcionalidades do simulador, assim como os testes realizados com o mesmo, são descritos detalhadamente por Fernandes (2013).

Embora o simulador não substitua o uso de robôs reais, o mesmo pode ser utilizado para praticar conhecimentos inerentes ao uso de robôs. Seu uso neste trabalho

permite a realização de uma avaliação automática, diminuindo a necessidade de inserção de dados por parte do professor.

5.2 Mudanças no simulador S-Educ

Para adicionar a metodologia do sistema avaliativo ao simulador S-Educ, foi necessário realizar algumas modificações no mesmo, adicionando novas funcionalidades e modificando a interface gráfica.

A Figura 5.2 mostra a tela inicial do sistema, na qual professores e alunos podem visualizar as atividades já cadastradas e cadastrar novas atividades. Comparando a versão anterior com a versão atual, podemos ver que adicionamos uma área para indicar que critérios de avaliação serão analisados, além de uma área para separação em turmas, funcionalidade esta não presente na versão anterior do simulador. Com isto, é possível que o professor cadastre uma atividade apenas para um determinado grupo de alunos.

A Figura 5.3 mostra a tela de simulação do sistema, na qual o usuário pode adicionar robôs e elementos virtuais, e escolher os programas que serão executados pelos robôs adicionados. Na tela antiga (Figura 5.3a), o usuário deveria utilizar os menus para modificar entre os dois ambientes: inserção de elementos e execução de programas. Para facilitar este processo modificamos toda a interface (Figura 5.3b), facilitando o acesso aos programas, simplificando a inserção de elementos virtuais, e permitindo que os professores cadastrem os parâmetros dos critérios de avaliação da atividade.

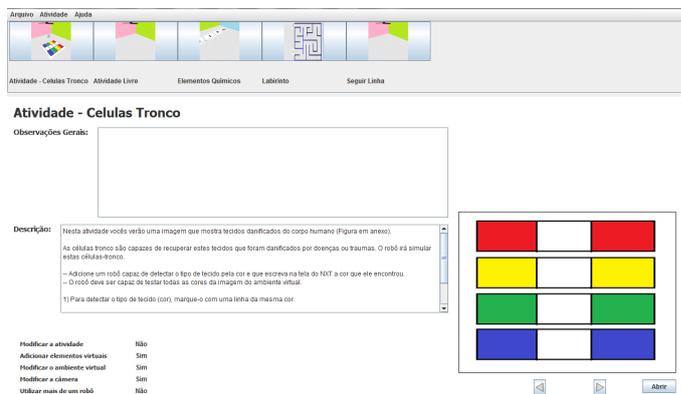
Além das modificações na interface gráfica do simulador, modificamos o banco de dados, permitindo armazenar informações sobre alunos, professores, turmas, além das informações acerca das avaliações dos alunos. O novo modelo do banco de dados foi discutido na Seção 5.3 e pode ser visto na Figura 5.7.

Ao se cadastrar no sistema, o professor pode visualizar as atividades e as turmas cadastradas (Figura 5.2b). Como podemos ver, ao cadastrar ou editar uma atividade, o professor deve indicar quais critérios de avaliação serão analisados pelo simulador, e qual o peso de cada um deles para a definição da nota final do aluno. Quando uma atividade for cadastrada, o professor pode associá-la a uma turma, indicando qual o nível desta atividade naquela turma. Apenas alunos inseridos naquela turma terão acesso às atividades cadastradas nela.

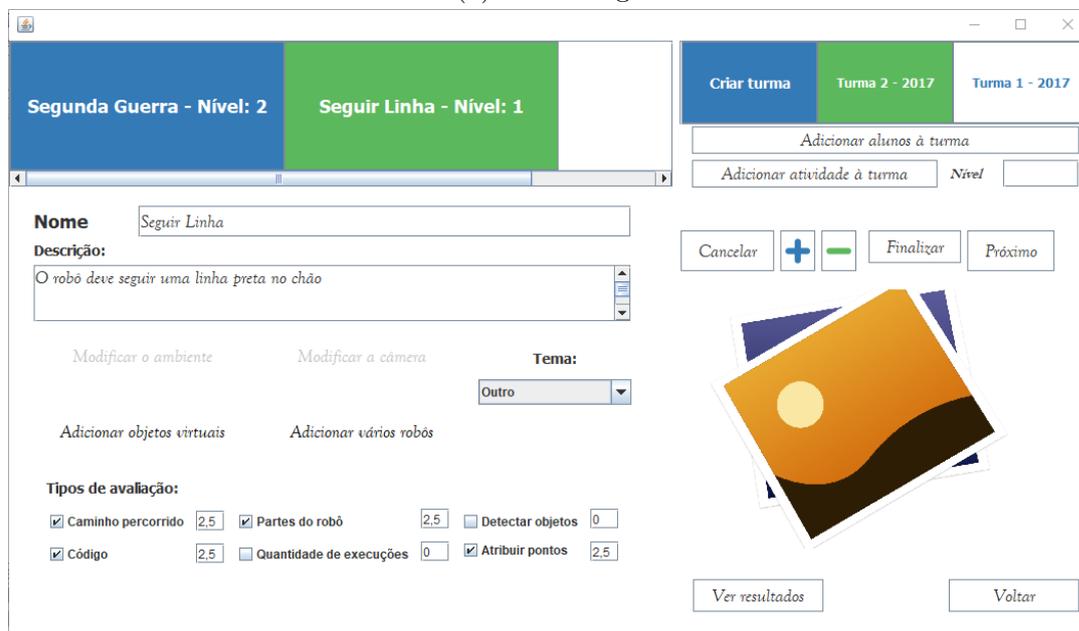
O professor pode também adicionar novas turmas (Figura 5.4) e adicionar alunos a turmas existentes (Figura 5.5).

Na Figura 5.6 representamos o Diagrama de Fluxo de Dados do simulador S-Educ em sua versão anterior e em sua nova versão. Este diagrama nos permite analisar a estrutura do sistema, ou seja, as relações entre os dados e os processos que transformam esses dados.

Como podemos ver na Figura 5.6a, a versão anterior do S-Educ possuía apenas um sistema armazenador de dados. As atividades cadastradas podiam ser executadas com a escolha dos robôs e dos programas.



(a) Tela antiga



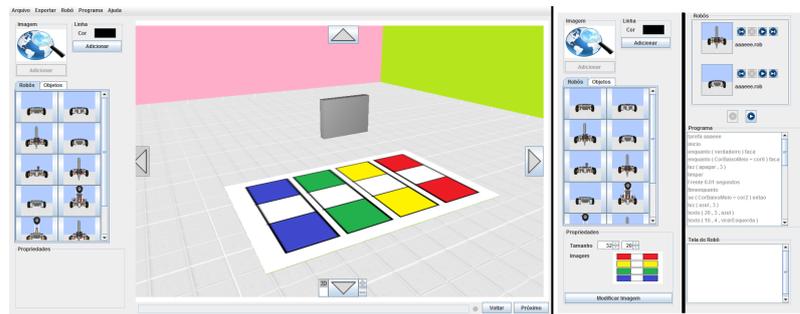
(b) Tela atual

Figura 5.2: Tela inicial do simulador S-Educ.

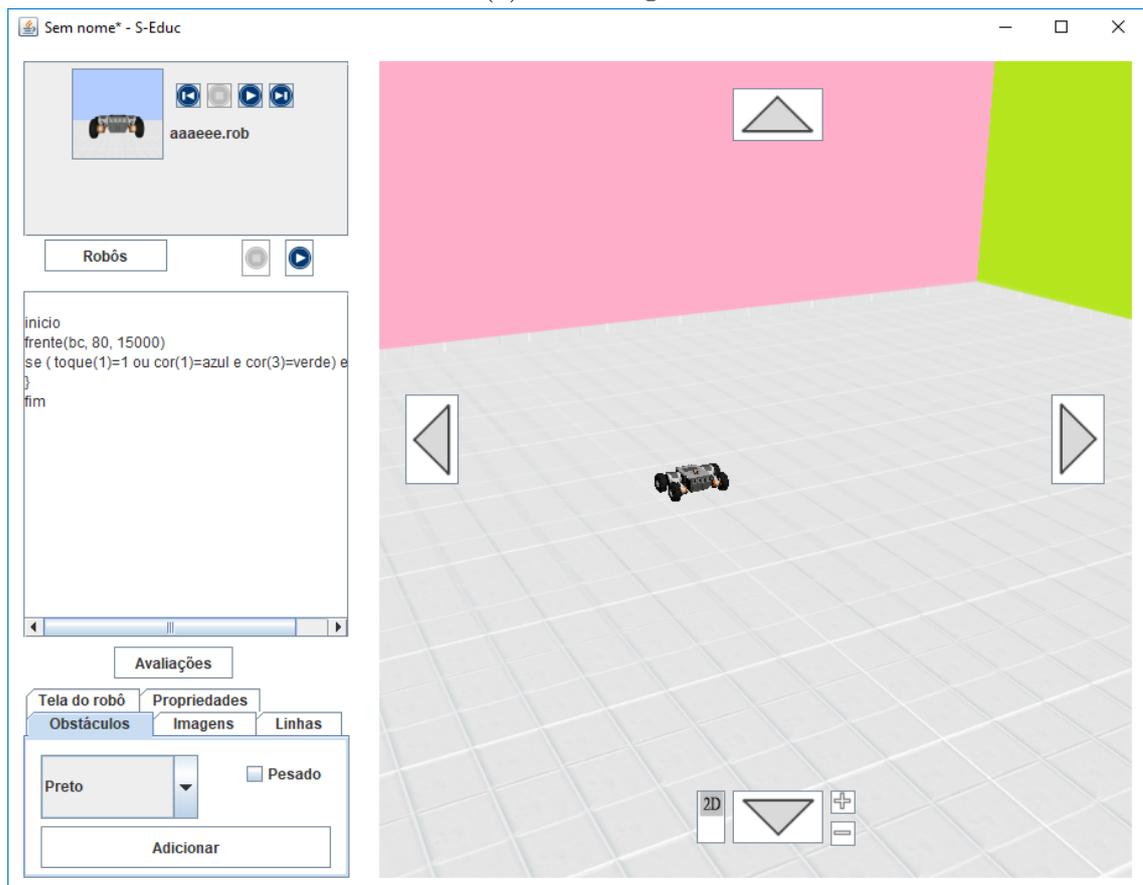
Já na sua versão atual (Figura 5.6b), além das atividades, o sistema armazena os dados sobre as turmas, as associações entre atividades e turmas, as avaliações cadastradas nas atividades, e as avaliações de cada aluno quando executa uma atividade. O sistema permite também que sejam feitas consultas para verificar as avaliações dos alunos.

5.3 Modelagem do banco de dados

O banco de dados foi desenvolvido baseado nas necessidades do sistema de avaliação, já descritas neste Capítulo. O diagrama entidade-relacionamento utilizado para a modelagem do banco de dados pode ser visto na Figura 5.7.



(a) Tela antiga



(b) Tela atual

Figura 5.3: Tela de simulação do simulador S-Educ.

Neste diagrama não foram representados todos os atributos das entidades, apenas aqueles considerados mais significativos para o entendimento do banco de dados criado.

Como podemos ver no diagrama, ao professor são associadas turmas e atividades. A uma turma são associadas atividades, e nesta associação é indicado o nível a que esta atividade pertence e se a mesma está ou não finalizada. O aluno terá acesso apenas às atividades associadas às turmas nas quais está inserido. Atividades

Nome da turma:

Ano:

Cancelar **Criar turma**

Figura 5.4: Criação de novas turmas.

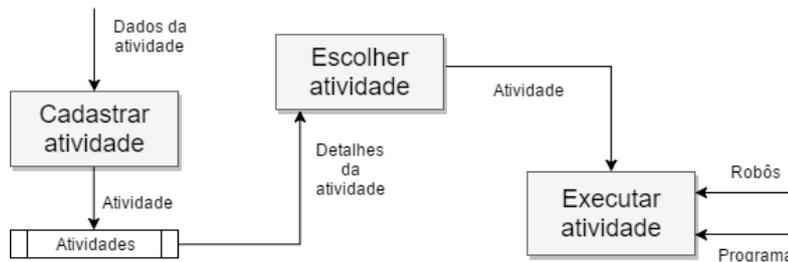
Nome da turma: Turma 2 - 2017

Aluno:

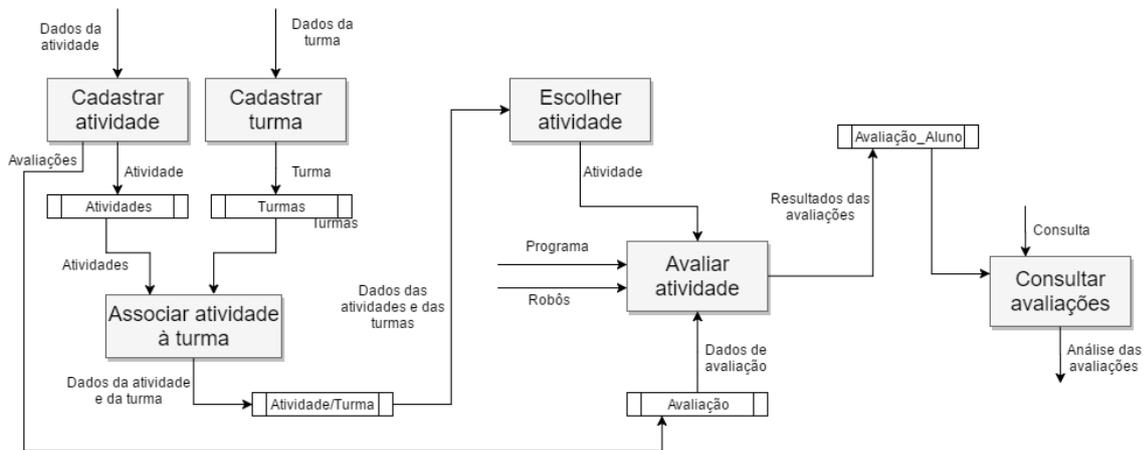
João
Maria
Pedro

Cancelar **Adicionar alunos à turma**

Figura 5.5: Adicionar alunos a turmas.



(a) Sistema antigo



(b) Sistema novo

Figura 5.6: Diagrama de fluxo de dados do S-Educ.

finalizadas não podem mais ser executadas pelos alunos daquela turma.

Para cada atividade podem ser cadastradas várias avaliações, indicando o tipo da avaliação, ou seja, o critério avaliativo utilizado, e o valor de referência. Dependendo do tipo de avaliação, uma atividade pode conter mais de uma avaliação do mesmo tipo. Por exemplo, só pode ser associada a uma atividade uma avaliação de atribuição de notas subjetivas, enquanto que podem ser associadas várias avaliações

de análise de código.

A medida que uma atividade é inserida em uma turma, são criadas avaliações ligadas à turma e à atividade. Quando um aluno executa uma atividade que possua avaliações, são criadas associações entre o aluno e a avaliação cadastrada pelo professor, nas quais são atribuídas as notas do aluno naquele critério, e dependendo do critério avaliativo esta nota é atribuída no início da execução do programa desenvolvido ou no final.

O banco também armazena os programas criados por cada aluno, e que estão acessíveis apenas a ele para a realização das atividades.

5.4 Diagramas UML

Para projetar o sistema, criamos diagramas UML que nos auxiliaram no entendimento do funcionamento do mesmo. Dentre os diagramas desenvolvidos destacamos a importância do diagrama de atividades, que permite entender o funcionamento de um determinado caso de uso, indicando que atividades são executadas para que este caso de uso seja finalizado, e também quais os atores responsáveis por executá-las.

Na Figura 5.8 é mostrado o diagrama de atividades do caso de uso Criação de Atividade, executado pelo professor. Ao criar a atividade, o professor deve indicar que avaliações serão realizadas nesta atividade, e quais serão os seus pesos na definição da nota final. Em seguida, caso seja necessário, o professor realiza modificações no ambiente virtual para adaptá-lo à atividade que está sendo criada. Por fim, o professor deve cadastrar no sistema os parâmetros para cada critério de avaliação que será analisado na atividade. O professor só pode finalizar a criação da atividade caso todos estes parâmetros sejam definidos.

O único critério ao qual não deve-se definir um parâmetro é o critério 2, de atribuição de nota subjetiva. Neste caso o professor simplesmente atribui as notas que julgar necessárias ao aluno antes de finalizar a atividade.

Na Figura 5.9 é mostrado o diagrama de atividades do caso de uso Execução de Atividade. Este caso de uso envolve não apenas o aluno como também o sistema, responsável por realizar as avaliações automaticamente. Inicialmente o aluno escolhe a atividade que irá realizar, definindo em seguida o robô que será utilizado e o programa que ele irá executar.

Assim que o aluno habilita a execução do programa, o sistema pode realizar três avaliações: detecção de elementos no robô, análise do código de programação e quantidade de tentativas de execução. Estes critérios serão avaliados apenas se tiverem sido previamente cadastrados pelo professor.

Após estas avaliações iniciais o robô começa a executar o programa definido pelo aluno. Caso o critério de detecção da posição do robô ou detecção de elementos virtuais tenha sido cadastrado, esta avaliação será frequentemente atualizada, à medida que o robô passe pelos pontos definidos pelo professor ou que detecta objetos utilizando sensores. Se o programa finalizar ou se o aluno parar a execução do programa, o critério de movimentação de objetos é avaliado, caso tenha sido cadastrado.

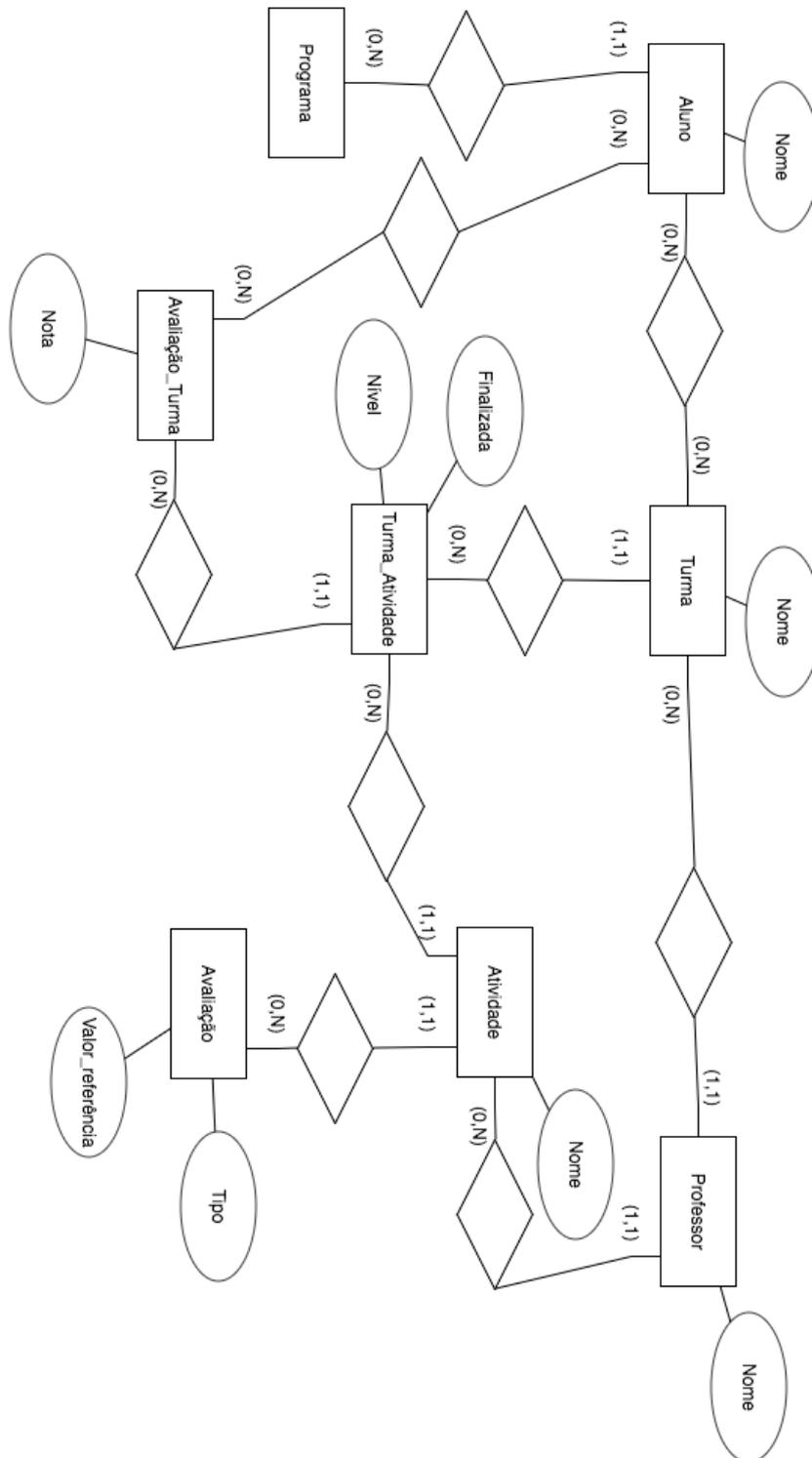


Figura 5.7: Diagrama entidade-relacionamento.

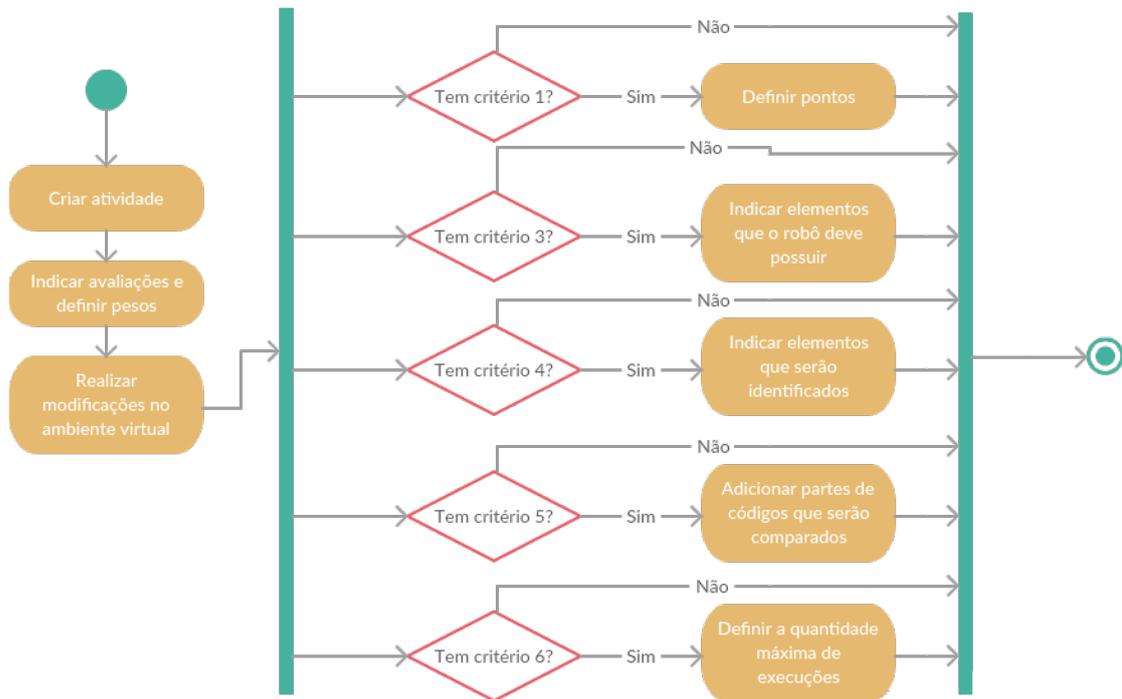


Figura 5.8: Diagrama de atividades do caso de uso Criação de Atividade.

Na Figura 5.10 é mostrado o diagrama de atividades do caso de uso Finalizar Atividade, que envolve o professor e o sistema. Neste diagrama podemos ver que, uma vez que o professor inicia a finalização da atividade, o sistema ainda analisa um critério de avaliação, atribuindo as notas do critério 2 (atribuição de notas subjetivas) para todos os alunos da turma, caso estas notas ainda não tenham sido atribuídas.

A última atividade realizada neste ponto é a atribuição da nota final do aluno na atividade.

Como podemos ver, as notas dos critérios 3 e 5 são atribuídas no momento em que é iniciada a execução de um programa, a nota do critério 1 é atribuída durante a execução de um programa, e a nota do critério 4 é atribuída ao fim da execução de um programa. Todas estas notas podem ser alteradas caso o robô execute outros programas. A nota do critério 2 é atribuída apenas quando o professor finaliza a atividade.

Para o critério 6 é sempre armazenado o seu valor, ou seja, a quantidade de tentativas de cada aluno. A sua nota final, utilizando os parâmetros vistos na Seção 4.2.6, é calculada para atribuir a nota final do aluno toda vez que o professor desejar visualizar as notas dos alunos.

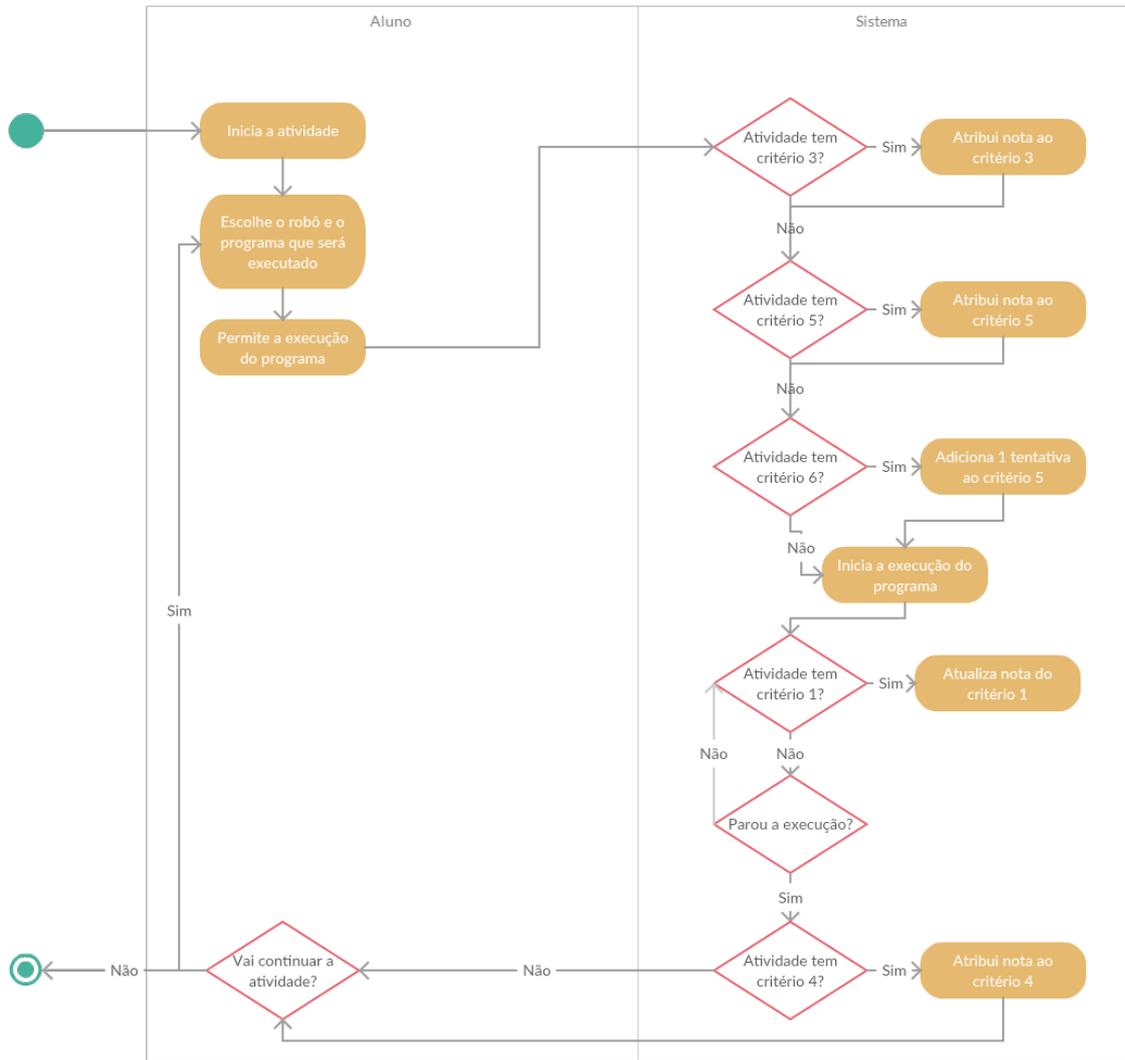


Figura 5.9: Diagrama de atividades do caso de uso Execução de Atividade.

5.5 Cadastro de critérios de avaliação

Como foi descrito na Seção 4.2, foram inseridos seis critérios de avaliação no simulador S-Educ:

1. Detecção da posição do robô
2. Atribuição de notas subjetivas
3. Detecção de elementos no robô
4. Movimentação e detecção de objetos no ambiente virtual
5. Análise do código de programação
6. Quantidade de tentativas de execução

A escolha destes critérios levou em consideração os parâmetros de avaliação que

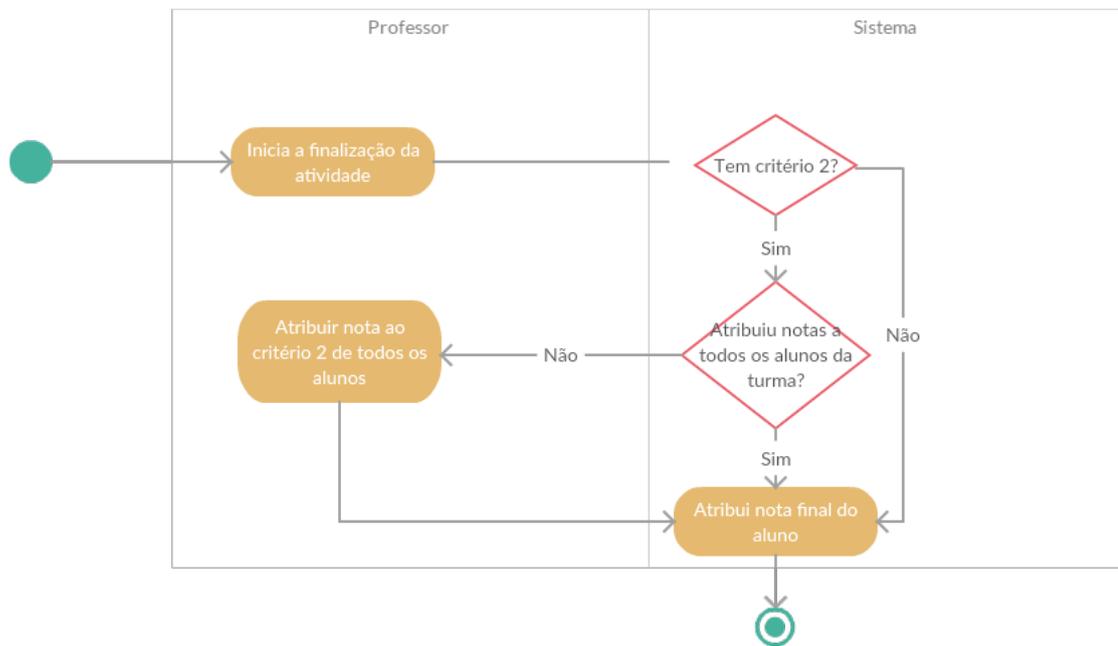


Figura 5.10: Diagrama de atividades do caso de uso Finalizar Atividade.

se deseja avaliar em uma aula de robótica educacional: montagem do robô, programação do robô e execução da atividade. O uso de um simulador de robótica facilita a análise destes critérios, permitindo uma avaliação automática do aluno.

Embora estejamos interessados em uma avaliação automática, é importante que sejam cadastrados parâmetros de avaliação, ou seja, critérios que o sistema computacional deve utilizar como base para definir a nota final do aluno. Por esta razão o professor tem um papel tão importante neste sistema, considerando que ele é o responsável por definir estes parâmetros de avaliação. Sem isso, não seria possível que o sistema fornecesse ao professor os dados necessários para analisar o desempenho dos alunos.

É necessário que o professor possua conhecimento sobre informática e sobre o simulador de robótica para que possa cadastrar tais parâmetros. Caso o professor não domine estes pontos, esta tarefa pode ser delegada a outro usuário, seja outro professor ou algum coordenador responsável, desde que este entenda o funcionamento da atividade proposta, e como o professor deseja que a avaliação seja realizada.

Na tela de simulação (Figura 5.3b) o professor pode visualizar as avaliações cadastradas naquela atividade (Figura 5.11), e a partir daí definir cada um dos critérios avaliativos.

5.5.1 Detecção da posição do robô

Para definir o critério de detecção da posição do robô, o professor deve cadastrar no ambiente virtual os pontos nos quais o robô deve passar ao executar uma

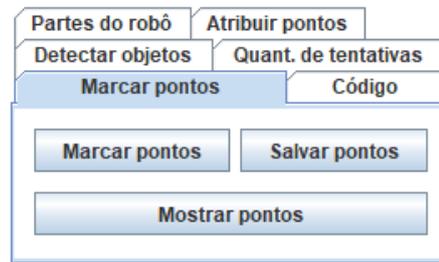
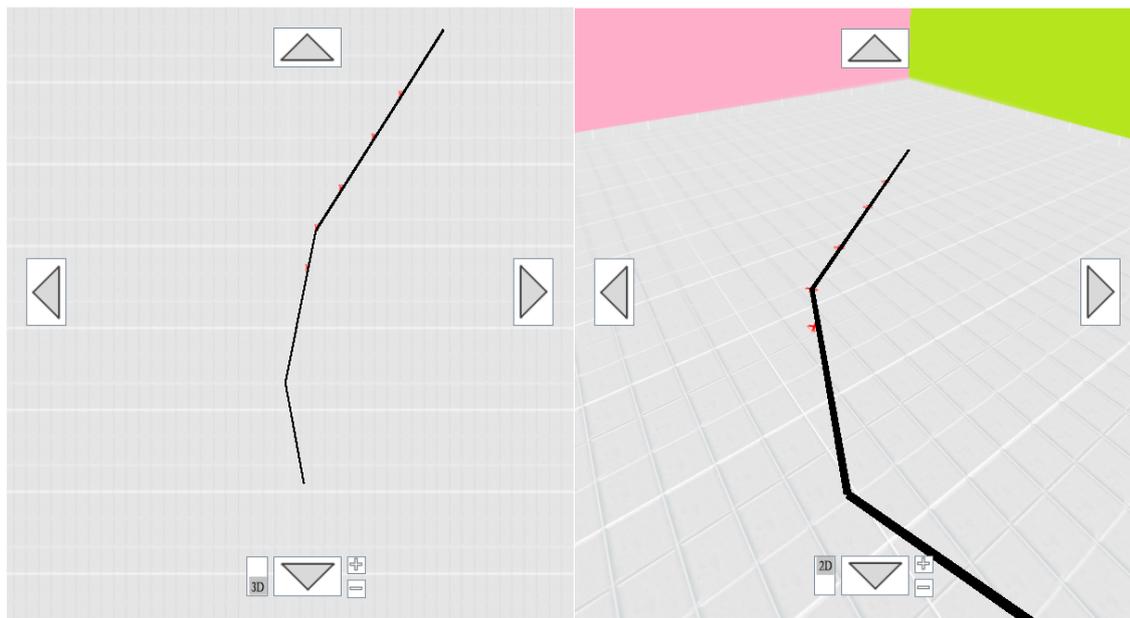


Figura 5.11: Avaliações cadastradas na atividade.

atividade. Os pontos cadastrados devem estar na ordem nas quais eles serão percorridos pelo robô. A Figura 5.11 mostra as opções do professor: marcar pontos, salvar pontos e mostrar pontos.



(a) Marcar pontos

(b) Mostrar pontos

Figura 5.12: Cadastro do critério 1.

Caso o professor escolha marcar pontos, o simulador automaticamente modifica para uma visualização bidimensional, facilitando a definição dos pontos, representados por uma marcação vermelha (Figura 5.12a). Quando todos os pontos tiverem sido marcados, o professor pode salvá-los, inserindo estas informações no banco de dados.

A qualquer momento, o professor pode escolher a opção de mostrar pontos, visualizando na tela os pontos que foram marcados (Figura 5.12b). Caso queira, o professor pode também limpar os pontos e escolher novas posições para serem utilizadas como parâmetro neste critério de avaliação.

Como dito na Seção 4.2.1, ao percorrer o ambiente virtual o simulador assume uma margem de erro, garantindo que o aluno possa receber uma boa avaliação sem precisar passar exatamente aonde o professor definiu, e sim em suas proximidades.

5.5.2 Atribuição de notas subjetivas

A Figura 5.13 mostra a atribuição de notas subjetivas. A nota pode ser atribuída aos alunos da turma selecionada. O sistema permite que o professor visualize e edite as notas que já foram atribuídas aos alunos.

(a) Visualizar notas

(b) Atribuir notas

Figura 5.13: Cadastro do critério 2.

5.5.3 Detecção de elementos no robô

Para a detecção de elementos no robô, o professor deve cadastrar os sensores e atuadores que o robô deve possuir. No caso dos sensores, o professor deve indicar também a posição no robô em que este sensor deve estar (Figura 5.14).

Quando um programa for executado o simulador irá comparar os elementos presentes no robô com os cadastrados pelo professor, gerando então a nota do aluno neste critério. Enquanto o aluno estiver executando a atividade, sua nota pode ser atualizada, caso ele escolha um novo robô.

Como vimos na Seção 4.2.3, elementos corretos geram um fator positivo na nota do aluno, enquanto que elementos incorretos geram um fator negativo. Esta estratégia visa a criação de uma solução menos trabalhosa e custosa por parte do aluno.

5.5.4 Movimentação e detecção de objetos no ambiente virtual

Foi adicionado ao sistema avaliativo um critério que permite que o professor indique objetos virtuais que devem ser movimentados pelo robô, ou elementos que devem ser detectados por sensores. Estas duas opções permitem que o sistema

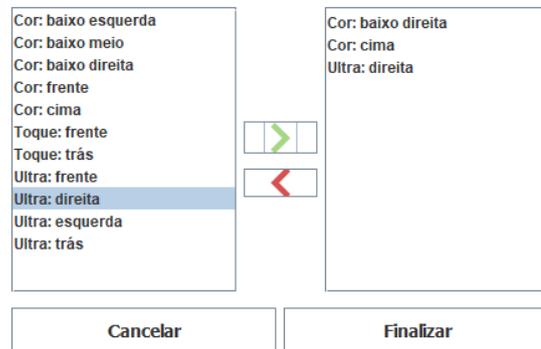


Figura 5.14: Cadastro do critério 3.

análise não apenas a resolução da atividade, mas também o código de programação e os elementos inseridos no robô.

A Figura 5.15 mostra as opções do professor ao cadastrar este critério: marcar posição, detectar com sensor ou visualizar dados.



Figura 5.15: Cadastro do critério 4.

Quando o professor escolhe marcar posição, ele deve selecionar um objeto leve e então marcar a sua posição final, ou seja, a posição que ele deve se encontrar ao fim da execução do programa. Para escolher esta posição o simulador modifica para uma visualização bidimensional, facilitando para o professor a marcação da nova posição, representada por uma marcação vermelha (Figura 5.16a).

Quando o professor escolhe detectar com sensor, ele também deve selecionar um elemento virtual, que pode ser um objeto leve ou pesado, ou uma linha no chão. Em seguida, o professor deve indicar qual sensor deve ser utilizado para detectar o elemento escolhido (Figura 5.16b).

A qualquer momento, o professor pode visualizar e remover elementos que foram inseridos para definir este critério avaliativo (Figura 5.17).

5.5.5 Análise do código de programação

O professor pode adicionar códigos que serão comparados com o código do aluno para analisar se ele utilizou as mesmas estruturas de programação esperadas pelo

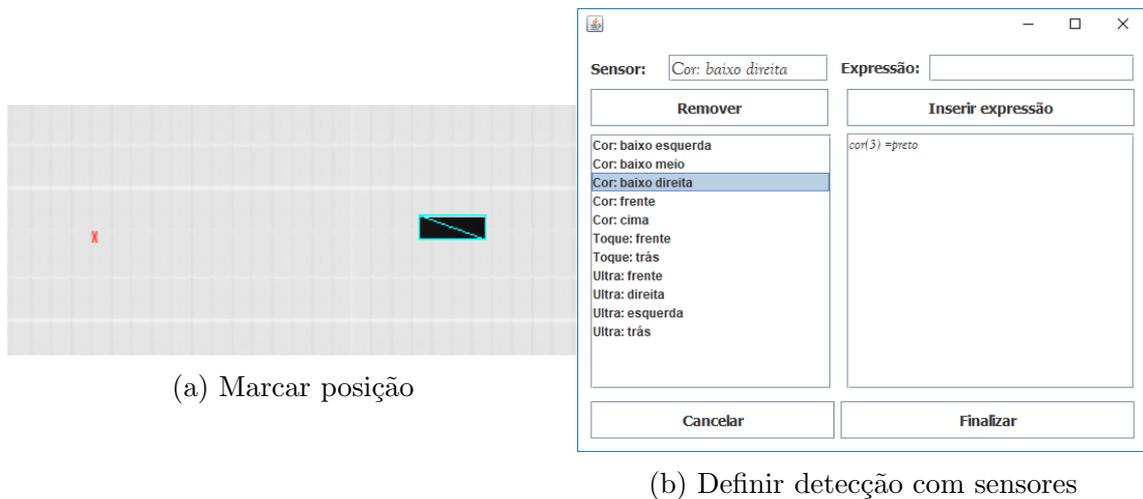


Figura 5.16: Cadastro do critério 4: definição dos parâmetros.

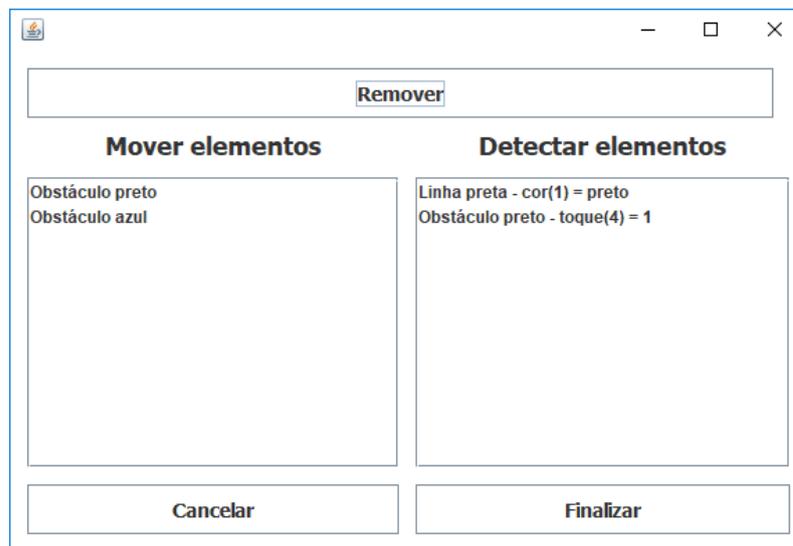


Figura 5.17: Cadastro do critério 4: visualização dos parâmetros avaliativos.

professor, e se ele executou as mesmas ações. Desenvolvemos um método de comparação de códigos de programação, descrito detalhadamente na Seção 4.2.5.

Permitimos a adição de vários códigos em uma atividade visto que um problema de programação pode ser resolvido de várias formas diferentes. Quanto mais completo for o conjunto de códigos adicionado pelo professor, mais precisa será a nota dos alunos. O código do aluno é comparado com todos os códigos adicionados pelo professor, e aquele que obtiver o melhor resultado de comparação será utilizado para atribuição da nota final.

A Figura 5.18 mostra como estes códigos são adicionados ao banco de dados. Para adicionar códigos, o sistema fornece ao professor as estruturas que serão comparadas com os códigos do aluno, auxiliando o mesmo a criar seus códigos (Figura 5.18a).

O professor pode visualizar, editar ou remover os códigos previamente adicionados (Figura 5.18b).

(a) Adicionar códigos

(b) Visualizar códigos

Figura 5.18: Cadastro do critério 5.

5.5.6 Quantidade de tentativas de execução

Adicionamos ao sistema de avaliação um critério de quantidade de tentativas de execução, em que o professor define a quantidade máxima de vezes que o aluno deve executar um programa com o objetivo de concluir a atividade.

Para definir a quantidade de tentativas, fornecemos ao professor a média da quantidade de tentativas dos alunos e a quantidade de tentativa de cada um deles separadamente (Figura 5.19). Com estas informações, o professor pode modificar o parâmetro deste critério de avaliação para melhor adaptar-se à realidade da turma, caso julgue necessário.

5.6 Avaliação automática

Para realizar a avaliação automática dos alunos, o sistema leva em consideração os dados adicionados pelo professor ao cadastrar os critérios de avaliação, mais detalhadamente descrito na Seção 5.5.

A nota do aluno nos critérios de avaliação escolhidos é atualizada cada vez que um programa é executado. Como descrevemos na Seção 5.4, os critérios são avaliados

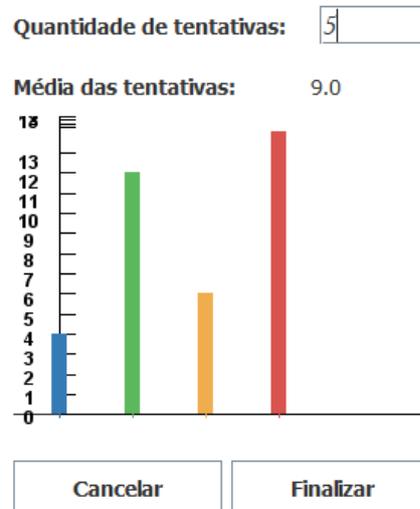


Figura 5.19: Cadastro do critério 6.

em diferentes momentos durante a execução de um programa:

- Detecção de elementos no robô e análise do código de programação: analisados antes de um programa iniciar;
- Detecção da posição do robô e detecção de elementos virtuais: analisado durante a execução do programa;
- Movimentação de elementos virtuais: analisado ao fim da execução de um programa;
- Atribuição das notas subjetivas: pode ser feito a qualquer momento pelo professor, até a atividade ser finalizada; e
- Análise da quantidade de tentativas de execução: analisada apenas quando a atividade é finalizada.

5.7 Visualização dos resultados

Para facilitar uma análise do desempenho dos alunos nas aulas de robótica educacional, o sistema permite que o professor visualize as notas obtidas pelos alunos nas atividades cadastradas e nos critérios de cada atividade. A Figura 5.20 mostra esta tela de visualização, na qual o professor pode escolher uma turma e o sistema fornece todos os dados de avaliação das atividades associadas a esta turma.

Ao escolher visualizar os resultados, o sistema mostra todos os alunos cadastrados naquela turma, e suas notas finais em cada uma das atividades, mostrando também a média da turma. O professor pode escolher filtrar entre as atividades cadastradas na turma por nível (Figura 5.21).

Esta tabela de visualização permite que o professor acesse detalhes sobre cada aluno ou sobre cada atividade. O professor pode selecionar na tabela uma nota

The screenshot shows a web application window with the following elements:

- At the top left, there is a small icon.
- At the top right, there are standard window control buttons: minimize, maximize, and close.
- Below the controls, there are two dropdown menus:
 - Turma**: Set to "Turma 3 - 2017".
 - Nível**: Set to "Todos".
- Centered below the dropdowns is a button labeled **Visualizar Resultados**.
- Below the button is a table with three columns: **Nome**, **Atividades**, and **Nota final**. The table body is currently empty.
- Below the first table is another table with eight columns: **Nome**, **Critério 1**, **Critério 2**, **Critério 3**, **Critério 4**, **Critério 5**, **Critério 6**, and **Nota final**. This table body is also empty.

Figura 5.20: Tela de visualização dos resultados.

de um aluno em uma determinada atividade, tendo acesso aos dados mostrados na Figura 5.22. Com isto, o sistema fornece as notas do aluno em cada um dos critérios de avaliação cadastrados naquela atividade, mostrando ao professor em que pontos tal aluno deve melhorar.

O gráfico do lado esquerdo compara a nota final do aluno com a nota da turma, mostrando a nota dele, a menor nota, a maior nota e a média de notas da turma. Ao selecionar um determinado critério de avaliação, o sistema realiza também uma comparação entre o aluno e a turma, desta vez analisando apenas um determinado critério, como mostrado no gráfico do lado direito.

O professor pode visualizar também os resultados de um aluno em todas as atividades cadastradas na turma (Figura 5.23). Neste caso, o sistema fornece as notas do aluno em cada um dos critérios separados por atividade. O gráfico do lado esquerdo mostra as notas finais do aluno em cada uma das atividades. Caso o professor selecione algum critério de avaliação, o sistema mostra o gráfico do lado direito, no qual o professor pode visualizar as notas do aluno por atividade em um determinado critério.

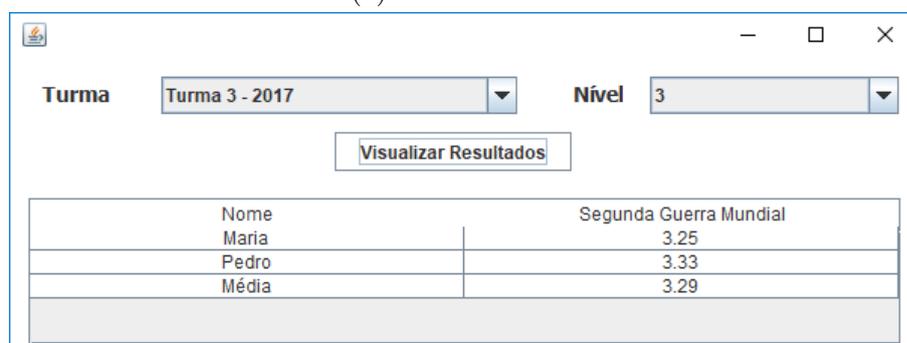
Por fim, o professor pode visualizar também o resultado de todos os alunos da turma em uma única atividade (Figura 5.24). O gráfico do lado esquerdo mostra as notas de cada um dos alunos naquela atividade, mostrando também a média da turma. Caso selecione algum critério de avaliação, o sistema mostra ao professor o gráfico do lado direito, em que podemos ver as notas de todos os alunos da turma naquele determinado critério.



The screenshot shows a web application window with a title bar. Below the title bar, there are two dropdown menus: 'Turma' set to 'Turma 3 - 2017' and 'Nível' set to 'Todos'. A 'Visualizar Resultados' button is centered below the dropdowns. Below the button is a table with three columns: 'Nome', 'Segunda Guerra Mundial', and 'Seguir Linha'. The table contains three rows of data.

Nome	Segunda Guerra Mundial	Seguir Linha
Maria	3.25	5.4
Pedro	3.33	4.66
Média	3.29	5.02

(a) Todos os nível



The screenshot shows the same web application window as in (a), but with the 'Nível' dropdown menu set to '3'. The 'Visualizar Resultados' button is still present. The table below it now only displays two columns: 'Nome' and 'Segunda Guerra Mundial', with three rows of data.

Nome	Segunda Guerra Mundial
Maria	3.25
Pedro	3.33
Média	3.29

(b) Apenas nível 3

Figura 5.21: Visualização de atividades por nível.



Figura 5.22: Visualização de resultados de um aluno em uma atividade.

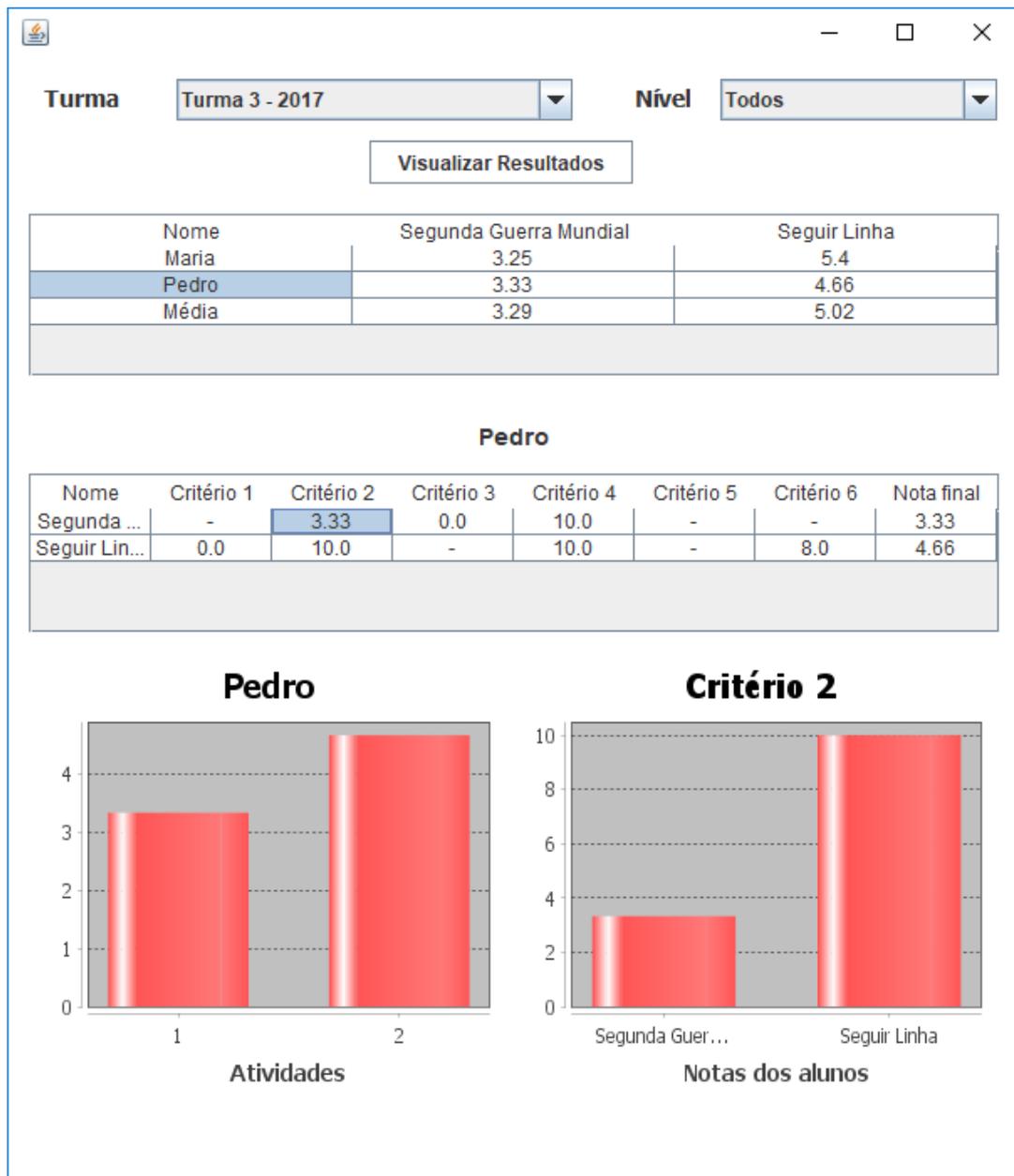


Figura 5.23: Visualização de resultados de um aluno.



Figura 5.24: Visualização de resultados de uma atividade.

Capítulo 6

Experimentos e Resultados

Percebemos a necessidade de desenvolver uma metodologia avaliativa que inclua uma série de critérios avaliativos capazes de tornar a avaliação da aula de robótica educacional o mais proveitosa possível para o professor, disponibilizando para ele informações úteis para a análise da sua aula e dos seus alunos.

Neste Capítulo apresentaremos testes realizados em duas atividades de robótica criadas no simulador S-Educ, nos quais os professores podem avaliar diferentes critérios e analisar seus resultados depois que a atividade for executada pelos alunos.

6.1 Plataforma de testes

Desenvolvemos uma plataforma de testes para ser utilizada pelos professores ao realizar os testes com o simulador S-Educ, com o objetivo de analisar a metodologia de avaliação automática inserida no simulador. O objetivo desta plataforma de teste não é testar as funcionalidades do simulador, funcionalidades estas já testadas, e sim testar como os dados obtidos pelo simulador podem auxiliar o professor a analisar o desempenho dos alunos nas aulas de robótica educacional.

Os experimentos realizados com o simulador S-Educ mostram que professores e alunos apresentaram interesse em utilizar esta ferramenta em aulas de robótica educacional para a realização de testes, produzindo soluções que poderiam em seguida ser utilizadas com *kits* de robótica. Os usuários sugeriram algumas funcionalidades que foram em seguida implementadas no sistema, permitindo a realização de aulas cada vez mais similares àquelas que utilizam *kits* de robótica. Maiores detalhes sobre os resultados dos testes realizados com o simulador, envolvendo professores e alunos, foram documentadas por Fernandes (2013).

O teste é dividido em duas etapas: definição dos critérios avaliativos e análise dos dados obtidos após a avaliação automática do sistema. Na primeira etapa, o professor deve definir os critérios avaliativos que serão utilizados nas atividades já inseridas no sistema, e em seguida definir os parâmetros avaliativos para cada um destes critérios.

O Apêndice A mostra o documento apresentado aos professores para a realização deste teste, com a descrição da atividade na qual eles devem cadastrar os critérios avaliativos, descrevendo que critérios serão avaliados. Preparamos tal documento

para garantir que o professor utilize todas as funcionalidades de avaliação inseridas no sistema, auxiliando-o, já que este é o primeiro contato do professor com esta ferramenta.

Como podemos ver no documento do Apêndice A, o professor é responsável por cadastrar os critérios avaliativos em duas atividades diferentes: Seguir Linha e Segunda Guerra Mundial.

Na primeira atividade, o professor deve cadastrar os critérios de detecção da posição do robô, atribuição de notas subjetivas, detecção de elementos no robô e análise do código de programação. Na segunda atividade, o professor deve cadastrar os critérios de detecção da posição do robô, detecção de elementos no robô, quantidade de tentativas de execução e movimentação e detecção de objetos no ambiente virtual.

Em seguida, os professores devem participar da segunda etapa de testes: análise dos dados obtidos após a avaliação automática do sistema. O Apêndice B mostra a descrição do passo-a-passo desta etapa de testes. O professor deve analisar os resultados de duas atividades, com o objetivo de identificar como os critérios avaliativos podem auxiliá-lo na avaliação dos alunos e da turma.

Por fim, os professores devem responder um questionário qualitativo/quantitativo (Apêndice C) no qual devem descrever a utilização do sistema desenvolvido.

6.1.1 Descrição das atividades

O professor é responsável por cadastrar e analisar os parâmetros avaliativos de duas atividades: Seguir Linha e Segunda Guerra Mundial. A escolha destas atividades se deu por serem atividades já previamente testadas por nossa equipe em aulas de robótica educacional, e que permitem a análise de diferentes critérios avaliativos.

Na primeira atividade, o aluno deve utilizar dois sensores de cor voltados para baixo para que o robô se mova seguindo uma linha preta no chão, que pode ou não conter descontinuidades. Este tipo de atividade é amplamente utilizado em competições de robótica no Brasil e no mundo, tais como a Olimpíada Brasileira de Robótica e a RoboCup.

O objetivo desta atividade é garantir que o aluno tenha conhecimento sobre sensores, encontrando a melhor forma de utilizar sensores de cor para resolver o problema. Além disso, avalia o conhecimento do aluno sobre programação, já que o programa desenvolvido pelo aluno utiliza estruturas de condição e repetição.

Os códigos de programação sugeridos na atividade (Apêndice A) são apenas exemplos de códigos que o professor pode cadastrar. Sabemos que, para esta atividade, há diversos tipos de códigos diferentes que podem ser cadastrados. Sugerimos que o professor cadastre a maior quantidade possível de códigos, com o objetivo de tornar este tipo de avaliação o mais precisa possível.

Com relação à avaliação das notas subjetivas dos alunos, não sugerimos a definição destas notas no ato do cadastro da atividade, considerando que os alunos ainda não terminaram a atividade. Sugerimos que o professor cadastre as notas dos alunos na segunda etapa de testes, em que o professor deve finalizar a turma, e só assim

definir as notas subjetivas dos alunos.

Na segunda atividade, o aluno deve utilizar um sensor ultrassônico para se movimentar como se estivesse em um labirinto, mas utilizar também um sensor de cor voltado para baixo para reconhecer marcações que representam países, e um sensor de cor voltado para cima, para indicar um alerta quando o robô chegar em algum destes países.

O objetivo desta atividade é garantir que o robô passe pelos países, e que saiba identificar como cada um deles participou da Segunda Guerra Mundial. O aluno deve perceber a necessidade de utilizar um sensor ultrassônico para se mover nas paredes, e um sensor de cor virado para baixo para identificar as linhas no chão.

6.1.2 Grupo de testes

Como mencionado anteriormente, foram realizados testes com um grupo de professores de robótica educacional da área de ciências exatas, formado por cinco professores. O objetivo deste trabalho é testar a metodologia de avaliação que deve ser utilizada pelos professores em sala de aula. Os professores são responsáveis por cadastrar os critérios avaliativos e avaliar os resultados fornecidos pelo sistema.

Por este motivo, não vimos necessidade de realizar testes com alunos, dado que testes com alunos avaliariam apenas a execução de atividades de robótica em um simulador robótico. Embora tenhamos feito modificações no simulador S-Educ, as funcionalidades básicas do mesmo já foram objetivo de teste (Fernandes 2013).

O teste com professores é composto por duas etapas, sendo a segunda etapa a análise dos resultados da avaliação fornecidos pelo sistema. Os dados utilizados na fase de testes foram obtidos pelo próprio sistema, gerados a partir da execução das atividades em contas de alunos fictícios.

6.2 Análise dos resultados

Para realizar uma análise da metodologia de avaliação desenvolvida e inserida no simulador S-Educ, criamos uma plataforma de teste descrita na Seção 6.1. Ao fim dos testes, os professores responderam um questionário qualitativo/quantitativo (Apêndice C), com o objetivo de analisar os seguintes tópicos:

- Interesse dos professores em aplicar esta metodologia de avaliação em sala de aula;
- Identificação do benefício da aplicação desta metodologia avaliativa em aulas de robótica educacional;
- Identificação da importância de cada um dos critérios avaliativos;
- Facilidade na definição dos parâmetros de cada um dos critérios avaliativos;
- Comentários sobre os critérios avaliativos;
- Sugestão de novos critérios avaliativos;
- Comentários sobre a visualização dos dados após a avaliação automática; e
- Comentários gerais.

Todos os professores mostraram interesse em aplicar esta metodologia de avaliação em sala de aula, o que valida nossa hipótese inicial dentro deste universo selecionado. Um deles destacou, inclusive, que gostaria de utilizá-la em atividades extra-classe, como atividades de casa, opção esta limitada ao utilizar métodos de avaliação tradicionais em aulas de robótica educacional.

Com relação aos benefícios da aplicação da metodologia avaliativa, todos os professores indicaram que acreditam apresentar muitos benefícios. Dentre os comentários mais relevantes, vale destacar um professor que acredita que este tipo de avaliação será estimulante aos alunos, que irão procurar as melhores formas de resolver o problema para obter melhores resultados.

Sobre a importância dos critérios avaliativos inseridos no sistema, o gráfico mostrado na Figura 6.1 representa a importância que os professores deram a cada critério avaliativo. Como podemos ver, o critério de quantidade de tentativas de execução obteve a pior avaliação. Nos comentários sobre os critérios avaliativos, os professores indicaram que este critério poderia estimular os alunos a resolver o problema o mais rápido possível, mesmo que com uma solução incorreta, objetivando uma boa nota neste critério.

Os critérios de atribuição de notas subjetivas, detecção de elementos no robô e análise do código de programação foram os mais bem avaliados. Em seus comentários, os professores destacaram a importância das notas subjetivas, permitindo que uma parcela da nota seja obtida através da experiência do professor. Destacaram também a importância da análise do código de programação, facilitando o trabalho do professor. No entanto, neste critério um professor sugeriu que ele pudesse cadastrar apenas uma parte do código que quer testar, como por exemplo indicar que quer apenas uma estrutura de repetição testando um determinado sensor.

O critério de detecção da posição do robô foi muito bem avaliado, e um professor indicou que isso representa diretamente o conhecimento do aluno acerca do assunto tema da aula. Por fim, o critério de movimentação e detecção de objetos virtuais gerou notas diversas, embora os comentários dos professores fossem positivos acerca deste critério avaliativo. Um professor mencionou que não saberia bem como aplicar este critério em suas aulas de robótica.

Em seguida, os professores indicaram a facilidade na definição dos parâmetros avaliativos para cada um dos critérios. O resultado desta pesquisa pode ser visto no gráfico mostrado na Figura 6.2.

Os critérios de detecção da posição do robô, atribuição de notas subjetivas, detecção de elementos no robô e quantidade de tentativas de execução foram considerados muito fáceis para definir os parâmetros avaliativos. Alguns professores destacaram as vantagens da visualização da quantidade de tentativas dos alunos para a definição deste parâmetro.

Com relação à análise do código de programação, alguns professores indicaram que a única dificuldade que percebem é a necessidade de cadastrar vários códigos para uma avaliação precisa. No entanto, um professor sugeriu em seus comentários que a descrição da atividade já indique o tipo de estruturas de programação que ele quer.

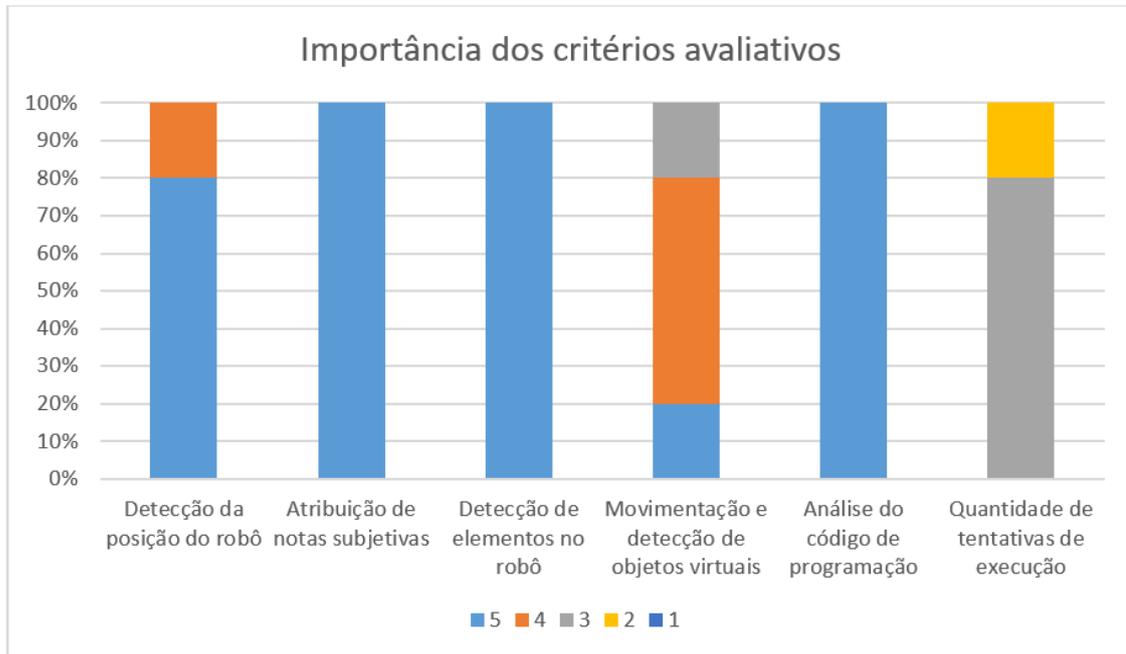


Figura 6.1: Gráfico 1: importância dos critérios avaliativos.

Por fim, o critério de movimentação e detecção de objetos virtuais obteve notas intermediárias. Alguns professores foram além dos testes propostos e realizaram também o cadastro da movimentação de objetos virtuais, e descreveram esta tarefa como simples.

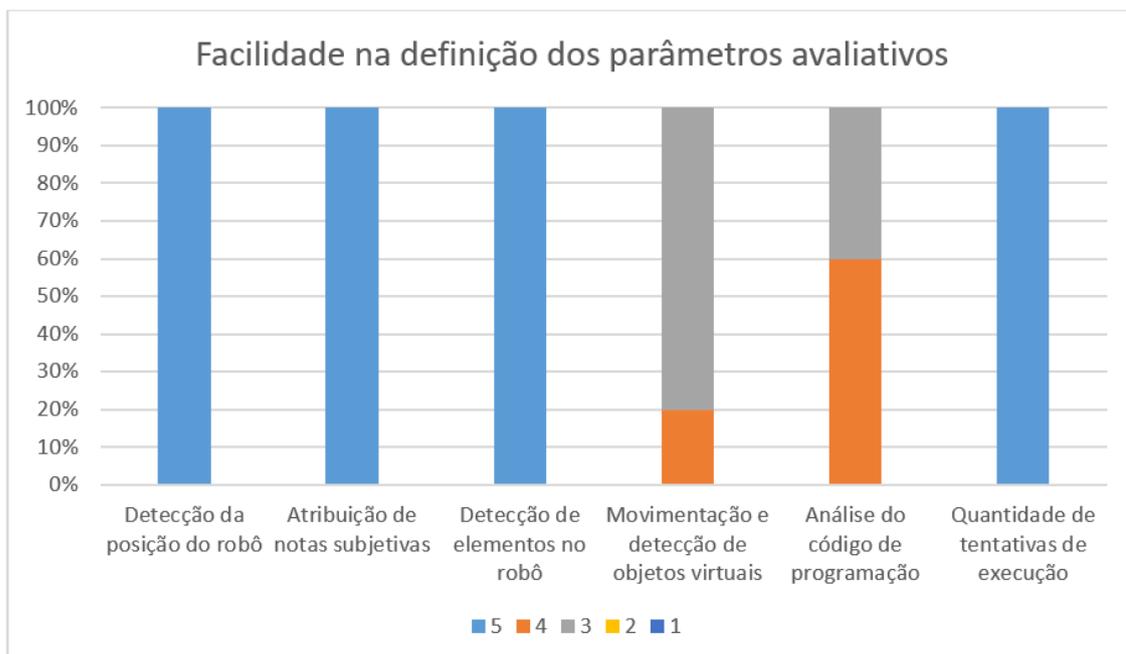


Figura 6.2: Gráfico 2: facilidade da definição dos parâmetros avaliativos.

Com relação à inserção de novos critérios avaliativos, apenas um professor sugeriu a ampliação da análise do código de programação, como já mencionado acima.

Sobre a visualização dos dados obtidos pelo sistema de avaliação, os professores realizaram comentários positivos, indicando que estas informações podem ser utilizadas para reformular suas aulas e focar em alunos com mais dificuldades. Uma sugestão que consideramos importante foi a possibilidade de geração de um relatório para o professor com os dados visualizados nesta área, para que possa ser anexado aos documentos da turma.

Um professor sugeriu também a possibilidade de criação de atividades apenas para um aluno, já que visto um atraso do mesmo com relação à turma, a realização de atividades extra podem ajudá-lo.

Com os dados obtidos nesses experimentos, podemos concluir que todos os professores identificaram benefícios em aplicar esta metodologia de avaliação em aulas de robótica educacional, auxiliando o processo de ensino-aprendizagem, e facilitando uma reformulação de suas aulas de robótica a partir dos dados fornecidos pelo sistema. Nenhum se opôs ao uso da metodologia.

A aplicação desta metodologia de avaliação em sala de aula permite que o professor ajuste sua aula de acordo com o desempenho dos alunos, realizando modificações no processo de ensino focadas para a turma como um todo ou apenas para alguns alunos que apresentaram maiores dificuldades. Os dados fornecidos podem ser utilizados também como ferramenta classificatória, possibilitando a evolução dos alunos em níveis, conforme seu desempenho.

Capítulo 7

Conclusão

É notório que robótica educacional permite estabelecer um ambiente de aprendizagem rico, interdisciplinar, capaz de contribuir para o desenvolvimento da criatividade, do pensamento investigativo, do raciocínio lógico e da coordenação motora. Para a realização de uma aula de robótica seguindo o modelo tradicional, geralmente é necessário o uso de *kits* de robótica, que permitem a criação dos robôs. No entanto, a maioria dos *kits* possuem custo elevado ou restrições com relação à qualidade de *software* e *hardware*.

Almejando tornar a robótica educacional mais acessível às escolas brasileiras, planejamos e desenvolvemos um simulador robótico, chamado S-Educ, que pode ser utilizado em aulas de robótica educacional em substituição ou em complementação ao uso de *kits* de robótica, tradicionalmente utilizados.

Embora a robótica educacional seja uma área em crescente desenvolvimento, com a criação de diversos projetos de pesquisa na área e com a inserção da robótica nas escolas, nos deparamos ainda com uma situação que nos levaram ao problema de pesquisa apresentado no início desta tese: como avaliar os alunos em uma aula de robótica educacional? Será que as técnicas tradicionais de avaliação são suficientes para este tipo de aula?

Visualizando esta lacuna no uso de robótica em sala de aula, criamos uma metodologia de avaliação para aulas de robótica educacional, partindo da hipótese enunciada (e aqui repetida) de que é possível criar uma metodologia de avaliação automática voltada para aulas de robótica educacional, considerando as particularidades deste tipo de aula, que envolvem desde a análise do robô utilizado até o desenvolvimento do programa utilizado para controlá-lo, desde que isto seja realizado de forma guiada pelo professor que, em última instância é o sujeito mediador do processo ensino-aprendizagem.

A metodologia de avaliação desenvolvida pode ser utilizada para a realização de uma avaliação diagnóstica, formativa ou somativa, auxiliando o professor a identificar as deficiências dos alunos, a realizar as mudanças necessárias no processo de ensino e a classificá-los conforme seu rendimento no desenvolvimento das atividades.

Para definir os critérios avaliativos, analisamos os pontos de uma aula de robótica educacional: montagem do robô, programação do robô e entendimento do tema interdisciplinar. Com bases nestes tópicos, criamos seis métricas para avaliação de uma aula de robótica: detecção da posição do robô, atribuição de notas subjetivas,

detecção de elementos do robô, movimentação e detecção de objetos no ambiente virtual, análise do código de programação e quantidade de execuções.

Cada critério avaliativo possui uma formulação matemática para a atribuição da nota do aluno, sendo as notas individuais de cada critério utilizadas para a atribuição da nota final. O professor, ao cadastrar a atividade, pode definir que critérios avaliativos serão utilizados para definir a nota final, e qual o peso de cada um deles nesta nota.

A avaliação de critérios de caráter social e comportamental, como trabalho em equipe, coordenação motora, criatividade, entre outros, pode ser avaliado através da atribuição de uma nota de forma subjetiva pelo professor.

A metodologia desenvolvida pode ser aplicada em sala de aula com o uso de *kits* de robótica, no entanto isto requer ainda bastante envolvimento e interação com o professor, que deve analisar o passo-a-passo de cada grupo individualmente, o que pode se tornar um trabalho bastante custoso.

Com o objetivo de facilitar o processo avaliativo, fornecendo aos alunos um *feedback* rápido, neste trabalho utilizamos essa metodologia juntamente com o simulador de robótica S-Educ, permitindo uma avaliação automática do aluno à medida que ele utiliza o simulador para realizar alguma atividade. Realizamos modificações no simulador para permitir a realização da avaliação automática, modificando também o modelo do banco de dados utilizado.

O fato do simulador realizar uma avaliação automática não exclui o professor do processo avaliativo. Além de ser o responsável por definir os parâmetros avaliativos, o professor pode também atribuir notas subjetivas aos alunos, notas estas que levam em consideração o desempenho do aluno, o comportamento, o trabalho em grupo, entre outros fatores. Além disso, o professor é o responsável por analisar os dados fornecidos pelo simulador. Estes dados podem ser utilizados para guiar o professor para definir novas atividades e auxiliá-lo a identificar em que pontos a turma ou um determinado aluno tem mais dificuldades.

Embora tenhamos utilizado o simulador S-Educ para realizar uma avaliação automática, a metodologia de avaliação proposta pode ser futuramente inserida em qualquer sistema computacional, considerando as particularidades deste sistema. Esta metodologia avaliativa pode também ser aplicada em sala de aula, com o uso de *kits* de robótica, de forma não-automática pelo professor. No entanto, esta estratégia de avaliação exige uma atenção do professor a todos os alunos e todas as suas soluções.

Para validar a nossa hipótese e a ferramenta, realizamos testes utilizando o simulador S-Educ com professores de robótica. O teste foi dividido em duas etapas: definição de parâmetros avaliativos em atividades já inseridas no sistema e análise dos resultados fornecidos pelo sistema depois da execução da atividade por alunos.

Após a realização dos testes, os professores responderam um questionário para fornecer informações sobre a aplicação da metodologia proposta juntamente com o simulador S-Educ. Os professores mostraram muito interesse em aplicar esta metodologia avaliativa em suas aulas de robótica educacional, relatando benefícios de tal metodologia.

Com relação aos critérios avaliativos escolhidos, os critérios de detecção da posição do robô, atribuição de notas subjetivas, detecção de elementos no robô e análise do código de programação apresentaram os melhores resultados com relação à importância dos mesmos em uma aula de robótica educacional. Com relação à facilidade de definição dos parâmetros avaliativos, todos os critérios obtiveram bons resultados.

Por fim, os professores apresentaram grande interesse em analisar os dados fornecidos pelo sistema, relatando que estas informações representam o desempenho do aluno na realização da atividade.

7.1 Trabalhos Futuros

Daremos continuidade a este trabalho, considerando as melhorias sugeridas pelos professores nos experimentos realizados. Para tornar a ferramenta mais abrangível, iremos desenvolver uma versão *online* da mesma, que depende do desenvolvimento do simulador também em sua versão *online*.

Realizaremos também algumas modificações nos critérios avaliativos adicionados ao sistema, com o objetivo de realizar uma avaliação mais completa, fornecendo dados cada vez mais válidos para a avaliação dos alunos. Em trabalhos futuros possibilitaremos que o professor sobrescreva a nota do sistema, recompensando alunos que desenvolveram soluções criativas, diferentes das esperadas.

Com relação ao critério de análise da quantidade de tentativas, possibilitaremos que o professor visualize a quantidade de tentativas dos alunos, mesmo que este critério não seja utilizado na atribuição da nota final, com o fim de fornecer informações que possam auxiliar o professor na análise da atividade proposta.

Pretendemos também permitir que a avaliação subjetiva do professor seja subdividida em diversas categorias, permitindo que o professor avalie pontos como criatividade, trabalho em equipe, habilidades manuais, entre outros critérios. Dessa forma, é possível realizar uma avaliação de critérios sociais que são estimulados em uma aula de robótica educacional. Para estimular também o trabalho em equipe, iremos permitir que alunos executem uma atividade em grupo, permitindo que a nota seja atribuída a todos eles.

Acreditamos que com o trabalho desenvolvido e ainda a desenvolver, podemos contribuir muito, atraindo mais alunos às carreiras tecnológicas, além de estarmos proporcionando melhorias no processo tradicional do ensino-aprendizagem. A criação de novas metodologias de avaliação é de extrema importância quando consideramos uma aula pouco ortodoxa como é uma aula de robótica educacional, na qual apenas a realização de uma prova não é capaz de avaliar o aluno.

Referências Bibliográficas

- ActiveSoft* (n.d.), <http://www.activesoft.com.br/>. Accessed: 2017-05-18.
- Alimisis, Dimitris & George Boulougaris (2014), Robotics in physics education: fostering graphing abilities in kinematics, *em* ‘Proceedings of the 4rd International Workshop Teaching Robotics, Teaching With Robotics 5th International Conference Robotics in Education’, Padova, IT.
- Aroca, Rafael Vidal (2012), Plataforma robótica de baixíssimo custo para robótica educacional, Tese de doutorado, Universidade Federal do Rio Grande do Norte - Programa de Pós-Graduação em Engenharia Elétrica e de Computação.
- Barbosa, João & Anabela Neves (2006), ‘Fantasmas, mitos e ritos da avaliação das aprendizagens’, *Revista Portuguesa de Pedagogia* (40-3), 219–235.
- Barnes, Tiffany, Heather Richter, Eve Powell, Amanda Chaffin & Alex Godwin (2007), Game2learn: building cs1 learning games for retention, *em* ‘ACM SIGCSE Bulletin’, Vol. 39, ACM, pp. 121–125.
- Barroca, Itamir, Gibeon Aquino & José Guilherme Santa Rosa (2013), Sigaa mobile—o caso de sucesso da ferramenta de gestão acadêmica na era da computação móvel, *em* ‘Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)’, Vol. 24, p. 92.
- Becker, Guilherme Lage (2010), ‘Desenvolvimento de um simulador para um veículo autônomo’, *Monografia (Graduação)*. UFMG, Belo Horizonte .
- Benitti, Fabiane Barreto Vavassori (2012), ‘Exploring the educational potential of robotics in schools: A systematic review’, *Computers & Education* **58**(3), 978–988.
- Benitti, Fabiane Barreto Vavassori, Adilson Vahldick, Diego Leonardo Urban, Matheus Luan Krueger & Arvid Halma (2009), Experimentação com robótica educativa no ensino médio: ambiente, atividades e resultados, *em* ‘Anais do Workshop de Informática na Escola’, Vol. 1, pp. 1811–1820.
- Boggino, Norberto (2009), ‘A avaliação como estratégia de ensino. avaliar processos e resultados’, *Sísifo. Revista de Ciências da Educação* **9**, 79–86.

- Buriasco, Regina Luzia Corio, Pamela Emanuelli Alves Ferreira & Andréia Büttner Ciani (2009), 'Avaliação como prática de investigação (alguns apontamentos)', *Bolema: Mathematics Education Bulletin= Bolema: Boletim de Educação Matemática* **22**(33).
- Caldas, Vanessa Martins & Eloi Luiz Favero (2009), Uma ferramenta de avaliação automática para mapas conceituais como auxílio ao ensino em ambientes de educação a distância, *em* 'Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)', Vol. 1.
- Chiou, Andrew (2012), Teaching technology using educational robotics, *em* 'Proceedings of the Australian conference on science and mathematics education (formerly UniServe Science Conference)', Vol. 10.
- Chueiri, Mary Stela Ferreira (2008), 'Concepções sobre a avaliação escolar', *Estudos em Avaliação Educacional* **19**(39), 49–64.
- Class Marker* (n.d.), <https://www.classmarker.com/>. Accessed: 2017-05-12.
- Fernandes, Carla da Costa (2013), S-educ: Um simulador de ambiente de robótica educacional em plataforma virtual, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.
- Fernandes, Domingos (2011), 'Articulação da aprendizagem, da avaliação e do ensino: questões teóricas, práticas e metodológicas'.
- Fornaza, Roseli (2016), Robótica educacional aplicada ao ensino de física, Dissertação de mestrado, Programa de PósGraduação em Ensino de Ciências e Matemática (PPECiMa) Mestrado Profissional em Ensino de Ciências e Matemática, da Universidade de Caxias do Sul (UCS), Caxias do Sul.
- Frangou, Stassini, Kyparissia Papanikolaou, Liliane Aravecchia, Luc Montel, Silviu Ionita, Javier Arlegui, Alfredo Pina, Emanuele Menegatti, Michele Moro, Nello Fava et al. (2008), Representative examples of implementing educational robotics in school based on the constructivist approach, *em* 'SIMPAN Workshop on Teaching with robotics: didactic approaches and experiences, Venice, Italy'.
- Gomes, Rafael Beserra, Luiz Marcos Garcia Gonalves & Bruno Motta de Carvalho (2008), Real time vision for robotics using a moving fovea approach with multi resolution, *em* 'Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on', IEEE, pp. 2404–2409.
- Gonçalves, Paulo C (2007), 'Protótipo de um robô móvel de baixo custo para uso educacional', *Programa de Pós-Graduação da CAPES. Universidade Estadual de Maringá*.
- Ihantola, Petri, Tuukka Ahoniemi, Ville Karavirta & Otto Seppälä (2010), Review of recent systems for automatic assessment of programming assignments, *em*

- ‘Proceedings of the 10th Koli Calling International Conference on Computing Education Research’, ACM, pp. 86–93.
- Jesus, Elieser Ademir (2010), Avaliação Empírica da Utilização de um Jogo para Auxiliar a Aprendizagem de Programação, Dissertação de mestrado, Universidade do Vale do Itajaí.
- Kraemer, Maria EP (2005), ‘A avaliação da aprendizagem como processo construtivo de um novo fazer’, *Revista da Avaliação da Educação Superior, Campinas* **10**(2), 137–147.
- LightBot* (n.d.), <https://lightbot.com/flash.html>. Accessed: 2017-05-14.
- Lin, Chin-Yew & Eduard Hovy (2003), Automatic evaluation of summaries using n-gram co-occurrence statistics, *em* ‘Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1’, Association for Computational Linguistics, pp. 71–78.
- Maratona de Programação* (n.d.), <http://maratona.ime.usp.br/>. Accessed: 2017-05-14.
- Matte, Marília Luiza (2011), ‘A linguagem logo como possibilidade de aprendizagem em matemática’. Trabalho de conclusão de curso - Licenciatura de Matemática, UFRS (Universidade Federal do Rio Grande do Sul), RN, Brasil.
- Michel, Olivier (2004), ‘Cyberbotics Ltd. webotsTM: professional mobile robot simulation’, *International Journal of Advanced Robotic Systems* **1**(1), 5.
- Miranda, Leonardo Cunha de (2006), ‘Robofácil: Especificação e implementação de artefatos de hardware e software de baixo custo para um kit de robótica educacional’, *Rio de Janeiro-RJ* **124**.
- Miranda, Leonardo Cunha, Fábio Ferrentini Sampaio & José Antonio dos Santos Borges (2011), ‘Robofácil: Especificação e implementação de um kit de robótica para a realidade educacional brasileira’, *Brazilian Journal of Computers in Education* **18**(03), 46.
- Montero, Jose Antonio, Francesc Alias, David Badia, David Fonseca & Lluís Vicent (2014), A method for designing automatic assessment systems based on teachers reasoning for evaluating subjective engineering student’s competences, *em* ‘Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on’, IEEE, pp. 1–6.
- Moodle* (n.d.), <https://moodle.org/>. Accessed: 2017-05-10.
- Moreira, Mireille Pinheiro & Eloi Luiz Favero (2009), Um ambiente para ensino de programação com feedback automático de exercícios, *em* ‘Workshop sobre Educação em Computação (WEI 2009)’, Vol. 17.

Mota, Marcelle Pereira, Silvana R de Brito, Mireille Pinheiro Moreira & Eloi Luiz Favero (2009), Ambiente integrado a plataforma moodle para apoio ao desenvolvimento das habilidades iniciais de programação, *em* 'Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)', Vol. 1.

Multiple Choice Scanner (n.d.), <http://www.multiplechoicescanner.com>. Accessed: 2017-05-12.

Neves Júnior, Othon da Rocha et al. (2011), 'Desenvolvimento da fluência tecnológica em programa educacional de robótica pedagógica'.

Nguyen, Tien Dzung, Quyet Hoang Manh, Phuong Bui Minh, Long Nguyen Thanh & Thang Manh Hoang (2011), Efficient and reliable camera based multiple-choice test grading system, *em* 'Advanced Technologies for Communications (ATC), 2011 International Conference on', IEEE, pp. 268–271.

Obst, Oliver & Markus Rollmann (2004), Spark—a generic simulator for physical multi-agent simulations, *em* 'German Conference on Multiagent System Technologies', Springer, pp. 243–257.

Okada, Hugo Kenji Rodrigues & Clayton André Maia Santos (2014), 'Robótica educativa: um sistema de apoio ao aprendizado através de hardware livre', *Anais do Computer on the Beach* pp. 404–406.

Pacheco, José Augusto (1998), 'Avaliação da aprendizagem'. In Leandro Almeida e José Tavares (org.). *Conhecer, aprender e avaliar*. Porto: Porto Editora, pp. 111-132.

Paiva, Taynara Pinheiro de (2016), 'Proposta de implementação de um sistema gamificado voltado para o ensino'. Trabalho de conclusão de curso - Engenharia de Computação, UFRN (Universidade Federal do Rio Grande do Norte), RN, Brasil.

Pedrosa, Eurico Farinha (2010), Simulated environment for robotic soccer agents, Dissertação de mestrado, Universidade de Aveiro.

Pelz, Fillipi D, Elieser A de Jesus & André LA Raabe (2012), Um mecanismo para correção automática de exercícios práticos de programação introdutória, *em* 'Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)', Vol. 23.

Pitta, Renata (2008), 'Roboeduc-uma ferramenta para programação de robôs lego', *Trabalho de conclusão de curso, Universidade Federal do Rio Grande do Norte*.

Pitta, Renata (2011), Evolução, avaliação e validação do software roboeduc, Dissertação de mestrado, Master's thesis, Universidade Federal do Rio Grande do Norte.

- Prado, José Pacheco de Almeida (2008), 'Robôs estarão disponíveis para estudantes brasileiros'. Disponível em: <<http://www.acessasp.sp.gov.br/html/modules/news/article.php?storyid=466>>. Acesso em: 10 de maio de 2017.
- QCM direct* (n.d.), <https://neoptec.com/produits/qcm-direct/>. Accessed: 2017-05-12.
- Queiroz, Dalva Maria de (2010), A avaliação como acompanhamento sistêmico da aprendizagem: uma experiência de investigação-ação colaborativa no ensino fundamental, Tese de doutorado.
- RoboCode* (n.d.), <http://robocode.sourceforge.net/>. Accessed: 2017-05-14.
- RoboCup 2016* (n.d.), <http://www.robocup2016.org/en/>. Accessed: 2017-05-10.
- Rocha, Francisco Edson Lopes da (2007), Avaliação da aprendizagem: uma abordagem qualitativa baseada em mapas conceituais, ontologias e algoritmos genéticos, Tese de doutorado, Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal do Pará.
- Romeiro, Alice de La Rocque (2000), 'Um olhar sobre a avaliação hoje', *Salto para o futuro: Um olhar sobre a escola; Secretaria de Educação a Distância, Brasília: Ministério da Educação, Seed*.
- Sá, Sarah Thomaz de Lima (2013), Ambiente de desenvolvimento web multiplataforma configurável para robótica educacional, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.
- Santos, Carmen Faria & Crediné Silva de Menezes (2005), A aprendizagem da física no ensino fundamental em um ambiente de robótica educacional, em 'Anais do Workshop de Informática na Escola', Vol. 1.
- Santos, João Francisco Severo (2006), 'Avaliação no ensino a distância', *Revista Iberoamericana de Educación* **38**(4), 1–9.
- Santos Jr, Eugene, Ahmed A Mohamed & Qunhua Zhao (2004), Automatic evaluation of summaries using document graphs, em 'Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL 2004) workshop on text summarization branches out, Barcelona, Spain', pp. 66–73.
- Santos, Leonor (2003), 'Avaliar competências: uma tarefa impossível', *Educação e Matemática* **74**, 16–21.
- Silva, Alzira Ferreira, Alynara Silva, Luiz Marcos Garcia Gonçalves, Ana Maria Guimaraes Guerreiro & Dennis Barrios-Aranibar (2008), Utilização da teoria de vygotsky em robótica educativa, em 'em 'Caracas-Venezuela. Congresso Iberoamericano de Informatica Educativa RIBIE 2008'.

- Silva, Alzira Ferreira da (2009), RoboEduc: Uma metodologia de aprendizado com Robótica Educacional, Tese de doutorado, Programa de Pós-Graduação em Engenharia Elétrica e de Computação - Universidade Federal do Rio Grande do Norte.
- Soares, Renato Ferreira & Marcos Augusto Francisco Borges (2011), Robótica: aprendizado em informática de forma lúdica, em 'WEI-XXXI Congresso da Sociedade Brasileira de Computação, Natal-RN: SBC'.
- Souza, Draylson Micael, José Carlos Maldonado & Ellen Francine Barbosa (2012), Aspectos de desenvolvimento e evolução de um ambiente de apoio ao ensino de programação e teste de software, em 'Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)', Vol. 23.
- Sá, Sarah Thomaz de Lima (2016), W-Educ: um ambiente web, completo e dinâmico para robótica educacional, Tese de doutorado, Universidade Federal do Rio Grande do Norte.
- Tavares, Paula Correia, Elsa Ferreira Gomes & Pedro Rangel Henriques (2015), Animation and automatic evaluation in supporting the teaching of programming, em 'Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on', IEEE, pp. 1-4.
- Wolf, Denis Fernando, EV Simões, Fernando S Osório & Onofre Trindade Junior (2009), Robótica móvel inteligente: Da simulação às aplicações no mundo real, em 'Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC', p. 13.
- Yujian, Li & Liu Bo (2007), 'A normalized levenshtein distance metric', *IEEE transactions on pattern analysis and machine intelligence* **29**(6), 1091-1095.
- Zampirolli, Francisco, Valério Ramos Batista & José Artur Quilici-Gonzalez (2016), An automatic generator and corrector of multiple choice tests with random answer keys, em 'Frontiers in Education Conference (FIE), 2016 IEEE', IEEE, pp. 1-8.
- Zanatta, Ronnie Petter Pereira (2013), A robótica educacional como ferramenta metodológica no processo ensino-aprendizagem: uma experiência com a segunda lei de newton na série final do ensino fundamental, Dissertação de mestrado, Universidade Tecnológica Federal do Paraná.
- Zilli, Silvana do Rocio et al. (2004), A robótica educacional no ensino fundamental: perspectivas e prática, Dissertação de mestrado, Programa de Pós-Graduação em Engenharia de Produção - Universidade Federal de Santa Catarina.

Apêndice A

Descrição dos testes: definição dos critérios avaliativos

Atividade 1: Seguir Linha

Objetivo: o robô deve seguir uma linha preta no chão.

- Entre no sistema como professor
 - Usuário: professor_teste
 - Senha: 123456
- Selecione a turma Turma 1 - 2017
- Selecione a atividade Seguir Linha
- Adicione os seguintes critérios avaliativos (escolha os pesos que preferir):
 - Caminho percorrido
 - Código
 - Partes do robô
 - Atribuir pontos
- Vá para a área de simulação para definir os parâmetros de cada um dos critérios avaliativos
- Vá para Avaliações
- Cadastrar avaliação: **Marcar pontos:**
 - Marcar cinco pontos em cima da linha
 - Salvar os pontos no banco de dados
 - Limpar tela, para remover os pontos da tela
 - Mostrar pontos, para visualizar se os pontos foram salvos
- Cadastrar avaliação: **Código:**
 - Adicionar um código (Algoritmo 5)
 - Adicionar um código (Algoritmo 6)
- Cadastrar avaliação: **Partes do robô:**
 - Cor: baixo esquerda
 - Cor: baixo direita

Algoritmo 5: Código 1 para a Atividade 1: Seguir Linha.

```
1 enquanto (1=1) farei {
2     se (cor(1)=branco e cor(2)=branco) entao {
3         frente
4     }
5     se (cor(1)=branco e cor(2)=preto) entao {
6         girar
7     }
8     se (cor(1)=preto e cor(2)=branco) entao {
9         girar
10    }
11    se (cor(1)=preto e cor(2)=preto) entao {
12        frente
13    }
14 }
```

Algoritmo 6: Código 2 para a Atividade 1: Seguir Linha.

```
1 enquanto (1=1) farei {
2     se (cor(1)=branco e cor(2)=branco ou cor(1)=preto e cor(2)=preto)
3     entao {
4         frente
5     }
6     se (cor(1)=branco e cor(2)=preto) entao {
7         girar
8     }
9     se (cor(1)=preto e cor(2)=branco) entao {
10        girar
11    }
12 }
```

Atividade 2: Segunda Guerra Mundial

Objetivo: o robô será colocado em cima de um mapa mundi, e se guiará através de paredes (obstáculos). Ao encontrar uma linha no chão de uma determinada cor, o robô deve acender uma luz da cor da linha identificada, e deve escrever na tela qual o país e qual foi a sua participação na guerra. Estas linhas representam países que participaram da Segunda Guerra Mundial.

- Voltar à tela principal
- Selecione a turma Turma 1 - 2017
- Selecione a atividade Segunda Guerra Mundial
- Adicione os seguintes critérios avaliativos (escolha os pesos que preferir):
 - Caminho percorrido
 - Partes do robô
 - Quantidade de execuções
 - Detectar objetos
- Vá para a área de simulação para definir os parâmetros de cada um dos critérios avaliativos
- Vá para Avaliações
- Cadastrar avaliação: **Marcar pontos:**
 - Marcar três pontos em cima das linhas
 - Salvar os pontos no banco de dados
 - Limpar tela, para remover os pontos da tela
 - Mostrar pontos, para visualizar se os pontos foram salvos
- Cadastrar avaliação: **Partes do robô:**
 - Cor: cima
 - Cor: baixo meio
 - Ultra: frente
- Cadastrar avaliação: **Quant. de tentativas:**
 - Definir 15 tentativas
- Cadastrar avaliação: **Detectar objetos:**
 - Selecionar cada uma das linhas coloridas e indicar que deve ser detectada com: **Cor: baixo meio**
 - Expressões:
 - * = vermelho
 - * = azul
 - * = verde

Apêndice B

Descrição dos testes: análise dos resultados dos alunos

- Entre no sistema como professor
 - Usuário: professor_teste
 - Senha: 123456
- Selecione a turma Turma 2 - 2017
- Selecione a atividade Seguir Linha
- Finalize a turma
- Atribua notas aos alunos da turma que não tem nota ainda
- Selecione a turma Turma 2 - 2017
- Selecione a atividade Segunda Guerra Mundial
- Finalize a turma
- Vá para a área de visualização de resultados
- Analise os resultados das atividades da Turma 2 - 2017

Apêndice C

Questionário aos professores

1. Sobre o uso de avaliação automática, classifique o seu interesse em utilizar esta ferramenta em uma aula de robótica educacional, onde (1) não possui interesse e (5) possui muito interesse:

(1) (2) (3) (4) (5)

Comente:

2. Sobre o uso de avaliação automática, classifique os benefícios do uso desta ferramenta em uma aula de robótica educacional, onde (1) não possui benefícios e (5) possui muitos benefícios:

(1) (2) (3) (4) (5)

Comente:

3. Sobre os critérios de avaliação, classifique a importância deles para a avaliação de alunos em aulas de robótica educacional, onde (1) não é importante e (5) é muito importante:

(a) Detecção da posição do robô (1) (2) (3) (4) (5)

(b) Atribuição de notas subjetivas (1) (2) (3) (4) (5)

(c) Detecção de elementos no robô (1) (2) (3) (4) (5)

(d) Movimentação e detecção de objetos virtuais (1) (2) (3) (4) (5)

(e) Análise do código de programação (1) (2) (3) (4) (5)

(f) Quantidade de tentativas de execução (1) (2) (3) (4) (5)

4. Sobre os critérios de avaliação, classifique a facilidade da definição dos parâmetros avaliativos, onde (1) é muito difícil e (5) é muito fácil:

(a) Detecção da posição do robô (1) (2) (3) (4) (5)

(b) Atribuição de notas subjetivas (1) (2) (3) (4) (5)

(c) Detecção de elementos no robô (1) (2) (3) (4) (5)

(d) Movimentação e detecção de objetos virtuais (1) (2) (3) (4) (5)

(e) Análise do código de programação (1) (2) (3) (4) (5)

(f) Quantidade de tentativas de execução (1) (2) (3) (4) (5)

5. Comente sobre os critérios avaliativos inseridos no sistema.

6. Que outros critérios de avaliação você sugere inserir no sistema?

7. Comentários sobre a visualização dos dados obtidos pelo sistema de avaliação:

8. Comentários gerais:
