

Uma Metodologia Baseada em Semiótica para Elaboração e Análise de Práticas de Ensino de Programação com Robótica Pedagógica

Humberto A. P. Zanetti¹, Rodrigo Bonacin^{1,2}

¹Faculdade Campo Limpo Paulista – FACCAMP – Campo Limpo Paulista – SP – Brasil

²Centro de Tecnologia da Informação Renato Archer (CTI) – Campinas – SP – Brasil

hzanetti@gmail.com, rodrigo.bonacin@cti.gov.br

Abstract. *This paper presents a methodology for conception, elaboration and analysis of pedagogical practices for teaching computer programming using Educational Robotics. This methodology uses Organizational Semiotics concepts and methods aiming to provide a systematic process to analyse, from the semiotics' point of view, the development of programming skills on practices integrated to disciplines of technological and higher education courses. The paper also illustrates the application of the process in a practice with six students of a higher education technical school.*

Resumo. *Este artigo apresenta uma metodologia para a concepção, elaboração e análise de práticas pedagógicas para ensino de programação de computadores por meio de Robótica Pedagógica. Essa metodologia utiliza métodos e conceitos provenientes da Semiótica Organizacional com o objetivo de constituir um processo sistemático para analisar, do ponto de vista da semiótica, o desenvolvimento de habilidades de programação em práticas integradas a disciplinas de cursos superiores em tecnologia e cursos técnicos do ensino médio. O artigo também ilustra a aplicação do processo em uma prática com seis alunos de uma escola técnica.*

1. Introdução

Grande parte dos alunos ingressantes em cursos de Computação tem contato com programação de computadores pela primeira vez nas disciplinas introdutórias nos primeiros semestres. Infelizmente, como aponta o trabalho de Gomes, Henriques e Mendes (2008), essas disciplinas são responsáveis por altos índices de evasão e reprovações, que promovem muitas vezes a desmotivação por parte do aluno. Vale destacar que começar a ensinar detalhes sintáticos antes que os alunos percebam qual são as finalidades e utilidades dos conceitos de programação pode potencializar essa desmotivação.

O processo de aprendizagem dos conceitos iniciais de programação é complexo e, muitas vezes, conforme destaca o trabalho de Gomes et al. (2008), necessita de um nível de abstração que não está presente na maioria dos alunos iniciantes. Várias pesquisas apontam soluções que visam colaborar para a criação de um ambiente mais diversificado e motivador para o aluno. D'Abreu (2002) e Benitti (2012) indicam que entre as soluções e ferramentas possíveis destaca-se a Robótica Pedagógica (RP). A RP

apresenta um vasto conjunto de recursos e possibilidades para o desenvolvimento de práticas pedagógicas mais estimulantes que as tradicionais execuções simuladas em quadro negro. No aspecto teórico da Educação, Papert (1993) aponta que a utilização de robôs como artefatos educacionais tem um grande potencial para prover um ambiente favorável ao aprendizado dentro da sala de aula.

Entretanto, conforme destacado por Benitti (2012), ainda é preciso um método sistemático de elaboração e avaliação de práticas pedagógicas utilizando robótica. Em um currículo de cursos técnicos ou de graduação a construção dessas práticas devem refletir o conteúdo programático envolvido, bem como os conceitos fundamentais que devem ser aprendidos. Consequentemente, há a necessidade de criar meios avaliativos que sejam efetivos e que colabore para a apresentação de resultados fidedignos.

Neste contexto, o artigo propõe uma metodologia baseada em semiótica para concepção, elaboração e análise de práticas. Como parte desta metodologia, é proposto um processo que visa desenvolver meios para pesquisadores analisar os efeitos destas práticas com alunos. O objetivo é apresentar uma proposta e discutir como a semiótica pode auxiliar na identificação de benefícios e desafios do uso da RP no ensino de programação. Para tanto, são empregados métodos da Semiótica Organizacional (SO) em conjunto com a RP. O artigo ainda apresenta um estudo com seis alunos e respectivas respostas a questionamentos sobre o uso da RP no processo proposto.

O artigo apresenta na Seção 2 uma breve descrição sobre a RP e sua aplicação no ensino de programação de computadores; na Seção 3 são abordados métodos da SO; na Seção 4 é apresentada a metodologia; na Seção 5 é destacada a aplicação em uma prática pedagógica; e por fim, na Seção 6 são apresentadas as considerações finais.

2. A Robótica Pedagógica no ensino de Programação de Computadores

A RP já está presente em diversas instituições de ensino e pesquisa, sendo usada como artifício para criar um ambiente de ensino mais atraente e diversificado. Segundo d'Abreu (2002), a RP propicia a montagem, automação e controle de dispositivos mecânicos. Ela também auxilia na interação entre aluno e professor com as ferramentas tecnológicas, construindo conhecimento e possibilitando um ambiente pedagógico que não existia a princípio. Assim, com a RP os alunos podem trabalhar a conexão de peças mecânicas e de componentes, além de estimular o desenvolvimento de habilidades de liderança e trabalho em equipe [Papert 1994].

A aprendizagem de programação de computadores é um processo difícil, sendo necessário um alto nível de abstração e compreensão de linguagens com sintaxe complexa. Gomes, Henriques e Mendes (2008) salientam que falta aos alunos iniciantes competências necessárias para resolução dos problemas. Um dos maiores entraves está no fato de um conteúdo com caráter dinâmico ser apresentado de forma estática.

Métodos tradicionais de ensino de programação utilizam fluxogramas e pseudocódigos para demonstração de conceitos abstratos. A adoção de RP traz novos recursos de ensino que ajudam a apresentar esses conceitos. Com a RP é possível observar o impacto de instruções em um objeto concreto e associar os comandos abstratos aos movimentos e ações do mesmo [Gomes, Henriques e Mendes 2008]. Segundo Zilli (2004) a utilização da RP contribui para o desenvolvimento de

competências necessárias para alunos iniciantes em programação, como: raciocínio lógico, representação e comunicação, resolver problemas por meio de erros e acertos, aplicação das teorias formuladas em atividades práticas e capacidade crítica.

A RP pode ser utilizada em conjunto com o Construcionismo, teoriadesenvolvida por Seymour Papert que baseia-se no indivíduo criar significados por meio de experiências e ações de autoaprendizagem, sendo elemento ativo na construção do seu conhecimento [Papert 1994]. Segundo Valente (1993), o Construcionismo promove o aprendizado através de um processo de formulação de hipóteses, testes e avaliação.

Com um foco mais prático e complementar à teoria do Construcionismo, temos a metodologia *Problem Based Learning* (PBL), que é eficaz para o ensino de conceitos tecnológicos e que propõe o “aprender a aprender”. Segundo Soares e Borges (2011), nessa metodologia o próprio aluno é responsável pela criação do aprendizado, e o principal papel do professor é agir como um intermediador ou facilitador.

3. A Semiótica Organizacional aplicada ao Ambiente de Ensino

A SO utiliza conceitos e técnicas baseadas na Semiótica para analisar organizações e o comportamento organizado. De acordo com Liu (2000), no contexto da SO entende-se como organização qualquer domínio que demande comportamento organizado, tal como um ambiente de ensino. Para Stamper (1973) a comunicação pode ser dividida em seis níveis que formam um framework semiótico: social, pragmático, semântico, sintático, empírico e físico.

O MEASUR (*Methods for Eliciting, Analysing and Specifying User Requirements*) é um conjunto de métodos orientados às normas para lidar com signos e seus significados, intenções e consequências sociais. Com o PAM (*Problem Articulation Methods*) é possível identificar os objetos a serem trabalhados no ambiente de ensino nos diversos níveis do *framework* semiótico. Enquanto o método SAM (*Semantic Analysis Method*) tem como foco os agentes e suas invariantes de comportamento, como por exemplo, invariantes de comportamentos do professor e aluno pertinentes ao contexto de aplicação da RP. O NAM (*Norm Analysis Method*) possibilita a especificação de padrões comportamentais dos agentes em sociedade que participam do sistema organizacional e a análise é realizada por meio da especificação na forma de normas.

Os conceitos fundamentais do MEASUR incluem agentes, *affordances* e normas. Para Liu (2000), agente é algo responsável por *affordances* e pode-se tomar como exemplo “aluno”, “professor” ou mesmo a instituição de ensino. O conceito de *affordance*, introduzido por Gibson (1968), pode ser utilizado para denominar a propriedade ou o comportamento de um objeto, sistema ou organismo combinada com as características de seu ambiente. O conceito de Gibson foi expandido por Stamper (1973) para incluir invariantes que são percebidas no mundo social. Uma “estrutura de seleção” é um exemplo de um *affordance* do agente “aluno”, que só é possível pelas habilidades do aluno com o “ambiente de programação” (que é outro *affordance*).

O conceito de normas define as responsabilidades dos agentes envolvidos em alguma tarefa, ou as condições sob as quais certas ações podem ou não ser realizadas.

Normas correspondem no nível social à ideia de *affordance* no nível individual. Esse conceito baseia-se na teoria de normas criada por Stamper e outros (2000), que inclui normas: perceptuais, cognitivas, avaliativas, denotativas e comportamentais. No processo proposto, estas normas resultam em requisitos-chaves de aprendizagem a serem explorados nas práticas com alunos, assim como elementos¹ a serem analisados após as práticas. Demais ferramentas utilizadas (e.g., Cebola Semiótica, lógica deôntica, e *framework* semiótico) são detalhadas em Liu (2000).

4. A Metodologia e Processo Proposto

Estudos preliminares [Zanetti e Bonacin 2014] exploraram a aplicação do *framework* semiótico e do NAM para a descrição de atividades de RP. Neste artigo, é proposto sistematizar o emprego do PAM, SAM e NAM no processo PDAP-RP (Processo de Desenvolvimento e Análise de Práticas com uso de Robótica Pedagógica). A Figura 1 apresenta uma visão geral do PDAP-RP, com linhas tracejadas que indicam fases distintas conforme descritas a seguir:

1. *Identificação do cenário e problemas* – Cenário são os elementos e ambientes necessários para a prática e problemas são as dificuldades na obtenção de um determinado objetivo. Nessa fase, o professor-pesquisador faz o levantamento inicial dos recursos existentes e quais são os objetivos. Ela é composta por três atividades: (1.1) *Delimitação do Cenário/Problema*: descrição do cenário e levantamento dos principais problemas; (1.2) *Análise de Stakeholders*: analisa os *stakeholders* (partes interessadas) envolvidos através da “cebola semiótica” [Liu, 2000]; e (1.3) *Uso de Framework Semiótico*: como guia para elicitação e análise de problemas e soluções das práticas usando RP. A fase visa classificar e organizar os problemas e soluções relacionadas ao domínio dos objetivos didáticos do nível físico ao nível social.
2. *Definição e modelagem de agentes, comportamentos e dependências ontológicas* – É realizada a análise e modelagem dos conceitos relacionados ao SAM. Em uma prática com RP os agentes (e.g., alunos, professor) possuem *affordances* relacionados aos recursos de programação (e.g., estruturas de repetição, seleção) e a interação com o robô (e.g., reconhecimento de obstáculo, movimento). Esta fase inclui as seguintes atividades (inspiradas em Liu (2000)): (2.1) *Definição de Agentes e Affordances*: onde são listados candidatos a agentes e *affordances* a partir das descrições do problema; (2.2) *Agrupamento de Candidatos*: análise das unidades semânticas; (2.3) *Detalhamento de comportamentos e ações*: mapeamento as dependências existentes entre as unidades semânticas; e (2.4) *Diagramação de Ontologias*: criação de um modelo de representação gráfica para conceitos como: agentes (círculos), *affordances* (retângulos) e dependências (linhas lidas da esquerda para direita).
3. *Análise e modelagem de normas* – Primeiramente são modeladas as normas comportamentais que irão compor os quadros de descrição das tarefas a serem realizadas pelos alunos. As normas referem-se essencialmente às relações dos agentes com seus *affordances*. Grande parte dessas relações dizem respeito ao aluno, ao robô e à lógica de programação, incluindo aspectos relacionados aos recursos de

¹ “Elementos” no contexto de artigo são entendidos como objetos necessários podendo ser físicos (e.g., arena, obstáculos) e conceituais (e.g., objetivos, problemas).

hardware disponível, à comunicação e à linguagem usada entre aluno e robô. Esta fase inclui as seguintes atividades: (3.1) *Análise de Normas*: é a identificação, modelagem e análise das normas que guiam a prática pedagógica; (3.2) *Elaboração de quadro com normas comportamentais*: onde são produzidos quadros [Zanetti e Bonacin 2014] contendo os agentes envolvidos, os *affordances* e descrição em Lógica Deontica; (3.3) *Elaboração de quadro de respostas (demais normas)*: Uma vez definidas as normas comportamentais, é criado um quadro com os demais tipos de normas com os resumos das respostas dos alunos e suas impressões quanto aos aspectos perceptuais, cognitivos, denotativos e avaliativos. O objetivo é identificar se todas as ações foram realizadas, percebidas e compreendidas corretamente.

4. *Aplicação de práticas* – Nessa fase é aplicada a prática pedagógica em si. Para tanto, é definido qual o ambiente de aplicação. Essa fase engloba a avaliação e verificação dos resultados, que podem levar à revisão de outros pontos da metodologia e modelos em um processo iterativo. Esta fase inclui as atividades: (4.1) *Definição das práticas usando RP*: onde é definido o ambiente de aplicação da prática e quais elementos que o robô deverá possuir (e.g., sensores e estrutura física); e (4.2) *Avaliação dos resultados*: o professor-pesquisador observa e determina, através dos pontos definidos nos quadros de normas, se as ações desenvolvidas pelos alunos e robô atenderam ao que era esperado. Questionários e entrevistas sintetizados através de quadros de respostas complementam a avaliação.

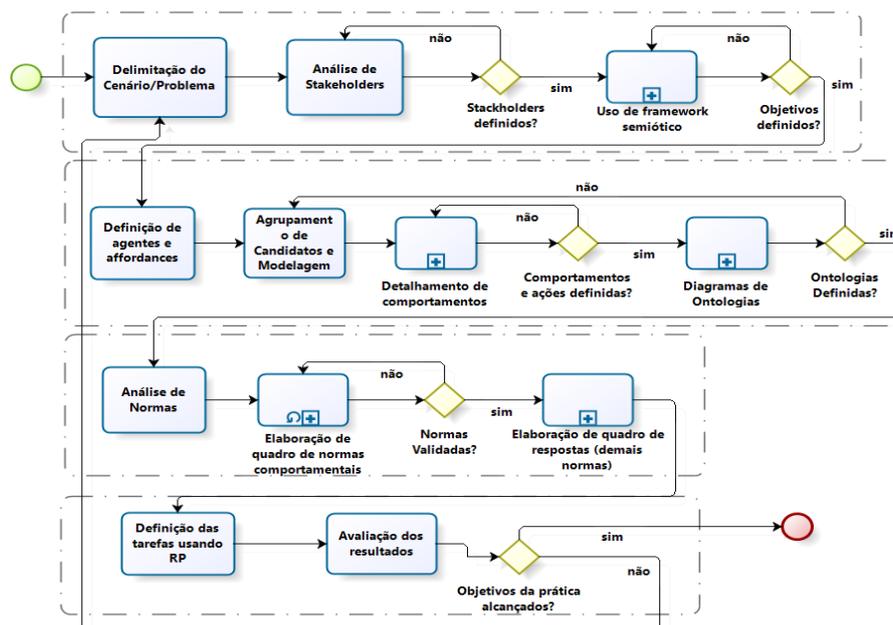


Figura 1. Visão geral do PDAP-RP

5. Aplicação e Resultados

A prática realizada teve como objetivo trabalhar dois conceitos de programação: (1) a estrutura de seleção (*if-else*) e a estrutura de repetição (*while*). Conforme ilustra a Figura 2, ela foi executada em uma arena na cor branca, sendo demarcada uma faixa na cor preta que indicava o percurso que o robô devia seguir. O robô contava com um sensor identificador de cor, para verificar se o mesmo se encontra sobre a faixa. Para coordenar

a movimentação do robô sobre o percurso foi necessário desenvolver uma programação com a seguinte verificação: “se encontrar branco, gire 30° graus para a esquerda até encontrar a faixa preta, caso contrário volte à posição inicial e gire 30° à direita até encontrar a faixa preta”. A subseção 5.1 apresenta a aplicação do método durante esta prática e a subseção 5.2 faz uma breve análise dos resultados obtidos.



Figura 2. Visão da arena e preparação dos alunos

5.1. Aplicação do Método PDAP-RP

Para a realização da prática foram selecionados seis alunos (duas alunas e quatro alunos) do curso Técnico em Informática integrado ao Ensino Médio, da Escola Técnica Rosa Perrone Scavone, localizada na cidade de Itatiba-SP. Todos os alunos têm idade entre quinze e dezessete anos, atualmente estão no 2º ano do curso, já cursaram disciplinas básicas de Lógica de Programação e Programação de Computadores e possuíam contato prévio com robótica. O grupo é formado por alunos em processo de aprendizagem de programação e ainda são afetados pelas dificuldades em disciplinas relacionadas à programação de computadores. A prática foi realizada em um dos laboratórios com duração de aproximadamente três horas.

A Tabela 1 apresenta a aplicação da Fase 1 do PDAP-RP, sendo composta pela delimitação do cenário/problema, análise do principais *stakeholders* e o uso de *framework* semiótico. Como mostra a Figura 3, na “Cebola Semiótica” (lado esquerdo) os problemas relacionados à prática e *stakeholders* estão categorizados de acordo com os níveis técnico (*e.g.*, manuseio e comunicação), formal (*e.g.*, procedimentos de ensino) e informal (*e.g.*, cultura do ambiente). A Figura 3 também exhibe o uso do *framework* semiótico (lado direito), onde aspectos da plataforma tecnológica (*e.g.*, recursos de hardware e software) e do sistema de informação humano (*e.g.*, atuação profissional do aluno) estão categorizados nos seis níveis do *framework*.

Tabela 1. Atividades e resultados da Fase 1 do PDAP-RP

Fase 1: Identificação do cenário e problemas	
1.1. Delimitação do Cenário/Problema	Descrição do cenário listando os principais conceitos a serem trabalhados e a descrição dos objetivos a serem alcançados de maneira ampla.
1.2. Análise de Stakeholders	Lista com a definição dos papéis dos <i>stakeholders</i> professor e alunos. Nesse contexto, além de aplicar a prática, o professor tem o papel de facilitador e orientador. Os alunos têm como objetivo a resolução do problema proposto, identificação da solução e execução através da programação do robô. A Figura 3 (esquerda) apresenta a Cebola Semiótica utilizada na análise.
1.3. Uso de Framework Semiótico	Problemas e soluções classificadas de acordo com os níveis do <i>framework</i> . Como exemplo, no nível físico, foram apontados quais os recursos necessários ao robô (componentes estruturais, motores e sensores); e no pragmático, a listagem das principais soluções de programação necessárias que os alunos devem abstrair (<i>e.g.</i> , estrutura de seleção, conceito de operadores lógicos, estruturas de repetição). O <i>framework</i> é apresentado na Figura 3 (direita).

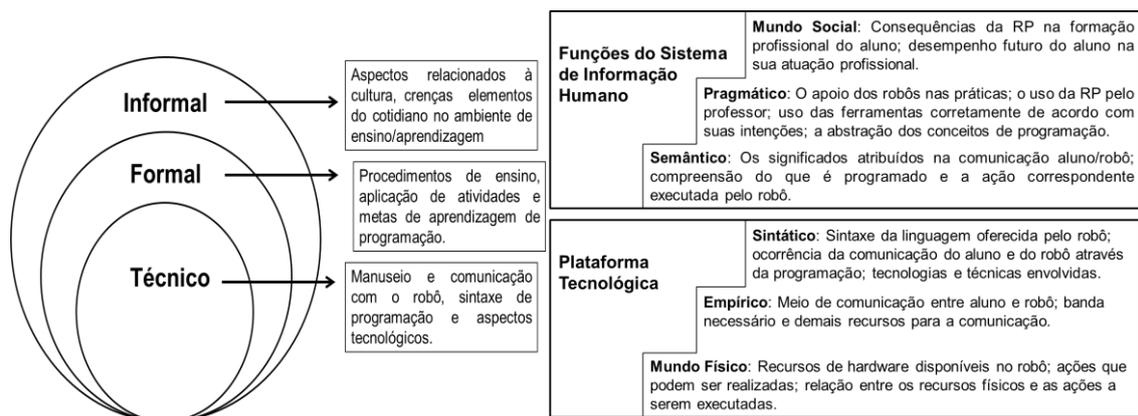


Figura 3. Cebola e Framework Semiótico

A Tabela 2 apresenta a Fase 2, cujo objetivo principal é analisar os agentes e seus *affordances*. A Figura 4 apresenta uma versão simplificada do diagrama de ontologia com agentes e papéis (e.g., professor e aluno de programação), *affordances* a serem trabalhados (e.g., identificar traçado e desviar de obstáculos) e dependências ontológicas (e.g, a programação do robô depende do robô e do aluno).

Tabela 2. Atividades e resultados da Fase 2 do PDAP-RP

Fase 2: Definição e modelagem de agentes, comportamentos e dependências ontológicas	
2.1. Definição de Agentes e Affordances	Foram definidos os agentes, incluindo “aluno” e “professor” como os principais responsáveis por vários <i>affordances</i> .
2.2. Agrupamento de Candidatos e modelagem de ontologias	Tabela listando os agentes e <i>affordances</i> encontrados na etapa anterior. Organizamos a tabela em duas colunas, sendo uma para as palavras candidatas a se tornarem <i>affordances</i> , e outra coluna com as classificações e informações que possam explicar suas funções.
2.3 Detalhamento de comportamentos e ações	Refinamento do artefato da etapa anterior, listando as dependências existentes entre as unidades semânticas, a partir de suas funções, comportamentos e ações.
2.4 Diagrama de Ontologia	Diagrama de Ontologia contendo as relações entre agentes e <i>affordances</i> (Figura 4).

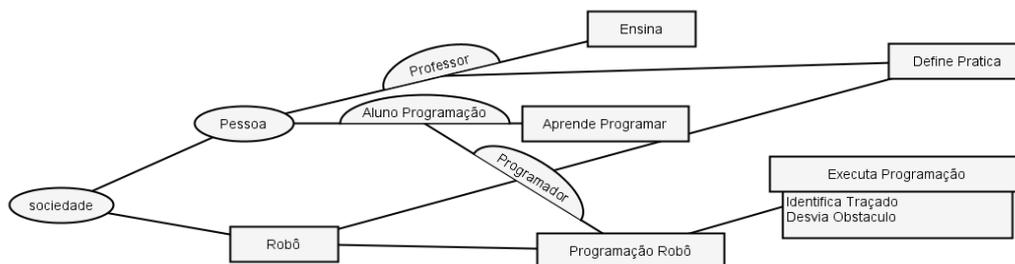


Figura 4. Diagrama de Ontologia

A Tabela 3 descreve a execução da Fase 3 incluindo a análise de normas e a elaboração de quadros descritivos das normas comportamentais, como no exemplo apresentado na Tabela 4. Esse exemplo detalha que o aluno é obrigado a descrever uma estrutura de seleção para o robô desviar de um obstáculo.

Tabela 3. Atividades e resultados da Fase 3 do PDAP-RP

Fase 3: Análise e modelagem de normas	
3.1. Análise de Normas	Descrição informal dos comportamentos dos agentes nos níveis pragmático e social.
3.1.1. Elaboração de quadro com normas comportamentais	Desenvolvimento de quadros que descrevem as normas comportamentais, contendo especificação sobre quais agentes estão envolvidos no cenário, quais são os <i>affordances</i> e o detalhamento dessa norma através do uso da Lógica Deontica para definir regras e ações, conforme Tabela 4.
3.1.2. Elaboração de	Tabela qualificando as diferentes normas. Nela consta questões relacionadas às diferentes normas

<i>quadro de respostas</i>	organizadas na seguinte maneira: Normas Perceptuais: O resultado da tarefa foi facilmente percebido? Teve alguma dificuldade em perceber os efeitos dos conceitos de programação? Normas Cognitivas: Houve dificuldades na compreensão da tarefa? Conseguiu compreender plenamente os efeitos dos conceitos no robô? Normas Avaliativas: Os resultados obtidos foram satisfatórios? Como avalia os efeitos obtidos? Normas Denotativas: Os conceitos de programação foram escolhidos adequadamente de acordo com a categoria/tipo de tarefa?
----------------------------	--

Tabela 4. Exemplo de quadro de normas comportamentais

Norma	<i>Desviar de um obstáculo na arena</i>
Agentes	Aluno (no papel de programador)
Affordances	Robô (<i>affordance</i> da Sociedade); Programação Robô (<i>affordance</i> aluno com o Robô); Desvia Obstáculo (<i>affordance</i> aluno por meio da programação do robô)
Detalhamento da norma	sempre que robô identificar um obstáculo na arena; se é preciso desviar desse obstáculo; então , o aluno; é obrigado a; utilizar uma estrutura de seleção para determinar o desvio

A Tabela 5 apresenta a Fase 4, com as atividades relacionadas à aplicação das práticas. Os resultados são descritos e comentados na próxima subseção.

Tabela 5. Atividades e resultados da Fase 4 do PDAP-RP

Fase 4: Aplicação de Práticas	
4.1. Definição das tarefas usando	Descrição da arena e tarefa conforme destacado anteriormente.
4.2. Avaliação dos resultados	Tabulação das respostas dos alunos a partir das normas perceptuais, cognitivas, avaliativas e denotativas (a próxima subseção descreve respostas às normas perceptuais e cognitivas).

5.2. Análise dos Resultados

Para analisar os resultados obtidos na prática foram elaboradas quatro questões relacionadas às normas perceptuais e cognitivas: (1) “O resultado da tarefa feita foi facilmente percebido?”, (2) “Teve alguma dificuldade em perceber os efeitos dos conceitos de programação?”, (3) “Houve dificuldades na compreensão da tarefa?”, e (4) “Conseguiu compreender plenamente os efeitos dos conceitos no robô?”.

A Tabela 6 sintetiza as respostas dos alunos às questões apresentadas. Ao analisar as respostas é possível identificar que houve uma percepção consistente quando considerado a compreensão dos conceitos de programação e o efeito positivo na utilização do robô para isso.

Tabela 6. Exemplo de quadro de respostas

	O resultado da tarefa foi facilmente percebido?	Teve alguma dificuldade em perceber os efeitos dos conceitos de programação?	Houve dificuldades na compreensão da tarefa?	Conseguiu compreender plenamente os efeitos dos conceitos no robô?
Aluno 1	<i>Sim, foi fácil.</i>	<i>Não houve dificuldade em relação a usar os conceitos de programação.</i>	<i>Não, entender o que o robô deveria realizar durante sua execução não foi difícil.</i>	<i>Sim, ficaram bastante claros.</i>
Aluno 2	<i>Sim, o objetivo foi facilmente compreendido.</i>	<i>Era possível identificar quais ferramentas precisavam ser utilizadas.</i>	<i>Não.</i>	<i>Houve algumas variações na resposta do robô a certos comandos, mas basicamente ele correspondia ao que era esperado.</i>
Aluno 3	<i>Sim, foi facilmente percebido.</i>	<i>De certa forma sim, mas por causa do ambiente de programação que não trabalhava diretamente com o código.</i>	<i>Não.</i>	<i>Sim, embora devido a erros do robô algumas instruções eram executadas de forma que não correspondiam ao que era programado.</i>

Aluno 4	<i>Sim.</i>	<i>Após os primeiros testes foi mais fácil identificar.</i>	<i>Não.</i>	<i>Quase sempre. Só tive problemas quando o robô respondia de maneira estranha.</i>
Aluno 5	<i>Foi fácil perceber.</i>	<i>Os efeitos da programação no robô eram claros pra mim.</i>	<i>Não tive dificuldades.</i>	<i>Sim, consegui ver as ações dele de maneira correta.</i>
Aluno 6	<i>Sim.</i>	<i>Consegui perceber quais eram os conceitos necessários e usá-los.</i>	<i>Eu não tive dificuldades em compreender.</i>	<i>Sim, consegui ver as suas ações corretamente. Em alguns momentos ele não respondia corretamente, mas acredito que não era erro da programação.</i>

De acordo com as respostas da Tabela 6, o robô conseguiu proporcionar um ambiente concreto amenizando a barreira da abstração. O retorno positivo pelos alunos também demonstra indicativos da viabilidade da metodologia na análise tanto dos aspectos comportamentais, quanto dos cognitivos e perceptuais. Uma análise das respostas também demonstra indícios iniciais de melhoria na aprendizagem, apontando uma compreensão positiva dos efeitos dos conceitos de programação junto ao robô.

6. Considerações Finais

O PDAP-RP mostrou que o processo tem potencial para o desenvolvimento e o fornecimento de instrumentos de avaliação de práticas com RP. O diferencial deste processo é a utilização SO na formatação de práticas, assim como a avaliação de resultados através de uma perspectiva orientada a normas.

Vários aspectos que envolvem professores e alunos podem ser descritos e analisados por meio dessa perspectiva. Levando-se em conta que a utilização da RP para o ensino de programação de computadores não é algo trivial, é essencial ter meios para auxiliar nesse processo. O uso da metodologia proporciona uma visão na qual o ambiente de aprendizagem se apresenta como um sistema de signos, o qual pode ser analisado através de seus aspectos semânticos, pragmáticos e sociais descritos em *affordances* e normas. A compreensão desse sistema auxilia na identificação de quais são as etapas fundamentais para a construção de práticas que possam contemplar toda estrutura didática, da concepção à avaliação. Além disso, a metodologia contribui potencializando a uso da RP, indo além dos aspectos motivacionais, proporcionando a possibilidade de estabelecer atividades organizadas que podem ser revistas para uma melhoria contínua. A busca pela sistematização da criação de práticas possibilita identificar pontos específicos no qual houve erros durante sua elaboração.

É possível destacar algumas limitações da metodologia e processo, por exemplo, em situações (embora não frequentes) com respostas inesperadas ou inconsistentes em ações do robô, sem que a programação fosse responsável por tais erros. Outra limitação chave é que ainda não foi possível obter as respostas no nível de granularidade necessária a uma avaliação estatística quantitativa dos resultados da prática.

Entretanto, os resultados preliminares obtidos na prática mostram o potencial do PDAP-RP trazer benefícios no estudo de práticas de RP. As etapas possibilitam um encadeamento de trabalho que favorece a identificação dos objetivos pedagógicos no contexto de programação de computadores, responsabilidades dos alunos e professor envolvidos, definição de metas de desempenho e recursos de avaliação satisfatórios em relação às habilidades técnicas e cognitivas desenvolvidas pelo aluno. Como trabalho futuro, espera-se um refinamento maior da metodologia e processo, objetivando melhorar os indicativos de desempenho do aluno. Para tanto, já estão sendo realizados

experimentos com grupos distintos de alunos (iniciante e com experiência), bem como a aplicação com um número maior de participantes.

7. Referências

- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988.
- D'Abreu, J. V. (2002). Integração de Dispositivos Mecatrônicos para Ensino-Aprendizagem de Conceitos na Área de Automação. Tese de Doutorado em Engenharia Mecânica, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas-SP.
- Gibson, J. J. (1968). *The Ecological Approach to Visual Perception*. Houghton Mifflin Company, Boston, Massachusetts, 127-143.
- Gomes, A., Henriques, J., e Mendes, A. J. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias*.
- Gomes, A., Areias, C., Henriques, J., e Mendes, A. J. (2008). Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. *Revista Portuguesa de Pedagogia*, v. 42, n. 2.
- Liu, K. (2000). *Semiotics in Information System Engineering*. Cambridge: Cambridge University Press.
- Papert, S. (1994). *A máquina das crianças: repensando a escola na era da informática*. Porto Alegre: Artmed.
- Soares, R. F., e Borges, M. A. F. (2011). Robótica: aprendizado em informática de forma lúdica. In *WEI-XXXI Congresso da Sociedade Brasileira de Computação, Natal-RN: SBC*.
- Stamper, R. K. (1973). *Information in Business and Administrative Systems*, Hoboken: John Wiley & Sons.
- Stamper, R., Liu, K., Hafkamp, M., & Ades, Y. (2000). Understanding the roles of signs and norms in organizations-a semiotic approach to information systems design. *Behaviour & Information Technology*, 19(1), 15-27.
- Valente, J. A. (1993). Por quê o computador na educação. *Computadores e Conhecimento: repensando a educação*. Campinas: Gráfica da UNICAMP.
- Zanetti, H.A.P.; Bonacin, R.. Uso de semiótica e análise de normas em práticas de ensino de Programação de Computadores utilizando Robótica Pedagógica. *Revista Eletrônica de Tecnologia e Cultura*, v. 1, n. 1, 2014.
- Zilli, S. R. (2004). *A robótica educacional no ensino fundamental: Perspectivas e prática*. Dissertação de Mestrado em Engenharia, Universidade Federal de Santa Catarina, Centro Tecnológico, Florianópolis-SC.