



A Filosofia LOGO no Ensino Superior

Adriana da Silva Nogueira

(UNESA)

Stella Maria Peixoto de Azevedo Pedrosa

(UNESA)

Resumo

O presente artigo apresenta a utilização da Filosofia LOGO e da Linguagem LOGO como metodologia de ensino para a disciplina Algoritmos e Programação I ministrada no primeiro período dos cursos relacionados à computação. Foi realizada uma pesquisa-ação que teve como foco a observação do desenvolvimento das estruturas lógicas de pensamento dos alunos. Verificamos que a Filosofia LOGO pode propiciar uma aprendizagem agradável ao permitir que os alunos interajam com autonomia no gerenciamento de seu processo de aprendizado, contribuindo para a melhoria do processo. Ajustes na ementa da disciplina e a adoção da metodologia de ensino descrita colaborariam para o desenvolvimento das estruturas lógicas de pensamento e, conseqüentemente, para a formação acadêmica e profissional do aluno.

Palavras-chave: aprendizagem, Lógica de Programação, Filosofia LOGO

Abstract

This article presents the use of LOGO and LOGO language philosophy as a teaching methodology for Algorithms and Programming I subject taught in the first period of the courses related to computing. An action research which focused on the observation of the development of logical structures of thought of the students was held. We conclude that the LOGO philosophy can provide an enjoyable learning by allowing students to interact with autonomy in managing their learning process, contributing to the improvement of the process. Adjustments on the menu of discipline and the adoption of the described teaching methodology collaborate for the development of logical structures of thought and hence to the academic and professional training of the student.

Keywords: Learning, Programming Logic, LOGO Philosophy.



Introdução e Contextualização

Em cursos superiores de áreas relacionadas à computação, o domínio de conceitos referentes à lógica de programação é fundamental para o bom desempenho acadêmico do aluno em todo o curso. A introdução desses conceitos ocorre nos primeiros períodos, pois eles são a base para a compreensão das demais disciplinas, especialmente, as relacionadas à programação de computadores.

A construção de algoritmos e a assimilação desses conceitos depende das estruturas de raciocínio lógico desenvolvidas ao longo da educação básica e de toda a vida do indivíduo. Tais estruturas apresentam-se desenvolvidas em maior ou menor grau dependendo das características individuais e da estimulação dos seres humanos

Gomes (2008) aponta alguns fatores que podem interferir negativamente no processo de aprendizagem de programação. Restrições temporais e dificuldade de personalizar o ensino, levando em conta o ritmo de aprendizado de cada um, é um problema latente tendo em vista que, em geral, o docente acaba por ter uma maior preocupação em ensinar a linguagem e sua sintaxe ao invés de focar na resolução de problemas.

Existe, ainda, de acordo com o mesmo autor, um déficit de conhecimentos lógico-matemáticos e da própria estrutura da linguagem que acarreta dificuldades em detectar erros sintáticos. Para programar é necessário um alto nível de abstração e o programador precisa se preocupar tanto quanto a sintaxe, quanto com a construção do programa. Influenciam nesse aprendizado os aspectos psicológicos como desmotivação e a instabilidade inicial ao ingressar em um curso superior.

O papel do docente deve contemplar a percepção dos elementos críticos de sua turma e promover ferramentas que transformem esses elementos em oportunidades de aprendizado. Buscando tais ferramentas e verificando a necessidade de reduzir os problemas mencionados, baseamos nosso estudo na mudança da metodologia de ensino por meio da utilização da Filosofia LOGO (PAPERT, 1988), o que nos permitiu criar condições favoráveis à aprendizagem, melhorando todo o processo por meio da contextualização do conteúdo.



1. A Filosofia LOGO

Papert (1988) em seu trabalho com crianças caracteriza o computador como uma máquina capaz de transformar a aprendizagem pela forma como aprendemos a utilizá-la. O autor menciona que qualquer criança pode programar, considerando que "a programação nada mais é do que a comunicação com o computador por meio de uma linguagem que ele é capaz de compreender" (PAPERT, 1988, p.18).

Para Piaget (1972), é através da ação que o sujeito desenvolve continuada e progressivamente sua inteligência. O "[...] conhecimento não procede, em suas origens, nem de um sujeito consciente de si mesmo, nem de objetos já constituídos (do ponto de vista do sujeito) que a ele se impoariam. O conhecimento resultaria de interações que se produzem a meio caminho entre os dois[...]" (PIAGET, 1972, p.14).

Tais ações possibilitam a interiorização do conhecimento, pela assimilação e acomodação, definidos como pressupostos para a construção do conhecimento. A assimilação está relacionada à ação do sujeito sobre um objeto, que permite a incorporação de novas experiências ou informações às já existentes. A acomodação é um processo de modificação de suas estratégias de ação, ideias e conceitos, baseando-se nas novas informações/experiências adquiridas, gerando novas estruturas cognitivas. Entre esses dois processos ocorre a equilíbrio, caracterizada como adaptação. Cabe ressaltar que Papert desenvolveu suas teorias com base nos estudos que realizou com Piaget e aperfeiçoando-as por meio da utilização do computador.

Papert ao falar sobre a construção do conhecimento não prioriza o ensino, mas a aprendizagem e aborda a necessidade de, ao invés de disponibilizar informações ou conteúdos para o aluno, criar situações favoráveis à construção do conhecimento.

Desse modo, o autor (2008, p. 134) aborda o construcionismo como uma filosofia na qual deve-se obter "a maior aprendizagem a partir do mínimo de ensino", assim o docente deve apresentar as ferramentas necessárias para que o aluno construa o conhecimento e se sinta impulsionado a buscar as respostas que necessita. Com base nas perspectivas construcionistas, são oferecidas atividades agradáveis, que partam de experiências do mundo real, para que se aprenda através da resolução de problemas.



A filosofia LOGO, concebida por Papert (1988) a partir do construcionismo, apresenta o ambiente de programação como um “lugar” onde o aluno poderá controlar sua aprendizagem e “ensinará o computador a pensar”, embarcando assim, “numa exploração de como ele mesmo pensa”. Ele propõe que o aluno use a tecnologia para se motivar e criar condições de desenvolvimento autônomo e autoral, criando assim suas próprias atividades e consequentemente desenvolvendo suas estruturas lógicas de pensamento.

Sugere-se, então, a utilização dos computadores para romper as barreiras associadas ao fracasso escolar, pois o fato de crescermos em uma sociedade onde se acredita que existam pessoas mais capazes que as outras e onde “tudo é preparado para que as crianças atribuam suas primeiras experiências de aprendizagem desagradáveis ou mal sucedidas à sua própria inabilidade” (PAPERT, 1988, p. 63), nos leva, a cada experiência aparentemente negativa, consolidar estas ideias e fazer com que os alunos acreditem que são mais ou menos capazes de aprender determinado conteúdo.

Com base nas teorias construcionistas, foi desenvolvida a linguagem LOGO que representa uma família de linguagens que possuem diversos programas ou ambientes de programação que permitem sua utilização de diversas formas. Papert (1988, p. 21) afirma que “LOGO é uma filosofia de educação, que é possível graças a uma família sempre crescente de linguagens de computação que acompanham essa filosofia”.

Dentre outros aspectos importantes, destacamos o fato de que durante o período de operação desta linguagem o aluno “ensina” comandos ao computador, podendo, assim, escrever sobre a linguagem existente uma nova linguagem com base em sua melhor assimilação, o que permite ao aluno desenvolver novos comandos com uma sequência de instruções já existentes.

Ao dar uma instrução desconhecida pela linguagem, ele rapidamente obtém um feedback que atribui para a própria linguagem o erro. A resposta dada é: “Não aprendi...”, o que faz com que o aluno perceba que está no controle da situação, pois é ele quem ensina ao computador.

Sob a abordagem da Filosofia Logo, o docente não é um elemento central. Este e os alunos exercem constantes interações no processo de ensino e aprendizagem.



2. O Ensino de Lógica de Programação

O ensino de Lógica de Programação merece cuidadosa atenção, pois a aprendizagem dos conceitos básicos apresentados pode ser um problema especialmente para alunos que não possuem qualquer conhecimento anterior - por meio de uma formação técnica ou realização de cursos livres - sobre programação de computadores.

Buscando minimizar essa dificuldade, trabalhamos os conceitos de lógica de programação por meio da Filosofia LOGO. A seguir descreveremos as atividades realizadas com a turma em questão e os resultados obtidos.

2.1. Resumo das Atividades Realizadas

A presente pesquisa foi realizada em uma turma de primeiro período - com cerca de 60 alunos e com quatro tempos semanais com duração de 40 minutos cada - que abrangia os cursos de Bacharelado em Sistemas de Informação e Licenciatura em Computação de uma instituição particular da Zona Oeste da cidade do Rio de Janeiro.

No primeiro encontro foi apresentado aos alunos o que seria ministrado durante o semestre e como as aulas seriam conduzidas. Neste momento foi feito um diagnóstico sobre as características da turma, por meio do qual foi possível verificar se tinham conhecimento de programação, se conheciam alguma linguagem, se já haviam programado comercialmente e como aprimoram seu conhecimento acerca das linguagens conhecidas.

Na segunda aula os alunos realizaram um teste por nós considerado como um teste diagnóstico. Nele foram apresentadas questões relacionadas a problemas do cotidiano que nos permitiram verificar até que ponto os alunos conseguiam solucionar o que foi proposto, ainda que sem nenhuma linguagem específica. Nosso foco foi verificar se conseguiriam ou não estruturar as soluções.

Na terceira aula, um dos exercícios realizados e que trouxe um resultado bastante positivo foi o que chamamos de "Robô Pintor", onde supostamente deveriam programar um robô para pintar uma parede. Esta proposta inicialmente foi considerada simples, mas logo os alunos concluíram que haveria necessidade de estabelecer uma lista de ações para ensinar o



robô a pintar a parede. Desse modo, eles puderam reconhecer as várias possibilidades de se abordar um problema e, ainda, os vários pontos de vista que geraram novas formas de resolvê-lo. A turma se mostrou motivada e evidenciando o desejo de avançar tanto no conteúdo quanto no que se refere à programação.

As aulas seguiram com os conceitos básicos de programação e a introdução da Linguagem LOGO. Utilizamos os programas SuperLOGO e SCRATCH, que fazem parte da família de linguagens LOGO, além do VisuAlg. Nas aulas com LOGO a turma se deparou com novas possibilidades e concebeu novos problemas a partir dos inicialmente propostos.

A metodologia de ensino teve como suporte a proposta de problemas e a apresentação de ferramentas capazes de fornecer meios para a solução destes. A linguagem LOGO foi utilizada como base para aprendizado de novos conceitos como repetições, procedimentos, entre outros.

No final do período, os alunos fizeram um novo teste diagnóstico que permitiu constatar que todos eles avançaram de modo significativo, embora não se possa atribuir tal avanço somente ao LOGO e sim à mudança de metodologia e ao enfoque na resolução de problemas.

2.2. Resultados Obtidos

Após este estudo, concluímos que a Filosofia LOGO pode se constituir como aliada ao processo de aprendizagem da disciplina *Algoritmos e Programação I* ou de qualquer outra disciplina relacionada à programação de computadores. As atividades propostas em forma de desafios e problemas reais permitiram que os alunos se mantivessem motivados, buscando ampliar seus conhecimentos para solucionar estes desafios.

Os testes aplicados no início do período letivo verificaram o grau de conhecimento que os alunos possuíam sobre o que será estudado e o nível de desenvolvimento das estruturas lógicas de pensamento. Desse modo, foi possível verificar como desenvolviam inicialmente as soluções e, em comparação com um teste aplicado ao final do período letivo, como passaram a estruturá-las.



Independente de os alunos terem conhecimentos anteriores de técnicas de programação, alguns não foram capazes de estruturar soluções mais complexas, o que nos levou a crer que as estruturas lógicas de pensamento necessárias ainda precisam amadurecer. Alguns alunos, mesmo sem conhecimento de programação, eram capazes de estruturar soluções que atingiam os objetivos, já no primeiro teste e só evoluíram a partir de então.

Certamente apenas as atividades necessárias para a aprendizagem de programação não são suficientes para o desenvolvimento das estruturas lógicas, porém a realização de um trabalho, desde o ensino básico, com base nas ideias de Papert contribuiria significativamente tanto para os que desejam seguir carreiras na área de exatas, quanto para solucionar questões do dia a dia. Por mais que todos vivenciem as mesmas experiências de aprendizado não é possível afirmar que todos alcançarão o mesmo nível de desenvolvimento.

Um ponto importante a ser destacado é a dificuldade dos alunos em interpretar as questões propostas e os enunciados dos problemas, o que muitas vezes está relacionado à capacidade interpretativa referente linguagem escrita. Sendo assim, alguns alunos não podem solucionar o problema pois sequer conseguem compreendê-lo. Outros embora o compreendam, não sabem como solucioná-lo.

3. Uma proposta Metodológica para o Ensino de Lógica de Programação

Após este estudo verificamos a necessidade de reformular a metodologia de ensino utilizada no ensino de Lógica de Programação, considerando que esta proposta não deve se restringir à disciplina inicial.

Em primeiro lugar, a disciplina torna-se mais interessante, do ponto de vista do aluno, quando há uma ferramenta que possibilite o teste das soluções desenvolvidas. Assim, compreendemos que é necessário que os alunos estejam alocados em um laboratório com capacidade para acomodar, no máximo, dois alunos por computador. O uso de recursos multimídia, como o datashow, também auxilia positivamente favorecendo o processo.



À ementa devem ser acrescentados elementos que contemplem: introdução ao raciocínio lógico, resolução de problemas no contexto computacional e técnicas de interpretação de problemas algorítmicos.

Aulas conduzidas por meio do construcionismo e das propostas da Filosofia LOGO alcançam resultados mais satisfatórios, o que indica que o conteúdo apresentado deve permitir a elaboração de algoritmos com foco na resolução de problemas. Portanto, é importante a criação de um ambiente favorável à aprendizagem, no qual o aluno possa “falar a linguagem” do conteúdo que está aprendendo, desde o ensino básico para que assuma o controle da construção do seu conhecimento. Isso, além de torná-lo mais independente, favorece o processo de aprendizagem e não permite que ele perca o foco.

Devido ao fato de alguns alunos avançarem mais que outros, por já dominarem conceitos de programação ou por terem as estruturas lógicas de pensamento mais desenvolvidas bem como para evitar que se sintam desmotivados, seria interessante estabelecer 3 grupos de atividades, onde cada aluno poderia escolher o grupo daquelas que, por sentir-se mais seguro, desejasse realizar.

As atividades estariam divididas em fáceis (nível 1), medianas (nível 2) e avançadas (nível 3), ou seja, organizadas de modo independente de tal forma que não seja obrigatório passar por todos os níveis. Desse modo, alguns alunos poderiam optar por realizar as atividades mais avançadas sem a obrigatoriedade de passar pelos níveis mais simples.

Essa parte da proposta decorre do fato de ter sido verificada, nos testes iniciais, a existência de três grupos de alunos: os que não possuíam nenhum conhecimento, os que possuíam algum conhecimento e os que já conheciam o conteúdo previsto. A partir da comparação desses resultados iniciais com os obtidos ao final do curso, observaram-se os seguintes perfis:

- No teste inicial o aluno não demonstrou conhecimento e no teste final o aluno alcançou todos os objetivos propostos.
- No teste inicial o aluno não demonstrou conhecimento e no teste final o aluno alcançou parcialmente os objetivos propostos.



- No teste inicial o aluno não demonstrou conhecimento e no teste final o aluno não atingiu os objetivos, não atendendo aos requisitos mínimos para prosseguir.
- No teste inicial o aluno demonstrou algum conhecimento e no teste final o aluno demonstra que apenas manteve seus conhecimentos e não alcança os objetivos propostos.
- No teste inicial o aluno demonstrou algum conhecimento e no teste final o aluno demonstra que ampliou seus conhecimentos e alcança os objetivos propostos.
- No teste inicial o aluno demonstrou algum conhecimento e no teste final o aluno demonstra que ampliou seus conhecimentos acima das expectativas e alcança plenamente os objetivos propostos.
- No teste inicial o aluno já programa e/ou conhece o conteúdo previsto e no teste final o aluno demonstra que manteve seus conhecimentos e alcança os objetivos propostos.
- No teste inicial o aluno já programa e/ou conhece o conteúdo previsto e no teste final o aluno demonstra que ampliou seus conhecimentos acima das expectativas e, alcança plenamente os objetivos propostos.

O importante é alocar os alunos nas categorias corretas, com base em um teste inicial e um permanente acompanhamento, sugerindo-se que os resultados, com base no seu aproveitamento, sejam expressos de forma qualitativa de seguinte forma:

- EX - Excelente – aproveitamento entre 90% (noventa por cento) a 100% (cem por cento)
- MB - Muito Bom – aproveitamento entre 80% (oitenta por cento) a 89% (oitenta e nove por cento)
- B - Bom – aproveitamento entre 70% (setenta por cento) a 79% (setenta e nove por cento)
- S - Suficiente – aproveitamento entre 60% (sessenta por cento) a 69% (sessenta e nove por cento)
- I - Insuficiente – até 59% (cinquenta e nove por cento).

Outro ponto essencial a ser destacado é a questão da avaliação. O desenvolvimento de projetos, acompanhados pelo professor em todas as suas fases, permite uma avaliação



superior do que aquela obtida por uma simples conferência de uma prova registrada em papel em um determinado momento, dentro de um tempo estabelecido.

Tais projetos partiriam de um problema proposto pelo professor que permitisse ao aluno desenvolvê-lo até um nível que lhe fosse interessante. Também poderia ser dada ao aluno a liberdade para escolher um problema que, com base nas suas experiências, tivesse interesse em solucionar. Acreditamos que avaliar a evolução individual do aluno dessa forma contribui para o empenho do aluno e torna a avaliação mais transparente.

Considerações Finais

Um trabalho focado na aprendizagem do aluno realizado no primeiro período da faculdade pode minimizar a ideia de que os cursos da área de informática são cursos difíceis de serem concluídos, evitar a evasão e garantir que os alunos estejam melhor preparados para os demais períodos.

Considera-se que os resultados dessa pesquisa indicam a relevância de que, a partir das ideias propostas, a metodologia no ensino da programação seja modificada incorporando, gradativamente, os elementos apresentados e acompanhada uma avaliação permanente, de modo que seja possível verificar sua eficácia para o aprimoramento do processo.

Mesmo com turmas de ensino superior, ao se iniciar uma disciplina é necessário conhecer o perfil dos alunos e acompanhar seu desenvolvimento para que, se necessário, sejam realizadas alterações na metodologia aplicada.

Acreditamos que não podemos dar nosso estudo por completo e pretendemos dar continuidade ao mesmo, porém consideramos que a modificação e o aprimoramento da prática já realizados contribuíram significativamente para a turma em questão. Deve ser feita no cotidiano da sala de aula, testando hipóteses e melhorando o processo como um todo.



Referências Bibliográficas

GOMES, A. et al. **Aprendizagem de programação de computadores- dificuldades e ferramentas de suporte**. Revista Portuguesa de Pedagogia: Tecnologias Educacionais e da Comunicação. Ano 42-2, 2008. Disponível em: <http://hdl.handle.net/10316.2/4672>. Acesso em: 30/03/2014.

PAPERT, S. **Logo: Computadores e Educação**. 3ª ed. São Paulo: Editora Brasiliense, 1988.

_____. **A máquina das Crianças: Repensando a Escola na Era da Informática**. Ed. Rev. Porto Alegre: ArtMed, 2008.

PIAGET, J. **A Epistemologia Genética**. Ed. Vozes, Petrópolis, RJ, 1972.