

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/266492721>

Avaliação de uma metodologia alternativa para a aprendizagem de programação

Article

CITATIONS

52

READS

4,958

1 author:



[Marcos Augusto Francisco Borges](#)

University of Campinas

74 PUBLICATIONS 309 CITATIONS

SEE PROFILE

Avaliação de uma metodologia alternativa para a aprendizagem de programação

Marcos Augusto F. Borges¹ - marcosborges@ig.com.br
Faculdade Campo Limpo Paulista

Resumo

Este trabalho apresenta uma metodologia alternativa para a aprendizagem de conceitos de programação de computadores em cursos de nível superior.

A metodologia tradicional de ensino que contempla os conteúdos de programação estruturada possui uma série de inconvenientes em uma disciplina inicial de programação. Esta nova metodologia busca utilizar o paradigma de aprendizado construcionista como base para uma disciplina que trabalhará, prioritariamente, conteúdos de programação orientada a objetos.

O trabalho apresenta inicialmente um resumo desta nova metodologia. A seguir é relatado o resultado de uma pesquisa feita com alunos que cursaram a disciplina segundo a nova metodologia. Ao final, os resultados são analisados, discutindo a adequação em se usar esta nova estratégia no ensino de programação.

Abstract

This work presents a new methodology of the process of learning computer programming skills in high degree's courses.

The traditional learning methodology and the structured programming contents have a number of problems to fit in the first programming course. This new methodology try to use the constructivism learning as a basis to the learning process of object oriented programming skills.

This work presents a research with students attended the first programming course based on this methodology and analyze the results of this research, discussing the use of the methodology.

Palavras-Chave

Aprendizagem, Programação, Orientação a Objetos

¹ Rua Barão de Paranapanema, 400, ap. 112, Campinas-SP, 13026-010

1. Introdução

O ensino de programação no nível superior normalmente apresenta conceitos de programação estruturada. A postura didática do professor normalmente segue o que Borges (1998) definiu como “modo tradicional” (figura 1). É uma postura muito comum em todas as áreas de conhecimento, mas especialmente nos cursos da área de computação. Apesar de estar sendo usado há muito tempo, possui uma série de inconvenientes.

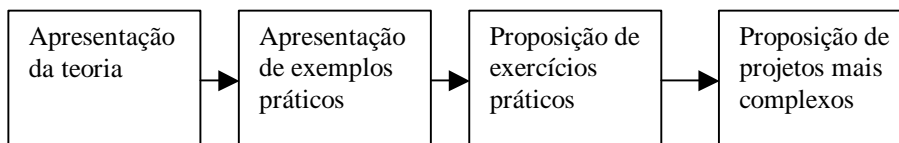


Figura 1: a sequência de passos típica na apresentação de uma disciplina

Alguns currículos apresentam os conteúdos baseando-se em uma linguagem (comumente, o Pascal). Outros apresentam os conceitos, ilustrando-os apenas com algoritmos ou português estruturado, sem se basear em uma linguagem.

O modo tradicional não consegue facilmente motivar os alunos a se interessar pela disciplina. Não é claro para alunos em seu primeiro ano de curso, especialmente àqueles que não têm experiência profissional em informática, a importância de certos conteúdos para sua formação. E essa falta de motivação torna-se especialmente crítica quando o conteúdo é apresentado sem uma linguagem para exemplificar: torna-se ainda mais difícil para os alunos perceber a utilidade dos conteúdos apresentados.

Quando é apresentada uma linguagem, é comum a existência de aulas em laboratório. Em muitos casos, apesar de, em princípio, o ensino em ambientes de laboratório possa significar um cunho diferenciado da concepção tradicional, as ações não diferem da lógica presente na maioria dos espaços do ensino superior. Os professores não exploram a diversidade dos equipamentos disponíveis, suas performances e autonomia, com reais e concretas práticas, mas agem do seguinte modo: "na frente dos alunos - postados tal qual uma orquestra de teclados - um professor, como um maestro de batuta na mão, está dizendo *Agora todos teclem...*" (Rodrigues, 1998, p.21). Baseia-se em uma concepção dos professores de ensino pronto e acabado, ou seja, um tipo de conhecimento baseado em uma única verdade, fazendo com que, em determinados momentos, ele entenda que somente ele sabe ou tenha alguma coisa nova a dizer.

Não é motivador para os alunos desenvolver programas sem uma interface gráfica mais elaborada, uma vez que muitos deles já trabalham com um ambiente gráfico (como o Windows). Como alguém pode motivar-se em fazer um programa, se ele em nada se assemelha com aqueles que se pretende construir depois de formado? Além disso, alguns típicos exemplos de programas sugeridos para disciplinas de programação baseiam-se em conceitos matemáticos tais como porcentagem e fatorial, que em nada motivam a maioria dos alunos (especialmente aqueles que possuem uma base matemática deficiente).

Quando um programador desenvolve a habilidade de programar segundo o paradigma estruturado, pode ser mais difícil para ele reaprender a programar no paradigma orientado a objetos (OO). Esta dificuldade pode gerar programadores que programam basicamente de forma estruturada, explorando muito pouco da potencialidade do paradigma OO. Porque, então, ensinar o paradigma estruturado e depois ensinar o paradigma orientado a objetos? Será

que não seria mais razoável os alunos aprenderem diretamente o paradigma mais moderno, que é também o mercado mais necessita?

Este trabalho apresenta uma proposta que tenta resolver alguns inconvenientes da abordagem tradicional, utilizando-se de uma abordagem de aprendizagem mais próxima do construtivismo *piagetiano*, mais de acordo com as necessidades de aprendizado da sociedade atual (Mazzone, 1993). Apresenta também o resultado de um levantamento feito entre os alunos a respeito desta nova prática.

2. A nova proposta

Segundo esta nova proposta, o curso é todo desenvolvido em um ambiente gráfico, desenvolvido especificamente para o uso em um curso de introdução a programação, e baseado no livro Karel ++ (Bergin et al., 1996). Os alunos, desde o início, trabalham com o paradigma OO.

Segundo esta nova proposta, a disciplina começa com algumas aulas (no exemplo em que foi baseado o texto, um mês de 4 aulas semanais) onde são trabalhados os conceitos básicos de orientação a objetos, tais como objetos, classes, hierarquias, atributos, métodos e mensagens. Os alunos aprendem também a trabalhar com a técnica de modelagem de objetos (*object modelling technich*, OMT) (Rumbaugh et al., 1991). Os primeiros problemas são relacionados com o mundo real e, pouco a pouco, os exemplos vão sendo adaptados ao problema de robôs, que serão usados durante toda a disciplinas.

Nas aulas posteriores, esta proposta reordena as fases do ensino tradicional (figura 1), sendo que o resultado desta reorganização de conteúdos é apresentado na figura 2.

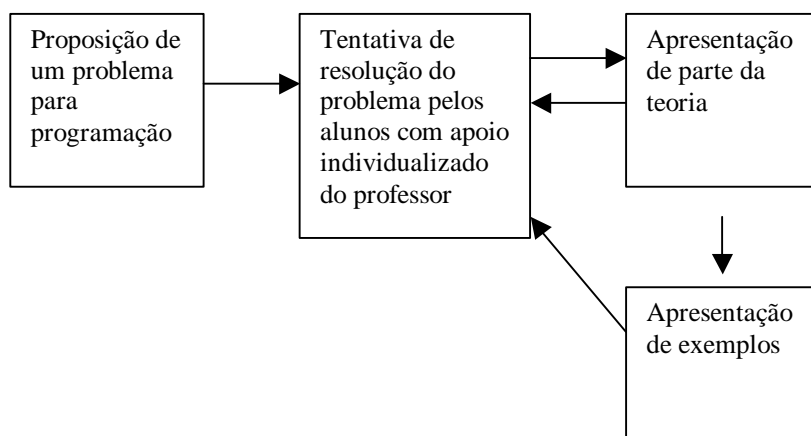


Figura 2: a nova proposta para a apresentação de uma disciplina

Nesta nova proposta, cada aula começa com a apresentação de um pequeno problema de programação. Os alunos tentam resolvê-lo com os conhecimentos que possuem, sendo estimulados pelo professor para trabalhar, preferencialmente, em pequenos grupos. Após um tempo suficiente para que estes alunos consigam a solução para o problema proposto ou detectem a necessidade de algo novo, o professor apresenta esse novo conteúdo. O ciclo retorna com mais uma oportunidade para que os alunos possam tentar aplicar o apresentado. Durante este período, o professor abandona o controle sobre o ambiente de aprendizado, buscando identificar pontos que possam ser esclarecidos para cada aluno, apoiando, principalmente, aqueles alunos que demonstrem maior dificuldade. Ao final, o professor apresenta uma solução para o problema usando os novos conceitos, preferencialmente usando

um exemplo de algum aluno que tenha chegado a solução. O curso segue em um ciclo, que pode ser visto como uma adaptação do ciclo de aprendizagem da “estética Construcionista” (Valente, 1993).

2.1. O ambiente "mundo do robô"

O mundo do robô é uma biblioteca² de classes de Object Pascal (ambiente Delphi) que implementa a base para que os alunos possam desenvolver tarefas semelhantes às tarefas propostas pelo livro Karel ++ (Bergin et al., 1996).

Esta biblioteca define classes para os robôs, o mundo onde estes robôs habitam (uma "sala" limitada em duas dimensões, quadriculada), paredes (que os robôs não conseguem transpor) e bolas (que os robôs podem pegar e colocar em uma bolsa ou tirar desta bolsa e deixar na posição). Os robôs, as paredes e as bolas podem ser dispostos livremente pelo mundo. A figura 3 apresenta um exemplo de mundo com um robô, bolas e paredes.

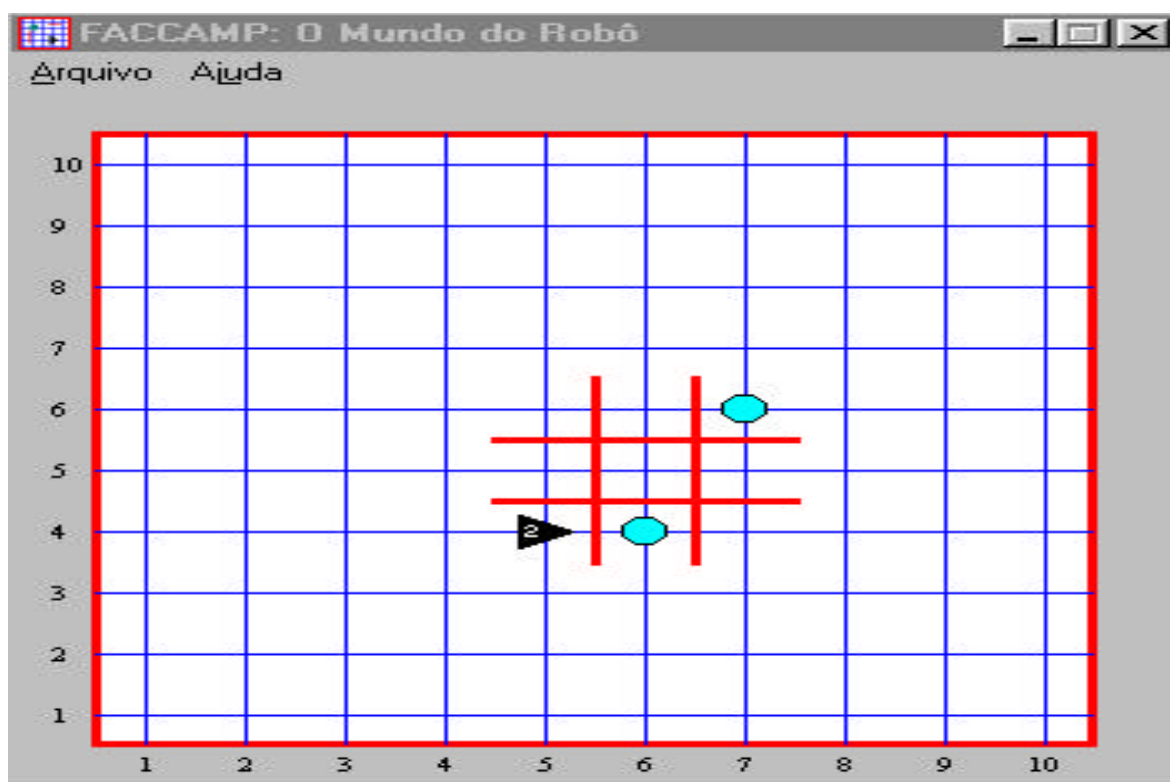


Figura 3. Um exemplo de "mundo" (com 4 paredes, duas bolas no chão e duas bolas na bolsa do robô)

São definidas duas classes de robôs: a "robô padrão", que tem métodos apenas para andar uma posição para frente, girar 90° a esquerda, pegar e soltar bolas; a "robô", derivada da robô padrão, que também sabe verificar se o robô está com a frente livre, se está sobre uma bola, se tem bolas em sua sacola, etc. Todos os movimentos dos robôs são animados de modo que os alunos possam acompanhar passo a passo a execução de seus programas.

² Desenvolvida por Osvaldo Luiz de Oliveira, em 1999.

Foi definido um programa básico onde é apresentada a estrutura de um programa e a partir do qual os alunos podem derivar os seus (anexo 2). O objetivo deste programa é facilitar a atividade inicial dos alunos, que tem um modelo de como os programas devem ser.

2.2. A prática das aulas

O programa da disciplina depois da introdução de OMT, não inicia com aulas de algoritmos ou sintaxe, como é comum. Como um dos objetivos é tornar as aulas mais motivadoras, desde o início os alunos trabalham com exercícios que podem ter seus resultados apresentados graficamente na tela.

Deste modo, os primeiros exercícios baseiam-se em fazer certas tarefas com robôs das classes previamente definidas, tais como: "faça o robô dar uma volta na sala", "faça o robô soltar bolas em uma certa disposição", etc. Segue-se a definição de classes de robôs mais elaboradas e seu uso. Um exemplo de exercício é definir um robô que saiba girar 90° a direita, ou andar para trás, baseando-se apenas nos métodos herdados, definidos nas classes robô e robô padrão. Assim, a princípio são trabalhados apenas os conceitos de orientação a objetos.

A sintaxe é trabalhada mais fortemente a partir deste ponto. Até esse momento, os alunos preocupam-se apenas com seguir o modelo apresentado e algumas informações passadas informalmente na sala. Mas a partir desse ponto, todas as estruturas são apresentadas segundo uma representação de sua definição sintática.

Só então são trabalhados os conceitos tradicionais desta disciplina, como variáveis, comandos condicionais e de repetição. Isto não implica, no entanto, em um atraso no programa, uma vez que quando são apresentados a estes conteúdos, os alunos estão mais preparados e motivados, o que faz com que as aulas sejam mais efetivas.

Os conceitos tradicionais são apresentados quando da proposição de problemas. Por exemplo: "faça um robô que pegue uma bola apenas se estiver sobre uma". Os alunos tentam resolver este problema mas percebem que não tem ferramentas para isso: neste momento são apresentados os comandos condicionais. Um outro exemplo, onde os conceitos de iteração são trabalhados: faça um robô andar por todo o mundo. Sem iteração, isto resultaria em um número grande de comandos normais. Parte dos próprios alunos a preocupação com a necessidade de se repetir os comandos. Apresenta-se então os comandos de repetição e os alunos precisam usar dois comandos de repetição aninhados (um para linhas, outro para colunas) para resolver este problema.

Todas as aulas são divididas entre sala e laboratório (duas aulas semanais para cada). Em sala, os alunos devem, trabalhando em pequenos grupos, buscar escrever os programas propostos, preferencialmente explorando os últimos conceitos trabalhados. No laboratório, o professor não apresenta uma aula de fato, mas os alunos (cada um em um micro) estão livres para tentarem implementar suas soluções feitas em classe e verificar a correção destas. Ao professor cabe identificar momentos onde um apoio possa ser efetivo na solução de alguma dificuldade.

Além das quatro aulas obrigatórias, existem outras duas aulas opcionais em laboratório, onde os alunos podem exercitar os conceitos trabalhados nas aulas ou mesmo ir além, contando com o apoio do professor.

3. Avaliação da metodologia pelos alunos

3.1. Materiais e métodos

Foi passado um questionário (apresentado no anexo 1) aos alunos presentes em uma aula da disciplina de algoritmos e programação de computadores da Faculdade Campo Limpo Paulista, no primeiro semestre do curso (ingressantes em 1999), onde esta proposta foi inicialmente implementada. Os alunos tiveram que responder o questionário em classe. Dos 22 alunos matriculados, 18 responderam o questionário, mas apenas 12 destes foram considerados: os outros não responderam questões consideradas relevantes pelo autor, como experiência prévia em informática, idade e sexo.

3.2. Resultados

Do universo de 12, 83% (10) são homens, o que é uma proporção válida com relação ao tamanho da classe (de 22 alunos, 86% (19) são homens no total). A idade dos alunos que preencheram corretamente o levantamento varia de 18 a 42 anos, tendo como média 23 e mediana 21. Os resultados não mostraram alguma tendência das mulheres claramente diferente dos homens. Isto pode ser justificado pelo pequeno número de mulheres analisadas (2) e por elas possuírem experiências prévias em informática diferenciadas (uma já programava, enquanto a outra já sabia instalar aplicativos, usar a Internet, mas não programar).

Apesar de ser uma classe relativamente jovem, 50% dos alunos que responderam corretamente (a partir desse momento, serão chamados apenas de alunos) iniciaram o curso com noções de programação e todos já haviam usado computador, sendo que entre os 50% que ainda não sabiam programar, 25% tinham alguma experiência em instalar aplicativos e/ou usar a Internet e 25% já havia, pelo menos, trabalhado com um editor de textos. Informalmente, o autor levantou que muitos já possuíam o título de técnicos em processamento de dados e outros já haviam iniciado cursos em outras faculdades da região. O fato de Campo Limpo Paulista ser uma cidade próxima da grande São Paulo, pequena, mas industrializada, onde não existia até então um curso superior na área de informática pode justificar, em parte, o alto índice de alunos que não só tem experiência com computador, mas também trabalham na área. Como o curso é novo, profissionais da área que moram nessa região foram os primeiros a se interessar por ele. Certamente, é uma realidade bem diferente de outras regiões.

Avaliando a metodologia de ensino utilizada na disciplina, 17% a consideraram tradicional, 50% parcialmente diferente da tradicional e 32% totalmente diferente do tradicional. Isto pode demonstrar que, entre os alunos, não há um consenso do que seja uma metodologia tradicional de ensino. 17% acreditam que a metodologia é diferente, porque a linguagem trabalhada no curso é diferente. Demonstaram, assim, acreditar que a metodologia de ensino é decorrente do conteúdo trabalhado, e não de uma estratégia didática adotada.

Entre os que acham que a metodologia de ensino não é tradicional, 20% a consideram menos eficaz e 40% mais eficaz. Os que acharam menos eficaz são alunos que já sabiam programar, ao ingressar. Um aluno indicou que a metodologia ajuda a programar segundo o paradigma de orientação a objetos. Outro que o método "força o raciocínio e estimula o aprendizado". Um aluno com experiência em programação, no entanto, acredita que os alunos novatos estão tendo dificuldades com conceitos básicos de programação, o que, curiosamente,

não foi indicado por estes "novatos" (mas note que pelo menos 50% dos alunos não são "novatos").

Entre os alunos que já haviam aprendido programação, 33% acreditavam que a metodologia é tradicional em comparação com as disciplinas por eles cursadas anteriormente, enquanto 17% acreditavam que é parcialmente diferente e 50% que é totalmente diferente.

Entre os alunos que já haviam trabalhado com programação e que acharam a metodologia utilizada diferente (pelo menos parcialmente), 25% indicaram que a metodologia utilizada é menos eficaz que a utilizada em outros cursos de programação e o mesmo número que é mais eficaz. Um aluno indicou que a metodologia usada em seu curso de programação anterior fazia com que o aprendizado só se adequasse a linguagem ensinada. 17% acharam interessante o ensino de OO, mas ressaltaram que "é questionável sua aplicação à alunos sem conhecimento de programação". Isto demonstra uma saudável preocupação dos alunos mais experientes com os outros, preocupação esta muito salutar e que pode ter sido apoiada pelo fato de ser incentivada a discussão em grupos da solução.

92% responderam uma questão sobre a dificuldade da disciplina. 17% que já sabiam programar consideraram a disciplina fácil, outros 17% alunos consideraram a disciplina nem fácil nem difícil e 58%, difícil, incluindo 17% que já sabiam programar, 17% com experiência razoável com computadores e os 25% com apenas conhecimentos de editores de textos.

92% consideraram a disciplina essencial para a formação e 8% (um aluno com menos experiência com informática) consideraram a disciplina importante, mas não essencial. Isto demonstra que um conhecimento prévio maior de informática deixa clara a importância de programação para um profissional da área.

50% dos alunos indicou querer mais aulas teóricas, especialmente 66% dos alunos com menor conhecimento prévio na área (33% não querem mudanças nesse sentido). No entanto, 64% gostariam de uma disciplina mais prática, incluindo todos os que possuem menor nível de conhecimento prévio. 90% quer mais aulas em laboratório com atividade planejada pelo professor (apenas 8%, com experiência anterior, querem menos) e 67% quer também mais aulas livres em laboratório (mas apenas 33% entre os com menor conhecimento anterior). Isto indica que os alunos acreditam que irão atingir um progresso maior com mais prática, mas, principalmente aqueles com menor experiência, acreditam precisar da presença do professor para que esta prática seja válida.

73% indicaram querer a disciplina mais aprofundada e nenhum menos. 42% queriam que a disciplina apresentasse mais conteúdo no mesmo tempo, incluindo um aluno com menor conhecimento prévio. Por outro lado, entre os 42% que indicaram preferir uma menor quantidade de conteúdo no período, surpreende a existência de alunos que já sabiam programar previamente.

Todos os alunos com menor conhecimento prévio (e apenas eles) indicaram acreditar que terão dificuldades superiores para acompanhar a disciplina em relação a turma. Entre os que acreditam que terão maior facilidade, todos já possuem experiência de programação. É importante ressaltar que pelo menos um aluno sem nenhuma experiência prévia em programação teve resultados superiores em nota a média da turma, mas, para isso, participou mais ativamente das atividades não obrigatórias em laboratório.

4. Discussão

Neste capítulo esta proposta é analisada, com relação a pesquisa realizada com os alunos e aos problemas indicados na introdução.

4.1. Motivação

O professor deve ter como preocupação constante a manutenção da motivação da classe. O uso de um ambiente gráfico, onde o aluno pode "ver seu programa rodando", foi planejado para manter este nível de motivação.

O fato de, nas avaliações, a maior parte dos alunos indicar querer ter mais aulas (tanto teóricas quanto práticas), de muitos desejarem conteúdos mais detalhados na disciplina e a ampla maioria definir a disciplina como essencial podem ser indícios de que os alunos estão motivados.

Por outro lado, como é exigida uma nova postura dos alunos, que devem ser mais ativos, buscando resolver problemas de um modo mais independente, o professor deve preocupar-se em trabalhar a auto-estima destes, para que eles sempre acreditem serem capazes de resolver os problemas propostos.

É muito interessante observar que, muitas vezes, os alunos identificam qual o tipo de informação que eles necessitam: é muito motivador também para o professor apresentar-se uma teoria solicitada pelos alunos.

4.2. Efetividade

A resolução de problemas é uma forma de aprendizagem interessante porque aprende-se mais profundamente técnicas para resolver problemas que são importantes, questionando, buscando respostas em diferentes fontes, considerando diversas perspectivas, trocando visões com outros e construindo seu próprio conhecimento (Kay, 1995). Os aprendizes desejam e precisam aplicar os novos conhecimentos em uma experiência autêntica e relevante para a vida (Nicol, 1990; Woolf, 1996). O fato de as aulas começarem com a tentativa de resolução de um problema por grupos de alunos faz com que essas qualidades sejam viabilizadas neste processo de aprendizado.

Fornecendo um apoio mais individualizado quando necessário, o professor consegue inserir os conteúdos em momentos mais próximos da zona de desenvolvimento proximal dos alunos (Oliveira, 1997). Conseguir a informação no momento em que ela se apresenta necessária é interessante, pois é neste momento que o cérebro indexa mais rapidamente estas informações (Ellis, 1994). Sabe-se que uma informação que nunca é utilizada durante o processo de aprendizado pode ser difícil de recuperar e usar quando se necessitar dela no mundo real (Schank e Kass, 1996).

No levantamento, a maioria dos alunos indica acreditar que esta metodologia é tão ou mais eficaz que a tradicional.

4.3. A cultura instrucionista dos alunos

Como o autor já discutiu em (Borges, 1998), romper com a cultura instrucionista das salas de aula é algo complexo.

Alguns alunos demonstraram desconforto por ter que tentar desenvolver raciocínios baseados em conceitos apresentados na mesma aula, sem um número grande de exercícios de fixação. Além disso, em alguns casos, por exemplo, o problema sugerido não era solúvel com o que havia sido apresentado. O objetivo disto era trazer os alunos para a zona de desenvolvimento proximal, antes de apresentar um novo conteúdo. Apesar de isso ter causado, algumas vezes, um mal-estar ("por que não disse isso antes?"), o autor acredita que isto tenha dado bons resultados e sempre que surgia um comentário como este, buscou-se explicar o porque da estratégia. Estes questionamentos devem justificar a preocupação de certos alunos mais experientes com o uso desta didática com alunos menos preparados.

Rodrigues (1998) propõe a inversão da lógica habitual da prática pedagógica dos cursos, que normalmente iniciam os conteúdos teóricos em aula para depois exercitá-los através de computadores no laboratório. Nesta proposta, a prática vem antes da teoria, mas não a prática em computador, e sim a prática de escrever e pensar em um programa no papel. E não é sem motivo: primeiramente, não se deseja incentivar que os alunos trabalhem no computador sem antes pensar no problema, algo comum entre programadores, mas com resultados bastante discutíveis. Mas, principalmente, porque é muito mais fácil promover-se a colaboração entre alunos em sala, onde as carteiras podem ser livremente movimentadas e não existem computadores que não podem ser movimentados.

O fato de existir um computador por aluno no laboratório apoia ainda mais um trabalho individualizado. Poderia ser mais didático o trabalho de duplas por computador, mas existe uma resistência muito grande dos alunos a esse respeito. Basta analisar propagandas de cursos livres de informática, que sempre exploram o fato de ser "um aluno por micro".

4.4. O conteúdo

A organização dos conteúdos deve ser mais flexível, adaptável ao dia-a-dia. Com esta nova metodologia, muitas vezes parte dos alunos uma necessidade de aprendizado, sendo muito mais efetivo apresentar-se um conteúdo quando ele é necessário para os aprendizes (uma vez que eles estão na zona de desenvolvimento proximal), mesmo que alguns pré-requisitos não tenham sido trabalhados ou que a ordem previamente estabelecida tenha que ser invertida. Assim, o professor deixa de lado uma "seqüência lógica" baseada exclusivamente na complexidade dos conteúdos e passa a priorizar a manutenção da motivação entre os alunos e a efetividade do processo de aprendizagem. Evita-se a apresentação de conteúdos muito teóricos ou densos no início do curso, trazendo os alunos primeiramente à zona de desenvolvimento proximal para depois apresentar estes conteúdos. Consegue-se assim, manter os alunos motivados.

O volume de conteúdos apresentados foi satisfatório, mesmo comparando-se com uma disciplina que só apresentasse programação estruturada. No entanto, como foram trabalhados e exercitados os conceitos de orientação a objetos, o conteúdo apresentado foi mais extenso que o previamente esperado. No entanto, os alunos indicaram que gostariam de ter mais aulas e uma disciplina com um conteúdo mais aprofundado, o que demonstra que eles acreditam que precisam trabalhar com mais conteúdos.

Como indicado por vários alunos, o fato de trabalhar com orientação a objetos desde o início é interessante, uma vez que o mercado de trabalho está exigindo profissionais com este tipo de habilidade. No dia-a-dia com os alunos não foi detectado nenhum problema decorrente do fato de iniciar-se com orientação a objetos. Pelo contrário: os alunos percebem o potencial deste paradigma desde o início e buscam usá-lo, preferindo explorar a orientação a objetos a resolver problemas apenas com soluções estruturadas, muitas vezes muito mais complexas e muito menos reutilizáveis.

4.5. A postura do professor

O professor precisa mudar sua postura em relação as aulas para que esta proposta seja efetiva. Ele deixa de ser, durante boa parte do tempo, o "centro da atenção" na sala. Passa a ser um elemento de apoio ao processo de aprendizado dos alunos nos momentos em que estes buscam resolver um problema proposto. Para ser mais efetivo neste apoio, precisa ser mais informal, estar mais próximo dos alunos, sentando com os grupos, lançando desafios que sejam motivadores mas que não induzam a competição, buscando conversar com os alunos e perceber pontos nos quais eles estão com problemas, incentivando a formação de grupos e um trabalho mais colaborativo, etc. Enquanto está apoiando os alunos em suas tarefas, o professor deve evitar a apresentação de conceitos pré-estabelecidos, explorando técnicas de aprendizado mais construtivistas e usando, quando possível, um método socrático de seqüenciar perguntas para que o próprio aluno descubra as suas respostas.

5. Conclusão

Este trabalho apresenta uma proposta alternativa de metodologia para o aprendizado em disciplinas de introdução a programação que busca adequar o conteúdo à orientação a objetos e aumentar a motivação dos alunos.

As análises feitas indicam que os primeiros resultados com o uso desta metodologia são positivos, não obstante eles exijam um acompanhamento por um período de tempo mais longo para que seus resultados possam ser efetivamente comprovados.

Como os alunos possuíam um bom conhecimento prévio em informática, seria um trabalho interessante a avaliação desta metodologia em outras realidades. Também seria interessante adaptar esta metodologia a outras disciplinas de programação, como estrutura de dados.

Embora esta proposta exija uma mudança significativa da parte do professor (tanto na preparação das aulas, quanto na postura didática), o autor acredita que as vantagens apresentadas justificam esta adaptação.

É importante ressaltar, no entanto, que a adoção efetiva desta metodologia só foi possível porque a direção da Faculdade Campo Limpo Paulista apoiou a divisão da turma (originalmente de 50 alunos) em duas salas para esta disciplina, viabilizando um atendimento mais individualizado. O fato de a direção não só apoiar a manutenção de dois professores para a mesma disciplina, como manter também o plantão do professor em horários extra-sala foi essencial para o sucesso desta proposta.

Agradecimentos

O autor agradece a Osvaldo Luiz de Oliveira, pelo desenvolvimento da biblioteca de classes que viabilizou a adoção desta proposta, assim como pela preparação das apostilas e listas de exercícios.

O autor agradece também aos alunos que responderam prontamente ao levantamento e, principalmente, ao professor Nelson Gentil, diretor da Faculdade Campo Limpo Paulista, por viabilizar a adoção desta metodologia e por preocupar-se sempre com a qualidade do curso.

Bibliografia

- **Bergin, J., Stehlik, M., Roberts, J., Pattis, R.** *Karel ++ A gentle introduction to the art of object-oriented programming*. New York: John Wiley and Sons, 1996.
- **Borges, M.A.F.** *Uma nova abordagem para o ensino de banco de dados*. Anais do Workshop de Ensino de Informática WEI98/CSBC98, Volume 1, pp.445-453, 1998.
- **Ellis, J.E.** *Roger Schank wants your child's mind*. Business week, p.36-37, july 18, 1994.
- **Kay, A.C.** *Computers, networks and education*. Scientific american, p.148-155, 1995.
- **Mazzone, J.** *O sistema "Enxuto" e a Educação no Brasi"*. Em *Computadores e Conhecimento - Repensando a Educação*, J.A. Valente (ed.), Gráfica Central da Unicamp, 1993.
- **Nicol, A.** *Interfaces for learning: what do good teachers know that we don't*. In: Laurel, B. (Ed.). *The art of human-computer interface design*. Massachusetts: Addison-Wesley, p.113-122, 1990.
- **Oliveira, M.K.** *Vygotsky - Aprendizado e desenvolvimento. Um processo sócio-histórico*. São Paulo-SP, Brasil: Scipione, 1997.
- **Rodrigues, F.P.M.** *A prática do professor no ensino de informática*. Pelotas-RS, Brasil: Educat, 1998.
- **Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.** *Object-oriented modeling and design*. Prentice-Hall International, 1991.
- **Schank, R.C., Kass, A.** *A goal-based scenario for high-school students*. Communications of the ACM, v.39, n.4, p.28-29, april 1996.
- **Valente, J.A.** *Diferentes usos do computador na educação*. In: Valente, J.A.(ed), *Computadores e conhecimento: repensando a educação*. Campinas,SP, Brazil, Gráfica Central da Unicamp,1-23, 1993.
- **Woolf, B.P.** *Intelligent multimedia tutoring systems*. Communications fo the ACM, v.39, n.4., p.30-31, april 1996.

Anexo 1: Questionário

Avaliação da Metodologia

(Aluno: esta avaliação faz parte de um projeto que visa analisar a efetividade da metodologia de ensino que está sendo adotada. Colabore, respondendo com cuidado).

1. Qual o seu conhecimento prévio (antes de ingressar na faculdade) na área de informática?
 - a. ☐ Nenhum, nunca usei um computador
 - b. ☐ Usuário básico, já sabendo usar editores de texto, por exemplo
 - c. ☐ Usuário *expert* em informática, já sabendo instalar aplicativos, usar internet, etc.
 - d. ☐ Profissional de informática, tendo, no mínimo, noções de programação
2. Você considera a metodologia de ensino utilizada nesta disciplina:
 - a. ☐ Tradicional, seguindo as mesmas estratégias de ensino utilizadas em outras disciplinas
 - b. ☐ Parcialmente diferente
 - c. ☐ Totalmente diferente
3. Responda esta apenas se você respondeu b ou c na segunda questão. Comparando com a metodologia tradicional de ensino, você considera a metodologia que está sendo utilizada:
 - a. ☐ Menos eficaz que a metodologia tradicional
 - b. ☐ Tão eficaz quanto a tradicional
 - c. ☐ Mais eficaz

Por que?

Responda as questões 4 e 5 apenas se você respondeu **d** na questão 1:

4. Comparando com a metodologia de ensino de programação com a qual você já teve contato, você considera a metodologia que está sendo utilizada:
 - a. ☐ Tradicional
 - b. ☐ Parcialmente diferente
 - c. ☐ Totalmente diferente

Por que?

5. Comparando com a metodologia de ensino de programação com a qual você já teve contato, você considera a metodologia que está sendo utilizada:
 - a. ☐ Menos eficaz que a metodologia tradicional
 - b. ☐ Tão eficaz quanto a tradicional
 - c. ☐ Mais eficaz

Por que? _____

6. Você considera esta disciplina:
 - a. ☐ Fácil
 - b. ☐ Normal
 - c. ☐ Difícil
 - d. ☐ Extremamente difícil

7. Você considera esta disciplina
- a. ☐ Essencial para sua formação
 - b. ☐ Importante para sua formação
 - c. ☐ Interessante para sua formação
 - d. ☐ Inútil para um profissional de ciência da computação
8. Você gostaria de ter nesta disciplina (coloque + ou - na frente da cada tópico):
- ☐ Aulas teóricas
 - ☐ Aulas em laboratório com proposta de atividade feita pelo professor
 - ☐ Aulas livres em laboratório
9. Você gostaria que esta disciplina fosse (coloque + ou - na frente da cada tópico):
- ☐ Prática (teórica)
 - ☐ Aprofundada
 - ☐ Rápida (conteúdo apresentado por tempo)
10. Como você está se sentindo em relação a turma:
- a. ☐ Com dificuldades superiores para entender o conteúdo
 - b. ☐ Equilibrado
 - c. ☐ Com maior facilidade de aprendizado
11. Qual a sua idade: _____
12. Qual o seu sexo: ☐ Masculino ☐ Feminino
13. Faça aqui qualquer outros comentários que você achar interessante sobre a metodologia de ensino.

Muito obrigado por responder este questionário!
Professor Marcos Borges

Anexo 2: Template (program Template, unit TemplateF e janela)

```
program Template;

uses
  Forms,
  TemplateF in 'TemplateF.pas' {Form1};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

unit TemplateF;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, RoboF;

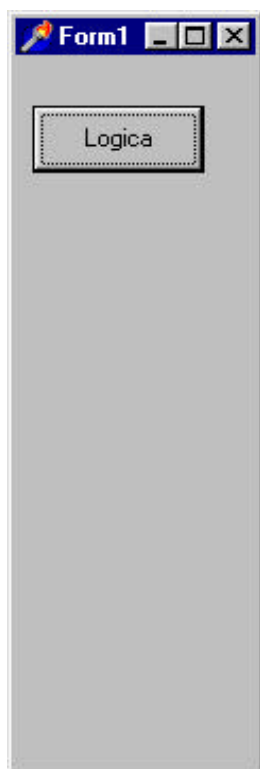
type
  TForm1 = class(TForm)      // classe janela derivada da classe TForm
    Logica: TButton;         // botão na janela
    procedure LogicaClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;             //declare aqui os objetos
  M: TMundo;                 //objeto janela
  Robo: TRoboPadrao;          //objeto mundo do robo
  Robo: TRoboPadrao;          //objeto robo

implementation

{$R *.DFM}
procedure TForm1.LogicaClick(Sender: TObject);
begin //O que acontece quando se aperta o botão
  // Ponha aqui a lógica do seu programa
  Robo.AndarFrente; //exemplos de comandos
  Robo.AndarFrente;

end;
procedure TForm1.FormCreate(Sender: TObject);
begin //o que acontece quando o programa é iniciado
  // O Mundo
  M := TMundo.Criar;
  {M.Beepers.InserirBeeper (1, 4); //exemplos de comandos
  M.Paredes.ParedeHorizontal (5, 5, 5);}
  // Cria robos
  Robo := TRoboPadrao.Criar (1, 2, Leste, clBlack, 1, M);
end;
end.
```



Janela do template: apenas um botão, que executará a lógica solicitada nos exercícios