

O Ensino do Polimorfismo nos Cursos de Licenciatura em Ciência da Computação: Proposta de um material instrucional

Adelito Borba Farias¹, Jefferson Aurélio Ferreira e Silva¹, Flávia V. Costa Souza¹

¹Departamento de Ciências Exatas – Universidade Federal da Paraíba (UFPB)
58297-000 - Rio Tinto - PB - Brasil

(adelito.farias, jefferson.aurelio, flavia)dce.ufpb.br

Abstract. *In front of several reports about the difficulty of understanding the content of programming - OOP in particular - the idea to build an instructional material to address one of the concepts related to the theme of this article. Through a survey of undergraduate students in computer science, the concept was chosen for the polymorphism. This paper presents a proposal to build an environment where students can learn in practice and theory, and share knowledge.*

Key words: *Learning Environment, OO Paradigm, Distance Education.*

Resumo. *Devido a diversos relatos sobre a dificuldade de compreensão do conteúdo de programação Orientada a Objeto, surgiu a ideia de construir um material instrucional para abordar um dos seus conceitos. Através de uma pesquisa com alunos do curso de graduação em Licenciatura em Ciência da Computação foi escolhido o conceito de polimorfismo. Este trabalho apresenta uma proposta para construção de um ambiente onde os alunos possam aprender, praticando e compartilhando conhecimento.*

Palavras Chaves: *Ambiente de Aprendizagem, Paradigma OO, Educação à distância.*

1. Introdução

Os cursos de Computação são constituídos de conhecimentos diversos interligados que buscam desenvolver diferentes competências e habilidades necessárias para exercício da profissão (Júnior et al, 2004). Dentre as várias áreas, os graduandos em Licenciatura em Ciência da Computação, convivem com dois eixos distintos durante o seu processo de formação profissional: ciência da computação e educação.

Dentro do fluxograma dos cursos de Computação, a Programação Orientada a Objeto (POO) é uma área que desenvolve nos estudantes habilidades específicas para resolver situações ou problemas de natureza complexa encontradas no mundo real. A abordagem orientada a objetos tem se mostrado mais adequada, comparativamente às demais, para ser empregada no desenvolvimento de sistemas de softwares complexos e de grande porte. Neste sentido, a POO possui papel fundamental na preparação dos estudantes da área de Computação (Filho, 2004).

Porém, a grande dificuldade de absorção dos conteúdos de POO, provoca desmotivação nos estudantes, fazendo-os abandonar e/ou migrar de curso (Junior,

2010). Esta dificuldade é comprovada pelo grande índice de reprovação e/ou evasão da disciplina de programação.

Visando mensurar as principais dificuldades dos alunos no aprendizado de POO dos cursos de Licenciatura em Ciência da Computação da UFPB – Campus IV, foi desenvolvida uma pesquisa com os estudantes do referido curso, apontando como complicadores, as técnicas algorítmicas que, exploram o raciocínio lógico na construção de sistemas computacionais. Analisando os dados obtidos da pesquisa, uma das questões que teve índice elevado na compreensão do conteúdo, foi o uso do conceito de Polimorfismo.

Segundo Ferreira (2006), na perspectiva OO, poliformismo é um conceito que trata da possibilidade de uma função ser habilitada para operar em uma variedade de tipos de dados, no qual, o efeito de polimorfismo está ligado aos conceitos de tipo abstrato de dados e de herança.

Cardelli e Wegner (1985) classificam o polimorfismo em linguagens orientadas a objetos de acordo com a Figura 1. Ela exhibe de forma esquemática esta classificação que está detalhada a seguir.

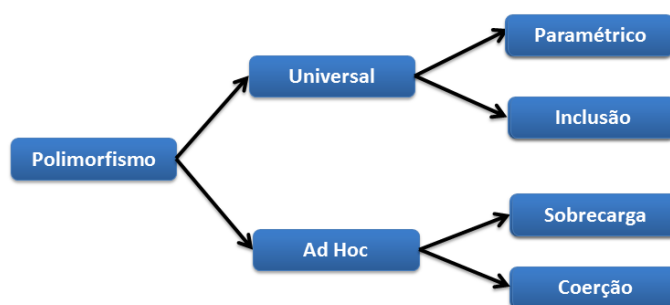


Figura 1. Classificação de Cardelli e Wegner (1985) para polimorfismo.

Para o polimorfismo universal a mesma função pode ser executada para um conjunto possivelmente ilimitado de tipos de dados diferentes. Já no polimorfismo "ad hoc" uma mesma função poderá apenas ser executada para um conjunto restrito de tipos de dados. O polimorfismo universal está subdividido em: Paramétrico - no qual, a partir de uma única definição de uma função ela pode trabalhar uniformemente. Também conhecido como genericity; e Inclusão - consiste em permitir que um objeto possa "ser visto" como um objeto de diversas classes, fornecendo comportamentos diferentes de acordo com o objeto real em tempo de execução. E o polimorfismo ad hoc é composto por: Coerção – é um cast, no qual a operação converte um argumento de um determinado tipo para outro que está sendo esperado; e Sobrecarga - consiste em prover diferentes implementações para funções com o mesmo nome, no qual os tipos dos argumentos na chamada são utilizados para escolha do método a ser executado.

Este artigo descreve a proposta de um ambiente para a aprendizagem e prática do Polimorfismo. A proposta foi desenvolvida buscando proporcionar ao usuário a aplicabilidade do conceito, através da resolução de exercícios. O material também buscará mostrar aos alunos o quão útil essa estrutura é utilizada para o desenvolvimento de códigos flexíveis. Para isso, pretendemos fazer uso de roteiros. Os roteiros irão conduzir os usuários em situações de busca de soluções. Isso ajudará os mesmos a perceberem a utilidade do conceito na criação dos algoritmos experimentais. O

ambiente disponibilizará de um fórum, para que seja compartilhada a construção do conhecimento entre os envolvidos. A fim de indicar o nível de assimilação do conteúdo, haverá exercícios informando a porcentagem de acerto do questionário.

O presente trabalho está estruturado em quatro seções: a primeira trata das iniciativas de melhoria do ensino de programação; a segunda seção apresenta os problemas do ensino de programação; a terceira mostra a metodologia usada para criação da proposta; e por último, considerações finais serão apresentadas.

2. O Ensino de Programação

Na conjuntura da pesquisa realizada, foram encontrados trabalhos que tratam de materiais instrucionais para auxiliar no aprendizado do paradigma OO. Porém, os trabalhos focam outros conceitos que não é o propósito deste artigo, o Polimorfismo. Dentre os variados tipos de ferramentas e ambientes propostos na literatura com a finalidade de facilitar o aprendizado de lógica e linguagens de programação, podemos destacar que os ambientes buscam minimizar a abstração do conteúdo. A seguir apresentamos alguns desses trabalhos.

O HabiPro (Habits of Programming) (Vizcaíno et al., 2000) é ambiente colaborativo que visa desenvolver nos estudantes “bons hábitos” em programação, ao estimular os estudantes a adquirirem habilidades como a observação e reflexão sobre a estrutura do algoritmos necessários para se tornarem bons programadores.

Um protótipo de sistema de ensino e aprendizagem de programação proposto por Kemp et al. (2003), apresenta programas de computadores previamente prontos e o estudante deve criar um diagrama correspondente ao programa já implementado. Neste sistema, os estudantes devem associar as estruturas de programação com a representação correspondente aos diagramas.

O projeto do ambiente ASTRAL (Rezende e Garcia 1995) visou atender as necessidades de preparação de diversos exercícios de implementação de estruturas de dados utilizando animações gráficas, de modo homogêneo sob uma interface consistente.

Outro trabalho relacionado foi proposto por Iepsen (2010), em que é apresentado um sistema que recomenda uma série de exercícios de acordo com o nível de conhecimento do estudante. O Persona-Algo, realiza uma personalização dos conteúdos para ficar proporcional ao estágio no qual está o estudante, desenvolvendo habilidades específicas para a elaboração de algoritmos.

A maioria destas aplicações estão baseadas na exposição e animação de algoritmos, podendo ser utilizadas para tarefas tão diversas, a exemplo: reter a atenção dos alunos durante as apresentações, explicação dos conceitos de forma visual e encorajar um processo de aprendizagem firmado na prática. Este último aspecto parece extremamente relevante, uma vez que os ambientes de programação (IDEs como Eclipse¹, Netbeans²) são direcionados aos profissionais, e a informação que fornecem é normalmente bastante complexa para ser compreendida por estudantes recém introduzidos na esfera do mundo computacional.

¹<http://www.eclipse.org/downloads/>

²<http://netbeans.org/downloads/>

O trabalho de Vahldick (2007) exhibe que diante da necessidade de um ambiente mais próximo da realidade dos estudantes, a utilização de ambientes mais simples, com interface acessível e interação pode contribuir para estimular os alunos a experimentar, avaliar e corrigir os seus próprios algoritmos, o que os auxilia no processo de aprendizagem de programação.

Uma das colunadas deste artigo, é a pesquisa realizada por um grupo de estudantes do curso de Licenciatura em Ciência da Computação da UFPB – Campus IV que, ao perceberem quão grande era o número de estudantes reprovados e/ou repetentes das disciplinas de POO, decidiram por fazer tal pesquisa. Os alunos que participaram da entrevista, pertenciam aos quarto e quinto período do mesmo curso. A pesquisa aponta que um dos conteúdos considerados como grande obstáculo de aprendizagem e prática foi o Polimorfismo.

Amparados por essa pesquisa e pelo fato dos materiais encontrados não tratam diretamente do polimorfismo, decidimos trabalhar na construção de um material instrucional focado no referido conceito, buscando expor quão valioso é para compreensão de reuso de código, também como a sua aplicação em sistemas computacionais.

2.1 Problemas no Ensino de Programação

Para a construção deste projeto, buscamos compreender como ocorre a aprendizagem de algoritmos, e elaboração de material instrucional voltados para este tema. É pertinente para uma nação que pleiteia acender do posto de consumidor tecnológico, para ocupar o cargo de produtor de tecnologia que, seja melhorada a qualidade do processo de ensino/aprendizagem.

Para os cursos da área de Computação, também é válida esta observação, considerando que um dos maiores entraves é o elevado número de reprovação e/ou evasão das disciplinas de algoritmo e programação. Este é um dentre as várias causas que fazem com que a quantidade de alunos iniciantes seja inversamente proporcional à quantidade de estudantes concluintes (Júnior, 2005).

A linguagem de programação é o conteúdo predominante nos cursos das áreas correlatas de Computação. O aprendizado de programação é um processo difícil e exigente, e vai além da aquisição de conhecer a sintaxe e a semântica de uma linguagem de programação (França et al, 2010).

Linguagens de programação possuem níveis de complexidade elevado. A abstração também é uma das principais dificuldades dos alunos. Segundo Vahldick (2003, apud Bruce e Michiels, p. 84), a Programação Orientada a Objetos é um tópico essencial na Ciência da Computação, pois é o paradigma mais presente nas linguagens e ambientes de programação na atualidade (C++, Java, C#, Python, Ruby, PHP, Perl).

Esses fatos nos levaram a propor um material instrucional que viesse a auxiliar no ensino de POO. O fato de existirem poucos materiais voltados para o ensino do Polimorfismo e por esse conceito ser importante, propomos um ambiente para aprendizagem e prática do mesmo.

3. Metodologia Proposta

Esse trabalho fez uso de técnicas qualitativas de pesquisa e adotou um processo de design instrucional para elaboração da proposta do material aqui tratado.

No processo de design instrucional seguimos modelo proposto por Filatro (2008). Nosso procedimento foi composto das seguintes etapas: levantamento de requisitos, elaboração do planejamento do material e elaboração de StoryBoard.

Iniciamos com etapa de levantamento de requisitos. O produto final dessa etapa foi a elaboração de um relatório de análise de necessidades que apresentava a demanda e a caracterização do público-alvo. Baseados nesse relatório iniciamos o planejamento do material didático. Nessa etapa elaboramos uma matriz instrucional com os seguintes elementos: objetivos de aprendizagem de cada unidade, conteúdo a ser abordado, descrição das atividades, ferramentas utilizadas, duração de cada unidade e forma de avaliação. Por fim, elaboramos um Storyboard com objetivo de obter uma sequência completa de ação, a fim de termos a oportunidade de compreender e testar o material proposto antes da etapa de implementação final.

Após a análise dos dados coletados e da investigação realizada com estudantes que já cursaram a disciplina de POO foi possível compreender o nível de dificuldade da aprendizagem. Aproximadamente 67% dos entrevistados possuem dificuldade sobre o conceito foco deste trabalho, Polimorfismo. Também buscamos identificar os estilos de aprendizagem dos alunos através da aplicação de um questionário. Cerca de 48% dos alunos que participaram preferem estudar por meio de software que auxilie no processo de aprendizagem. Estes dados serviram de estímulo para elaboração deste material instrucional.

4. Análise dos dados e Resultados

Os dados obtidos da pesquisa possibilitaram conhecer melhor os estudantes, bem como ajudaram a encontramos estratégias para desenvolver atividades que viessem a facilitar a compreensão da abstração algorítmica. Desta forma, foi realizado um design instrucional, que culminou nas fases a seguir.

4.1 Relatório de Análise de Necessidades

Durante essa etapa buscamos identificar os requisitos necessários para a criação e desenvolvimento do material instrucional. Realizamos as seguintes atividades: revisão bibliográfica sobre dificuldades de aprendizagem de POO e seus conceitos; aplicação de questionários com alunos que já cursaram a disciplina e alunos que ainda estão cursando; aplicação de questionário de estilos de aprendizagem; análise de competidores dos softwares e ambientes que buscam ensinar POO.

Desta forma, após a constatação da dificuldade dos discentes em suas atividades, referentes à aplicação do conceito de polimorfismo em suas soluções algorítmicas. Os dados obtidos também ajudaram na proposta do tipo de material. Identificamos que os alunos preferem estudar praticando, usando softwares e que também gostam dos tutoriais.

4.2 Matriz instrucional

Elaborada para ter uma visão ampla de cada tópico de aprendizagem. Durante essa etapa definimos as atividades essenciais para atingir as metas. Nossa matriz apresentou 4 unidades, que estão apresentadas a seguir:

Unidade 1: Revisão dos conceitos básicos de herança e interface – Tem como objetivo gerar compreensão dos conceitos básicos sobre herança e interface e recordar o assunto que serve como pré-requisito para as unidades posteriores nos usuários.

Unidade 2: O uso do polimorfismo na Programação Orientada a Objetos – O objetivo é exemplificar o uso do polimorfismo na POO, usando simulações ou analogias para a compreensão do conteúdo, fazendo com que o estudante compreenda a utilidade deste conceito, bem como as aplicações no mundo OO, a promover motivação para o uso deste conceito.

Unidade 3: Aplicação do polimorfismo – A identificação dos casos onde se aplica o conceito é o principal objetivo desta unidade, bem como, resolução de roteiros que mostrem a utilidade no desenvolvimento de softwares.

Unidade 4: Roteiros – Aplicar os conhecimentos para a implementação de códigos usando o conceito de polimorfismo; Construir novos algoritmos com uso do polimorfismo; Implementação de códigos usando o conceito de polimorfismo

4.3 Storyboard

A construção do Storyboard serviu para a compreensão de como seria realizada a inserção dos conteúdos montados na matriz instrucional, descrevendo os por menores da estrutura e do fluxo da informação, dos conteúdos e da interface (Filatro, 2008). Assim, neste intuito, é possível identificar possíveis erros de interação e usabilidade, antes da implementação do software.

Abaixo é exibida duas telas, que, respectivamente, são a Figura 2: tela inicial, onde o usuário escolherá o tópico para estudo; e a Figura 3: tela de um tópico escolhido, no qual o usuário terá acesso aos materiais do tópico. Nesta tela, o usuário poderá optar por acessar o conteúdo de forma textual, e posteriormente poderá resolver exercícios para fixar o conteúdo. Em outra, ele terá a opção de acessar o conteúdo no formato de áudio.

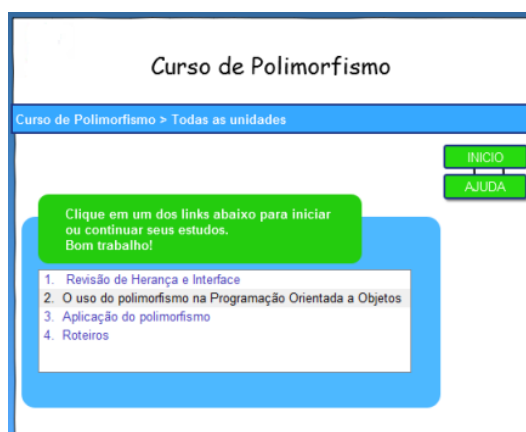


Figura 2

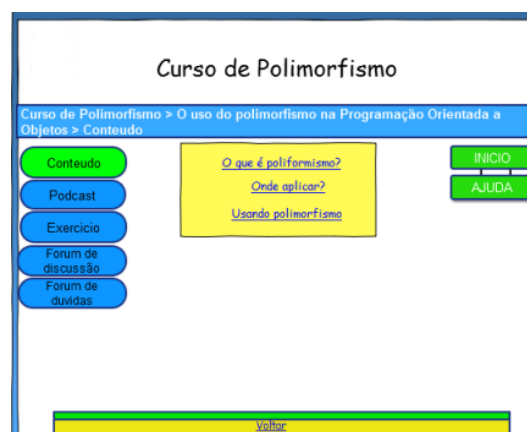


Figura 3

4.4 A Proposta

O material proposto é um tutorial, que aborda o conteúdo de Polimorfismo. Esse material apresenta os conceitos principais de polimorfismo. Também apresenta exercícios com aplicações reais do conceito, e alguns textos para autoavaliação dos estudantes, no qual, eles poderam verificar o andamento do seu aprendizado.

Ao término, o indivíduo tem a possibilidade de criar e aplicar estratégias de utilização do conteúdo em atividades (roteiros). Dessa forma, o aluno poderá desenvolver o senso crítico para poder identificar a melhor forma de solucionar os problemas com o uso do Polimorfismo.

5.Considerações Finais

Portanto, nosso trabalho teve por objetivo aprimorar o aprendizado do conceito de Polimorfismo, dando suporte ao usuário na compreensão do conteúdo por meio de uma metodologia que envolva a resolução e o compartilhamento de exercícios, focando na aplicabilidade do conceito. Com isso, o estudante é estimulado a formular resoluções de problemas que farão uso dessa estrutura.

Como trabalho futuro, propomos o uso de técnicas de prototipação a fim de realizar testes de usabilidade e aceitação do material, para após essa etapa, implementarmos o material.

Referencias

- Arias, J.O.C.; Yera, A.P. O que é a Pedagogia Construtivista? Rev. Educ. Pública., Cuiabá, v. 5, n. 8, jul./dez. 1996.
- Cardelli, L.; Wegner, P. On understanding types, data abstraction, and polymorphism. 1985. ACM Comput. Surv. (ACM) 28: 150. doi:10.1145/242224.242415. ISSN 0360-0300. Disponível em: <http://lucacardelli.name/Papers/OnUnderstanding.A4.pdf>. Acesso em 06 out 2011.
- De Souza, C. R. B. Conceitos de Orientação a Objetos. Disponível em: <http://www.ufpa.br/cdesouza/teaching/es/3-OO-concepts.pdf>. Acesso em: 06 out. 2011.
- Farinelli, F. Conceitos básicos de Programação Orientada a Objetos. 2007. Disponível em: <http://www.riopomba.ifsudestemg.edu.br/dcc/dcc/materiais/1662272077_POO.pdf>. Acesso em: 14 jul. 2011.
- Ferreira, A.P.L.; Ribeir,L. Programação Orientada a Objeto com Grafos. 2006. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/jai/2006/009.pdf>. Acesso em: 06 out. 2011
- Filatro, Andrea (2008). Design Instrucional na prática. Pearson: São Paulo
- Filho, A.M.S. Introdução à Programação Orientada a Objetos. Revista Espaço Acadêmico, Nº 35, abril/2004. Mensal. ISSN 1519.6186.

- França, E.L.; Felix, Z.C.; Souza, M.S.; Carneiro, T.B.; Sousa, P.R.C; Filho, C.A.P.D. Utilização de Objetos de Aprendizagem em Sistemas Tutores Inteligentes para o ensino da Programação. VII SEGeT - Simpósio de Excelência em Gestão e Tecnologia. 2010.
- Junior, D.P., Freitas, R.L. Estratégias para melhorar os processos de abstração na disciplina de Algoritmos. XXI Simpósio Brasileiro de Informática na Educação. João Pessoa, PB, Brasil, 2010.
- Júnior, J.C.R.P.; Rapkiewicz, C.E.; Delgado, C.; Xexeo, J.A.M. Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio. XXV Congresso da Sociedade Brasileira da Computação, 2005. Disponível em: <<http://200.17.137.110:8080/licomp/Members/jeanemelo/plonelocalfolderng.2006-04-10.7475913377/PEP/Aula5/arq0033-2005-medio.pdf>>. Acesso em: 08 jul 2011.
- Kemp, R., Todd, E. & Lu, J.Y. A novel approach to teaching an understanding of programming. In U. Hoppe, F. Verdejo & J. Kay (eds.), Proc. 11th Int. Conf. on AIED, pp. 449-451, Amsterdam, IOS Press, 2003.
- Lemos, D. G. Material Instrucional em Educação a Distância (EAD) para Professores-formadores. Revista da Alfabetização Solidária, Nº 5. 2005. Disponível em: <http://www.cereja.org.br/pdf/revista_v/Revista_DelbaGuariniLemos.pdf>. Acesso em: 14 jul. 2011.
- Mendes, A. J. N. Software educativo para apoio à aprendizagem de programação. Disponível em: <http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla_mendes.htm>. Acesso em: 27 mai. 2011.
- Neto, R. M .F. Polimorfismo e Acoplamento Dinâmico. [data desconhecida]. Disponível em: <http://www.catalao.ufg.br/cc/roberto/apostilas/polimorfismo.pdf>. Acesso em: 06 out. 2011.
- Rezende, P.J. ;Garcia, I.C. Astral: Animação Gráfica de Algoritmos e Estruturas de Dados - Uma Abordagem Construtiva, VIII Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, 317-318. 1995.
- Vahldick,A. Uma Experiência Lúdica no Ensino de Programação Orientada a Objetos. XVIII Simpósio Brasileiro de Informática na Educação. São Paulo, SP, Brasil, 2007.
- Valadares, C. Ambiente de Educação a Distância Sobre a Linguagem de Programação Java. Disponível em: <http://www.tise.cl/archivos/tise2006/20.pdf>. Acesso em: 27 mai. 2011.