

Simulação para Ensino de Conceitos da Orientação a Objetos

Mariane Fogaça Galhardo (FACENS)
marianefg@yahoo.com.br

Luciana Aparecida Martinez Zaina (FACENS)
luciana@facens.br

Resumo. Existem conteúdos que apresentam graus de dificuldades de assimilação diferentes. Na área de programação de computadores, como por exemplo, no paradigma orientado a objetos, a falta de prática pode trazer dificuldades durante o processo de aprendizagem da mesma. Utilizar uma ferramenta como auxílio ao ensino nesta área pode trazer grandes contribuições.

Este artigo tem como objetivo apresentar uma ferramenta que simula os conceitos da Orientação a Objetos permitindo a interação do aluno durante o processo de aprendizagem.

Palavras-chave: Simulação, Tecnologia para Ensino, Programação Orientada a Objetos.

1. Introdução

A velocidade com que as mudanças tecnológicas vêm acontecendo atualmente e a necessidade de se estar cada vez mais especializado no mercado de trabalho, vêm mudando e aumentando a importância do uso da tecnologia na educação já há algum tempo, principalmente no ensino de linguagem de programação. A tecnologia auxilia na capacidade de transpor os conceitos estudados para uma linguagem de programação de forma prática.

Além disso, a necessidade de receber estímulos durante o processo de aprendizagem é evidente nos alunos de um modo geral. Estas necessidades podem surgir por diversos motivos como a falta de tempo, dificuldades na compreensão de assuntos complexos, entre outros. Por isso, muitos docentes têm utilizado recursos tecnológicos para tornar suas aulas mais motivadoras e interativas, proporcionando maiores vantagens educacionais para o professor, o aluno e também para a instituição de ensino de um modo geral.

Entretanto, não basta só a existência do recurso tecnológico para que o processo de ensino-aprendizagem ocorra de maneira satisfatória. É necessária a participação do aluno e do docente durante este processo para que o mesmo apresente resultados gratificantes.

Dentre os vários recursos tecnológicos utilizados, o computador tem o seu lugar reservado, pois possibilita, por exemplo, simulações de situações próximas da realidade.

A simulação é muito utilizada em várias áreas de ensino, não só na informática. Durante este trabalho a simulação será apresentada como ferramenta de apoio para alunos iniciantes em linguagem de programação, mais especificamente, alunos iniciantes na área de Programação Orientada a Objetos.

A disponibilização da ferramenta será em uma plataforma que permitirá sua utilização através da Web, possibilitará ao aluno visualizar como se constrói uma classe usando técnicas da orientação a objetos, sem a necessidade de um conhecimento prévio de nenhuma linguagem de programação deste paradigma.

Através de algumas interações, o aluno informará os dados solicitados pelo simulador, o sistema transportará estes dados traduzindo-os para um formato de uma linguagem orientada a objetos: a linguagem Java. Assim, o aluno poderá visualizar como o código é gerado no formato

de uma linguagem específica através da tradução das informações que ele determinou de maneira abstrata.

2. A utilização da Tecnologia no Ensino

Apenas a existência do recurso tecnológico para o que processo de ensino-aprendizagem ocorra de maneira satisfatória não é o suficiente, pois também é necessária a participação do aluno e do docente durante este processo para que o mesmo apresente resultados gratificantes, segundo Guimarães (2003).

Quando alunos testam dados criados por eles mesmos, recebem resultados diferentes e podem com isso descobrir novos comportamentos, gerar erros ou apenas confirmar o que foi exposto durante aulas teóricas.

Os alunos desenvolvem habilidades cognitivas. Mas o mais interessante de tudo isso é que através desses dados aleatórios de cada aluno, todos aprendem mais, inclusive o professor; e a interação da classe como um todo aumenta, segundo Bridgeman (2000).

Assim, o importante não é apenas o uso da tecnologia, mas sim a aplicação sistemática e ordenada da mesma de forma a obter um resultado bem sucedido.

3. Ensino da Linguagem de Programação

O ensino de linguagens de programação às vezes apresenta conceitos difíceis de se entender, quando vistos em sala de aula, por serem muito abstratos.

Alunos iniciantes nesta área apresentam dificuldades durante o aprendizado de uma linguagem de programação. O domínio da lógica de programação se torna necessário para facilitar a assimilação do abstrato.

O fluxograma é uma ferramenta muito utilizada para auxiliar na compreensão da lógica de programação, pois utiliza-se de objetos gráficos para representar comandos.

No entanto, os fluxogramas não possibilitam que a lógica seja executada em um computador, necessitando com isso que o aluno imagine a execução do programa, segundo Silveira-Esmin (2003). Além disto, um fluxograma fica distante da real construção de um código.

Uma ferramenta de apoio possibilita ao aluno visualizar conceitos abstratos de forma prática e transparente e, conseqüentemente, torna o ensino mais proveitoso.

4. Simulação: Um apoio importante para o ensino

Silveira-Esmin (2003) mostra que determinadas disciplinas apresentam conteúdo de difícil assimilação, por serem constituídas de assuntos abstratos, complexos ou até mesmo cansativos.

Utilizar a simulação como apoio para tornar a aprendizagem mais dinâmica e atrativa pode facilitar a visualização de conceitos complexos além de permitir o entendimento de conceitos abstratos através de exemplos concretos, segundo Schimiguel-Araújo-Amaral (2003).

Schimiguel-Araújo-Amaral mostra que simulações podem ser feitas utilizando-se vídeos, animações e imagens em movimento. Esse recurso tecnológico pode ser utilizado tanto em aulas presenciais como a distância, ou em sistemas híbridos (mesclando aulas presenciais e a distância).

5. Ferramenta de Apoio

A partir das dificuldades encontradas na transição do ensino da programação imperativa para o ensino orientado a objetos decidiu-se desenvolver uma ferramenta que apoiasse os alunos

iniciantes neste novo paradigma. A ferramenta tem como foco principal o apoio aos conceitos estudados durante as aulas presenciais de um curso de Engenharia da Computação.

A ferramenta é constituída por páginas HTML que explicam e exemplificam os conceitos fundamentais da orientação a objetos e um simulador para transformação dos conceitos abstratos em concreto.

5.1. Definição dos Conceitos Fundamentais

A ferramenta tem como função explicar teoricamente os conceitos da Programação Orientada a Objetos, exemplificando de maneira simples e de fácil entendimento estes mesmos conceitos ao longo das páginas HTML, conforme pode ser observado na Figura 1. Estes conceitos dão suporte para o aluno interagir com a ferramenta, resolvendo os exercícios propostos e construindo seus próprios códigos.

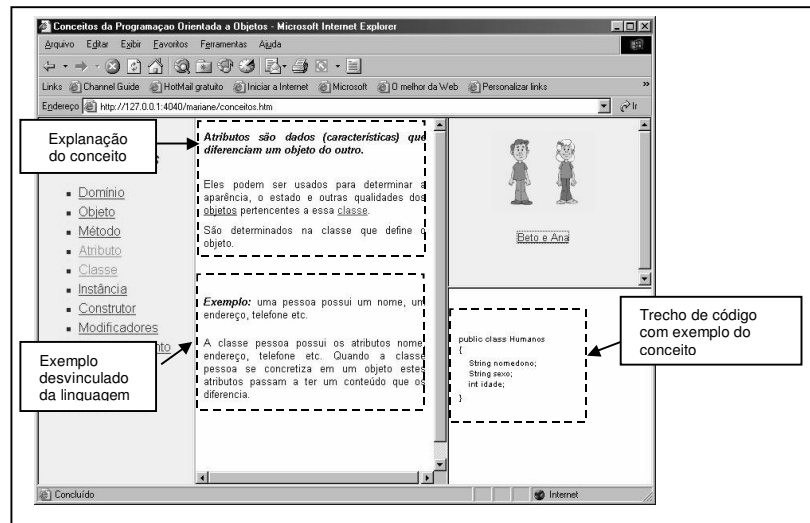


Figura 1: Apresenta o conceito de atributo e seus respectivos exemplos

O protótipo apresenta uma lista de conceitos básicos sobre orientação a objetos que darão suporte ao aluno para interagir com a ferramenta.

5.2. Resolução de Exercícios

O simulador receberá um enunciado do professor, que poderá ser salvo de forma a poder utilizá-lo mais vezes (Figura 2). O aluno, por sua vez irá buscar o enunciado proposto e salvo pelo professor (Figura 3).

É importante salientar que não houve a preocupação na distinção entre o usuário professor e aluno, através de mecanismos de controle de autenticação de senhas, tendo visto que este não era o objetivo principal do projeto. Este controle será implementado na nova fase da ferramenta.



Figura 2: Cadastro dos enunciados de exercícios propostos pelo professor

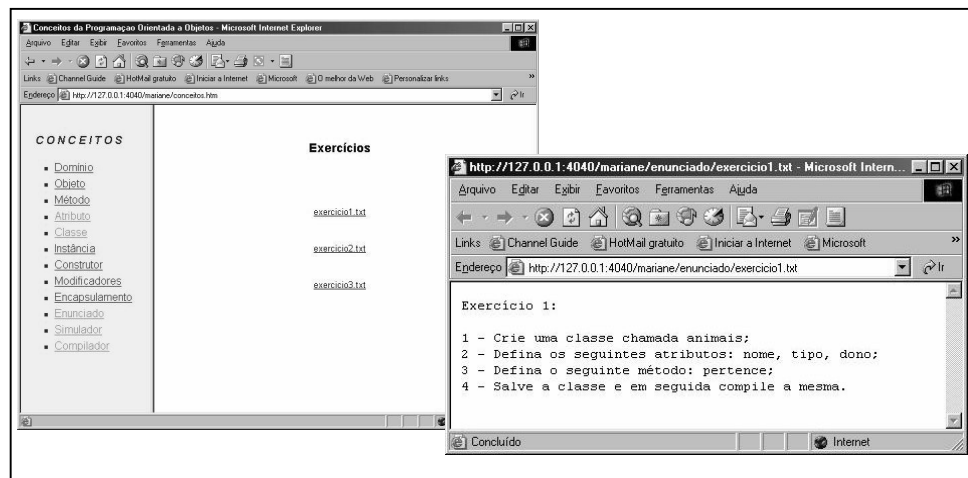


Figura 3: Enunciados dos exercícios disponíveis aos alunos

Após verificar o enunciado através do link “Enunciado” o aluno deve acessar o link “Simulador” para iniciar o processo de resolução.

Durante o processo de resolução do exercício o aluno fornecerá, gradativamente, todos os dados necessários para construção do código-solução, que partem das características gerais da classe, passam pela definição dos atributos e métodos e terminam com a definição dos valores iniciais dos atributos para serem utilizados pelo construtor da classe. O percurso percorrido pela ferramenta é apresentado na Figura 4.

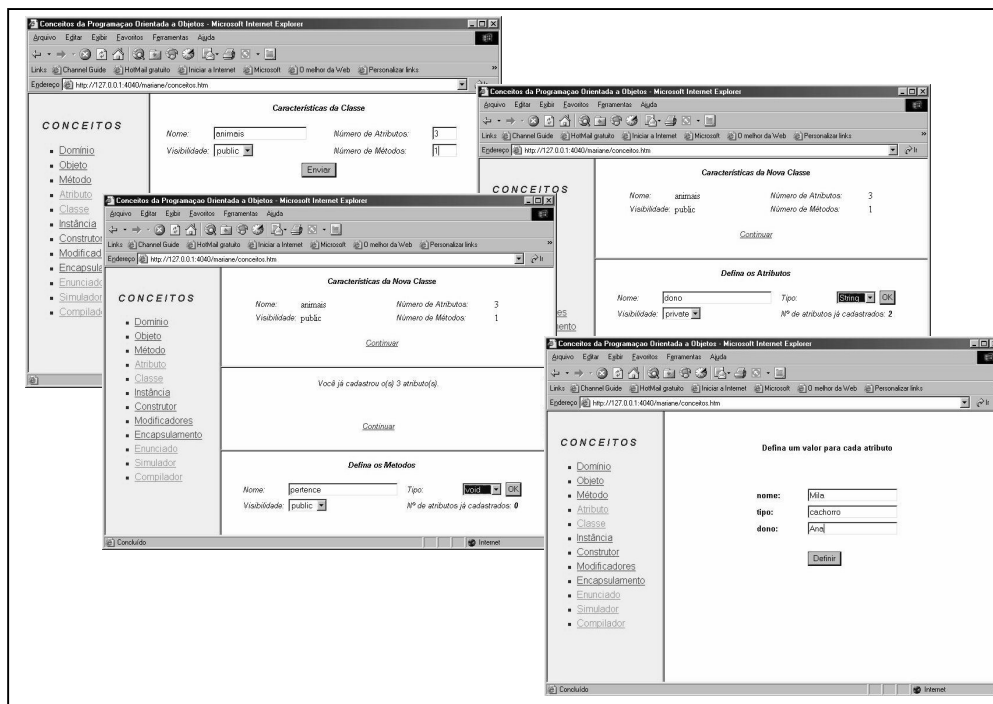


Figura 4: Construção da solução do exercício passo a passo

Quando o aluno terminar de fornecer todos os dados solicitados para a resolução do exercício, a ferramenta irá gerar para o aluno o código correspondente aos dados que ele informou em Java.

Esse código poderá ser visualizado e, se o aluno quiser poderá alterar o mesmo. A classe gerada pode ser salva no diretório desejado (Figura 5).

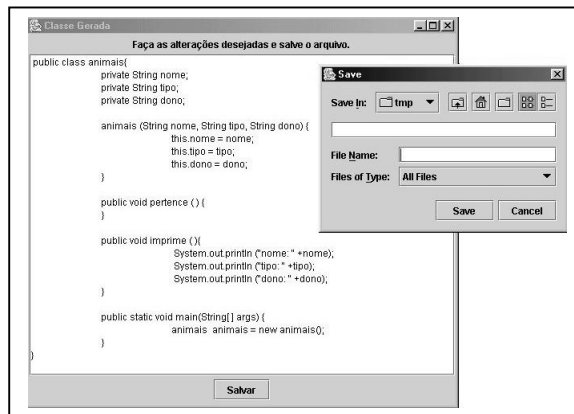


Figura 5: Código gerado pela ferramenta que pode ser armazenado pelo aluno em qualquer pasta

Após ter construído a solução para o exercício o aluno terá a possibilidade de analisar o código gerado e comparar o mesmo com as definições e exemplos dos conceitos da orientação a objetos disponibilizados na ferramenta. Desta forma, ele pode realizar um paralelo do abstrato (conceitos) para o concreto (código gerado).

O resultado obtido é de grande valia ao aluno, pois mostra que a partir de dados abstratos por ele informados, um código numa linguagem orientada a objetos foi gerado. Ressaltando dessa forma a importância do estudante ter bem sedimentado os conceitos da programação orientada a objetos antes de construir seus códigos. O entendimento das conceituações passa a ser fundamental para a geração do resultado final.

Além disto, fazer a transcrição de abstrações para a geração concreta de um código é muito valorosa para que o aluno entenda o processo de construção que deve ser utilizado para a resolução de problemas.

5.3. Compilando os Códigos Gerados pela Ferramenta

Caso o aluno tenha feito alguma modificação na classe, possíveis erros poderão ser detectados através da compilação do código gerado (Figura 6). Recurso este também disponibilizado pela ferramenta, que irá apresentar ao final da compilação um relatório de erros ou uma mensagem de sucesso da compilação.

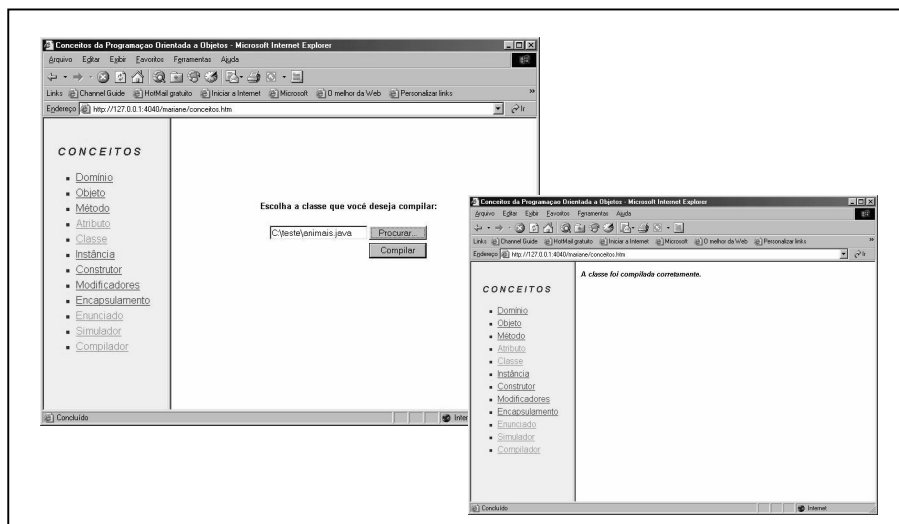


Figura 6: Compilação do código gerado pela ferramenta

6. Conclusões

O aluno que está iniciando em curso de programação normalmente apresenta maior afinidade com o paradigma imperativo. Isto porque o mesmo se aproxima mais dos processos lógicos utilizados na resolução de problemas.

Desta forma, é importante apresentar aos alunos, diferentes alternativas que auxiliem a aprendizagem de um novo paradigma de programação.

O uso da tecnologia no ensino abre espaço para um novo cenário que enriquece o processo de ensino-aprendizagem. Nesse novo cenário professor e aluno devem interagir, participando dinamicamente do processo, construindo juntos o caminho da aprendizagem.

O aluno passa a não ser um objeto estático como antigamente e o professor necessita abordar os conceitos de uma forma mais dinâmica.

Através de um simulador é possível se verificar a mudança de comportamento de ambos os lados, pois esta ferramenta proporciona um relacionamento mais interativo entre aluno e professor. Além de estimular o aluno a aprender, auxilia o professor no processo de ensino promovendo um crescimento pessoal para ambos.

O protótipo apresentado neste trabalho prioriza que os esforços dos alunos iniciantes na programação orientada a objetos sejam concentrados nos conceitos fundamentais durante o estudo. Por isso, a ferramenta disponibiliza os conceitos iniciais e necessários ao aluno para desenvolver códigos modelados em uma linguagem orientada a objetos.

Uma das vantagens da ferramenta é o aluno não precisar aprender a sintaxe de uma linguagem em específico para criar uma classe, será necessário apenas o conhecimento dos conceitos básicos da programação orientada a objetos para que ele possa preencher as características solicitadas e ter como retorno sua classe criada em uma determinada linguagem. Tal aplicação permite que o aluno visualize aquilo que foi estudado, fortalecendo assim os alicerces que irão sustentar sua aprendizagem no assunto.

Referências Bibliográficas

BRIDGEMAN, S.; GOODRICH, M.T.; KOBOUROV, S.G.; TAMASSIA, R.; **PILOT: An Interactive Tool for Learning and Grading:** SIGCSE Bulletin Conference Proceedings – The 31° SIGCSE Technical Symposium on Computer Science Education. 2000. 5f.

CORTELAZZO, I.B.C.; **Redes de Comunicação e Educação: Mudanças no Paradigma:** Revista Brasileira de Aprendizagem Aberta e a Distância.

COUTINHO, L.M.; **Educação à Distância: Algumas Considerações:** Revista Brasileira de Aprendizagem Aberta e a Distância. 2003.

GUIMARÃES, A.S.; **Novo Ecossistema Cognitivo: Pensamentos sobre Tecnologia de Informação e Comunicação e a metamorfose do Aprender:** Revista Brasileira de Aprendizagem Aberta e a Distância. 2003.

KORHONEN, A.; MALMI, L.; **Algorithm Simulation with Automatic Assessment:** SIGCSE Bulletin – The 5° Annual SISCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education. 2000. 4f.

MATTA, A.E.R.; **Comunidades em redes de computadores: abordagem para a Educação a Distância – EAD acessível a todos:** Revista Brasileira de Aprendizagem Aberta e a Distância. 2003.

OLIVEIRA, I.C.A.; ARAÚJO, C.F.; LÉDON, M.F.P.; **Objetos de Aprendizagem para o Ensino de Programação em Engenharia e Computação:** 3rd International Conference on Engineering and Computer Education. São Paulo, 2003. 5f.

PIERSON, W.C.; RODGER, S.H.; **WEB-based Animation of Data Structures Using JAWAA:** 2001. 5f.

RODRIGUEZ, M.I.; **Para tratar EAD com o devido Respeito:** Revista Brasileira de Aprendizagem Aberta e a Distância. 2003.

ROMISZOWSKI, A.; **E-learning: a diferença entre teoria e realidade:** Revista Brasileira de Aprendizagem Aberta e a Distância. 2003.

SANTOS, A.I.; **A Eficácia dos Fóruns de Discussão na Aprendizagem On-Line:** Revista Brasileira de Aprendizagem Aberta e a Distância. 2003.

SCHIMIGUEL, J.; ARAÚJO, C.F.; AMARAL, L.H.; **Desenvolvimento de Simulações para o Aprendizado em cursos na Web:** 3rd International Conference on Engineering and Computer Education. São Paulo, 2003. 5f.

SILVEIRA, I.J.; ESMIN, A.A.A. **AVA – Um ambiente Visual para a construção de algoritmos:** 3rd International Conference on Engineering and Computer Education. São Paulo, 2003. 6f.

STERN, L.; SONDERGAARD, H.; NAISH, L.; **A Strategy for Managing Content Complexity in Algorithm Animation:** SIGCSE Bulletin – The 4º Annual SISCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education. 1999. 4f.

TORI, R.; **O Virtual que marca Presença:** Revista Brasileira de Aprendizagem Aberta e a Distância. 2003.