

# **Criando Oportunidades de Aprendizagem em Linguagens de Programação através de Tutoria - uma pesquisa em andamento, mas em busca de parceria**

Aury de Sá Leite  
Júlio Santana Antunes  
Maria de Fátima C.L.Santos  
{aury, santana, flacaz}@feg.unesp.br  
Unesp/Campus de Guaratinguetá

## **Sumário**

Este artigo faz uma introdução sobre a utilização do computador como um auxiliar do trabalho docente, com a finalidade de exemplificar este papel, relata-se uma experiência bastante abrangente que vem sendo desenvolvida há quase dez anos pelos autores, sobre o ensino/aprendizagem de linguagens de programação através de tutoramento via computador. Neste processo, as concepções teóricas das linguagens FORTRAN, Pascal, C e C++ são apresentadas através de apostilas eletrônicas com alguns recursos de hipertexto. Módulos tutoriais contendo textos explicativos, programas prontos para "rodar", e sugestões de novos exercícios, podem ser carregados diretamente nos ambientes integrados de programação, podendo ser compilados, modificados e executados pelos estudantes.

Os bons resultados desta experiência educacional animou os autores a propor uma continuação deste projeto visando a integração dos diversos conjuntos de tutoriais já desenvolvidos e testados, num único sistema que permita ou o estudo em separado de cada uma das linguagens ou o estudo comparado das mesmas. Uma das formas de dar continuidade ao trabalho já realizado seria a de uma parceria com outros pesquisadores interessados no assunto.

**Palavras-Chave:** Aprendizagem e instrução auxiliada por computador; Computadores como auxiliares do trabalho docente; aprendizagem de Linguagens de Programação; Tutoria.

## **1.- Introdução**

No mundo todo, e rapidamente, a informática mudou, e vem mudando, os conceitos de trabalho, seja na área de produção, seja na área de prestação de serviços. Seria impossível imaginar-se que as escolas pudessem ficar alheias aos incríveis e até mesmo terríveis reflexos deste fenômeno. Na verdade, estas novas concepções vêm abalando, apesar de o fazerem com algum atraso, os alicerces pedagógicos e didáticos mais sólidos. Só para citar um exemplo, as máquinas de calcular quase que totalmente proibidas nas escolas de primeiro grau, aceitas com restrições nas escolas de segundo grau, e exigidas em algumas disciplinas, nas escolas de terceiro grau, deixaram de ser o centro de discussões de carácter pedagógico, pois foram ultrapassadas, e em muito, pela discussão sobre a utilização dos computadores em salas de aula.

Isto nos mostra que o antigo e ingênuo questionamento: "*deve-se utilizar máquinas de calcular nas salas de aula?*" foi substituído por um outro questionamento muito mais contundente: "*como utilizar os computadores em sala de aula?*". Deve-se notar que o foco

destas questões passou do "deve-se?" para o "como?", fazendo-nos perceber nitidamente que agora, nem ao menos se discute se os computadores devem, ou não, entrar porta a dentro pelas escolas [Setzer 1984]. Esta nova tecnologia, com todo o seu incontestável poder de subverter os conceitos de trabalho, veio também para subverter a prática pedagógica e os procedimentos didáticos, trazendo com isto a obrigação improrrogável de nos fazer repensar sobre as práticas escolares e as novas possíveis formas de criar oportunidades de aprendizagem, de acesso à informação e ao conhecimento e, finalmente, à cidadania responsável e consciente.

## **2.- Computadores nas Escolas**

Computadores nas escolas e novas formas de se repensar a educação passaram a formar um único bloco muito sólido e indissociável. Não basta somente introduzir-se os computadores na escola é preciso saber urgentemente como incluí-los no processo educacional.

Para se iniciar este tipo de discussão poderemos estabelecer duas formas básicas de se pensar o papel dos computadores nas escolas: a de *auxiliares ao trabalho docente* ("intelligent tutoring assistants") [Nunes et alii 1993] e a de *máquinas a serem estudadas, exploradas e compreendidas*, tentando-se envolver nesta última forma de estudo e exploração uma porção substancial da Informática, como ciência. Estas duas formas de pensar-se o papel dos computadores podem, ou não, ser excludentes. O que vai-se apontar neste artigo é que, mesmo podendo ser excludentes, estas formas de se utilizar os computadores podem ser harmonizadas.

## **3.- Computadores como Auxiliares do Trabalho Docente**

Nas universidades não se discute apenas a adequação dos conteúdos em função do desenvolvimento tecnológico provocados pela utilização da informática, mas as novas formas de se disponibilizar as oportunidades de aprendizagem através ou com o uso dos computadores. Da discussão sobre o uso ou não das máquinas de calcular passou-se a discutir a utilização dos denominados CAS (Computer Analytic Systems) - os Sistemas Analíticos Computacionais, tais como o Mathematica, o Derive e o Mapple como oráculos nas disciplinas ligadas ao Cálculo Diferencial e Integral e à Física, sendo que nas demais disciplinas se discute a utilização de Tutoria Inteligente [Wenger 1987; Polson 1988; Sá Leite 1999] ou de Softwares Educacionais Dedicados, como auxiliares ao trabalho docente e, mais, rediscute-se o papel das Linguagens de Programação na aprendizagem das Ciências Exatas e sua aplicabilidade na pesquisa científica.

## **4.- Computadores Como Máquinas Programáveis**

A aprendizagem de Linguagens Computacionais na Universidade vem passando por momentos cruciais. Já não se discute mais se o que deve ser ensinado é o FORTRAN, o Pascal ou o C. Discute-se hoje como inserir neste ensino as linguagens orientadas a objetos e as orientadas por eventos, pois dificilmente um projeto de iniciação científica que envolva programação poderá prescindir do uso de interfaces visuais amigáveis como aquelas produzidas aparentemente sem "muito esforço" pelas chamadas linguagens visuais. O Visual Basic, o Delphi, o Visual C++ e o Java, são alguns dos exemplos destas novas linguagens praticamente exigidas em qualquer projeto de engenharia de software.

O que se percebe é que é necessário o quanto antes se ensinar mais e melhor, ou seja, dar oportunidades de aprendizagem cada vez mais eficazes e eficientes aproveitando os recursos e potencialidades dos computadores [Valente 1993]. Os conceitos de eficácia e eficiência aplicável a este tipo de ensino poderão, aqui, ser convenientemente emprestados da Teoria de Programação: eficácia diz respeito à resolução do problema proposto e eficiência diz respeito a fazê-lo com um custo mínimo.

Em síntese, o problema que se apresenta no ensino atual de Linguagens de Programação envolve, no mínimo, a necessidade de aprendizagem integrada e coerente de uma linguagem imperativa (procedural e estruturada), uma linguagem orientada a objetos e finalmente uma linguagem orientada por eventos (uma linguagem visual). Podendo-se citar como exemplo desta integração a aprendizagem da Linguagem Pascal e do Delphi. Acrescente-se ainda a este quadro, somente a título de exemplo, as linguagens necessárias para algum tipo eventual de programação em sistemas dedicados, como o CAS (Computer Analytic Systems - Sistemas Analíticos Computacionais) e CAD (Computer Aided Design - Projeto Assistido por Computador), sistemas de Controle Numérico e Analógico e sistemas robóticos.

## **5.- Os Tutoriais de Linguagens de Programação**

A disciplina denominada Computação e Cálculo Numérico que é atualmente ministrada nos segundos anos das engenharias Mecânica, Civil e Elétrica na UNESP, Campus de Guaratinguetá, prevê a aprendizagem de uma linguagem computacional imperativa no primeiro semestre, com a finalidade de se vir a utilizá-la na implementação de programas relativos ao Cálculo Numérico já no segundo semestre. Atualmente foi criada, no curso de Engenharia de Produção, com uma ementa mais ambiciosa, a disciplina Linguagem de Programação, mas praticamente com o mesmo objetivo daquela anteriormente citada. Outras disciplinas, do rol das optativas, completam a formação de alguns dos alunos interessados, ministrando a disciplina denominada Programação Avançada de Computadores que envolve a aprendizagem de uma linguagem orientada a objetos e a disciplina Programação em Linguagens Visuais.

O que sempre se percebeu nestas disciplinas é que os estudantes, a partir da aprendizagem de alguma teoria sobre a linguagem de programação, passavam a entender-se diretamente com os computadores, passando o professor a ser um mero consultor e, assim mesmo, quando os estudantes não conseguiam resolver o problema entre eles próprios. Foi assim que se passou a ministrar tópicos teóricos em sala de aula seguido da proposição de tarefas a serem realizadas diretamente nos computadores. Inicialmente era dado a estas tarefas o tratamento típico de *exercícios de laboratório* - tarefas com passos bem definidos e quase-algorítmicos, sugestões e proposta de exercícios complementares e a fatal emissão de um relatório ou da entrega dos resultados, para avaliação.

O processo funcionava, mas era lento e burocrático. Alguns alunos venciam as etapas de forma rápida e queriam mais, mas tinham que se adaptar ao ritmo do grupo e das aulas, onde a teoria era apresentada. Era preciso pensar-se uma maneira de se criar oportunidades de aprendizagem que satisfizessem às exigências e necessidades individuais de cada um dos estudantes. Alguns desejam aprender programação para se dedicar a projetos de Iniciação Científica, outros porque estavam fortemente motivados para aquele tipo de aprendizagem ou estavam fascinados pela possibilidade de programar computadores, alguns porque possuíam um computador em casa e tinham a possibilidade de utilizá-los em tempo integral naquele novo tipo de aprendizagem.

A solução foi surgindo ao longo destes anos todos e de acordo com o próprio progresso da tecnologia [Sá Leite, A. et alii 1993]. Uma apostila apresentava a teoria de forma resumida.

Ela deveria ser utilizada como um manual da Linguagem. Poderia conter um histórico, um programa exemplo básico, dados técnicos sobre a linguagem envolvendo as características a sintáticas e semânticas, seguida de exemplos a serem digitados, compilados e executados. Os laboratórios foram transformados em pequenos tutoriais. Estes tutoriais continham basicamente um pequeno texto introdutório normalmente abordando aspectos teóricos e comentários, um programa executável via ambiente de programação e sugestões de modificações a serem feitas diretamente no programa ali apresentado. O estudante carregava o tutorial diretamente no ambiente de programação estudava o texto, executava o programa e analisava os resultados. Em seguida fazia as modificações solicitadas como exercício no programa anteriormente executado e conferia os novos resultados.

Esta maneira de apresentar os cursos de Linguagens de Programação hoje abrange as linguagens FORTRAN (em fase de reestruturação), Pascal (com 50 tutores), C (com 40 tutores) e C++ (orientação a objetos - com 12 tutores).

As linguagens são apresentadas inicialmente através de apostilas eletrônicas com alguns recursos de hipertexto, contendo dados históricos e teóricos sobre as mesmas, seguidos de exemplos executáveis. Os módulos tutoriais contêm textos explicativos, programas prontos para "rodar", e sugestões de novos exercícios, e podem ser carregados diretamente nos ambientes integrados de programação, podendo ser compilados, modificados e executados.

## 6.- Idéias para o Futuro

O que se está buscando agora é uma forma de criar-se um conjunto de janelas padronizadas (um conjunto de *templates*), em linguagem visual, que permita ao usuário acessar qualquer uma das apostilas daquelas linguagens, de forma temática, bem como acessar os tutoriais, e independentemente dos ambientes de programação, poder executar os programas e visualizar seus resultados. E mais, que através de recursos de hipermídia, temas teóricos interdependentes bem como as possíveis diferenças entre as linguagens possam ser acessados em grupo ou seqüencialmente, para poderem ser estudados e comparados. Os programas já com as sugestões de modificação deveriam também poder ser acessados e executados.

A tarefa de se implementar estes *templates* é possivelmente fácil, a dificuldade seria encontrada na necessária homogeneização a ser dada aos diversos tópicos teóricos a serem abordados em cada uma das linguagens, no tratamento a ser dado a grupos de programas referentes a uma mesmo tópico teórico e no estabelecimento de correspondência entre os programas que utilizassem os mesmos algoritmos ou os mesmos conceitos. O tratamento específico a ser dado às características diferenciais de cada uma destas linguagens seria um outro desafio a ser enfrentado.

## 7.- Conclusões

O autor está aberto às sugestões e trocas de idéias, não estando descartada a possibilidade de uma parceria sob a forma de co-orientação em uma possível dissertação de mestrado, com a finalidade básica de estudar, de organizar e de desenvolver a proposta feita no item anterior deste artigo.

Cabe ressaltar que o perfil do candidato ao desenvolvimento deste projeto deve ser o de educador, com bom conhecimento de Linguagens de Programação no que diz respeito às linguagens imperativas, orientadas a objetos e visuais. No tocante aos aspectos educacionais necessários para o desenvolvimento do projeto, o estudo das teorias de Ausubel e Novak - da Aprendizagem Significativa e dos Mapas Conceituais [Ausubel, Novak & Hanesian 1979], de

Gagné - sobre as Habilidades Intelectuais e Mapas de hierarquia de Aprendizagem [Gagné & Medsker 1996; Gagné, Briggs & Wager 1992] de Vygotsky - sobre a Sócio-aprendizagem e a parceria mais competente [Vygotsky 1978; 1993] e a de Papert - sobre o Construcionismo [Harel & Papert 1993; Papert 1993] , precisaria estar em pauta, com a finalidade de dar embasamento científico ao trabalho.

## Bibliografia

- [Ausubel, Novak & Hanesian 1979] Ausubel, David P., Novak, J.D. & Hanesian, H. ***Educational Psychology : A Cognitive View***. New York, Holt, Rinehart and Winston, 1979, 2<sup>nd</sup> Ed.
- [Gagné, Briggs & Wager 1992] Gagné, Robert M., Briggs, Leslie J. Wager, Walter W. ***Principles of Instructional Design***. Fort Worth, Harcourt Brace & Company, 1992.
- [Gagné & Medsker 1996] Gagné, R.M. & Medsker, K.L. ***The conditions of learning – Training applications***. Fort Worth, TX: Harcourt Brace College Publishers, 1996.
- [Harel & Papert 1993] Harel, Idit & Papert, Seymour (editors). ***Constructionism- Research Reports and Essays, 1985-1990***. Norwood, New Jersey, Ablex, 1993.
- [Nunes et alii 1993] Nunes, Maria das Graças Volpe et alii. ***A Network-Based Model for Intelligent Tutoring Systems***. Porto Alegre, X Simpósio Brasileiro de Inteligência Artificial, 1993.
- [Papert 1993] Papert, Seymour. ***The Children’s Machine – Rethinking Scholl in the Age of the Computer***. New York, Basic Books, 1993.
- [Polson 1988] Polson, Martha C. & Richardson, J.Jeffrey (Eds). ***Foundations of Intelligent Tutoring Systems***. Hillsdale, New Jersey, Laurence Erlbaum, 1988.
- [Sá Leite, A. et alii 1993] Sá Leite, Aury et alii. ***Ensino-Aprendizagem de Linguagens de Programação em Cursos de Engenharia***. XXI Congresso Brasileiro de Ensino de Engenharia, Belo Horizonte, de 16 a 19 de novembro de 1993, vol. 2, páginas 912/920.
- [Sá Leite 1999] Sá Leite, Aury. ***Um Modelo de Sistema Educativo Cognitivista Baseado em Tutoria Inteligente Adaptativa Via Aderência Conceitual***. Tese de doutorado, ITA, 1999.
- [Setzer 1984] Setzer, Valdemar W. ***Manifesto contra o uso de computadores no ensino de 1º Grau***. São Paulo, Antroposófica, 1984.
- [Valente 1993] Valente, José Armando (organizador). ***Computadores e Conhecimento – Repensando a Educação***. UNICAMP, Gráfica Cent4al, 1993.
- [Vygotsky 1978] Vygotsky, L.S. ***Mind in society: The development of higher psychological process***. Cambridge, Massachusetts: Harvard University Press, 1978.
- [Vygotsky 1993] Vygotsky, L.S. ***Pensamento e Linguagem***. São Paulo, Martins Fontes, 1993.
- [Wenger 1987] Wenger, Etienne. ***Artificial Intelligence and Tutoring Systems – Computational and Cognitive Approaches to the Communication of Knowledge***. Los Altos, CA, Morgan Kaufmann Publishers, Inc, 1987.