

Portugol IDE – Uma ferramenta para o ensino de programação

António Manso^{*}, Luís Oliveira^{*}, Célio Gonçalo Marques^{}**

^{*} Escola Superior de Tecnologia de Tomar, Instituto Politécnico de Tomar, Quinta da Contador 2300-313 Tomar, Portugal

^{**} Escola Superior de Gestão de Tomar, Instituto Politécnico de Tomar, Quinta da Contador 2300-313 Tomar, Portugal

Email: manso@ipt.pt, loliveira@ipt.pt, celiomarques@ipt.pt

Resumo

O ensino da programação, geralmente, é realizado com recurso a linguagens de programação e a ambientes de desenvolvimento direccionados para o contexto comercial. Estas linguagens e ambientes quando utilizados na contexto educativo do ensino da programação de computadores nem sempre facilitam o desenvolvimento da lógica de programação. O Portugol IDE é um ambiente de aprendizagem direccionado, especialmente, para o ensino superior que possui uma linguagem de especificação de algoritmos baseada na língua materna dos alunos, o portugol; uma linguagem gráfica que reforça o conceito de execução sequencial, os fluxogramas; bem como um conjunto de características que o tornam recomendável para os primeiros passos na programação. Nesta comunicação é apresentado o ambiente de aprendizagem Portugol IDE, é descrita a sua utilização em contexto de aula e são referidas algumas inovações a introduzir futuramente.

Palavras-chave: Portugol IDE; linguagem fluxográfica; linguagem portugol; Ensino de Programação.

Abstract

Teaching of programming usually resorts to programming languages and development environments directed to commercial purposes. When used to teach programming basics, these languages and environments do not always facilitate the development of programming logic. Portugol IDE is a learning environment particularly directed to higher education which has an algorithm specification language based on the students' mother tongue (portugol), a graphic language that reinforces the concept of sequential execution (flowcharts), and a set of characteristics that makes it suitable for first steps in programming. This paper presents the learning environment Portugol IDE and its use in class environment as well as some innovations to be introduced in the future.

Keywords: Portugol IDE; flowchart language; portugol language; programming teaching.

1 Introdução

A programação de computadores é uma actividade complexa e com nível elevado de abstracção que, tradicionalmente, é ensinada recorrendo a linguagens de programação e a ambientes de desenvolvimento projectados para o contexto industrial da produção de software.

Os cursos e as unidades curriculares de programação de nível introdutório têm tradicionalmente elevadas taxas de reprovação (Butler & Morgan, 2007; Jenkins, 2002; Lahtinen, Mutka & Jarvinen, 2005). Este fenómeno é geral, não sendo exclusivo de um grupo de Instituições ou de um curso específico, ou mesmo de um grupo de alunos com um determinado perfil de formação. Este nível de insucesso no ensino da programação tem merecido a atenção de vários investigadores (Giangrande, 2007). A investigação vai desde os aspectos mais técnicos aos aspectos pedagógicos. Relativamente aos aspectos técnicos é o paradigma da programação que se deve usar, o que suscita mais divergências (Lister, Berglund, Clear, Bergin, Garvin-Doxas, Hanks, Hitchner, Luxton-Reilly, Sanders, Schulte & Whalley, 2006). Existem aspectos que são comuns aos paradigmas de programação orientados a objectos e procedimentais, nomeadamente, a definição de variáveis e de estruturas, a modularização, a definição de estruturas de dados e o uso de estruturas de controlo de execução. Com vista a dar resposta a estes problemas foi desenvolvido um ambiente de execução de algoritmos que designámos de Portugol IDE (Manso & Oliveira, 2006). A expressão dos algoritmos na língua materna do aluno, ou sob a forma gráfica, permite que a sua atenção e se concentre na expressão lógica do algoritmo, e não na linguagem e no ambiente de programação.

2 Portugol IDE

A reforma introduzida pelo processo de Bolonha em Portugal reduziu a carga horária presencial de muitas unidades curriculares do Ensino Superior e veio trazer alterações importantes nas metodologias de ensino-aprendizagem, dando grande ênfase à aprendizagem activa. O Portugol IDE é um ambiente de execução de algoritmos onde os alunos têm acesso a ferramentas que lhes permitem expressar os algoritmos, realizar a verificação formal e a sua depuração em caso de erro. Por conseguinte, este ambiente permite melhorar a eficácia do método tradicional de desenvolvimento e verificação de algoritmos baseado na resolução de problemas em papel.

O Portugol IDE possui duas linguagens de suporte aos algoritmos: o portugol, uma linguagem formal baseada no português estruturado que permite a definição de algoritmos computacionais; e a linguagem fluxográfica, uma linguagem gráfica constituída por formas executáveis e por ligações entre elas que representam o fluxo de execução do algoritmo. Estas linguagens permitem realizar as operações necessárias para a codificação de algoritmos simples, são compatíveis entre si e é possível alternar entre as duas nas fases de edição, execução e depuração.

2.1 Linguagem portugol

A definição da linguagem portugol norteou-se por três princípios: possuir as estruturas de programação necessárias para o desenvolvimento do raciocínio algorítmico; ser fácil de aprender e permitir uma transição suave para as linguagens de alto nível.

Na tabela 1 é apresentado um resumo das instruções definidas na linguagem portugol. As operações definidas são os blocos básicos do desenvolvimento de software e os algoritmos complexos formam-se pela aglutinação desses componentes.

Tabela 1: Definição da linguagem portugol (Manso & Oliveira, 2006).

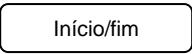

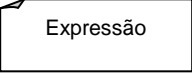
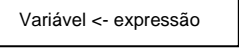


	Tipo de operações	Portugol
Definição de tipos de dados	Variáveis e constantes	Inteiro, real, lógico, carácter e texto
Operadores	Atribuição	<-
	Aritméticos	+, -, *, /, %, ^
	Relacionais	=, /=, >, >=, <, <=
	Lógicos	ou, e, não, xou
Entrada e saída de dados	Leitura	Ler
	Escrita	Escrever
Decisão condicional	Alternativa	Se então, senão
	Seleção	Escolhe, caso
Repetição	Controlo inicial	Enquanto faz Para de até
	Controlo final	Repete até
		Faz enquanto

A linguagem portugol permite que os alunos codifiquem os seus algoritmos numa linguagem próxima da que usualmente utilizam para comunicar, que aliada a uma sintaxe simples e flexível possibilita que se concentrem na resolução dos problemas e no raciocínio abstracto do algoritmo e não no ambiente e na linguagem onde expressam esse mesmo algoritmo. Por fim, a linguagem possui algumas características presentes nas linguagens de programação mais modernas, tais como a definição de variáveis quando elas são necessárias e a definição do tipo texto como básico.

2.2 Linguagem fluxográfica

A linguagem fluxográfica foi definida para fazer a representação gráfica das instruções presentes na linguagem portugol. Esta permite desenhar, visualizar e executar o algoritmo sob a forma de grafo dirigido. De forma a tornar a execução automática fizeram-se algumas alterações aos símbolos do fluxograma que tradicionalmente são utilizados para descrever os algoritmos. Procedemos ao desdobramento do símbolo de entrada/saída em dois: um símbolo para entrada de dados que permite que o utilizador introduza dados no algoritmo e um símbolo para impressão de informação na consola. De forma a manter a coerência com os fluxogramas tradicionais, o símbolo de entrada de dados também permite a definição de variáveis: se o utilizador introduzir uma variável que esteja definida na memória do algoritmo o símbolo altera o seu valor, se a variável não estiver definida, o símbolo define a variável e determina o seu tipo através do valor introduzido. O símbolo de processo é outro dos símbolos que ficou com as suas funcionalidades estendidas: para além de avaliar expressões e atribuir os seus resultados a símbolos que estão definidos em memória, também executa a definição de símbolos caso eles não existam. A tabela 2 apresenta os símbolos fluxográficos e a respectiva descrição.

Tabela 2: Símbolos e sintaxe da linguagem fluxográfica

Símbolo	Descrição funcional
	Terminal – Marca o início/fim do algoritmo. O fluxograma inicia-se com o símbolo terminal de início e termina com o símbolo terminal de fim.
	Ler – Lê a variável da consola e se necessário define-a em memória.
	Escrever - Escreve a expressão na consola.
	Processo - Calcula o valor de uma expressão, e se necessário define a variável de suporte em memória.
	Decisão - Desvia a fluxo de execução através do valor da condição. Marca o início das estruturas de decisão condicional e das estruturas de repetição com controlo inicial.
	Conector - Unificação de fluxo. Marca o início das estruturas de repetição com controlo final e o final das estruturas de decisão.

A linguagem fluxográfica permite a visualização dos blocos operacionais de um programa e o fluxo de execução que esses mesmos blocos geram. Os algoritmos são desenhados através de símbolos e fluxos tornando-se mais clara a execução sequencial das instruções dos programas. O desenho de algoritmos também é menos atreito a erros que as linguagens escritas uma vez que as formas são parametrizáveis e dotadas de funcionalidades inteligentes.

2.3 Evolução e caracterização do Portugol IDE

O Portugol IDE foi desenvolvido em 2004 no âmbito de um projecto de fim de curso e introduzido em 2005 nas unidades curriculares dos cursos de licenciatura e de especialização tecnológica da Escola Superior de Tecnologia de Tomar (ESTT)

do Instituto Politécnico de Tomar (IPT). O projecto iniciou-se com a definição da linguagem algorítmica (versão 0.5). Na versão 1.0 foi introduzido o núcleo de cálculo e de execução da linguagem algorítmica. Na versão 1.5 foram introduzidas as variáveis do tipo vector. Na versão 2.0 foi adicionado o editor, o parser e o executor de fluxogramas. A versão actual é a 2.2, que se caracteriza pelo reforço da compatibilidade entre o português e a linguagem fluxográfica. O Portugol IDE rege-se pela licença GNU e por conseguinte é de utilização livre e de código aberto. Esta forma de licenciamento permite que qualquer instituição o use livremente e que outros grupos de investigação adicionem novas funcionalidades.

O Portugol IDE dá especial importância ao desenvolvimento do raciocínio algorítmico, por isso, os algoritmos podem ser executados passo a passo, mostrando as instruções que são executadas e a forma como essas mesmas instruções afectam as variáveis que estão definidas em memória.

A figura 1 apresenta o algoritmo de Euclides para o cálculo do mínimo múltiplo comum. O módulo de fluxograma permite fazer a prototipagem rápida do algoritmo em que as variáveis manipuladas são deduzidas pelos valores introduzidos pelo utilizador ou calculados pelo algoritmo. O símbolo de “leitura” deduz e define as variáveis a e b e o símbolo de “processo” define a variável resto.

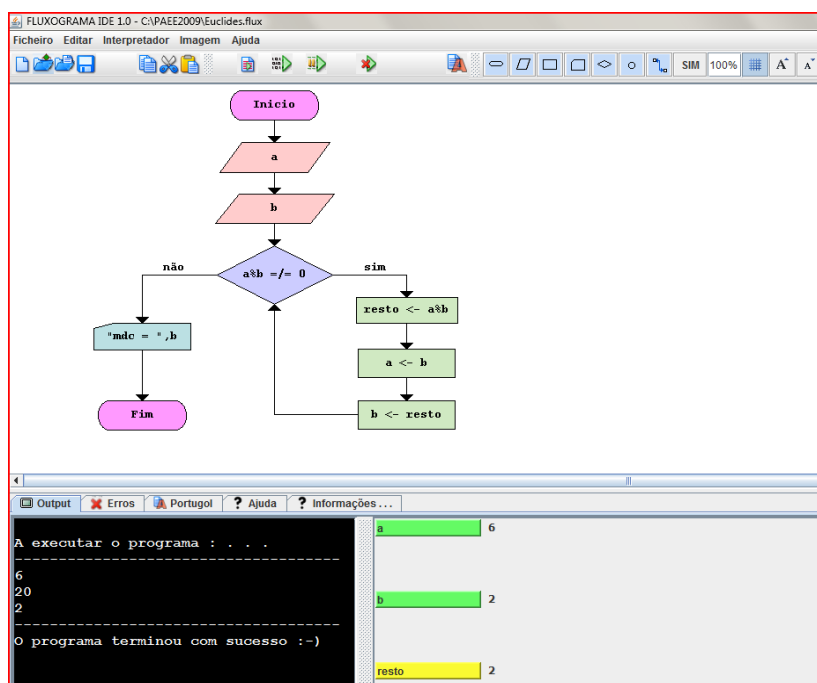


Figura 1: Algoritmo de Euclides em linguagem fluxográfica

Depois do utilizador correr pelo menos uma vez o fluxograma o sistema traduz o fluxograma para linguagem português com uma sintaxe correcta. Se o fluxograma não tiver sido executado, o sistema identifica as variáveis que são necessárias para a sua execução mas não consegue deduzir o tipo de dados que lhes está associado. A figura 2 apresenta o programa escrito em português que foi deduzido a partir do fluxograma da figura 1. A tradução da linguagem português para a linguagem fluxográfica não levanta qualquer ambiguidade uma vez que a cada instrução do programa escrito em português corresponde um símbolo na representação fluxográfica (Manso, Oliveira & Marques, 2009).

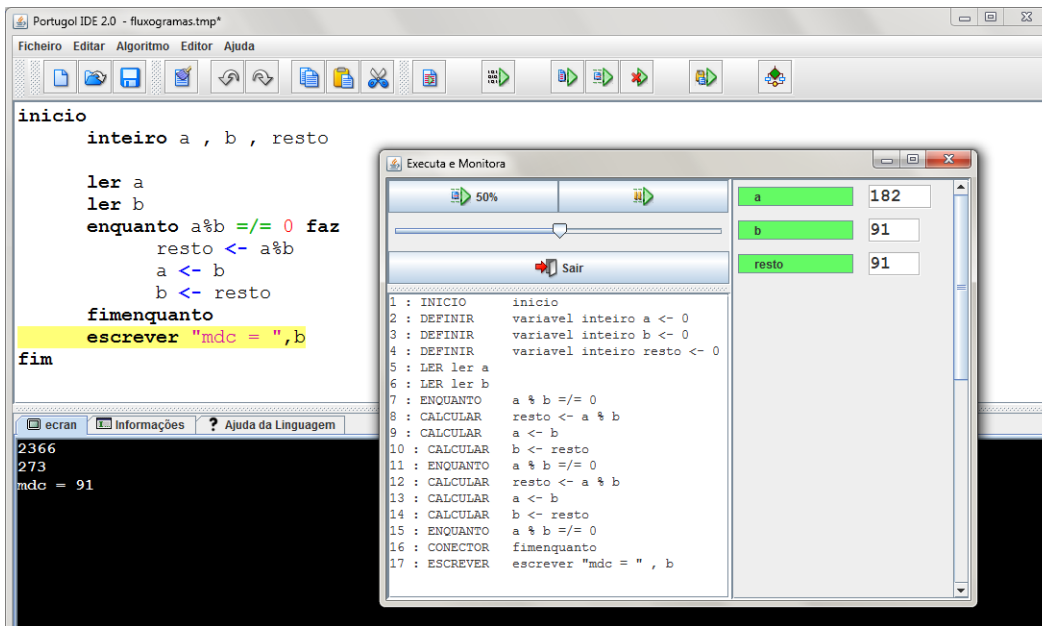


Figura 2: Algoritmo de Euclides escrito em portugol

O ambiente de edição, execução e depuração de algoritmos é mais simples que os ambientes desenvolvimento que servem de suporte às linguagens de programação mais tradicionais. O ambiente agradável e intuitivo do Portugol IDE, a simplicidade das linguagens de suporte e as ferramentas de auxílio à execução e depuração do algoritmo facilitam o desenvolvimento do raciocínio algorítmico.

3 Utilização em contexto de aula

A adequação dos cursos do Ensino Superior ao modelo de ensino e aprendizagem preconizado pelo Processo de Bolonha veio alterar profundamente os papéis do professor e do aluno. O aluno passou a ter um papel mais activo na aprendizagem, conduzindo o próprio processo de aprendizagem, ao invés, o professor passou a ser um orientador, um facilitador da aprendizagem. Pretende-se desta forma, transitar de um sistema de ensino baseado na ideia de transmissão de conhecimentos para um ensino baseado no desenvolvimento de competências (Decreto-Lei n.º 74/2006, de 24/03).

Nesta mudança de paradigma, assumem especial importância: a associação da carga horária lectiva das unidades curriculares ao trabalho autónomo dos alunos (e o conseqüente aumento do trabalho docente de acompanhamento da aprendizagem), a redução do número de horas de contacto em sala de aula, e a aprendizagem ao longo da vida (Guedes, Lourenço, Filipe, Almeida & Moreira, 2007).

Este modelo de ensino e aprendizagem pretende implementar uma abordagem construtivista, segundo a qual o conhecimento é construído pelos próprios alunos e não é algo que é transmitido pelo professor e memorizado pelos alunos (Arends, 1995).

Esta abordagem é baseada em dois pressupostos: a realidade é subjectiva e a aprendizagem resulta da construção que o sujeito faz do que o rodeia (Carvalho, 1999). Segundo a corrente construtivista, o conhecimento é temporário, passível de desenvolvimento, não objectivo, estruturado internamente e mediado social e culturalmente e a aprendizagem é como um processo auto-regulador do conflito entre o conhecimento preexistente do mundo e os novos conhecimentos com que o indivíduo se vai deparando (Fosnot, 1996). Esta abordagem implica que os conteúdos curriculares tenham que ser elaborados de acordo com necessidades dos alunos e que sejam facilmente personalizáveis para satisfazer necessidades individuais (Reigeluth, 1999).

As tecnologias de informação e comunicação assumem, por isso, um papel muito importante neste novo paradigma pois promovem uma aprendizagem centrada no aluno, exigindo-lhe uma atitude mais pró-activa no processo de ensino-aprendizagem.

É neste contexto que surge o Portugol IDE, um ambiente de execução de algoritmos onde os alunos têm acesso a ferramentas que lhes permitem expressar os algoritmos, realizar a verificação formal e a sua depuração em caso de erro. Esta ferramenta baseada na corrente construtivista permite que o aluno tenha um papel mais activo na aprendizagem, melhora a eficácia do desenvolvimento e verificação de algoritmos e promove o raciocínio algorítmico.

O ensino da programação na ESTT inicia-se no primeiro semestre com a unidade curricular de Introdução à Programação. A carga lectiva semanal é distribuída em 2 horas para a componente teórica e em 3 horas para a componente prática laboratorial, sendo também contempladas horas de acompanhamento tutorial.

O relatório Computing Curriculum 2001, elaborado pela ACM (Association for Computing Machinery) e pelo IEEE (Institute of Electrical and Electronic Engineers) identifica seis estratégias que podem orientar o ensino da programação: imperativa, objectiva, funcional, abrangente, algorítmica e maquinal (Guerreiro, 2003; IEEE Computer Society, 2008). A estratégia imperativa caracteriza-se por se concentrar nos aspectos relacionados com as instruções, procedimentos, estruturas de decisão e de controlo de fluxo. A estratégia objectiva, aborda numa primeira fase aspectos relacionados com a programação orientada a objectos. Na estratégia funcional, começa-se por abordar os aspectos relacionados com aspectos fundamentais à resolução de problemas computacionais como, por exemplo, a recursividade, a manipulação de estruturas de dados e de funções. Na estratégia abrangente, a programação é apenas um dos aspectos que são abordados na primeira disciplina de nível introdutório, a par desta são abordados assuntos relacionados com a arquitectura de computadores, bases de dados, entre outros. Os aspectos específicos da programação são abordados em disciplinas posteriores. Na estratégia algorítmica, não é usada uma linguagem de programação formal, em vez desta usa-se uma pseudo-linguagem. A linguagem de programação é introduzida numa fase posterior. Por último, surge a estratégia maquinal, na qual se começa por abordar aspectos relativos à arquitectura e ao funcionamento da máquina. Na estratégia maquinal são usadas linguagens de programação de baixo nível, as linguagens de alto nível são introduzidas depois.

Nas unidades curriculares de Introdução à Programação é usada a estratégia algorítmica. Segundo o ACM todas as estratégias são válidas mas todas apresentam problemas. No caso da estratégia algorítmica, um dos problemas identificados está relacionado com o facto de não existirem ferramentas que permitam a simulação dos algoritmos e a depuração de erros. Por conseguinte, a resolução dos problemas é realizada com recurso a papel e lápis. Este método não facilita o trabalho do aluno fora do período lectivo, porque não dispõe de ferramentas que lhe permitam verificar se a solução de um problema está correcta, e no caso de não estar onde estão os erros. O Portugol IDE pretende dar resposta a estes problemas.

Nas aulas laboratoriais não são resolvidos problemas no quadro, em vez disso é fomentado a aprendizagem activa. São distribuídos aos alunos guiões que contêm problemas e orientações que devem ser tidas em conta na sua resolução. Estes guiões são elaborados tendo por base que parte do trabalho é realizado pelo aluno fora do horário lectivo. A resolução dos guiões é classificada e entra no cálculo da nota final.

O Portugol IDE facilita a aprendizagem activa dentro e fora da sala de aula porque permite validar os algoritmos, facilita a depuração dos erros e suporta duas linguagens: linguagem portugol e linguagem fluxográfica. Acresce ainda o facto da utilização do Portugol IDE permitir que sejam propostos aos alunos exercícios mais complexos que suscitem mais interesse e que são fundamentais para o desenvolvimento do raciocínio algorítmico.

Os alunos sentem-se mais motivados já que o Portugol IDE permite a construção de algoritmos de uma forma mais rápida e agradável, bem como, uma correcção imediata dos mesmos. O professor deixou de ser o detentor da solução e passou a ter mais tempo para os orientar.

O facto de usarmos uma ferramenta de software para expressar algoritmos facilita a sua entrega através do nosso sistema de gestão de aprendizagem, o Moodle (<http://moodle.dei.estt.ipt.pt>).

Para procedermos à avaliação da ferramenta Portugol IDE relativamente à sua facilidade de utilização, eficiência, facilidade de recordar e satisfação (Nielsen, 2003) realizámos um estudo com 32 alunos do 1.º ano do curso de Engenharia Informática da ESTT. Foram utilizados dois questionários: questionário de opinião e questionário de ícones. Através do questionário de opinião pretendeu-se registar a posição dos sujeitos relativamente a diversas características da ferramenta. Com o questionário de ícones pretendeu-se verificar se o sistema é fácil de recordar. Ambos os questionários foram criados e distribuídos através da ferramenta da Web 2.0, SurveyMonkey (<http://www.surveymonkey.com>).

Os sujeitos consideraram o aspecto da interface agradável, a estrutura dos menus adequada, o tipo de letra de fácil leitura e os ícones sugestivos. Estes consideraram útil, o módulo de execução do fluxograma, a escrita em forma de fluxograma, a visualização do estado das variáveis e a ajuda na correcção dos erros. Segundo estes, os modos de execução ajudam muito na compreensão dos algoritmos e a execução em modo passo-a-passo auxilia muito a detecção de erros. Estes referiram ainda que é fácil codificar algoritmos em linguagem estruturada e em fluxograma no Portugol IDE.

Através do questionário de ícones constatámos que os sujeitos identificaram facilmente os ícones, o que é um bom indicador de que o sistema é fácil de lembrar.

O Portugol IDE está também a ser utilizado em outras instituições de ensino superior da comunidade lusófona, como a Universidade Regional Integrada e o UDF - Centro Universitário, e em várias escolas secundárias nacionais, entre elas, a Escola Secundária da Sertã e a Escola Secundária de Foz Côa.

4 Conclusão

Os métodos de ensino com recurso a tecnologias de informação e comunicação que fomentam a apreensão progressiva e intuitiva de conteúdos e a capacidade de raciocínio abstracto, estão a ter imenso sucesso na sociedade de informação em que hoje vivemos.

O Portugol IDE foi desenhado de forma a auxiliar o ensino e a aprendizagem das técnicas basilares de programação de computadores. A definição de um conjunto mínimo de instruções para as linguagens portugol e fluxográfica facilita a aprendizagem dos comandos e dos símbolos, mas não impede que estas linguagens sejam suficientemente poderosas para a construção de algoritmos complexos. O ambiente de desenvolvimento, execução e depuração foi concebido de forma a facilitar a tarefa de programar e de correcção dos erros de lógica e de sintaxe que eventualmente os alunos possam cometer.

Os resultados dos testes realizados com vista a apurar a aptidão do Portugol IDE, ou seja, a sua utilidade e usabilidade (Grudin, 1992) mostram que os alunos estão bastantes satisfeitos com este ambiente de execução de algoritmos considerando as suas inovações de grande utilidade para uma melhor aprendizagem da programação. Estes consideram também que se trata de um ambiente de estudo eficiente, fácil de utilizar, agradável de utilizar e fácil de recordar. Também os professores que ensinam programação e que utilizam o programa se encontram satisfeitos, fazendo chegar a sua opinião através de correio electrónico.

5 Trabalho futuro

O ensino de conceitos mais avançados de programação de computadores requer a utilização de estruturas de dados complexas e a modularização dos programas. Estamos neste momento a redefinir a linguagem algorítmica para dar suporte a estruturas de dados heterogéneas e á execução de funções iterativas e recursivas. A evolução do IDE passa de desenvolvimento um módulo de avaliação lógica dos algoritmos. Este módulo aumentará a autonomia da aprendizagem autónoma do aluno e está previsto o acesso a repositórios de problemas através de tecnologia Web.

Referências

- Arends, R. (1995). *Aprender a Ensinar*. Alfragide: McGraw-Hill.
- Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts , *ASCILATE 2007 Singapore*, 99-107.
- Carvalho, A. A. A. (1999). *Os Hipermedia em Contexto Educativo. Aplicação e Validação da Teoria da Flexibilidade Cognitiva*. Braga: Universidade do Minho, Instituto de Educação e Psicologia, Centro de Estudos em Educação e Psicologia.
- Fosnot, C. T. (1996). *Construtivismo e Educação: Teoria, Perspectivas e Prática*. Lisboa: Instituto Piaget.
- Giangrande, E. (2007). CS1 Programming Language Options, *Journal of Computing Sciences in Colleges*, 22 (3), 153-160.

- Grudin, J. (1992). Utility and Usability: Research Issues and Development Contexts, *Interacting with Computers*, 4, (2), 209-217
- Guedes, M. G., Lourenço, J. M., Filipe, A. I., Almeida, L., & Moreira, M. A. (2007). *Bolonha. Ensino e Aprendizagem por Projecto*. Lisboa: Centro Atlântico.
- Guerreiro, P. (2003). *Relatório de Programação I para a Licenciatura em Engenharia Informática*, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa. Acedido em Fevereiro 25 de 2009 de http://ctp.di.fct.unl.pt/~pg/docs/rel_p1.htm
- IEEE Computer Society (2008). *Educational Activities*. Computing Curricula Development. Acedido em Maio 10 de 2009 de <http://computer.org/education/cc2001/>.
- Jenkins, T. (2002). On the difficulty of learning to program. *Proc. of the 3rd Annual Conference of the LTSN Centre for Information and Computer Science*, 27-29.
- Lahtinen, E., Mutka, K. A., & Jarvinen, H. M. (2005). A Study of the difficulties of novice programmers, *Proc. of the 10 th Annual SIGCSE Conference on Innovation and Technology in Computer ITiCSE'05*, 14-18.
- Lister, R., Berglund, A., Clear, T., Bergin, J., Garvin-Doxas, K., Hanks, B., Hitchner, L., Luxton-Reilly, A., Sanders, K., Schulte, C. & Whalley, J. L. (2006). Research perspectives on the objects-early debate. *Annual Joint Conference Integrating*, 146 – 165.
- Manso, A. , Oliveira, L. (2006). *Portugol*. Acedido em Fevereiro 25 de 2009 de <http://orion.ipt.pt/~manso/Portugol/>
- Manso A., Oliveira, L., & Marques, C. G. (2009). Ensino da Programação através da Linguagem Algorítmica e Fluxográfica. *CISTI'2009 - 4ª Conferência Ibérica de Sistemas e Tecnologias de Informação, WCO&NTiHE - Workshop on Challenges, Opportunities and New Trends in Higher Education*. (Aceite).
- Nielsen, J. (1993). *Usability Engineering*. London: Academic Press.
- Reigeluth, C. M. (1999). *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory*. New Jersey: Lawrence Erlbaum Associates.