

Ambiente de Aprendizagem de Algoritmos – Portugol IDE

António Manso, Instituto Politécnico de Tomar, manso@ipt.pt

Luís Oliveira, Instituto Politécnico de Tomar, loliveira@ipt.pt

Célio Gonçalo Marques, Instituto Politécnico de Tomar, celiomarques@ipt.pt

Abstract

Computer programming is a complex activity involving high level of abstraction, which is usually taught resorting to programming languages and development environments directed towards the context of industrial software production. This paper presents a tool called Portugol IDE designed to teach introductory programming courses. Portugol IDE is an environment for the development of algorithms where special emphasis is given to the development of algorithmic reasoning as it has tools that favour algorithm coding, execution and depuration. This environment has a portuguese coding language, which is based on structured portuguese, and a graphic language based on flowcharts. This paper describes the learning environment for Portugol IDE and the student's opinion of its application in the programming courses of the Higher School of Technology, Tomar (ESTT).

keywords

Portugol IDE, learn to program, algorithms animation and simulation, program visualization, flowchart

Resumo

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. Neste artigo é apresentado uma ferramenta, designada por Portugol IDE, destinada ao ensino das cadeiras introdutórias de programação. O Portugol IDE é um ambiente de desenvolvimento de algoritmos onde se dá especial importância ao desenvolvimento do raciocínio algorítmico, porque possui ferramentas que favorecem a codificação, execução e depuração de algoritmos. Este ambiente possui uma linguagem de codificação de algoritmos em português, baseada no português estruturado e uma linguagem gráfica baseada em fluxogramas. Neste artigo é apresentado o ambiente de aprendizagem Portugol IDE e opinião dos alunos acerca da sua utilização nos cursos de programação na Escola Superior de Tecnologia de Tomar (ESTT).

Palavras-chave

Portugol IDE, ensino da programação, animação e simulação de algoritmos, visualização de programas, fluxograma.

1. Introdução

Os cursos de programação de nível introdutório têm tradicionalmente elevadas taxas de reprovação (Butler e Morgan, 2007; Lahtinen *et al.*, 2005; Jenkins, 2002). 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 de programação a usar o que mais divergências suscita (Lister *et al.*, 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.

Algumas ferramentas de aprendizagem usam representações gráficas animadas da execução dos algoritmos, permitindo aos alunos vários níveis de interacção que vão desde a validação dos algoritmos construídos à correcção de erros, passando pela possibilidade de teste de novas formas de resolução do problema. A representação gráfica e a animação de algoritmos podem ser usadas para atrair a atenção dos alunos durante as apresentações, explicar conceitos de uma forma visual e encorajar um processo de aprendizagem baseado na prática (Mendes, 2001).

Com vista a dar resposta a estes problemas foi desenvolvido um ambiente de aprendizagem de algoritmos que designámos de Portugal IDE.

2. Ferramentas de apoio à aprendizagem da programação

São várias as ferramentas, propostas pela comunidade académica que se destinam ao auxílio da aprendizagem da programação. Segundo Mendes (2001) e Marcelino et al. (2008) as ferramentas podem ser classificadas de acordo com o âmbito de aplicação, da estratégia que utilizam e do tipo de actividade que suportam. Estes autores classificam as ferramentas em:

- i) Mini-Linguagens – tratam-se de linguagens extraídas de linguagens de programação convencionais. São constituídas por um sub-conjunto da linguagem convencional à qual foi retirada a complexidade desnecessária para o processo de aprendizagem, podendo ser adicionadas extensões consideradas úteis. O Minijava (Roberts, 2001) é um exemplo deste tipo de linguagem. Estas ferramentas têm como principal desvantagem o facto das linguagens de programação usarem a língua inglesa, o que por si só já constitui uma resistência no processo de aprendizagem para os alunos cuja primeira língua não é esta. A principal vantagem reside na facilidade da transição para a linguagem convencional completa ou para outras similares.
- ii) Ambientes de desenvolvimento controlados – Têm como principal objectivo proporcionar ao aluno um ambiente de desenvolvimento mais simplificado relativamente aos ambientes destinados ao uso profissional. Como base pode ser usada uma linguagem de programação convencional. O BlueJ (Kolling e Rosenberg, 2001) é um dos exemplos de ambientes deste tipo. O facto de ser usada uma linguagem de programação convencional, ainda que em conjunto com outras formas de representação, pode constituir uma desvantagem. Também neste caso a transição para os ambientes destinados ao uso profissional está facilitada.
- iii) Mundos programáveis – É usado um ambiente virtual no qual é executado o algoritmo. A execução do algoritmo pode ser realizada através de uma personagem que se desloca no ambiente virtual de acordo com o algoritmo definido. A ferramenta Karel the Robot (Pattis, 1995) é o exemplo mais conhecido deste tipo de ferramenta. O facto de serem usadas outras formas de representação tem como vantagem a execução do algoritmo num ambiente simples e familiar. A abstracção da representação e execução do algoritmo num ambiente diferente de

um sistema computacional pode levar a que o aluno não aprenda conceitos básicos como, por exemplo, o conceito de variável e de atribuição.

iv) Ferramentas de animação específicas – Servem para representar de forma gráfica e animada algoritmos pré-definidos, por exemplo, algoritmos de ordenação e de pesquisa. Normalmente não permitem que o aluno especifique os seus próprios algoritmos, pelo que o grau de interactividade é baixo. Um directório de animações deste tipo pode ser encontrado em <http://www.cs.hope.edu/~algaanim/ccaa>.

v) Ferramentas de animação de programas – Este tipo de ferramentas permite a animação da execução dos algoritmos definidos pelos alunos. As animações vão desde da representação no código fonte da instrução que está a ser executada e dos seus resultados, até à representação do estado das variáveis. O Jeliot (<http://www.cs.helsinki.fi/research/aaps/Jeliot>) e as páginas Web interactivas suportadas em applets ou em tecnologias similares são alguns exemplos.

vi) Ferramentas de animação de algoritmos – São usadas outras linguagens de representação, por exemplo, o fluxograma para representar o algoritmo. Têm como principal vantagem o facto de poderem ser usadas formas gráficas para a representação do algoritmo. De forma a facilitar o processo de transição a ferramenta deve permitir a tradução do algoritmo expresso numa linguagem gráfica para uma forma de representação em código fonte, independentemente, de ser usada uma linguagem convencional ou pseudo-linguagem. A ferramenta SICAS (Gomes e Mendes, 2000) é um exemplo de uma ferramenta deste tipo.

O Portugol IDE reúne características de vários tipos de ferramentas anteriormente descritas, nomeadamente: representação e construção do algoritmo em fluxograma e em linguagem estruturada, visualização do estado das variáveis durante a execução de um algoritmo, e execução do algoritmo passo-a-passo.

3. Portugol IDE

O projecto Portugol IDE (<http://orion.ipt.pt/~manso/portugol>) assume-se como um ambiente de aprendizagem construído de raiz para o ensino da programação que, para além de utilizar o português para o desenvolvimento de algoritmos, disponibiliza um conjunto de ferramentas pedagógicas inovadoras para a aprendizagem de técnicas de programação de computadores. O Portugol IDE é constituído por duas linguagens de definição de algoritmos e um ambiente de exploração das mesmas.

3.1 Linguagem Algorítmica

A definição de uma linguagem algorítmica estruturada e em português permite que o aluno codifique os seus programas com instruções expressas no léxico da sua língua materna, simplificando a sua aprendizagem de técnicas de programação.

Esta linguagem algorítmica, que serve de suporte à definição de dados e aos algoritmos que os manipulam é inspirada no português estruturado, também conhecido por pseudo-

código ou português, e permite representar de forma clara e inequívoca as instruções a serem executadas pelo computador.

A linguagem algorítmica foi definida de forma a cumprir dois objectivos: o primeiro, e sem dúvida o mais importante, o desenvolvimento do raciocínio algorítmico numa linguagem formal simples; e o segundo, a modernização das linguagens que lhe serviram de inspiração de forma a incorporar características das mais recentes linguagens de programação. Esta representação foca o interesse do aluno no desenvolvimento do algoritmo e proporciona uma melhor transição para os ambientes de programação tradicionais.

3.1.1 Tipos de dados

Os tipos de dados definidos como básicos são aqueles que consideramos como fundamentais para a representação da informação e sobre os quais é possível a aplicação de operadores matemáticos, lógicos e relacionais. Os tipos básicos da linguagem estão definidos na tabela 1.

Tabela 1: Tipos de dados básicos.

Tipo	Descrição
Inteiro	Números inteiros
Real	Números reais
Lógico	Valores lógicos
Carácter	Caracteres da Tabela ASCII
Texto	Conjunto de caracteres

A escolha do tipo de texto como um tipo básico prende-se com a aplicação do operador de concatenação, uma das características que se vem tornando um padrão nas linguagens mais recentes como, por exemplo, o java ou o c#.

A linguagem algorítmica permite a definição de estruturas homogéneas de dados básicos multidimensionais. A definição e a utilização destas estruturas segue as regras definidas nas linguagens de alto nível mais populares que utilizam os parêntesis rectos para a sua definição e acesso indexado.

Esta linguagem permite a definição de símbolos constantes e variáveis em qualquer parte do algoritmo. A definição dos dados no início do algoritmo é uma prática correcta e que deve ser incentivada, no entanto, esconde conceitos fundamentais como o escopo das variáveis e a ocultação de variáveis em escopos diferentes que estão presentes na generalidade das linguagens de alto nível. Devido ao carácter didáctico da linguagem optou-se pela introdução desta característica.

3.1.2 Instruções executáveis

O português estruturado é caracterizado por uma falta de padronização semântica, existindo várias formas de representar as mesmas instruções como, por exemplo, a

instrução “escrever” que pode ser encontrada sob a forma de “escreve” ou “imprima”. Este problema de padronização tem sido objecto de estudo (Manzano, 2006).

A escolha dos termos foi feita tendo em conta que se trata de uma linguagem imperativa e com objectivo de tornar os programas próximos da linguagem portuguesa corrente.

Como instruções executáveis foram definidas as instruções para ler (ler); escrever na consola (escrever); estruturas de decisão condicional (se-então-senão) e selectiva (escolhe-caso); e estruturas de iteração (repete-até, faz-equanto, para-próximo e enquanto-faz).

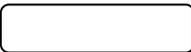
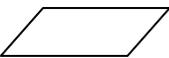
Este conjunto de instruções aliado aos tipos de dados definidos permite codificar um conjunto alargado de algoritmos. O conjunto de instruções anteriormente definido facilita a aprendizagem e a orientação do esforço para o desenvolvimento do raciocínio algorítmico que é a pedra basilar da programação.

3.2 Linguagem fluxográfica

A linguagem fluxográfica é inspirada nos fluxogramas e permite a definição e visualização dos algoritmos computacionais sob a forma gráfica, sendo composta pelos símbolos executáveis e o fluxo gerado pelos mesmos (tabela 2).

Tal como acontece com o português estruturado não existe um conjunto de símbolos fixo para a definição do fluxograma o que levou à tomada de decisões acerca da grafia dos símbolos, tendo em conta a sua execução computacional e a sua funcionalidade. A execução computacional levou ao desdobramento do símbolo de entrada/saída em dois símbolos distintos; e por questões funcionais existe apenas um símbolo de decisão que serve para codificar todas as estruturas de decisão e de iteração.

Tabela 2: Símbolos fluxográficos.

Símbolo	Descrição
	Início / Fim
	Ler
	Escrever
	Processo
	Decisão condicional
	Fluxo de execução
	Conector de Fluxo

Devido ao número limitado de símbolos e à necessidade de execução automática do fluxograma, existem símbolos que desempenham várias funções.

A linguagem fluxográfica é compatível com a linguagem algorítmica e permite que os algoritmos sejam escritos/desenhados e executados em qualquer uma delas. A tradução da linguagem algorítmica para a linguagem fluxográfica é unívoca, mas o mesmo já não acontece com a tradução inversa devido à multifuncionalidade dos símbolos gráficos.

3.3 Ambiente de exploração

O ambiente de exploração de algoritmos, Portugol IDE, permite o desenvolvimento, a execução e a depuração de algoritmos nas linguagens algorítmica e fluxográfica.

A depuração permite fazer a animação do algoritmo, com a apresentação das instruções em execução e do estado das variáveis em memória. Este modo possibilita ao aluno a visualização do fluxo de execução e o modo como as instruções afectam os dados do algoritmo.

Na figura 1 é mostrado o ambiente de desenvolvimento com a linguagem algorítmica e a execução do algoritmo passo-a-passo. Este modo de execução permite que os alunos sigam a execução das instruções e visualizem de que forma são afectadas as variáveis definidas em memória.

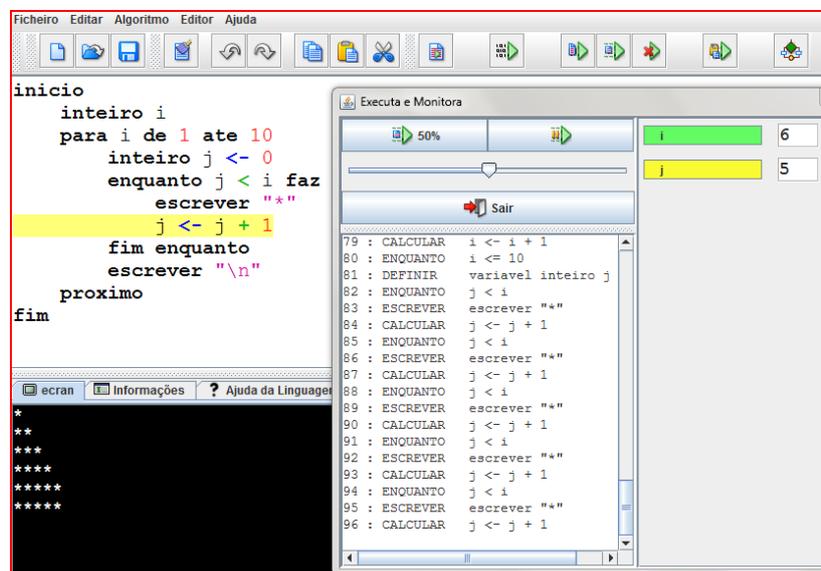


Figura 1: Ambiente de exploração da linguagem Algorítmica.

Na figura 2, está representado o ambiente de exploração que contém o mesmo algoritmo mas em linguagem fluxográfica.

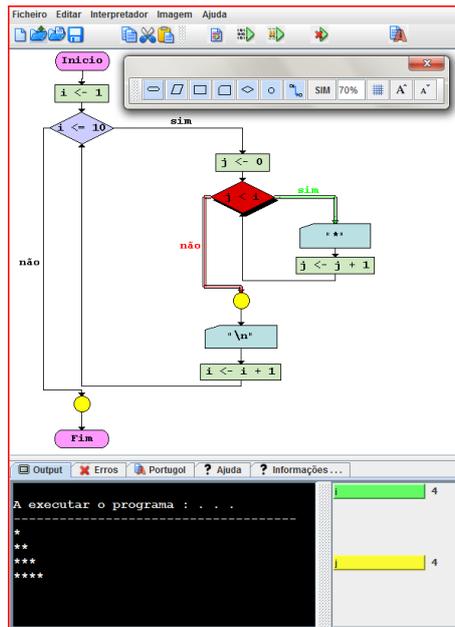


Figura 2: Ambiente de exploração da linguagem Fluxográfica.

O Portugol IDE assume-se como uma alternativa inovadora na aprendizagem de programação porque para além de utilizar uma linguagem estruturada baseada na língua portuguesa, são suportadas as seguintes representações gráficas animadas: i) Representação e construção do algoritmo em fluxograma; ii) Visualização do estado das variáveis durante a execução de um algoritmo e iii) Execução de um algoritmo passo-a-passo.

4. Opinião dos alunos sobre o Portugol IDE

Foi realizado um estudo com o objectivo de atentar nas reacções dos alunos à aprendizagem de programação através da ferramenta Portugol IDE. Começamos por caracterizar a amostra atendendo ao sexo, idade, número de linguagens de programação dominadas, experiência em programação e o número de horas dedicadas por semana à programação. Segue-se a descrição do estudo e a indicação dos instrumentos utilizados. Finalmente são apresentados os resultados do estudo.

4.1 Caracterização da amostra

A amostra deste estudo foi constituída por 32 alunos do 1.º ano do curso de Engenharia Informática da Escola Superior de Tecnologia de Tomar do Instituto Politécnico de Tomar.

Os sujeitos da amostra são predominantemente masculinos (75,0%) e no que respeita à idade, constatámos que a moda se situa na faixa etária dos 18 anos. A idade mínima situa-se nos 18 anos, a idade máxima nos 45 anos e a média é aproximadamente de 22 anos. Relativamente ao número de linguagens de programação dominadas, verifica-se que 50% dos sujeitos não domina nenhuma linguagem de programação, 21,9% domina

uma linguagem, 18,8% domina duas linguagens, 3,1% domina três linguagens e 6,3% domina cinco ou mais linguagens.

No que respeita à experiência em programação, verifica-se que mais de metade dos sujeitos (59,4%) começou a programar no 1.º da licenciatura em Engenharia Informática, 12,5% já programa há um ano, 3,1% há dois anos, 16,5% há três anos, 3,1% há quatro anos e 6,3% há cinco ou mais anos.

No que concerne ao número de horas dedicadas por semana à programação, constata-se que 62,6% dos sujeitos dedica entre 1 a 2 horas. Com 5 ou mais horas surgem 15,6%, com 4 horas surgem 3,1% e com 3 horas aparecem 9,4%. Existem 3 sujeitos que dedicam menos de 1 hora por semana à programação.

4.2 Descrição do Estudo

Neste estudo procedemos à avaliação da ferramenta Portugol IDE relativamente à sua facilidade de utilização, eficiência, facilidade de lembrar e satisfação (Nielsen, 2003). Os instrumentos utilizados foram dois questionários (questionário de opinião e questionário de ícones) criados e distribuídos através da ferramenta da Web 2.0, SurveyMonkey (<http://www.surveymonkey.com>).

Através do questionário de opinião pretendeu-se registar a posição dos sujeitos relativamente a diversas características da ferramenta. Optámos pela utilização de uma escala de diferencial semântico com valores de 1 a 7 (Nielsen, 1993; LaLomia e Sidowski, 1990). Esta escala utiliza dois termos antagónicos e os sujeitos são solicitados a escolher a posição, relativamente aos dois termos, que mais se adequa à ferramenta Portugol IDE. O número 1 indica o valor mais elevado positivo e o número 7 o valor mais elevado negativo.

Através do questionário de ícones pretendemos verificar se o sistema é fácil de lembrar. No futuro pretendemos também medir o tempo que determinados utilizadores casuais demoram a executar as mesmas tarefas após estarem algum tempo sem utilizarem o Portugol IDE. Os questionários foram validados por especialistas da área e a clareza e adequação das questões foi verificada num estudo piloto. Com base nos comentários recebidos foram realizadas algumas alterações.

4.3 Resultados do Estudo

Na tabela 6 verificamos que a maioria dos sujeitos considerou o aspecto da interface agradável, sendo a média 2,1. Existem quatro sujeitos que nem a consideraram agradável nem irritante e apenas um sujeito que a considerou próxima de irritante.

Tabela 6: Classificação do aspecto global da interface (agradável – irritante), n=32.

Diferencial semântico	Agradável ← → Irritante						
	1	2	3	4	5	6	7
Estadística descritiva							
F	14	9	4	4	0	1	0
%	43,8	28,1	12,5	12,5	0,0	3,1	0,0
Média	2,1						

Na tabela 7 verificamos que a maioria dos sujeitos considerou a estrutura dos menus adequada, sendo a média 2,0. Existem, no entanto, três sujeitos que nem a consideraram adequada nem inadequada.

Tabela 7: Classificação da estrutura dos menus (adequada – inadequada), n=32.

Diferencial semântico	Adequada ← → Inadequada						
	1	2	3	4	5	6	7
Estadística Descritiva							
F	12	12	5	3	0	0	0
%	37,5	37,5	15,6	9,4	0,0	0,0	0,0
Média	2,0						

Na tabela 8 constatamos que os sujeitos consideraram o tipo de letra de fácil leitura, sendo a média 1,8. Existem, no entanto, três sujeitos que nem a consideraram de fácil leitura nem de difícil leitura.

Tabela 8: Classificação do tipo de letra (fácil leitura – difícil leitura), n=32.

Diferencial semântico	Fácil Leitura ← → Difícil Leitura						
	1	2	3	4	5	6	7
Estadística Descritiva							
F	18	6	5	3	0	0	0
%	56,3	18,8	15,6	9,3	0,0	0,0	0,0
Média	1,8						

Na tabela 9 constatamos que os sujeitos consideraram os ícones sugestivos, sendo a média 2,0. Todavia, existem dois sujeitos que nem os consideraram sugestivos, nem nada sugestivos.

Tabela 9: Classificação dos ícones (sugestivos – nada sugestivos), n=32.

Diferencial semântico	Sugestivos ← → Nada Sugestivos						
	1	2	3	4	5	6	7
F	13	9	8	2	0	0	0
%	40,6	28,1	25,0	6,3	0,0	0,0	0,0
Média	2,0						

A maioria dos sujeitos considerou que o módulo de execução do fluxograma é útil, sendo a média 2,0 (tabela 10). Existe apenas um sujeito que nem o considerou útil nem inútil.

Tabela 10: Classificação do módulo de execução do fluxograma (útil – inútil), n=32.

Diferencial semântico	Útil ← → Inútil						
	1	2	3	4	5	6	7
f	15	14	2	1	0	0	0
%	46,9	43,8	6,3	3,0	0,0	0,0	0,0
Média	1,7						

Os sujeitos têm uma opinião idêntica à anterior no que se refere à escrita em forma de fluxograma. A média é igual e também só houve um sujeito que não a considerou útil nem inútil (tabela 11).

Tabela 11: Classificação da escrita em forma de fluxograma (útil – inútil), n=32.

Diferencial semântico	Útil ← → Inútil						
	1	2	3	4	5	6	7
f	15	12	4	1	0	0	0
%	46,9	37,5	12,5	3,1	0,0	0,0	0,0
Média	1,7						

A visualização do estado das variáveis é outro aspecto que os sujeitos consideram útil. A média é 1,6 e apenas um sujeito referiu que não é útil nem inútil (tabela 12).

Tabela 12: Classificação da visualização do estado das variáveis (útil – inútil), n=32.

Diferencial semântico	Útil ← → Inútil						
	1	2	3	4	5	6	7
f	18	10	3	1	0	0	0
%	56,3	31,3	9,4	3,0	0,0	0,0	0,0
Média	1,6						

Na tabela 13, também verificamos que os sujeitos consideram a ajuda útil na correcção dos erros. A média é 1,7.

Tabela 13: Classificação da utilidade da ajuda na correcção de erros (útil - inútil), n=32.

Diferencial semântico	Útil ← → Inútil						
	1	2	3	4	5	6	7
f	15	11	6	0	0	0	0
%	46,9	34,4	18,7	0,0	0,0	0,0	0,0
Média	1,7						

Os sujeitos consideram que os modos de execução ajudam muito na compreensão dos algoritmos, sendo a média de 1,7 (tabela 14).

Tabela 14: Classificação do auxílio dos modos de execução na compreensão dos algoritmos (muito - nenhum), n=32.

Diferencial semântico	Muito ← → Nenhum						
	1	2	3	4	5	6	7
f	15	13	3	1	0	0	0
%	46,9	40,6	9,4	3,1	0,0	0,0	0,0
Média	1,7						

Os sujeitos têm uma opinião idêntica no que se refere à ajuda para encontrar erros da execução em modo passo-a-passo, como se pode observar na tabela 15. A média é 1,6.

Tabela 15: Classificação do auxílio da execução em modo passo-a-passo para encontrar erros (muito - nenhum), n=32.

Diferencial semântico	Muito ← → Nenhum						
	1	2	3	4	5	6	7
f	17	12	3	0	0	0	0
%	53,1	37,5	9,4	0,0	0,0	0,0	0,0
Média	1,6						

Os sujeitos consideram que é fácil codificar algoritmos em Linguagem Estruturada no Portugol IDE, sendo a média de 1,8 (tabela 16). Dois sujeitos reponderam que nem era fácil nem difícil.

Tabela 16: Classificação da facilidade em codificar algoritmos em Linguagem Estruturada no Portugol IDE (fácil - difícil), n=32.

Diferencial semântico	Fácil ← → Difícil						
	1	2	3	4	5	6	7
f	14	13	3	2	0	0	0
%	43,8	40,6	9,4	6,2	0,0	0,0	0,0
Média	1,8						

Os sujeitos referem também que é fácil codificar algoritmos em fluxograma no Portugol IDE, como se pode observar na tabela 17. A média é de 1,6.

Tabela 17: Classificação da facilidade em codificar algoritmos em Fluxograma no Portugol IDE (fácil - difícil), n=32.

Diferencial semântico	Fácil ← → Difícil						
	1	2	3	4	5	6	7
f	17	12	3	0	0	0	0
%	53,1	37,5	9,4	0,0	0,0	0,0	0,0
Média	1,6						

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. Como se pode verificar na tabela 18, a percentagem de respostas correctas é superior a 90% em todos os ícones, alcançando os 100% no ícone “executa algoritmo”.

Tabela 18: Indicação da função correcta dos ícones, n=32.

Ícone	f	%
Executa o algoritmo	32	100%
Pára a execução do algoritmo	30	93,8%
Executa o programa passo a passo	29	90,6%
Acesso ao modo de fluxograma	30	93,8%

5. Conclusão

As primeiras etapas do ensino de programação devem estimular o desenvolvimento do raciocínio algorítmico num ambiente amigável para o aluno. Tradicionalmente estas etapas são executados com recurso a papel e a execução feita manualmente.

A ferramenta Portugol IDE, pretende ser um complemento, ou uma alternativa a esta metodologia apresentando um ambiente de desenvolvimento de algoritmos amigável, de fácil aprendizagem que possibilita a execução, animação e depuração do algoritmo de forma automática. A utilização da ferramenta em contexto de ensino/aprendizagem tem-se revelado importante na apreensão dos conceitos fundamentais de programação e no desenvolvimento do raciocínio abstracto. No estudo efectuado com alunos do 1º ano da Licenciatura em Engenharia Informática constatou-se que a ferramenta Portugol IDE é fácil e agradável de utilizar e fácil de lembrar. Na opinião dos alunos o módulo de escrita e execução em fluxograma, a visualização do estado das variáveis a ajuda na correcção de erros são úteis. Os alunos referem ser fácil codificar algoritmos e corrigir erros quando é usada esta ferramenta.

O ensino de conceitos mais avançados de programação requer a utilização de estruturas de dados complexas e a modularização dos programas. A extensão da linguagem algorítmica e fluxográfica estão em fase de projecto de forma a serem incorporadas no núcleo de execução do Portugol IDE.

A evolução do ambiente de desenvolvimento passa pela incorporação de um módulo de auxílio à correcção lógica do algoritmo e à introdução de tecnologia Web para acesso remoto a repositórios de problemas. Estas novas características visam a promoção do estudo autónomo da programação e o aumento das técnicas de programação suportadas.

6. Bibliografia

Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. In *ASCILATE 2007 Singapore*, 99-107.

- Giangrande, E. (2007). CS1 Programming Language Options, *Journal of Computing Sciences in Colleges*, 22 (3), 153-160.
- Gomes, A., & Mendes, A. (2000). Suporte à aprendizagem da programação com o ambiente SICAS. *Actas do V Congresso ibero-americano de informática educativa*, Viña del Mar, Chile.
- Jenkins, T. (2002). On the difficulty of learning to program. In *Proc. of the 3rd Annual Conference of the LTSN Centre for Information and Computer Science*, 27-29.
- Kolling, M., & Rosenberg, J. (2001). Guidelines for teaching object orientation with Java. In *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*, 70–74.
- Lahtinen, E., Mutka, K., & Jarvinen, H. (2005). A Study of the difficulties of novice programmers. In *Proc. of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer ITICSE'05*, 14-18.
- Lalomia, M., & Sidowski, J. (1991). Measurements of Computer Attitudes: A Review. *International Journal of Human-Computer Interaction*, 3 (2), 171-197.
- 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.
- Manzano, J. (2006). LPP – Linguagem de Projeto de Programação: proposta de padronização da estrutura sintática de uma linguagem de projeto de programação a ser definida para a área de desenvolvimento de software para países com idioma português. *Thesis*. Acedido em www.cantareira.br/thesis2/n6a3/manzano_thesis_6.pdf.
- Marcelino, M; Mihaylov, T., & Mendes, A. (2008). H-SICAS, a Handheld Algorithm Animation and Simulation Tool To Support Initial Programming Learning. In *38th ASEE/IEEE Frontiers in Education Conference*. Saratoga Springs, NY.
- Mendes, A. J. (2001). Software Educativo para apoio à aprendizagem de programação. In J. Sánchez (Ed.), *Taller Internacional de Software Educativo*. Universidad de Chile: Santiago.
- Nielsen, J. (1993). *Usability Engineering*. London: Academic Press.
- Pattis, R. (1981). *Karel the robot: A gentle introduction to the art of programming*. John Wiley & Sons.
- Ramani, K. V., & Rao, T. P. (1994). *A graphics based computer-aided learning package for integer programming: the branch and bound algorithm*, *Computers & Education*, 23.
- Roberts, E. (2001). An overview of MiniJava. In *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, 1–5.