

ENSINO DA PROGRAMAÇÃO ATRAVÉS DA LINGUAGEM ALGORÍTMICA E FLUXOGRÁFICA.

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

Resumo: O ensino da programação é considerado uma pedra basilar nos cursos de engenharia e em especial dos cursos de informática,. O insucesso demonstrado pelos alunos do curso de Engenharia Informática nas disciplinas de programação de nível introdutório na Escola Superior de Tecnologia de Tomar do Instituto Politécnico de Tomar levou ao desenvolvimento de uma ferramenta – Portugal IDE – com o objectivo de motivar os alunos e facilitar a aprendizagem da programação. Esta ferramenta permite a codificação dos algoritmos em linguagem algorítmica ou em linguagem fluxográfica e a sua execução automática. Neste trabalho são apresentadas as regras de tradução entre as duas linguagens e descrita a forma como a sua execução é feita.

Palavras chave: Programação, Portugol, Fluxogramas, Algoritmos, Ensino, Aprendizagem

INTRODUÇÃO

Conhecer uma linguagem de programação é diferente de saber programar. Conhecer uma linguagem de programação implica conhecer um conjunto de palavras reservadas e uma sintaxe. Saber programar está relacionado com a capacidade de resolver problemas cuja solução pode ser encontrada através da execução de um algoritmo computacional. A definição do algoritmo é o primeiro passo para a construção de um programa. Depois de definido, testado e otimizado, o algoritmo pode ser codificado numa linguagem de programação.

As disciplinas de programação de nível introdutório são caracterizadas por terem taxas elevadas de reprovação [1] [2] [3] e as disciplinas de programação de nível introdutório da Escola Superior de Tecnologia de Tomar (ESTT) do Instituto Politécnico de Tomar (IPT) não são excepção. A causa principal do insucesso está relacionada com a dificuldade sentida pelos alunos em analisar um problema e definir o algoritmo que o resolve devido ao elevado nível de abstracção que esta actividade envolve e não com aspectos próprios de uma ou outra linguagem de programação.

Têm sido criadas várias ferramentas com o objectivo de auxiliar os alunos a adquirirem competências na resolução de problemas através da definição de algoritmos. Segundo [4] e [5] há um conjunto de requisitos que uma boa ferramenta de desenvolvimento de algoritmos deve ter: i) ser interactiva, de forma a que o aluno tenha um papel activo durante o processo de

aprendizagem; ii) ser configurável, para que possa ser usada na resolução de vários problemas, recorrendo a várias soluções; iii) permitir representações alternativas, para estimular o aluno a observar o problema sob diferentes pontos de vista; iv) ser animada, para possibilitar ao aluno acompanhar a execução do algoritmo; v) ser simples óbvia e intuitiva; vi) ser portátil, para ser independente do hardware e do software onde vai ser executada; e por fim vii) ser económica, para alcançar o maior número de alunos. O Portugol IDE foi especificado para responder a todas as características anteriormente definidas e assim contribuir para a melhoria da aprendizagem de algoritmos nas disciplinas de programação de nível introdutório na ESTT. Este ambiente é constituído por um interface que manipula duas linguagens de programação (linguagem algorítmica e linguagem fluxográfica), permitindo ao aluno representar e executar os algoritmos nas duas linguagens. O Portugol IDE distingue-se das ferramentas analisadas: SICAS [5] [6], MiniJava [7], BlueJ [8], Karel the Robot [9], Jeliot [10], Ambap [11]; por possuir uma linguagem fluxográfica que se aproxima da execução computacional do algoritmo e uma linguagem de programação que se aproxima das linguagens de programação de alto nível. Esta característica permite fazer a ponte entre a execução computacional dos algoritmos e a expressão textual dos programas.

LINGUAGEM ALGORÍTMICA

A linguagem algorítmica é inspirada na linguagem de especificação de algoritmos conhecida como *português estruturado* ou *portugol*.

A linguagem algorítmica suporta um conjunto de operações que permitem que seja utilizada para aprender as técnicas básicas da construção de algoritmos computacionais de entre as quais salientamos a:

- Definição de tipos de dados;
- Entrada e saída de dados;
- Estruturas de decisão condicional;
- Estruturas de iteração;

A linguagem tem definido como básicos os tipos inteiros, real, lógico, carácter e texto. Sobre estes tipos de dados estão definidos um conjunto de operadores e funções que permitem a construção de expressões computacionais. O tipo texto foi definido como básico para poder ser manipulado com os operadores relacionais e o operador de concatenação (+), tal como acontece nas linguagens mais recentes (C# e Java). A linguagem permite ainda a definição e a manipulação de estruturas homogéneas multidimensionais de dados através do operador de indexação ([]). As estruturas para a manipulação de dados são as definidas no português e podem ser consultadas em [12].

LINGUAGEM FLUXOGRÁFICA

A linguagem fluxográfica foi especificada com dois objectivos muito concretos: fazer a prototipagem rápida de algoritmos nas primeiras etapas do ensino da programação e fazer a representação e execução visual da linguagem algorítmica.

A grafia da linguagem fluxográfica foi inspirada nos fluxogramas, depois de feitas as devidas adaptações para a tornar computacionalmente executável. Os símbolos definidos na linguagem e a sua sintaxe estão especificados na tabela 1.

Símbolo	Descrição
	Terminal – Marca o início/fim do algoritmo.
	Ler – Lê uma variável da consola.
	Escrever - Escreve a expressão na consola.
	Processo - calcula o valor de uma expressão.
	Decisão - Desvia a fluxo através do valor da condição
	Conector - Unificação de Fluxo

Tabela 1: Símbolos e sintaxe da linguagem fluxográfica

Através destes símbolos é possível descrever os blocos básicos dos algoritmos computacionais, e o seu reduzido número possibilita uma fácil compreensão das suas funcionalidades para o desenho de algoritmos.

Definição de tipos de dados

A linguagem fluxográfica não tem nenhum símbolo gráfico que permita fazer implicitamente a declaração de variáveis. Esta característica permite que se faça uma prototipagem rápida de um algoritmo sendo as variáveis implicitamente definidas pelos símbolos de “ler” e de “processo”. Quando é executada uma leitura ou um processamento o sistema verifica se já existe uma variável com o mesmo nome no programa; caso não exista, o sistema define a variável e deduz o tipo através do tipo de dados resultante da leitura ou do cálculo da expressão.

Entrada e saída de dados

A simbologia dos fluxogramas prevê apenas um símbolo para a entrada e saída de dados e o utilizador decide em cada caso qual o tipo de operação a executar. Para tornar a execução automática procedeu-se à introdução de dois símbolos distintos para cada uma das operações: “ler” e “escrever”.

Estruturas de decisão

A linguagem algorítmica tem duas estruturas de decisão: a decisão simples (se *então-senão*) [tabela 2], e a decisão em alternativa (*escolhe-caso*), [tabela 3]. A decisão simples tem uma transcrição imediata para a linguagem fluxográfica através de um símbolo de “decisão” que desvia o fluxo para a esquerda caso a condição seja falsa ou para a direita caso a condição seja verdadeira. O símbolo “conector” unifica o fluxo ramificado pela decisão.

Linguagem Algorítmica	Linguagem Fluxográfica
Se <condição> então Instruções1 Senão Instruções2 Fim se	

Tabela 2: Estrutura *se-então-senão*

A decisão em alternativa é transposta para a linguagem fluxográfica através de decisões

condicionais encadeadas na parte esquerda (Falso) dos símbolos de “decisão”.

Linguagem Algorítmica	Linguagem Fluxográfica
Escolhe <var> caso valor1: Instruções1 caso valor2: Instruções2 Defeito: Instruções3 Fim escolhe	

Tabela 3: Estrutura *escolhe-caso*

Estruturas de iteração

As estruturas de iteração são um conjunto de instruções que formam ciclos controlados por símbolos de “decisão”. A linguagem algorítmica tem dois tipos de estruturas de iteração: as estruturas de iteração com controlo inicial (*enquanto-faz* e *para-próximo*) [tabela 4] [tabela 5], e as estruturas com controlo final (*faz-enquanto* e *repete-até*), [tabela 6].

A estrutura *enquanto-faz* é traduzida para fluxograma utilizando um símbolo de “decisão” para o controlo da iteração, em que a parte direita aponta para o bloco a iterar e recebe o fluxo que sai do último símbolo.

Linguagem Algorítmica	Linguagem Fluxográfica
Enquanto <condição> faz Instruções Fim enquanto	

Tabela 4: Estrutura *enquanto-faz*

A codificação fluxográfica da estrutura *para-próximo* é feita através da sua transcrição para a estrutura *enquanto-faz*. O bloco começa com um símbolo de “processo” que inicializa a variável de iteração seguido de um símbolo de *decisão* condicional com a expressão lógica de término da iteração. O fluxo originado pela “condição” com o valor verdadeiro aponta para a bloco a iterar que por sua vez aponta para um bloco de “processo” que actualiza a variável de iteração e regressa ao símbolo de “decisão”.

Na tabela 5 é apresentada a tradução de uma estrutura *para-próximo* com incrementos de uma

unidade à variável de iteração. A tradução completa desta estrutura para linguagem fluxográfica é composta de várias excepções devido ao facto da estrutura iterar com incrementos positivos e negativos não unitários, e a sua explicação pormenorizada escapa ao âmbito desta comunicação.

Linguagem Algorítmica	Linguagem Fluxográfica
Para var de v1 ate v2 Instruções próximo	

Tabela 5: Estrutura *para-próximo* com incrementos positivos.

Nas estruturas de decisão com controlo final o bloco começa com um símbolo de conexão de fluxo seguido do bloco de iteração e do símbolo de “decisão”. Se o fluxo regressa ao símbolo de conexão através do fluxo gerado pelo valor verdadeiro da condição estamos na presença de uma estrutura *faz-enquanto* caso contrário estamos na presença de uma estrutura do tipo *repete-até*.

Linguagem Algorítmica	Linguagem Fluxográfica
Repete Instruções Até <condição>	
Faz Instruções Enquanto <condição>	

Tabela 6: Estrutura *repete-até* e *faz-enquanto*.

CONVERSÃO ENTRE AS LINGUAGENS ALGORÍTMICA E FLUXOGRÁFICA

Através das regras enunciadas no ponto anterior, a linguagem algorítmica pode ser traduzida pelo Portugol IDE em linguagem fluxográfica de uma forma simples e directa.

A conversão da linguagem fluxográfica para a linguagem algorítmica é um processo mais complexo devido a não haver uma correspondência unívoca entre símbolos e instruções. Os símbolos adquirem um significado na linguagem algorítmica através do contexto do fluxograma. Por exemplo, o símbolo de “decisão” pode corresponder a uma estrutura *enquanto-faz* ou *se-então-senão*.

Por outro lado a estrutura em grafo do fluxograma dificulta o processamento das estruturas computacionais uma vez que é necessário identificar onde as estruturas começam e terminam. Para a resolução deste problema utilizaram-se agentes inteligentes que navegam no fluxograma e que identificam as estruturas e o significado de cada símbolo no fluxograma. Após a identificação do significado de cada símbolo a tradução para a linguagem algorítmica baseia-se nas regras definidas no ponto anterior.

A linguagem algorítmica obriga á definição explícita dos dados antes da sua utilização, e esta norma não se aplica á linguagem fluxográfica pois os dados são deduzidos implicitamente. Quando o utilizador executa o fluxograma os tipos de dados são deduzidos e a conversão para a linguagem algorítmica gera um algoritmo executável. Caso contrário, o sistema identifica as variáveis mas não consegue deduzir o seu tipo e a definição desta variáveis fica com um tipo de dados indefinido, o que obriga o aluno a definir o tipo de dados na linguagem algorítmica.

EXECUÇÃO DO ALGORITMO

O Portugol IDE permite que a execução do algoritmo seja independente da linguagem onde o algoritmo foi escrito/desenhado.

A execução do algoritmo na linguagem algorítmica permite ao aluno visualizar a sequência de instruções executadas pelo algoritmo, [figura 1] enquanto que a execução em forma gráfica permite que o aluno siga o fluxo de execução do algoritmo e veja graficamente as estruturas de decisão e execução[figura 2].

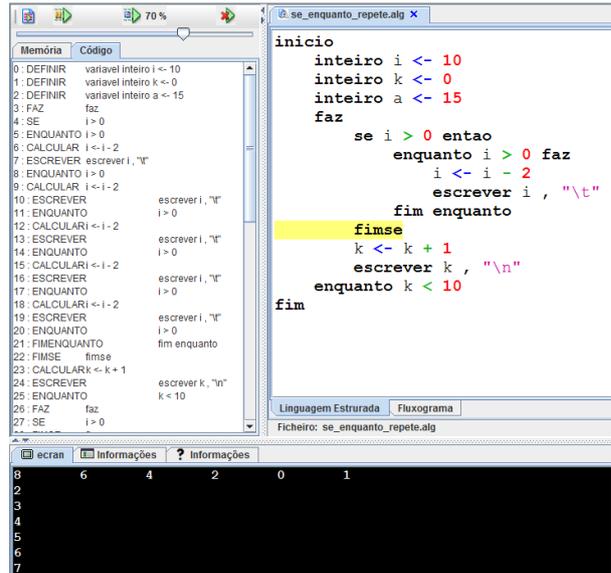


Fig. 1. Execução do algoritmo com a visualização da linguagem algorítmica, as instruções executadas e a consola.

A execução do algoritmo em modo de depuração permite a visualização do estado da memória do algoritmo e a forma como as instruções/símbolos afectam o seu estado.

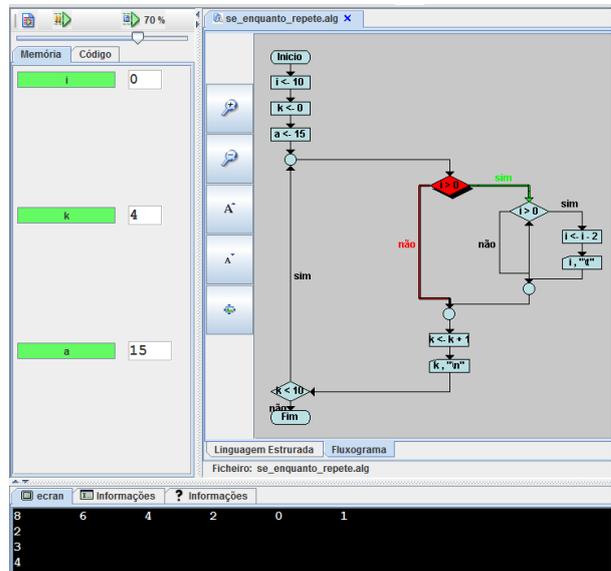


Fig. 2. Execução do algoritmo com a visualização da linguagem fluxográfica, o estado da memória e a consola.

UTILIZAÇÃO DO PORTUGOL NO ENSINO SUPERIOR

O Portugol é *freeware* e *open source* estando a versão 2.3 disponível para *download* em [12] [figura 3].



Fig. 3. Web site do Portugal

Desde que a ferramenta foi disponibilizada foram chegando várias reacções da comunidade que se podem enquadrar em dois campos: desenvolvimento e utilização.

No campo do desenvolvimento temos assistido à criação de projectos paralelos (é usado o núcleo do Portugal, mas são adicionadas outras funcionalidades), temos recebido pedidos de ajuda para a criação de projectos similares e temos ainda tido diversas ofertas de colaboração no projecto.

No campo da utilização temos recebido críticas muito positivas, pedidos de ajuda para codificação de algoritmos em Portugal, bem como pistas para a evolução do Portugal.

Actualmente o Portugal está a ser utilizado nas unidades curriculares de Introdução à programação nos cursos de engenharia da ESTT, nos Cursos de Especialização Tecnológica de Desenvolvimento de Produtos Multimédia e Tecnologias de Programação e Sistemas de Informação do Instituto Politécnico de Tomar e em várias instituições de Ensino Superior no Brasil, entre elas, o UDF - Centro Universitário.

Num estudo realizado em 2009 com 32 alunos do 1.º ano do curso de Engenharia Informática da ESGT constatou-se que a ferramenta Portugal é fácil e agradável de utilizar, é eficiente 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.

CONCLUSÃO

Programar é uma tarefa difícil que é aprendida gradualmente através da introdução de novos conceitos e técnicas. Nesta aprendizagem consideramos que o passo mais difícil é o

desenvolvimento do raciocínio algorítmico, o primeiro passo da programação e a base de todos os conceitos. A linguagem fluxográfica é uma linguagem simples de aprender devido ao seu pequeno número de símbolos e permite a criação de algoritmos de uma forma simples e imediata o que a torna adequada à iniciação do estudo da programação. Com base nestes pressupostos foi desenvolvido o Portugal IDE, com o objectivo de melhorar a aprendizagem da programação. Esta ferramenta utiliza duas linguagens de programação: a algorítmica e a fluxográfica. Através de agentes inteligentes, permite a execução dos algoritmos em linguagem fluxográfica e a sua tradução para uma linguagem próxima das linguagens de programação tradicionais com a grande vantagem de utilizar a língua materna dos alunos.

O suporte da ferramenta em ambiente Web é um dos principais aspectos que se encontram em desenvolvimento. Através da Web pretende-se disponibilizar dois serviços: i) um repositório de problemas e ii) um avaliador automático que permite validar, quase de imediato, os algoritmos submetidos pelos alunos. Estes dois serviços têm como principal objectivo fomentar o trabalho autónomo dos alunos.

Bibliografia

- [1] Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. In *ASCILATE 2007 Singapore*. p. 99-107.
- [2] 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*. p. 14-18.
- [3] 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*. p. 27-29.
- [4] 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.
- [5] 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.
- [6] 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.

- [7] Roberts, E. (2001). An overview of MiniJava. In *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, p. 1–5.
- [8] 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*. p. 70–74.
- [9] Pattis, R. (1981). *Karel the robot: A gentle introduction to the art of programming*. John Wiley & Sons.
- [10] Haajanen, J., Pesonius, M., Sutinen, E., Tarhio, J., Teräsvirta, T., & Vanninen, P. (1997). Animation of user algorithms on the Web. In: Proc. VL '97, IEEE Symposium on Visual Languages, IEEE 1997, p. 360-367.
- [11] s/a (s/d). Ambiente de Aprendizado de Programação, <http://www.ufal.br/tci/ambap/> (Março de 2009)
- [12] Manso, A. & Oliveira, L. . Portugol IDE (2009). Portugol IDE, <http://orion.ipt.pt/~manso/Portugol/> (Março 2009)