

Portugol Studio: Uma IDE para Iniciantes em Programação

Luiz F. Noschang¹, Fillipi Pelz¹, Elieser A. de Jesus¹, André L. A. Raabe^{1,2}

¹Mestrado em Computação Aplicada
²Programa de Pós Graduação em Educação
Universidade do Vale do Itajaí - UNIVALI
{noschang, fillipi, elieser, raabe}@univali.br

Abstract. *In this paper we argue that programming novices should use tools specifically designed for learners. Professional integrated development environments (IDEs) could lead the student to some troubles such as: too many advanced options; foreign language in the user interface; difficulties in the use and configuration of the debugger or lack thereof, among others. To minimize such problems we built the Portugol Studio IDE which aims to assist beginners in their first programming experiences. The Portugol Studio, detailed in this paper, is open source software that currently accounts for approximately 2.000 downloads and is being used by various groups of students. It can be obtained at <http://sourceforge.net/projects/portugolstudio/>.*

Resumo. *Neste artigo argumentamos que alunos iniciantes em programação devem utilizar ferramentas de programação especificamente projetadas para aprendizes. Os ambientes integrados de desenvolvimento (IDEs) profissionais podem gerar alguns problemas para alunos iniciantes, tais como: o excesso de configurações avançadas; uso do idioma inglês na interface gráfica quando muitos alunos iniciantes têm dificuldades com este idioma; dificuldades no uso e configuração do depurador ou inexistência deste, entre outros. Visando minimizar estes problemas foi construída a IDE Portugol Studio voltada para auxiliar os alunos iniciantes em suas primeiras experiências com lógica de programação. O Portugol Studio, cujos detalhes são apresentados neste artigo, é um software livre que atualmente contabiliza aproximadamente 2.000 downloads, e está sendo utilizado por várias turmas de estudantes. Pode ser obtido em <http://sourceforge.net/projects/portugolstudio/>.*

1. Introdução

A aprendizagem de programação introdutória é um dos pilares da formação em computação e áreas afins (CRISTÓVÃO, 2008) e o conhecimento adquirido pelos alunos nas disciplinas introdutórias de programação tende a influenciar o desempenho discente ao longo dos cursos (RAABE *et al.*, 2007).

Autores como Sheard *et al.* (2009) apontam a grande quantidade de literatura relacionada às dificuldades da programação introdutória como um indicador de que este assunto é difícil de aprender e de ensinar. As dificuldades tornam-se mais evidentes quando se observa taxas de reprovação entre 30% e 40% nestas disciplinas (BEAUBOEF e MASON; 2005, RAABE *et al.*; 2007).

Vários estudos vêm investigando as dificuldades na aprendizagem de programação introdutória sob diferentes aspectos (LAHTINEN *et al.*; 2005, DALE, 2006; HUANG *et al.*; 2013; NGUYEN *et al.* 2014). Neste artigo abordamos o problema da aprendizagem de programação com enfoque no uso de uma IDE projetada com finalidade didática.

Entende-se que o ensino de programação introdutória deve focar na promoção da habilidade de resolução de problemas e no desenvolvimento da lógica de programação dos estudantes, sendo a aprendizagem de uma determinada linguagem um aspecto secundário. Neste sentido, o uso de IDEs (*Integrated Development Environment*) profissionais que visam a produtividade e não especificamente a aprendizagem trazem dificuldades adicionais nos primeiros contatos que os estudantes têm com a tarefa de produzir algoritmos e traduzí-los em programas. Estas dificuldades, detalhadas adiante no artigo, tem como principais elementos o fato de as IDEs profissionais possuírem uma lógica voltada à construção de projetos de maior porte do que os abordados na programação introdutória e a baixa fluência no idioma inglês de boa parte dos estudantes, criando uma barreira para compreensão das mensagens de auxílio destas IDEs.

Sendo assim, neste artigo argumentamos que a utilização de IDEs projetadas com finalidade didática podem reduzir estes problemas e permitir que o tempo e a energia do estudante sejam investidos na aprendizagem da lógica propriamente dita. Esta hipótese serviu como diretriz para a concepção de uma IDE voltada especificamente para estudantes iniciantes em programação. A IDE didática chamada Portugol Studio prioriza o uso do idioma português, pois é voltada ao público com baixa fluência no idioma inglês, e em sua construção incluiu diversos elementos presentes nas IDEs profissionais, mas preservando sempre o enfoque na aprendizagem dos novatos em programação. Neste sentido, alguns elementos da interface auxiliam o docente no momento de ilustrar programas e conceitos como escopo e processos como a depuração.

O artigo detalha na seção 2 as principais dificuldades impostas por IDEs profissionais aos estudantes, na seção 3 fundamenta a escolha pela linguagem portugol e apresenta princípios adotados no projeto do compilador desenvolvido. A seção 4 detalha as funcionalidades da IDE assim como as crenças e diretrizes pedagógicas do projeto; a seção 5 considerações finais, resultados obtidos e trabalhos futuros.

2. Porque Não Usar IDEs Profissionais com alunos iniciantes?

Neste artigo defendemos que as IDEs profissionais disponíveis (Eclipse, Netbeans, VisualStudio, Code Blocks, etc.) são inadequadas para alunos iniciantes em programação. Várias destas IDEs apresentam todos os textos da interface gráfica (menus, ajuda, mensagens para o usuário, etc) no idioma inglês o que cria um obstáculo, em especial nos primeiros contatos com a ferramenta, aos estudantes que não possuem domínio deste idioma.

Ainda que no Brasil não existam índices oficiais caracterizando a baixa fluência no idioma inglês dos estudantes que concluem o ensino médio alguns indícios desta situação podem ser encontrados. O índice EPI (*English Proficiency Index*) em 2013 coloca o Brasil em 46º lugar dentre 54 nações pesquisadas (EPI, 2013). O MEC lançou o programa Inglês sem Fronteiras para aprimorar a proficiência em língua inglesa dos estudantes universitários brasileiros, com o objetivo de proporcionar-lhes oportunidades de acesso a universidades de países anglófonos por meio do Programa Ciência sem Fronteiras (MEC, 2014). O Exame Nacional do Ensino Médio (ENEM) a partir de 2012 passou a incluir questões referentes ao idioma inglês que demonstram uma preocupação em avaliar e fomentar o desenvolvimento deste idioma entre os estudantes do Ensino Médio.

Além do problema relacionado com idioma inglês mencionado anteriormente, IDEs profissionais possuem muitas opções avançadas, com inúmeras janelas e possibilidades de configurações, o que pode ser intimidador para um aluno iniciante.

Outro problema é a dificuldade dos alunos iniciantes em compreender as mensagens de erros no código que são apresentadas pelas IDEs profissionais. Estas mensagens são comumente mostradas em inglês, o que nos leva ao problema da barreira do idioma já mencionado anteriormente. Além disso, muitos compiladores/interpretadores geram mensagens de erro que são muito gerais e podem confundir o aluno iniciante durante a correção dos erros presentes no seu código. Em geral, os alunos levam algum tempo para familiarizarem-se com as mensagens de erro apresentadas em IDEs profissionais a ponto de poderem utilizá-las como dicas para a correção dos seus erros.

Por fim, outro problema que se observa em IDEs profissionais (quando utilizadas como ferramentas de aprendizagem) é a complexidade de uso e configuração do depurador (*debugger*). Defende-se aqui que o depurador é uma peça fundamental no kit de ferramentas do aprendiz, pois permite ao aluno inspecionar valores de variáveis durante a execução de um programa e verificar se o valor esperado (modelo mental) corresponde ao valor que realmente está sendo utilizado pela máquina durante a execução das instruções. Defende-se também que o depurador pode servir como mecanismo de auto-regulação da aprendizagem por propiciar ao aluno o exercício da metacognição. O aluno que através dos *feedbacks* fornecidos por um depurador consegue obter êxito na resolução de um determinado problema está auto regulando sua aprendizagem (MAXIMO; RAABE e BARONE, 2007).

Algumas IDEs profissionais não possuem depuradores e exigem uma série de configurações adicionais para que se possa usar um depurador externo. Nas IDEs que possuem depuradores nem sempre é fácil (para um aluno iniciante) usar este recurso. Os depuradores das IDEs profissionais costumam, assim como as próprias IDEs, ter uma grande quantidade de opções avançadas que são essenciais para um programador profissional mas que podem ser intimidadoras para um iniciante.

Além dos problemas relatados acima, as IDEs profissionais costumam suportar apenas linguagens de programação que também são profissionais, tais como Java, C++, PHP, Python, etc. Isto pode gerar mais uma dificuldade para o aluno iniciante, pois não são raros os casos em que a aprendizagem dos princípios mais fundamentais de programação é comprometida por dificuldades com a sintaxe desta ou daquela linguagem. Para evitar a complexidade sintática das linguagens profissionais alguns professores historicamente vêm adotando uma linguagem de programação simplificada chamada Portugol, cujos detalhes são discutidos a seguir.

3. Portugol como Linguagem de Programação Introdutória

O Portugol é uma notação utilizada para escrever programas em uma mescla de português e símbolos comuns em linguagens de programação de alto nível (operadores relacionais, operadores aritméticos, entre outros). Entre as vantagens da utilização do Portugol com alunos iniciantes estão a simplicidade da linguagem e o fato das palavras-chave e comandos estarem em português, evitando os problemas com o idioma inglês.

A pesquisa conduzida por Zanini e Raabe (2012) apontou que a maioria dos livros utilizados como referência básica nas disciplinas de introdução a programação em universidades brasileira possuem explicações usando alguma notação de portugol. Além

dos livros, observa-se que atualmente existem vários softwares que permitem ao aluno iniciante em programação escrever código em Portugol. Pode se mencionar as ferramentas G-Portugol¹, Visualg², WebPortugol³, Portugol Viana⁴ e Portugol IDE⁵ como exemplos de ferramentas construídas para apoiar a aprendizagem de programação introdutória através da linguagem Portugol. A existência dessas ferramentas deixa evidente que existe um interesse significativo por parte de professores e estudantes na utilização do Portugol como linguagem de apoio a aprendizagem de programação.

A IDE Portugol Studio, descrita na próxima seção do artigo, utiliza um compilador portugol chamado Portugol Núcleo que foi desenvolvido em projeto anterior a IDE. A sintaxe utilizada no Portugol Núcleo é mais próxima das linguagens como C, Java e PHP do que outras notações encontradas nos livros. Por exemplo, a declaração de uma função inicia com a palavra *função*, o equivalente de *function* em PHP. É necessário delimitar o código da função dentro dos símbolos de abre e fecha chaves, assim como na maioria das linguagens modernas.

Estas decisões de projeto da sintaxe da linguagem foram realizadas para reduzir o impacto na transição do Portugol para linguagens usadas após a fase inicial de aprendizagem, mas mantendo o cuidado de não inserir detalhes desnecessários e mantendo as palavras reservadas no idioma português.

A linguagem possui cinco tipos de dados: inteiro, real, caracter, lógico e cadeia (*string*). Possui suporte a constantes, vetores e matrizes destes cinco tipos. Possibilita a inicialização de vetores e matrizes com valores. Possui instruções para entrada e saída de dados via console, operações relacionais, aritméticas e lógicas, desvios condicionais, três tipos de loops, suporte a funções com passagem de parâmetros por valor e por referência, e suporte a recursividade. Possui suporte a bibliotecas que possibilitam ampliar o potencial da linguagem.

A linguagem não suporta orientação a objetos. Defendemos que o desenvolvimento da lógica de programação é um passo anterior a escolha do paradigma e que em pequenos programas (foco de nossa linguagem) não aparece vantagem na utilização do paradigma orientado a objetos.

A seguir discute-se a IDE PortugolStudio, onde os alunos iniciantes em programação podem codificar e depurar seus primeiros programas utilizando a linguagem Portugol.

4. O Ambiente Portugol Studio

O Portugol Studio foi contruído com foco na aprendizagem ao invés de focar na produtividade como fazem as IDEs profissionais. Sendo assim, foi projetado para ser uma IDE fácil de utilizar e para auxiliar a aprendizagem dos alunos iniciantes. Estas premissas nortearam várias das decisões de projeto ao longo do processo de desenvolvimento e aprimoramento do Portugol Studio.

A interface do ambiente de programação e teste de programs divide-se em três regiões principais: (i) Código Fonte, onde o estudante cria seu programa; (ii) Árvore de símbolos, onde pode visualizar as variáveis do programa e (iii) Console de Entrada e Saída/ Mensagens, onde visualiza mensagens de erro e onde ocorre a interação via console. A figura 1 ilustra estas regiões:

¹ <http://sourceforge.net/projects/gpt.berlios/>

² <http://www.apoioinformatica.inf.br/produtos/visualg>

³ <http://www.univali.br/webportugol>

⁴ <http://sourceforge.net/projects/portugolviana/>

⁵ <http://www.dei.estt.ipt.pt/portugol/>

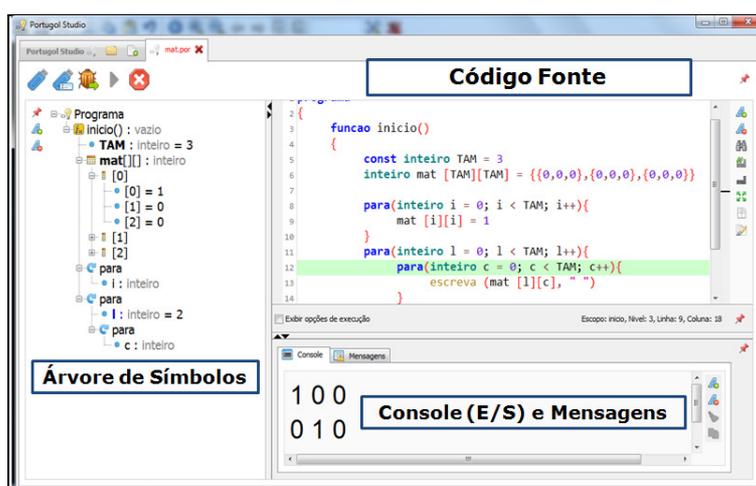


Figura 1. Visão geral da Interface da IDE Portugol Studio.

A seguir são descritas características do Portugol Studio que consideramos úteis como ferramenta de ensino e aprendizagem.

4.1. Mensagens de Erro

Como mencionado anteriormente, em geral os compiladores/interpretadores de linguagens profissionais apresentam mensagens de erro pouco informativas para alunos iniciantes, já que estas mensagens costumam ser carregadas de termos técnicos que o iniciante ainda desconhece ou está pouco familiarizado. Por este motivo, no Portugol Studio as mensagens de erro foram cuidadosamente elaboradas no sentido de apresentar dicas úteis para um aluno iniciante. Isto é possível em especial pela estratégia adotada no analisador semântico que percorre a árvore sintática abstrata, que representa o programa, de diferentes formas buscando identificar erros de maneira mais precisa. Na figura 2 é possível ver algumas mensagens de erro que exemplificam o tipo de linguagem que foi utilizada para descrever os erros encontrados nos códigos dos alunos.

Linha	Mensagem
5	Tipos incompatíveis! O parâmetro "expoente" da função "potencia" espera uma expressão do tipo "inteiro", mas foi passada uma expressão do tipo "cadeia".
7	A variável "resultado" está sendo utilizada como uma função
9	A inicialização do vetor "vet" deve possuir 5 elementos. Insira mais 2 elementos para corrigir o problema
12	A inicialização da matriz "mat" deve possuir 3 linhas. Insira mais 1 linha para corrigir o problema
12	A linha [1] na inicialização da matriz "mat" deve possuir 3 elementos. Remova 1 elemento para corrigir o problema

Figura 2. Mensagens de erro informativas do Portugol Studio.

4.2. Programas e não Projetos

No portugol Studio o estudante cria um programa e não um projeto. Muitas IDEs profissionais utilizam a abordagem de organizar os códigos fontes por projetos, onde um projeto é um conjunto de arquivos relacionados. Se por um lado isto é muito útil para projetos de médio ou grande porte, por outro, gera dificuldades para o aluno iniciante que, por exemplo, abre a IDE apenas para realizar um exercício onde deverá escrever um algoritmo de 15 a 30 linhas em apenas um arquivo.

No Portugol Studio evitou-se a criação de um projeto para cada exercício/atividade, e para programar o aluno precisa apenas criar um novo arquivo e iniciar a codificação, sem a necessidade das típicas configurações de projeto requeridas pelas IDEs profissionais (caminhos, nome de projeto, utilização de bibliotecas, etc).

4.3. Salientador de Sintaxe

Este é um recurso também frequentemente encontrado em IDEs profissionais e que por sua vez traz benefícios aos iniciantes e por isso foi adotado. Conforme o aluno vai digitando seu código ele vai sendo colorido e destacado. As palavras destacadas são os elementos fundamentais da sintaxe da linguagem, as palavras reservadas. Este recurso de salientação de sintaxe auxilia os alunos reduzirem erros de digitação e erros sintáticos. Por exemplo, a palavra *inteiro* deve ser utilizada para declarar uma variável numérica do tipo inteiro. Caso o aluno digite incorretamente esta palavra ela não será destacada pela IDE, dando rapidamente ao aluno uma indicação visual de que algo está incorreto.

Adotou-se também salientadores de erros semânticos (sublinhados) tais como: variável não declarada, atribuições incompatíveis, inicialização de vetores e matrizes com número errado de elementos (ilustrado na figura 3) e outros. Estes erros são apontados ao estudante ainda em tempo de digitação como ocorre em algumas IDEs profissionais.

```

1 programa
2 {
3     funcao inicio()
4     {
5         const inteiro TAM = 3
6         inteiro mat [TAM][TAM] = {{0,0,0},{0,0,0},{0,0,0},{1,1,1}}
7     }

```

The image shows a code editor window with a yellow highlight on line 6. The code on line 6 is `inteiro mat [TAM][TAM] = {{0,0,0},{0,0,0},{0,0,0},{1,1,1}}`. The IDE has underlined the number '4' in the second curly brace of the array initialization, indicating a semantic error because the array is declared with dimensions [TAM][TAM] (3x3) but is being initialized with 4 rows.

Figura 3. Indicação de erro semântico na inicialização de matrizes.

4.4. Apoio a Ilustração pelo Docente

Alguns recursos de interface do Portugol Studio foram concebidos para auxiliar o professor nas tarefas expositivas em sala de aula. Por exemplo, ao explicar um código fonte de um programa o tamanho dos textos é facilmente ajustável, seja na janela de código fonte, quanto na janela do console de entrada e saída de dados quanto na janela da árvore de símbolos. Isto possibilita que detalhes de construção e execução dos programas possam ser ilustrados aos alunos mesmo em projetores de baixa resolução. Além disso, o professor pode exibir somente a área do código fonte, escondendo os demais componentes da tela o que permite direcionar a atenção dos alunos ao código.

4.5. Depurador

Meslle (2010) menciona a utilização dos testes de mesa no ensino de programação para iniciantes desde 1960. O teste de mesa consiste em uma técnica de verificação de algoritmos onde os valores das variáveis são anotados (em papel) conforme a simulação mental da execução do algoritmo progride. O objetivo do teste de mesa é verificar se os valores anotados coincidem com os resultados esperados, ou seja, se os valores gerados pelo algoritmo são realmente os valores que ele deveria gerar. Meslle (2010) afirma que por razões que não são muito claras os testes de mesa não constam nos livros modernos ou mesmo em material de aula de professores, e não há uma razão aparente para o abandono desta ferramenta de ensino.

O principal problema com o teste de mesa é que ao anotar os valores das variáveis no papel o aluno pode cometer erros. Se o aluno que realiza o teste de mesa não compreende completamente o funcionamento de, por exemplo, uma estrutura de repetição (loop) então erros podem ser cometidos facilmente quando os valores da variável de controle (contador) forem anotados. Neste caso, os valores anotados não serão uma fonte confiável para a verificação do algoritmo. Por outro lado, entende-se que é possível utilizar o depurador como um teste de mesa onde os valores das variáveis são preenchidos pelo computador, e não pelo aluno, evitando as falhas mencionadas anteriormente.

Ao permitir a visualização de valores de variáveis o depurador está na verdade permitindo que se visualize o conteúdo da memória RAM, já que uma variável é a abstração de uma determinada área da memória principal do computador. Ou seja, o aluno que utiliza um depurador para visualizar o valor de uma variável está inspecionando o valor que realmente está sendo utilizado pelo programa, independentemente do nível de compreensão do aluno sobre o funcionamento do código. Dessa maneira, entende-se que o depurador provê os benefícios do teste de mesa para a aprendizagem e evita erros que poderiam ser cometidos por alunos iniciantes ao anotar equivocadamente os valores gerados durante a execução.

No Portugol Studio uma grande ênfase foi dada ao depurador. Optamos por criar os botões de execução e depuração com o mesmo tamanho na interface chamando igualmente a atenção do aluno e enfatizando a importância deste.

Durante a construção do programa o aluno visualiza na árvore de símbolos as variáveis do programa e também escopo destas representado pela hierarquia da árvore. Quando o aluno depura seu programa ele pode visualizar também os valores das variáveis que compõem o programa à medida que eles vão sendo alterados pelas instruções contidas no programa.

A figura 4 ilustra um código sendo depurado. O código preenche uma matriz de 3X3 com zeros para em seguida substituir os elementos da diagonal principal da matriz por um e exibi-a no console de saída de dados.

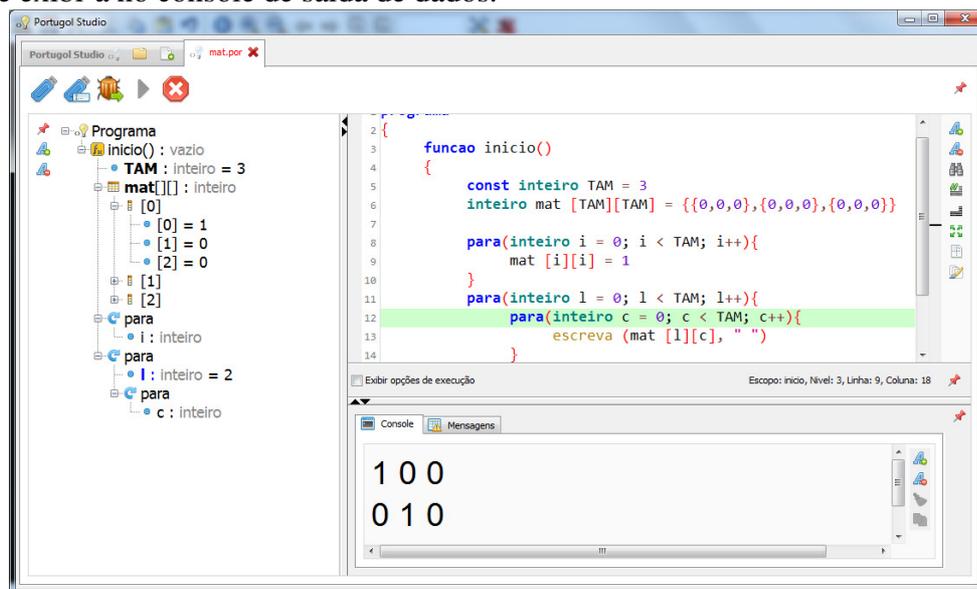


Figura 4 – Depuração em andamento em programa com matriz.

É possível notar na figura 4 que a árvore de símbolos representa hierarquicamente o escopo das variáveis que foram definidas dentro de loops. Representa ainda os valores contidos nas matrizes onde as linhas e colunas são apresentadas como vetores. A utilização do depurador demanda apenas a utilização do botão depurar (ícone de inseto na figura) que ao ser clicado inicia o processo de depuração e ao ser clicado novamente realiza um passo da depuração. É possível configurar a depuração para que ela seja detalhada (entrando nas funções) ou não.

A linha sendo depurada fica sempre no centro da tela de código, evitando que o aluno se desorienta na execução depurações de códigos com mais linhas e com loops.

4.6 Recursos de Apoio a Aprendizagem

Nesta seção apresentam-se outros recursos que foram agregados ao Portugol Studio e que contribuem para aprendizagem dos alunos iniciantes em programação.

Logo nos primeiros contatos com o Portugol Studio, é natural que o aluno ainda não conheça a sintaxe do Portugol, e por este motivo deu-se atenção especial para a ajuda da linguagem. Dentro da IDE Portugol Studio o aluno tem acesso ao conteúdo da ajuda. Na grande maioria dos tópicos de ajuda o aluno pode ver um exemplo de código que ilustra o tópico em discussão. Além disso, sempre que o código exemplo está presente o aluno pode clicar em botão “Tente você mesmo” para abrir o código automaticamente no editor do Portugol Studio, podendo executar o código, depurar, modificar, etc. Também criou-se um canal no *youtube* onde os alunos podem assistir vídeo-aulas sobre conceitos de programação introdutória usando o Portugol Studio. Ao todo são 11 vídeos.

O Portugol Studio também provê bibliotecas de funções para auxiliar o aluno na construção de programas mais sofisticados. Cabe destacar aqui as bibliotecas gráfica, teclado e sons que possibilitam a construção de jogos em duas dimensões utilizando o Portugol. Alguns jogos de baixa complexidade programados com a biblioteca gráfica foram disponibilizados como exemplos no Portugol Studio, permitindo que os alunos compreendam os princípios fundamentais da programação de jogos em duas dimensões. Além disso, os alunos podem abrir o código exemplo destes jogos e modificá-los, ou depurá-los.

5. Considerações Finais

Desde seu lançamento em 2013 a IDE Portugol Studio contabilizou quase 2.000 downloads em um período de oito meses. Como foi divulgado apenas em pequenos grupos acadêmicos, o número de downloads surpreendeu. Temos ciência que a IDE foi adotada como primeira opção em pelo menos uma Universidade Federal no estado de São Paulo e em nossa universidade aproximadamente 200 alunos de Ciência e Engenharia da Computação já utilizaram-na em sua formação.

O Portugol Studio é um projeto de código fonte aberto e conta atualmente com 10 colaboradores, entre alunos e professores. Uma dissertação de mestrado e quatro trabalhos de conclusão de curso já foram realizados em torno deste projeto, sendo que no momento em que este artigo é escrito outros dois trabalhos de conclusão de curso estão em andamento.

Neste sentido entende-se que além de ter o potencial de auxiliar a aprendizagem dos estudantes possui o potencial de se tornar um projeto de colaboração acadêmica entre universidades que o adotam. Seu desenvolvimento foi feito inteiramente na

linguagem Java e o código está disponível em <https://github.com/UNIVALI-L2S/Portugol-Studio>.

O Portugol Studio possui um núcleo que é responsável pela compilação, execução e depuração do código. Este núcleo é completamente desacoplado da interface gráfica da IDE, sendo possível reutilizá-lo em outros projetos. Este núcleo foi programado na linguagem JAVA, mas é possível utilizá-lo com outras linguagens de programação através da tecnologia CORBA, como demonstrado no trabalho de Noschang (2012).

Um dos trabalhos em andamento no Portugol Studio é um corretor automático para problemas algorítmicos previamente definidos pelo professor. Este corretor deverá, além de determinar se o algoritmo está correto ou não, gera dicas para auxiliar os alunos iniciantes na resolução de problemas pré-determinados pelo professor. Este mecanismo de correção já está em fase final de testes e futuramente pretende-se utilizá-lo em um ambiente integrado de aprendizagem e competição.

Outro aspecto que está sendo aprimorado no Portugol Studio é a criação automatizada de *break points* que possam auxiliar a compreensão do código durante a depuração, já que as IDEs profissionais exigem que os *break points* sejam adicionados manualmente.

6. Referências

BEAUBOUF, Theresa; MASON, John. Why the high attrition rate for computer science students: some thoughts and observations. In: ACM SIGCSE Bulletin, USA, v. 37, n. 2, p. 103-106, jun. 2005.

CRISTÓVÃO, Henrique Monteiro. Aprendizagem de Algoritmos num Contexto Significativo e Motivador: um relato de experiência. In: CONGRESSO DA SBC – WEI – WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 18., Belém do Pará, Pará. Anais... 2008.

DALE, Nell B. Most difficult topics in CS1: results of an online survey of educators. ACM SIGCSE Bulletin, USA, v. 38, n.2, p. 49-53, jun. 2006.

EPI - English Proficiency Index 2013. Disponível em: <http://www.ef.com.br/eipi/>. Acessado em nov, 2013.

HUANG, J.; PIECH, C.; NGUYEN, A.; GUIBAS, L. Syntactic and Functional Variability of a Million Code Submissions in a Machine Learning MOOC. 1st Workshop on Massive Open Online Courses at the 16th Annual Conference on Artificial Intelligence in Education, p. 25–32, Memphis, Tennessee, USA, 2013.

LAHTINEN, E.; ALA-MUTKA, K.; JÄRVINEN, H.-M. A study of the difficulties of novice programmers. Annual Sigcse Conference On Innovation And Technology In Computer Science Education, v. 37, n. 3, p. 14–18, Caparica, Portugal, 2005.

MAXIMO, Luis F.; RAABE, André. L. A.; BARONE, Dante A. C. Avaliação formativa assistida por computador no ensino a distância. RENOTE. Revista Novas Tecnologias na Educação, v. 5, p. 51-61, 2007.

MEC - Ministério da Educação e Cultura - Programa Inglês sem fronteiras. Disponível em: <http://isf.mec.gov.br/programa.html>. Acessado em abr, 2014.

MSELLE, L. Enhancing Comprehension by Using Random Access Memory (RAM) Diagrams in Teaching Programming: Class Experiment. In: Lawrance J, Bellamy R (eds.) *Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group*; 2010 Sep 19–21; University Carlos III of Madrid, Leganés. Madrid: Maria Paloma Díaz Pérez & Mary Beth Rosson.

NOSCHANG, Luis Fernando. Usando Corba para integrar um compilador desenvolvido em JAVA com ferramentas em outras linguagens de programação. *REAVI - Revista Eletrônica do Alto Vale do Itajaí*, Vol 2, nº 1, 2013.

NGUYEN, A.; PIECH, C.; HUANG, J.; GUIBAS, L. Codewebs: Scalable Homework Search for Massive Open Online Programming Courses. *International World Wide Web Conference*, Seoul, South Korea, 2014.

RAABE, A. L. A.; DAZZI, R. L. S.; SANTIAGO, R. de. Adquirindo experiência na construção de ferramentas de apoio a aprendizagem de algoritmos. In: *Workshop de Ambientes de Apoio a Aprendizagem de Algoritmos e Programação -*, 2007, São Paulo. XVIII Simpósio Brasileiro de Informática na Educação, 2007.

SHEARD, Judy et al. Analysis of research into the teaching and learning of programming. In: *INTERNATIONAL COMPUTING EDUCATION RESEARCH WORKSHOP*, 5., 2009, Berkeley, USA. *Proceedings...* USA: ACM, 2009. p. 93-104.

SCHULTE, Carsten; BENNEDSEN, Jens. What do teachers teach in introductory programming? In: *INTERNATIONAL WORKSHOP ON COMPUTING EDUCATION RESEARCH*, Canterbury, United Kingdom. *Proceedings...* USA:ACM, sep. 2006. p. 17-28

ZANINI, Adriana S. ; RAABE, André L.A. . Análise dos enunciados utilizados nos problemas de programação introdutória em cursos de Ciência da Computação no Brasil. In: *WEI - XX Workshop sobre Educação em Computação*, 2012, Curitiba. XXX Congresso da Sociedade Brasileira de Computação. Curitiba: SBC, 2012.