

Simulador Web para a Família de Processadores BIP

Bruno P. Peres, Cesar A. Zeferino, Paulo V. Vieira

Universidade do Vale do Itajaí (UNIVALI)
Rua Uruguai, 458 – 88302-901 – Itajaí, SC – Brasil

brunoperes@edu.univali.br, {zeferino,paulovieira}@univali.br

Abstract. *The BIP family and the integrated development environment Bipide were developed to facilitate the teaching of programming and Computer Organization & Architecture concepts through an interdisciplinary approach. However, the use of Bipide is limited to Microsoft Windows environments. This paper presents a web version of Bipide with the main features of previous versions and offers services for compiling programs written in high level language. The experimental and evaluation results demonstrate that this new version is 100% operational and that it was well accepted by the students.*

Resumo. *A família de processadores BIP e o ambiente de desenvolvimento integrado Bipide foram desenvolvidos com a finalidade de facilitar o ensino de programação e de conceitos de arquitetura e organização de computadores seguindo uma abordagem interdisciplinar. No entanto, o Bipide é uma IDE desktop com uso limitado a ambientes Microsoft Windows. Para contornar essa limitação, foi desenvolvida a versão web com suporte as principais características originais do Bipide e que ainda dispõe de serviços para compilação de programas escritos em linguagem de alto nível via web. Os resultados experimentais e de avaliação demonstram que a ferramenta é 100% operacional e que foi bem aceita pelos estudantes.*

1. Introdução

Compreender alguns dos conceitos básicos de arquitetura e organização de computadores mostra-se de grande valia para o melhor entendimento das abstrações envolvidas no ensino de algoritmos e programação. Apesar disso, percebe-se que as disciplinas que abordam esses conceitos nos cursos de graduação na área de Computação costumam ser ministradas após as disciplinas introdutórias de programação [Morandi et al. 2006].

Considerando essa característica, um grupo interdisciplinar de pesquisadores desenvolveu uma família de processadores básicos com recursos simples o suficiente para serem utilizados nas primeiras semanas de um curso de graduação de Computação. Isso permite que os alunos ingressantes possam visualizar e compreender os conceitos de arquitetura e organização de computadores necessários para um melhor entendimento dos conceitos básicos de programação [Morandi et al. 2006]. Atualmente a família de processadores BIP (Basic Instruction-set Processor) é composta de cinco processadores, sendo que cada um deles estende a arquitetura e organização de seu antecessor, adicionando novas funcionalidades. Também faz parte dessa iniciativa o desenvolvimento de um ambiente de desenvolvimento integrado educacional denominado Bipide (BIP Integrated Development Environment). Com o Bipide, é possível escrever programas em linguagem de alto nível e traduzi-los para a linguagem de montagem dos processadores BIP. Além disso, o Bipide oferece um simulador da arquitetura e da organização dos

processadores BIP com recursos de animação que ilustram claramente as etapas da execução de um programa no processador [Vieira et al. 2009].

As diferentes versões da ferramenta Bipide desenvolvidas até o momento apresentam algumas limitações quanto ao seu uso. A principal limitação refere-se às tecnologias utilizadas em sua implementação, as quais limitam seu uso a ambientes Microsoft Windows. Nesse contexto, este artigo descreve o desenvolvimento de uma versão *web* da ferramenta Bipide que pode ser executada em qualquer sistema operacional e acessada por meio da internet, sem a necessidade de instalação de *plugins* ou de configurações especiais. Essa versão, denominada Bipide Web, mantém os principais recursos presentes nas versões anteriores, tais como (i) suporte à linguagem de alto nível Portugol; (ii) a capacidade de compilar e simular graficamente a execução desses programas, demonstrando o funcionamento interno da organização do processador; e (iii) o suporte a todas as versões dos processadores BIP.

O restante deste artigo é organizado em cinco seções. A Seção 2 apresenta os processadores BIP e o ambiente integrado Bipide. Na Seção 3, é feito um posicionamento do ambiente proposto em relação a ferramentas similares. A Seção 4 descreve o desenvolvimento da versão *web* do Bipide. Na Seção 5, são apresentados os procedimentos de verificação e avaliação da ferramenta e, por último, na Seção 6, são apresentadas as conclusões deste trabalho.

2. Projeto BIP

O Projeto BIP disponibiliza uma família de processadores com arquitetura simplificada cujo o objetivo é auxiliar no ensino de conceitos básicos de Arquitetura e Organização de Computadores para alunos dos períodos iniciais dos cursos de graduação em Computação, de modo a facilitar a compreensão das abstrações utilizadas nas disciplinas de Algoritmos e Programação, tais como variáveis, constantes, atribuições e operações aritméticas, entre outros [Vieira et al. 2009].

Os recursos dos processadores que compõem a família BIP foram especificados de maneira incremental, onde cada processador estende a arquitetura de seu predecessor, adicionando novas funcionalidades. Assim, cada um dos processadores mantém seu foco na compreensão de conceitos específicos [Morandi et al. 2006]. Atualmente, a família de processadores BIP possui cinco versões distintas, sendo elas:

- **BIP I:** Suporta operações aritméticas (soma e subtração) e de acesso a memória;
- **BIP II:** Adiciona suporte desvios condicionais e incondicionais;
- **BIP III:** Inclui suporte a operações de lógica bit-a-bit e de deslocamento lógico;
- **BIP IV:** Adiciona suporte a procedimentos, manipulação de vetores e entrada-e-saída; e
- **μBIP:** Agrega características típicas de microcontroladores, tais como portas de entrada-e-saída, controlador de interrupções e temporizador.

Também faz parte do projeto BIP uma IDE denominada Bipide, a qual possibilita a escrita de programas em linguagem de alto nível, a tradução dos programas para a linguagem de montagem dos processadores BIP e a simulação passo-a-passo de tais programas, com o uso de animações. Esta abordagem permite ao aluno relacionar os

conceitos de programação em alto nível com os conceitos de baixo nível [Vieira, Raabe e Zeferino 2010].

A primeira versão do Bipide (v. 1.0) permitia ao usuário escrever programas em Portugol e simular tais programas sobre os processadores BIP I e BIP II. Em seguida, o Bipide v. 2.0 passou a permitir a programação diretamente em linguagem de montagem da família BIP. O Bipide v. 3.0 adicionou o suporte aos processadores BIP III e BIP IV. Por fim, a versão 4.0 da IDE adicionou suporte à linguagem de programação C e ao processador μ BIP [Oliveira et al. 2014].

Durante o processo de simulação dos algoritmos escritos na IDE Bipide, pode-se visualizar os conteúdos dos registradores do processador, da memória de dados e da memória de instrução. Além desses elementos, também pode-se visualizar o código-fonte em alto nível com seu equivalente em linguagem de montagem. Essas características permitem ao usuário criar relações e fazer comparações entre os elementos, facilitando o entendimento das abstrações envolvidas [Vieira, Raabe e Zeferino 2010]. A Figura 1 apresenta a interface do ambiente de simulação do Bipide v. 3.0.

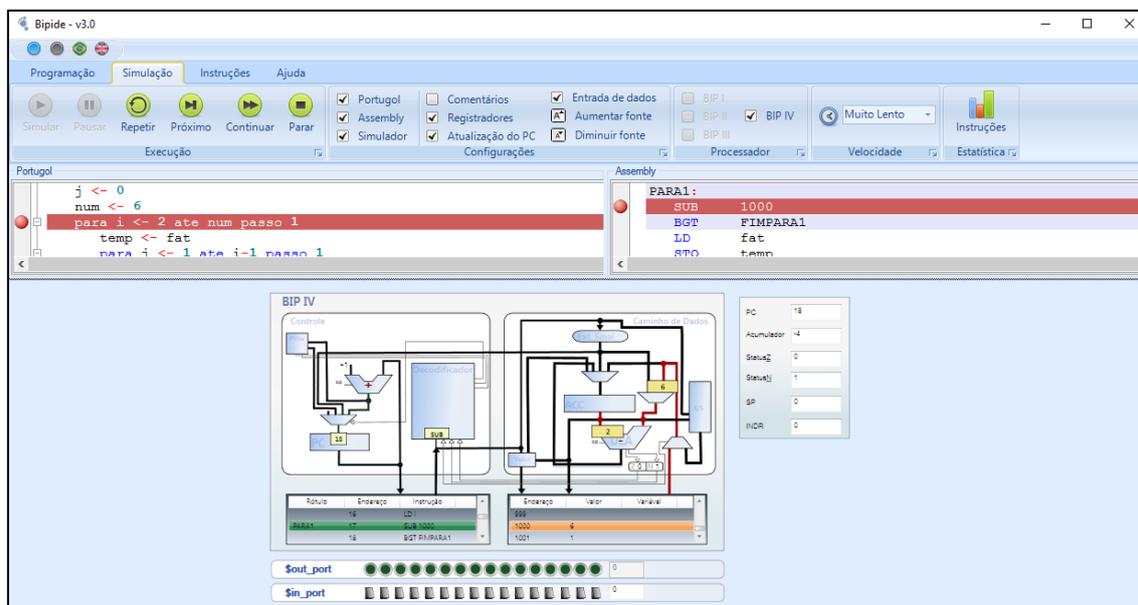


Figura 1. Ambiente de simulação do Bipide 3.0

3. Trabalhos Relacionados

Esta seção apresenta o resultado de uma análise realizada sobre outros simuladores de arquitetura e organização de computadores de propósito semelhante ao Bipide. A tabela comparativa presente nesta seção serviu como base para a definição de algumas das funcionalidades presentes na versão web do Bipide. O processo de seleção dos simuladores analisados levou em consideração apenas ferramentas disponíveis na internet, que contém simuladores de arquitetura e/ou da organização de computadores. Igualmente, foram consideradas apenas ferramentas que permitem ao usuário programar em linguagem de montagem e/ou de alto nível.

Os simuladores selecionados foram o FRISCjs [Zuzak 2016], o Simple 8-bit Assembler Simulator [Schweighauser 2016], o Basic MIPS Simulator [Morriswmz 2016], o Asm86 [Neves 2016] e o Little Man Computer [Higginson 2016].

A Tabela 1 apresenta as principais funcionalidades e características identificadas nos simuladores analisados e um comparativo com a versão *web* do Bipide. Pela tabela, observa-se que o Bipide Web e o Little Man Computer são os únicos que integram um simulador de organização e arquivos de ajuda, sendo que o Bipide Web destaca-se em relação ao Little Man Computer por suportar a programação em linguagem de alto nível. A configuração de *clock* de CPU e a pesquisa em endereço de memória não foram implementadas no Bipide Web pois entende-se que tais funcionalidades não são essenciais para o propósito de uso da ferramenta. Essas funcionalidades poderão ser implementadas em futuras versões da ferramenta.

Tabela 1: Comparação das funcionalidades dos simuladores analisados

Característica	FRISCjs	Simple 8-bit Assembler Simulator	Basic MIPS Simulator	Asm86	Little Man Computer	Bipide Web
Propósito acadêmico	X	X		X	X	X
Configuração de <i>clock</i> de CPU	X	X				
Programação em alto nível						X
Programação em assembly	X	X	X	X	X	X
Visualização do assembly	X	X	X	X	X	X
Visualização dos registradores	X	X	X	X	X	X
Execução passo-a-passo	X	X	X	X	X	X
Simulação da organização					X	X
Simulação da arquitetura	X	X	X	X	X	X
Pesquisar endereço de memória			X			
Identificação de erros	X	X	X	X	X	X
Arquivos de ajuda	X	X			X	X
Programas de exemplo	X	X	X	X	X	X

4. Desenvolvimento

A versão do Bipide descrita neste artigo foi desenvolvida na forma de um sistema *web*, respeitando a arquitetura cliente-servidor e o protocolo de comunicação HTTP (Hypertext Transfer Protocol – Protocolo de Transferência de Hipertexto). Sendo assim, o Bipide Web é composto de uma aplicação cliente e uma aplicação servidora.

Cada uma dessas aplicações possui responsabilidades bem definidas e são independentes entre si. Essa independência entre a aplicação cliente e a aplicação servidora permite que as funcionalidades implementadas na aplicação servidora, tais como o cadastro de usuário, criação e compilação de programas possam ser reaproveitadas em trabalhos futuros.

A Figura 2 apresenta uma visão geral do sistema. Nela, pode-se visualizar que a aplicação servidora é composta de: (i) um compilador Portugol; (ii) um montador; (iii) uma API REST [Fielding 2000], responsável pela comunicação com a aplicação cliente; e (iv) um banco de dados, responsável por armazenar dados dos usuários. A aplicação cliente foi desenvolvida utilizando as tecnologias HTML (HyperText Markup Language), CSS (Cascading Style Sheets) e JavaScript. A aplicação cliente é executada com o uso de um navegador *web*.

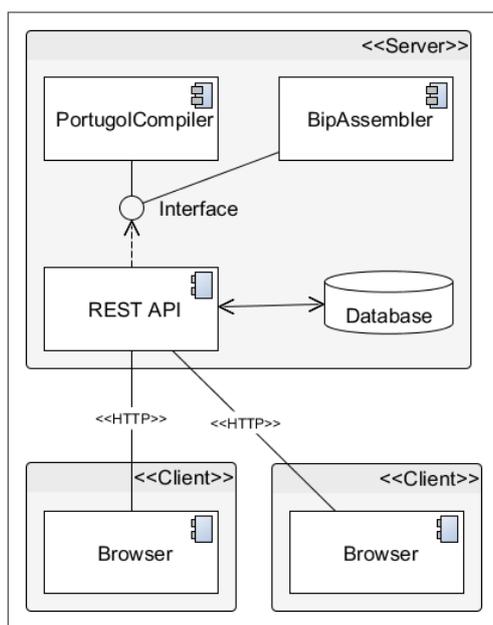


Figura 2. Diagrama de componentes do Bipide Web.

4.1 A Aplicação Servidora

A aplicação servidora da versão *web* do Bipide foi desenvolvida sobre a plataforma Java por esta ser uma tecnologia *open-source* e multiplataforma. Essa escolha também considerou o fato de que vários artefatos de software produzidos anteriormente e reaproveitados neste trabalho foram construídos sobre esta mesma plataforma, como é o caso do compilador Portugol utilizado. Assim, o uso da plataforma Java proveu uma melhor aderência dessas ferramentas ao projeto.

É de responsabilidade da aplicação servidora do Bipide Web: (i) permitir que a aplicação cliente crie novos cadastros de usuários; (ii) permitir que usuários se autenticem no sistema por meio do fornecimento de credenciais; e (iii) permitir a criação, recuperação e compilação de programas escritos na linguagem Portugol e na linguagem de montagem dos processadores BIP. A comunicação da aplicação cliente com a aplicação servidora ocorre por meio da API REST.

4.2 O Compilador e o Montador

A versão *web* do Bipide possui um compilador Portugol e um montador da linguagem de montagem dos processadores da família BIP. Ambos implementam a mesma interface Java, a qual possui um método específico que recebe por parâmetro o texto do programa fonte e retorna uma estrutura JSON (JavaScript Object Notation) representando o resultado da compilação.

A estrutura JSON devolvida contém toda a informação necessária ao simulador da aplicação cliente. Essa informação é constituída de: (i) uma lista na entrada de memória de dados; (ii) uma lista de instruções em *assembly* com metadados; (iii) uma lista de erros e avisos (quando existirem); e (iv) a versão mínima do processador BIP capaz de executar o algoritmo em questão.

O compilador Portugol tem a função de traduzir algoritmos escritos em Portugol para um equivalente em linguagem de montagem dos processadores da família BIP. O

processo de desenvolvimento desse compilador utilizou o Portugol Core [Oliveira 2014], com a finalidade de aproveitar a representação intermediária gerada por esse núcleo no formato de uma ASA (Árvore Sintática Abstrata). Dessa forma, o desenvolvimento do compilador Portugol da versão web do Bipide deu-se através da criação de uma classe Java capaz de percorrer a ASA gerada pelo Portugol Core e traduzir essa estrutura em um programa equivalente em linguagem de montagem dos processadores BIP.

A função do montador é a de garantir que os programas escritos em linguagem de montagem dos processadores da família BIP estejam corretos léxica, sintática e semanticamente. A linguagem fonte e a linguagem alvo são as mesmas (*assembly* dos processadores BIP). Assim, não é realizado nenhum processo de tradução do programa fonte, conforme acontece no compilador Portugol. O montador foi implementado com o uso da ferramenta ANTLR (ANother Tool for Language Recognition) em sua versão 3.5.2 (ANTLR, 2016).

4.3 A Aplicação Cliente

A aplicação cliente do Bipide é composta de três módulos: (i) Editor; (ii) Simulação; e (iii) Ajuda.

O Editor (módulo de edição de programas) permite ao usuário autenticado criar, editar e compilar programas escritos nas linguagens Portugol e *assembly*. Conforme pode ser visto na Figura 3, a interface do Editor é composta de: (1) uma listagem dos programas criados pelo usuário, separados nas categorias Portugol e Assembly; (2) uma barra de ferramentas contendo um conjunto de botões de ações; (3) um editor de código-fonte com destaque de sintaxe; e (4) um painel de saída em que mensagens de erro e avisos gerados durante a compilação são apresentados ao usuário.

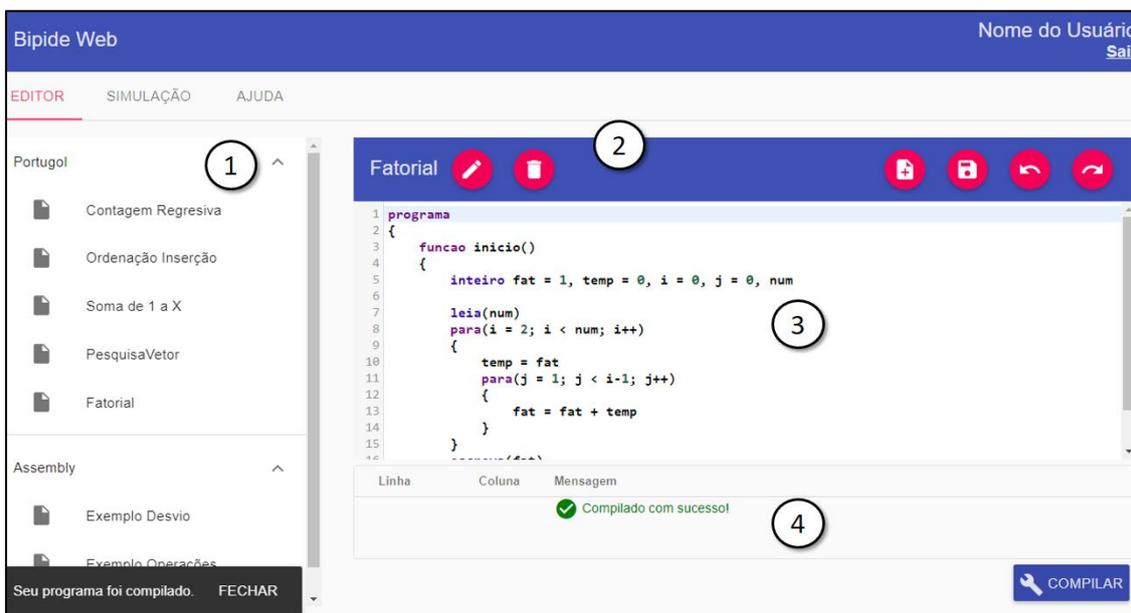


Figura 3. Interface do módulo de edição de programas.

O módulo Simulação permite ao usuário executar um programa escrito em Portugol ou em *assembly* e visualizar o comportamento dos componentes internos dos processadores BIP por meio de animações. Conforme pode ser visto na Figura 4, o módulo de simulação do Bipide Web é composto de: (1) uma área onde são apresentadas

as instruções do programa-fonte; (2) uma área onde apresentam-se as instruções do programa compilado; (3) uma barra de ferramentas com botões de controle e opções do simulador; (4) a representação da organização do processador BIP; (5) os valores atuais dos registradores do processador; (6) os valores atuais dos registradores de entrada-e-saída; (8) uma representação da memória de instruções; e (9) uma representação da memória de dados.

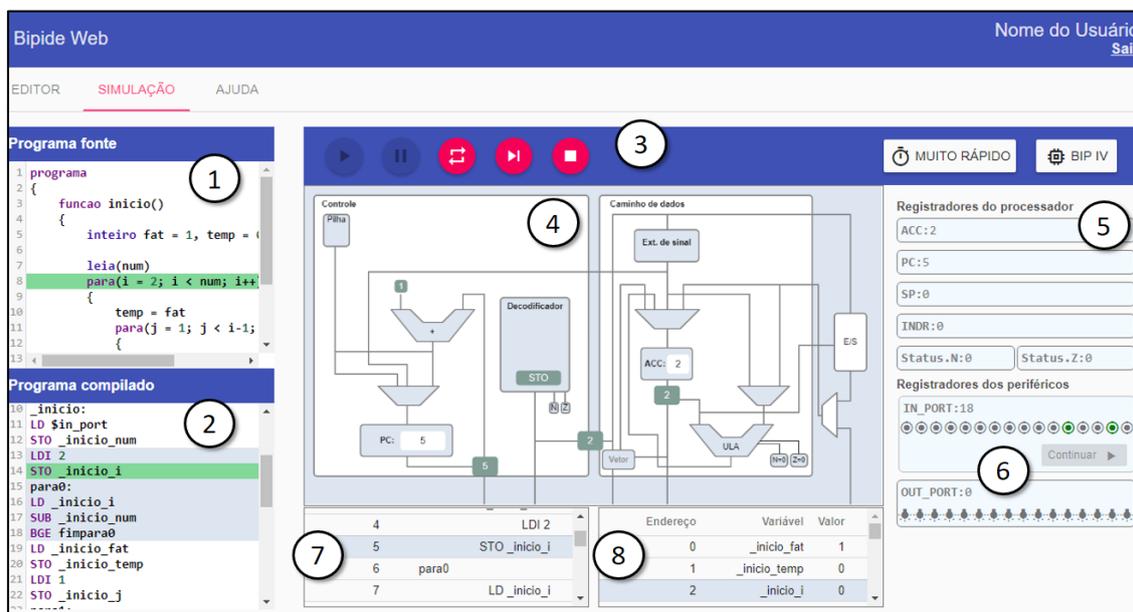


Figura 4. Interface do módulo de simulação.

Assim como nas versões anteriores do Bipide, essa nova versão permite ao usuário visualizar a relação existente entre uma instrução do programa-fonte e as instruções correspondentes geradas por tal instrução no programa compilado. Essa associação é feita por meio do destaque das linhas correspondentes do programa fonte e do programa compilado, conforme pode ser visto nos painéis laterais da Figura 4 (lado esquerdo).

O Bipide Web conta ainda com o módulo Ajuda, pelo qual o usuário pode obter informações sobre tópicos de Arquitetura e Organização de Computadores, descrição dos processadores BIP, aspectos da sintaxe da linguagem Portugol, algoritmos de exemplo e informações sobre as funcionalidades do Bipide Web.

5. Verificação e avaliação

5.1 Experimentos de verificação

Para realizar a verificação do compilador Portugol foram utilizados os algoritmos de teste disponibilizados em Mannes (2013). Cada um desses algoritmos foi compilado e executado no Bipide Web. Os resultados obtidos, incluindo os valores dos registradores e o estado final da memória de dados, foram comparados com os resultados esperados.

A verificação do montador deu-se por meio da execução dos algoritmos de teste presentes em Rech (2011). Assim como na validação do compilador Portugol, os algoritmos de teste escritos em *assembly* foram executados no Bipide Web. Após a execução de cada um dos algoritmos, verificou-se os valores dos principais registradores e o estado final da memória de dados do programa.

Os resultados obtidos durante o processo de verificação do Bipide Web correspondem aos resultados esperados, o que demonstra o correto funcionamento dos compiladores e da interface de simulação. Vale salientar que a interface de simulação foi validada nos navegadores Google Chrome (versão 54.0.2840.87), Mozilla Firefox (versão 49.0.2) e Microsoft Edge (versão 38.14393.0.0). O ambiente funcionou corretamente nos três navegadores, permitindo cadastrar novos usuários, editar, compilar e simular corretamente os programas desenvolvidos.

5.1 Avaliação com os Alunos

A nova versão do Bipide foi utilizada em duas disciplinas de graduação em Ciência da Computação, uma de Algoritmos e Programação (41 alunos) e outra de Arquitetura e Organização de Computadores (13 alunos). Os alunos puderam explorar a ferramenta e utilizá-la como apoio na realização de exercícios em laboratório de informática. Os exercícios propostos incluem a elaboração de algoritmos em Portugal e Assembly e a realização de testes de mesa.

Para avaliar a percepção dos alunos sobre a interface e a utilidade da ferramenta, foi elaborado um questionário estruturado composto de 11 questões, apresentadas na Tabela 2. O questionário foi disponibilizado na internet e os professores das disciplinas orientaram os estudantes a responde-lo após a utilização do Bipide Web.

Os itens do questionário utilizaram uma escala de cinco pontos do tipo Likert. A escala Likert requer que os entrevistados indiquem seu grau de concordância ou discordância com declarações relativas ao tema em análise [Ary et al. 2010]. Uma escala quantitativa foi atribuída a cada item, como segue: concordo totalmente (5), concordo parcialmente (4), não concordo e nem discordo (3), discordo parcialmente (2) e discordo totalmente (1).

De acordo com Boone Jr. e Boone (2012), uma das formas mais apropriadas para analisar um questionário composto por itens do tipo Likert é verificar a moda (resposta mais frequente) ou a mediana das respostas obtidas. Pode-se ainda agrupar quatro ou mais itens do questionário, formando categorias específicas que visam analisar as variáveis de interesse da pesquisa. Essas categorias podem então ser analisadas para obter a percepção dos respondentes por meio do cálculo da média e desvio padrão dos itens que compõem cada categoria [Boone Jr. e Boone 2012]. Considerando esse fato, os itens do questionário de avaliação foram agrupados por similaridade em duas categorias: (i) satisfação com a interface (Q1 a Q6) e (ii) utilidade da ferramenta (Q7 a Q11).

A Tabela 2 apresenta os resultados obtidos, onde observa-se a moda e a mediana obtidas para cada item do questionário e a média aritmética e desvio padrão das categorias analisadas (quanto mais próximo de 5, melhor). Percebe-se a satisfação dos estudantes em relação à interface do sistema (média de 4,41) e fica evidente que a ferramenta foi útil para a realização das tarefas solicitadas em sala de aula (média de 4,31).

Ao analisar as disciplinas separadamente, percebe-se que a média obtida na disciplina Arquitetura e Organização de Computadores é um pouco inferior, quando comparada com a disciplina de Algoritmos e Programação, como pode ser observado na Figura 5. Acredita-se que essa diferença se deve ao fato desses estudantes já terem utilizado a versão anterior do Bipide e estarem mais acostumados com sua interface. Porém, mesmo com essa diferença, percebe-se uma avaliação positiva da ferramenta.

Tabela 2: Resultados da Análise do Questionário de Avaliação

Questões	Moda	Mediana	Média por Categoria	Desvio Padrão por Categoria
Q1 - Foi simples usar o sistema	5	5		
Q2 - Eu me senti confortável usando o sistema	5	5		
Q3 - Foi fácil aprender a usar o sistema	5	5	4,41	0,11
Q4 - A interface do sistema é agradável	5	5		
Q5 - Eu gostei de usar o sistema	5	5		
Q6 - As funcionalidades do sistema estão muito bem integradas	5	4		
Q7 - O sistema me auxiliou na realização das tarefas	5	5		
Q8 - O sistema tem todos os recursos que eu precisei para realizar as tarefas	5	5		
Q9 - As mensagens de erro me auxiliaram de forma adequada	5	4	4,31	0,21
Q10 - As informações (tais como ajuda e mensagens na tela) são claras	5	4		
Q11 - A informação fornecida na Ajuda do sistema é fácil de ser entendida	5	4		

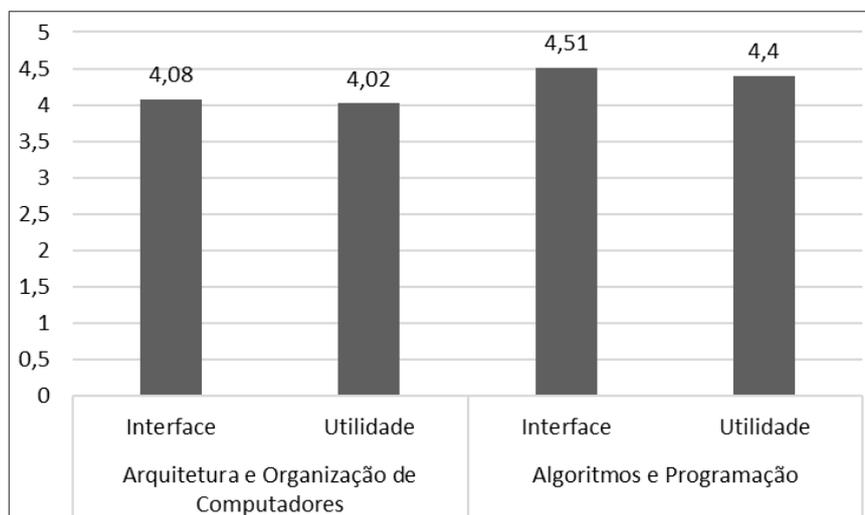


Figura 5. Médias obtidas em cada disciplina.

6. Conclusões

Este artigo apresentou o Bipide Web, uma versão online do ambiente de desenvolvimento integrado Bipide. Esta nova versão contorna limitações das versões anteriores do Bipide, possibilitando o uso da ferramenta em diferentes plataformas.

Outra característica que demonstra a relevância deste trabalho diz respeito a disponibilização de um serviço *web* capaz de compilar e traduzir programas escritos em Portugol para a linguagem de montagem da família BIP; bem como a possibilidade de analisar e validar códigos escritos na linguagem de montagem do BIP. Essas funcionalidades permitirão que outros desenvolvedores criem novas ferramentas e interfaces de usuários, eliminando a necessidade de recriar o processo de compilação.

Os resultados da avaliação em sala de aula apresentados neste artigo indicam que os alunos demonstraram satisfação com a interface da ferramenta e suas funcionalidades, além de evidenciar que a ferramenta foi útil para um melhor entendimento dos conceitos vistos em aula e para a realização das tarefas solicitadas. Contudo, pretende-se realizar

novas avaliações utilizando grupos de controle, com o intuito de verificar o impacto da ferramenta na aprendizagem.

A versão *web* ainda abre espaço para funcionalidades adicionais como aplicação de exercícios práticos em laboratório com avaliação *on-line*, bem como um melhor acompanhamento da comunidade de usuários (quantidade, instituição, tempo de uso da ferramenta, erros mais frequentes, entre outros).

Referências

- Antlr. (2016) “About The ANTLR Parser Generator”, <http://www.antlr.org/about.html>.
- Ary, Donald et al. (2015) Introduction to Research in Education. 8th edition.
- Boone Jr., Harry N. and Boone, Deborah A. (2012) “Analyzing Likert Data”, In *Journal of Extension*, v. 50, n. 2.
- Doglio, Fernando. (2015) Pro REST API Development with Node.js. 1st edition.
- Fielding, Roy Thomas. (2000) Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine.
- Higginson, Peter. (2016) “Little Man Computer”, <https://peterhigginson.co.uk/LMC/>.
- Mannes, Paula. (2013) Integração do Portugol Core com o Bipide. Trabalho Técnico-científico de Conclusão de Curso, Universidade do Vale do Itajaí, Itajaí.
- Morandi, Diana et al. (2006) “Um Processador Básico para o Ensino de Conceitos de Arquitetura e Organização de Computadores”. In *Hifen*, v. 30, p. 73-80.
- Morriswmz. (2016) “Simple MIPS Simulator in JavaScript”, <https://github.com/morriswmz/SimpleMIPS.js>.
- Neves, Carlos. (2016) “Assembly x86 Emulator”, <http://carlosrafaelgn.com.br/Asm86/>.
- Oliveira, Nereu P. et al. (2014) “Implementando Suporte a Novas Linguagens de Programação e outros Idiomas no Ambiente de Desenvolvimento Integrado Bipide”, In *International Journal of Computer Architecture Education*, v. 3, p. 5-8.
- Rech, Paulo R. M. et al. (2011) “BIP IV: especificação e suporte na ferramenta Bipide”. In: Proceedings of the 23rd International Symposium on Computer Architecture and High Performance Computing.
- Schweighauser, Marco. (2016) “Simple 8-bit Assembler Simulator”, <https://schweigi.github.io/assembler-simulator/>.
- Vieira, Paulo V. et al. (2009) “Bipide: Ambiente de Desenvolvimento Integrado para Utilização dos Processadores BIP no Ensino de Programação”, In: Anais do XX Simpósio Brasileiro de Informática na Educação.
- Vieira, Paulo. V., Raabe, André L. A. and Zeferino, Cesar A. (2010) “Bipide: ambiente de desenvolvimento integrado para a arquitetura dos processadores BIP”. In *Revista Brasileira de Informática na Educação*, v. 18, p. 32-43.
- Zuzak, Ivan. (2016) “FRISCjs: FRISC processor simulator in JavaScript”, <http://ivanzuzak.info/FRISCjs/webapp/>.