NeanderSIM: Simulador Gráfico de Apoio ao Ensino de Arquitetura de Computadores

Matheus Rudolfo Diedrich Ullmann, Ana Carolina Gondim Inocêncio, Edmundo Dias Moraes Neto, Marcelo Silva Freitas, Paulo Afonso Parreira Júnior

Universidade Federal de Goiás (UFG) – Regional Jataí
Br 364, Km 192 – CEP 75.801-615 – Jataí – GO – Brasil

{matheusullmann, anacarolina.inocencio, edmundodmneto, marcelo.caj.ufg}@gmail.com, paulojunior@jatai.ufg.br

Abstract. *Professors and students of some computer science disciplines, whose contents deal with theoretical and abstract aspects of computing, such as the "Computer Architecture", have found difficulties to teach/learn such contents only using non-interactive resources, such as blackboard and chalk, textbooks and slides. The literature has emphasized the usage of educational tools can help professors to minimize the obstacles of the teaching-learning process. This paper presents NeanderSIM, a graphic simulator to support teaching of Computer Architecture, which presents, in an interactive and animated way, the process of execution of instructions on a computer. In an evaluation conducted with eighteen students/professors of Computer Science, it was possible to notice that, on a scale ranging from one to seven, NeanderSIM has obtained the marks 6.42 and 6.22 regarding to its usefulness and easiness of use, respectively.*

Resumo. *Professores e estudantes de algumas disciplinas dos cursos de Ciência da Computação e áreas afins, tais como a "Arquitetura de Computadores", cujos conteúdos apresentam aspectos teóricos e abstratos de computação, têm encontrado dificuldades em ensinar/aprender tais conteúdos utilizando apenas recursos pouco interativos, como o quadro negro e giz, livros e slides. A literatura tem enfatizado o uso de ferramentas de ensino que podem ajudar os professores a minimizar os obstáculos do processo de ensino-aprendizagem. Este artigo apresenta NeanderSIM, um simulador gráfico para apoiar o ensino de Arquitetura de Computadores, que apresenta, de forma interativa e animada, o processo de execução de instruções em um computador. Em uma avaliação realizada com dezoito alunos/professores de Ciência da Computação, foi possível perceber que, em uma escala que varia de um a sete, NeanderSIM obteve as notas de 6,42 e 6,22 em relação à sua utilidade e facilidade de uso, respectivamente.*

1. Introdução

Em virtude da evolução dos recursos de Tecnologia da Informação e Comunicações (TICs), que têm se tornado cada vez mais interativos, os alunos têm apresentado menor interesse em aulas com apresentação de conteúdos apenas teóricos e abstratos e/ou quando recursos inadequados para exposição destes são utilizados (Mattos *et al.*, 2004). Uma das disciplinas da área de computação que pode ser prejudicada pela falta de recursos didáticos adequados é a disciplina de Arquitetura de Computadores, em especial, quando se trata de tópicos relacionados à organização dos componentes

internos do processador físico de um computador e à execução das instruções em nível de hardware, nestes componentes.

Isso acontece, pois o software é uma entidade lógica e não há uma representação palpável de sua execução nos componentes do computador. Sendo assim, é preciso simular a execução do software para esclarecer aos alunos os conceitos relacionados à organização de um computador e ao processo de execução do software nele. Ao ensinar tal processo de execução apenas utilizando lousa e giz, por exemplo, o professor terá poucos recursos para apresentar seus inúmeros passos, perdendo assim a interatividade e a representatividade necessária à explanação do conteúdo. Outro aspecto negativo é a falta de flexibilidade proporcionada por tais recursos, uma vez que exige-se um esforço significativamente alto por parte do professor para replicação da simulação com diferentes dados de entrada, isto é, outras instruções do computador. Além disso, o aluno não terá a oportunidade de simular a execução de outras instruções fora da sala de aula, se ele estiver utilizando apenas recursos pouco interativos, tais como livros-textos, vídeo-aulas, *slides*, entre outros.

Segundo Elias *et al*. (2011), ferramentas educacionais digitais têm sido desenvolvidas para auxiliar o ensino em várias áreas da computação e outras ciências. Tais ferramentas têm tornado-se cada vez mais sofisticadas e intuitivas, visando a apoiar o ensino em disciplinas cujo conteúdo possui natureza abstrata e dinâmica. Dessa forma, segundo Rodrigues e Martins (2008), no contexto da disciplina de Arquitetura de Computadores, o processo de ensino-aprendizagem pode ser facilitado com a utilização desse tipo de ferramenta.

A importância da realização de pesquisas sobre o ensino de Arquitetura de Computadores é notável. No Brasil, inclusive, há um evento específico sobre esse tema de pesquisa, a saber, o *Workshop* sobre Educação em Arquitetura de Computadores (WEAC)[1], que ocorre em conjunto com o *International Symposium on Computer Architecture and High Performance Computing*. Além disso, na literatura, há diversas ferramentas educacionais cujo objetivo é permitir ao aluno visualizar e entender melhor os conceitos relacionados a essa disciplina. Alguns exemplos são: *ProcSim* (ProcSim, 2012), *EKSMIPS* (Schemberger *et al*., 2010), *MIPS 32 bits* (Elias *et al*., 2011), *PS – CAS MIPS* (Maia *et al.*, 2009), *NeanderWin* (Borges e Silva, 2006) e *WNeander* (Weber, 2004).

Contudo, ainda há carência de ferramentas educacionais para auxiliar o ensino de determinados conceitos e tecnologias relacionados a esta disciplina. Como exemplo, pode-se citar a carência de ferramentas de apoio ao ensino de tópicos relacionados à organização e à execução de instruções no computador *Neander*. *Neander* é um computador desenvolvido para fins didáticos, proposto no livro de Weber (2004), e tem sido utilizado ao longo dos anos como livro-texto em diversas universidades do país (UFPE, 2014; UFRRJ, 2014; USP, 2014; UFRGS, 2014; UFSC, 2014). Um problema é a existência de diferenças significativas entre a arquitetura do computador *Neander* e a de outros computadores, como por exemplo, o *MIPS* (*Microprocessor without Interlocked Pipeline Stages*). Sendo assim, o aluno/professor fica impossibilitado de utilizar ferramentas educacionais que tenham sido desenvolvidas para computadores diferentes para ensinar tópicos relacionados ao *Neander*.

Neste contexto, o objetivo deste trabalho é apresentar a arquitetura e funcionamento da ferramenta educacional digital de apoio ao ensino de arquitetura de

[1] http://www.cin.ufpe.br/~sbac2013/weac/

computadores, denominada *NeanderSIM* (*Neander SIMulator*). Em particular, *NeanderSIM* aborda os conteúdos relacionados à arquitetura e à execução de instruções no computador *Neander*. Algumas justificativas para o desenvolvimento de outra ferramenta de apoio ao ensino-aprendizagem em arquitetura de computadores são: i) *NeanderSIM* contempla a arquitetura do computador *Neander*, ainda pouco explorada por ferramentas educacionais e amplamente utilizada em disciplinas de cursos de computação do Brasil; ii) as ferramentas educacionais existentes carecem de recursos apropriados de *feedback* ao aluno, principalmente quanto aos passos da execução dos processos simulados por elas, como pode ser visto na Seção 2 deste trabalho; e ii) as ferramentas pré-existentes foram desenvolvidas com tecnologias que hoje são consideradas obsoletas, como é o caso do *NeanderWin* (Borges e Silva, 2006), desenvolvido em Delphi, ou que não possuem recursos apropriados para construção de representações gráficas e animadas, o que pode dificultar sua manutenção/evolução, como é o caso do *NeanderWin* (Borges e Silva, 2006) e *WNeander* (Weber, 2004).

Para verificar a utilidade e a facilidade de uso do *NeanderSIM*, realizou-se uma avaliação dessa ferramenta com alunos e professores do curso de Bacharelado em Ciência da Computação da Universidade Federal de Goiás (Regional Jataí). Para isso, após terem passado por um treinamento e utilizado o *NeanderSIM*, os avaliadores responderam a um questionário construído com base no modelo de aceitação de tecnologia *TAM* (*Technology Acceptance Model*) (Davis *et al*., 1989). Como resultados, observou-se que a ferramenta *NeanderSIM* obteve aceitação positiva por parte dos avaliadores. Em uma escala de notas que vai de 1.0 (um) à 7.0 (sete), *NeanderSIM* obteve notas 6.42 e 6.22, respectivamente, quanto à utilidade e à facilidade de uso percebida pelos participantes da avaliação.

Este artigo está organizado da seguinte forma: na Seção 2 estão alguns trabalhos relacionados, que apresentam ferramentas educacionais de apoio ao ensino de arquitetura de computadores. Na Seção 3 a ferramenta *NeanderSIM* é apresentada, destacando-se sua arquitetura, interface e principais funções. Na Seção 4 é apresentada a avaliação do *NeanderSIM* e, por fim, na Seção 5 estão as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Existem diversas ferramentas para o ensino de arquitetura de computadores, cada uma com um enfoque específico em determinado processador. Nesta seção são apresentadas sucintamente as ferramentas *NeanderWin* (Borges e Silva, 2006), *WNeander* (Weber, 2004) e *ProcSim* (ProcSim, 2012). As justificativas para escolha dessas ferramentas são: i) as duas primeiras possuem enfoque no computador *Neander*, que é alvo de estudo deste trabalho. Sendo assim, faz-se necessário conhecer seus pontos fortes, limitações, bem como diferenças para a ferramenta desenvolvida neste trabalho; e ii) a última ferramenta (*ProcSim*) não trata do processador *Neander*, porém possui recursos gráficos e de animação mais sofisticados do que das duas primeira ferramentas. Portanto, também faz-se necessário conhecer os pontos fortes e limitações dessa ferramenta, bem como as principais contribuições do *NeanderSIM* sobre as características existentes no *ProcSim*.

NeanderWin é um ambiente educacional que inclui: i) um editor de código; ii) um montador (*assembler*); iii) um simulador de arquitetura e memória; iv) um utilitário para conversão de bases; v) um simulador de visor e painel de chaves; e vi) um gerador/carregador de imagem da memória simulada. Neste simulador, o código é escrito por meio do editor de texto ou através do tutor de programação, que ajuda a escrever o código em linguagem de montagem. Após isso, código feito em linguagem de máquina é

executado e são informados os resultados dessa execução por meio de visores (Figura 1). Algumas limitações dessa ferramenta são: i) ela não realiza verificação do código do usuário, ou seja, caso haja algum problema no código, o mesmo simplesmente não será executado e nenhum *feedback* será oferecido ao aluno; ii) a memória do computador é apresentada em hexadecimal, o que dificulta o entendimento dos valores armazenados na mesma; e iii) *NeanderWin* foi implementada em *Delphi*, uma linguagem de programação que foi descontinuada, desta forma, futuras manutenções que possam vir a ocorrer nesta ferramenta podem ser comprometidas.

WNeander é um simulador disponibilizado juntamente com o livro de Raul F. Weber "Fundamentos de Arquitetura de Computadores" (Weber, 2004). De modo análogo à ferramenta *NeanderWIN*, *WNeander* trata-se de uma ferramenta com poucos recursos gráficos, não sendo possível, por exemplo, visualizar como e por onde os dados trafegam no computador. Além disso, apenas os resultados da execução das operações são apresentados nos *displays* da ferramenta. Devido à limitação de espaço, a interface da ferramenta *WNeander* foi omitida deste texto.

ProcSim (*Processor Simulator*) é um simulador gráfico que permite ao usuário visualizar a dinâmica de execução das instruções no computador *MIPS* e identificar a funcionalidade de cada um de seus componentes (Figura 2). Após escrever o código *assembly* e iniciar a simulação, é possível ver como os dados trafegam dentro do computador, mas com um inconveniente, os registradores ficam dentro de uma área denominada "*Registers*" e muitos barramentos são ligados a ela, impedindo ao aluno visualizar o que cada registrador faz individualmente. Existem algumas outras ferramentas gráficas para ensino da arquitetura *MIPS*, tais como: i) *MIPS* 32 *bits* (Elias *et al*., 2011); ii) *EKSMIPS* (Schemberger *et al*., 2010); iii) *PS – CAS MIPS* (Maia *et al*., 2009), entre outras, porém essas ferramentas são para um tipo de computador distinto do que é tratado neste trabalho.

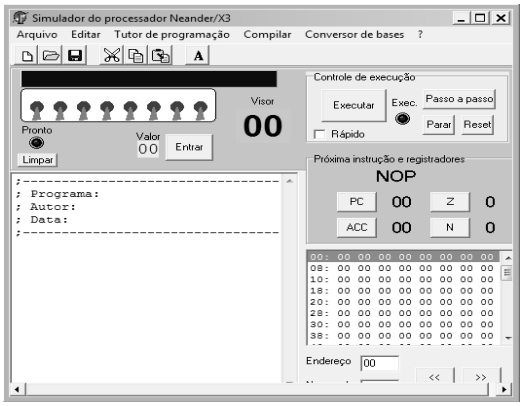

Figura 1. Interface do NeanderWin.

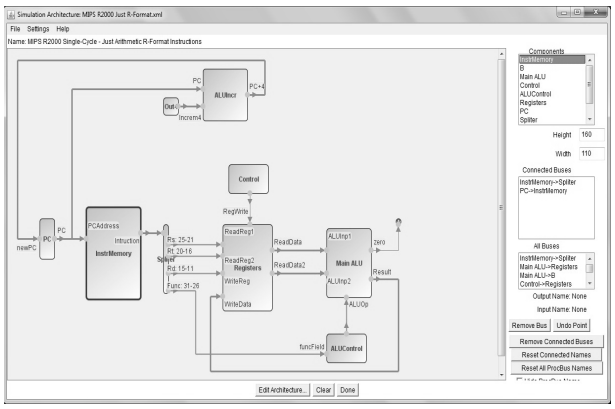

Figura 2. Interface do Interface do ProcSim.

O simulador desenvolvido neste trabalho, ***NeanderSIM***, apresenta as seguintes diferenças com relação às ferramentas citadas anteriormente: i) utiliza recursos gráficos e de animação para representar a execução das instruções nos componentes do computador *Neander*; ii) é uma ferramenta portável que pode ser executada via *web*, sem necessidade de instalação e que utiliza tecnologias atuais de programação, como a linguagem Java e o *framework* JavaFX, para construção de interfaces ricas; iii) oferece ao aluno um ambiente integrado e flexível, no qual ele possa, em uma única ferramenta, inserir as instruções que deseja executar e visualizar a execução das mesmas nos componentes do

computador; e iv) fornece *feedback* ao aluno de duas formas, alertando-o quanto aos erros cometidos durante a inserção do código fonte a ser executado no computador e informando-o, por meio de mensagens textuais, sobre os passos da execução das instruções inseridas por ele no computador.

3. *NeanderSIM*

NeanderSIM é uma ferramenta educacional gráfica baseada em simulação. Seu objetivo é simular as rotinas do computador *Neander* de forma animada e interativa, mostrando como os dados trafegam por meio dos barramentos e dos componentes do computador.

A. Interface Gráfica

O simulador *NeanderSIM* foi desenvolvido com a utilização da linguagem de programação *Java* e do *framework* para desenvolvimento de interfaces ricas, *JavaFX*. Estas tecnologias foram escolhidas, pois: i) possuem alta portabilidade; ii) permitem que produtos desenvolvidos nesta linguagem possam ser disponibilizados via *web*; e iii) têm sido atualizadas constantemente. Cabe ressaltar ainda que *NeanderSIM* é software livre e encontra-se disponível para *download* (código fonte e executável) via *web*[2].

A tela inicial do *NeanderSIM* é apresentada na Figura 3, na qual pode-se visualizar uma barra de *menu* no topo do simulador, que apresenta as opções "Arquivo" e "Ajuda" (Figura 3 - 1). A opção "Arquivo" possui o item "Exemplos Prontos", que trata-se de um conjunto de códigos de exemplo previamente cadastrados no simulador. A opção "Ajuda" possui o item "Manual", que consiste em um manual para aprimoramento do conhecimento dos usuários com relação ao funcionamento do *NeanderSIM* e aos conceitos de "Arquitetura de Computadores".

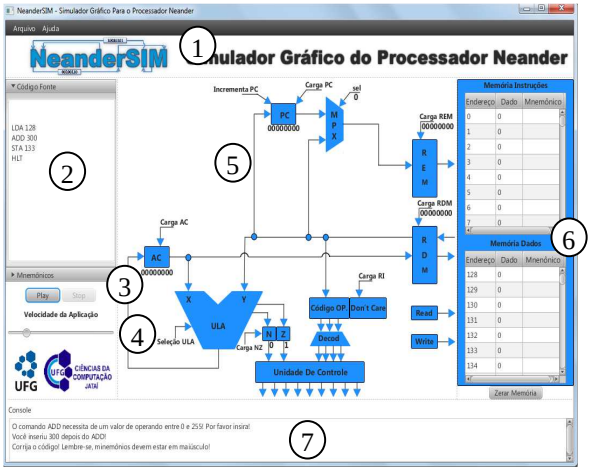

Figura 3. Interface do *NeanderSIM*.

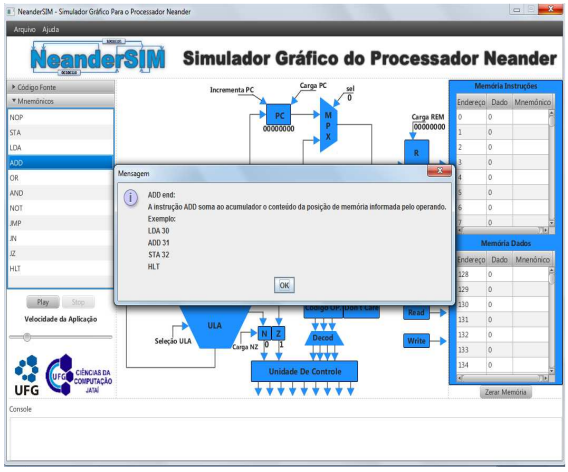

Figura 4. Texto Informativo e Exemplo.

Do lado esquerdo da Figura 3 (2), encontra-se o componente por meio do qual o usuário poderá selecionar as opções "Código Fonte" e "Mnemônicos". A opção "Código Fonte" exibe ao usuário uma área para construção do código em linguagem de montagem, com base nas instruções reconhecidas pelo computador *Neander*. A opção "Mnemônicos" apresenta a listagem de todas as instruções disponíveis no *Neander*. Ao clicar sobre uma destas instruções, abre-se uma janela com informações sobre esta instrução, bem como um exemplo de utilização da mesma (Figura 4).

[2] http:// paulojunior.jatai.ufg.br

Abaixo da área do código fonte e dos mnemônicos há dois botões para interação com a simulação, são eles “Iniciar” e “Parar” (Figura 3 - 3). Também pode-se visualizar um componente “*Slider*” (Figura 3 - 4), que permite regular a velocidade das animações durante o processo de simulação.

Na parte central da Figura 3 (5), tem-se a representação gráfica da arquitetura do computador *Neander*, mostrando os barramentos e componentes por onde os dados e instruções irão trafegar. Na parte direita da Figura 3 (6), tem-se a memória do computador. Nela estarão as instruções e os dados do programa do usuário; embora haja apenas uma memória no computador, para melhor entendimento, ela foi representada em duas: memória de instruções e memória de dados (essa é uma estratégia também adotada pelo autor da ferramenta *WNeander* – Weber, 2004).

Ao concluir a construção do código fonte e o preenchimento da memória com os dados iniciais do programa, o usuário poderá iniciar a simulação. Porém antes, será realizada uma verificação da sintaxe do código fonte informado pelo usuário. Havendo algum tipo de erro, o mesmo será informado com mensagens de texto localizadas em um componente denominado “Console” (Figura 3 - 7). O “Console” também é utilizado para manter o usuário informado sobre o que está ocorrendo durante a simulação do seu código fonte no computador. Esse é um recurso inovador da ferramenta *NeanderSIM*, com relação às demais ferramentas relacionadas.

B. Arquitetura

Outro ponto inovador da ferramenta *NeanderSIM* é sua arquitetura. Ela foi projetada em três módulos (camadas), com o intuito de facilitar futuras manutenções e incorporações de novas funcionalidades à ferramenta. Os módulos criados são (Figura 5): i) Módulo de Interface Gráfica; ii) Módulo Gráfico; e iii) Módulo Algorítmico.

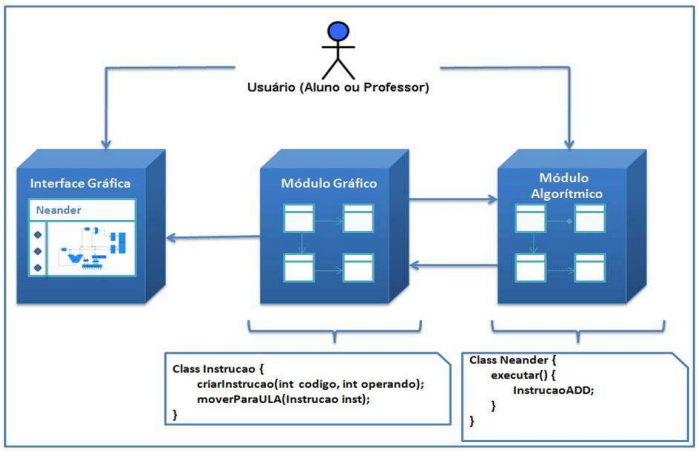

Figura 5. Arquitetura do *NeanderSIM*.

O **Módulo Interface Gráfica**, apresentado na Seção 3.A, representa a interface da ferramenta, por meio da qual o usuário poderá construir seu próprio código e simulá-lo no computador *Neander*. O **Módulo Gráfico** consiste em um conjunto de classes que permitem a criação de elementos gráficos utilizados pelos algoritmos do simulador, bem como a animação destes elementos pelos barramentos e componentes do computador *Neander*. Por exemplo, na Figura 5, observa-se a classe `Instrucao`, que possui métodos responsáveis pela: i) criação de uma instrução (`criarInstrucao(int codigo, int operando)`); e ii) movimentação dessa instrução para a unidade lógica aritmética deste

processador (`moverParaULA(Instrucao inst)`). Este método será responsável por movimentar, utilizando recursos de animação do *framework JavaFX*, a representação gráfica de uma instrução para a unidade lógica aritmética do computador *Neander*. O **Módulo Algorítmico** consiste de um conjunto de classes que representam os algoritmos responsáveis pela execução das instruções suportadas pelo *NeanderSIM*. Uma classe deste módulo, algorítmico, pode utilizar as classes do Módulo Gráfico para a execução gráfica de um determinado algoritmo. Por exemplo, na Figura 5, a classe `InstrucaoADD` é responsável pelo algoritmo que simula a execução de uma instrução do tipo ADD (adição) no computador *Neander*.

Com esse tipo de arquitetura, o *NeanderSIM* permite que seus usuários possam incrementar a sua funcionalidade, criando novos tipos de algoritmos, bastando ao usuário possuir conhecimentos prévios em *Java* e entender a arquitetura proposta no *NeanderSIM*. A vantagem é que os usuários poderão utilizar recursos gráficos e de animação disponíveis no *NeanderSIM* em seus algoritmos, sem a necessidade de conhecimento prévio de qualquer *API* ou *framework* para criação de elementos gráficos e animação. Por exemplo, supondo que o computador *Neander* foi aprimorado em uma nova edição do livro de Weber (Weber, 2004) e um novo tipo de instrução foi adicionado ao seu conjunto original, com a separação existente entre o Módulo Gráfico e o Módulo Algorítmico, o usuário poderá desenvolver e disponibilizar com maior facilidade, um algoritmo responsável pela simulação dessa nova instrução. Este recurso pode ser utilizado também para que o professor peça aos alunos, como trabalho prático da disciplina, que desenvolvam um determinado algoritmo para um tipo de instrução hipotético.

4. Avaliação

Para avaliação do *NeanderSIM*, o modelo de aceitação de tecnologia *TAM* (*Technology Acceptance Model*) (Davis *et al*., 1989) foi utilizado. Esse modelo possui como objetivo explicar o comportamento das pessoas no que diz respeito à aceitação de uma tecnologia ou sistema. O modelo *TAM* define dois constructos básicos (Davis *et al*., 1989): i) utilidade percebida, que mede o quanto uma pessoa acredita que usar determinada tecnologia aumenta seu desempenho no trabalho; e ii) facilidade de uso percebida, que mede o quanto uma pessoa acredita que o uso de determinada tecnologia é simples.

Além disso, modelo *TAM* sugere a criação de questionários, para os quais são atribuídas afirmações relacionadas à facilidade de uso e utilidade da tecnologia em análise. Para cada afirmação, o respondente poderá escolher uma dentre as seguintes opções “1 - Discordo Totalmente”, “2 - Discordo Fortemente”, “3 - Discordo Parcialmente”, “4 - Neutro”, “5 - Concordo Parcialmente”, “6 - Concordo Fortemente” e “7 - Concordo Totalmente”, conforme sua opinião sobre esta afirmação. Neste sentido, um usuário, após ter passado por um treinamento sobre a ferramenta *NeanderSIM* e a ter utilizado, pode mensurar qualitativamente seu “sentimento” de utilidade e facilidade de uso desta ferramenta, com base nas alternativas citadas anteriormente.

O questionário desenvolvido possui 26 (vinte e seis) questões, das quais, 3 (três) são para identificação dos usuários; 4 (quatro) correspondem ao treinamento dado aos usuários sobre o simulador *NeanderSIM*; 9 (nove) correspondem ao constructo

facilidade de uso percebida; e 10 (dez) ao constructo utilidade percebida. O questionário foi respondido por 18 (dezoito) usuários, dentre os quais, 17 (dezessete) eram alunos de graduação e 1 (um) era professor do curso de Bacharelado em Ciência da Computação da Universidade Federal de Goiás (Regional Jataí). As questões do questionário foram omitidas neste trabalho devido a limitações de espaço (mais detalhes podem ser encontrados em Ullmann (2013)). Contudo, ao longo desta seção, as questões mais relevantes são apresentadas com suas respectivas **notas finais**. As Notas Finais de cada Questão (*NFQ*) são obtidas por meio da Equação 1, onde: i) Q_i correspondente à quantidade de avaliadores que escolheram a alternativa *i* para essa questão; ii) *i* representa o valor dessa alternativa (número entre um e sete); e iii) *QU* consiste na quantidade total de avaliadores que responderam à questão. Sendo assim, o valor da nota final de uma determinada questão é um número entre 1.0 (um) e 7.0 (sete), sendo que, quanto mais próximo de sete, maior é a aceitação dos usuários para a questão em análise. Desse modo, a equação 1 determina a média ponderada para cada questão.

$$NFQ = \frac{\sum_{i=1}^{7} (Q_i * i)}{QU} \qquad \textbf{(1)}$$

Na Tabela 1 são apresentadas as notas de algumas questões do questionário obtidas por meio da equação 1, que são comentadas ao longo desta seção. Na Figura 6 estão as notas finais obtidas para cada constructo analisado, representadas pela média de todas as questões para cada constructo.

De acordo com a Figura 6, percebe-se que o constructo “Utilidade Percebida” obteve uma média alta (6.42), o que provê indícios de que a maioria dos avaliadores concordaram que o *NeanderSIM* possui utilidade no meio acadêmico. Evidência disso são as notas dadas para as questões 2 e 3 da Tabela 1, que receberam as maiores notas da avaliação. Quanto ao constructo “Facilidade de Uso Percebida”, mesmo tendo obtido média alta também (6.22), há indícios de que a usabilidade da ferramenta pode ser melhorada. Isso é evidenciado pelo fato de que a questão que recebeu menor nota final para esse constructo – questão 1 da Tabela 1 – abordava o seguinte: “os recursos de navegação do *NeanderSIM* estão todos claros e fáceis de achar”.

Tabela 1. Notas obtidas para algumas questões da avaliação.

#	Questão	Quantidade de pessoas que escolheram a opção:							
		1	2	3	4	5	6	7	Média
FACILIDADE DE USO									
1	Os recursos de navegação do *NeanderSIM* estão todos claros e fáceis de achar.	0	0	1	1	1	10	5	5,94
UTILIDADE PERCEBIDA									
2	Utilizar o *NeanderSIM* é importante e adiciona valor ao meu estudo/trabalho.	0	0	0	0	0	4	14	6,78
3	Usar o *NeanderSIM* pode aumentar meu desempenho durante os estudos ou ensino da disciplina de Arquitetura de Computadores.	0	0	0	0	0	4	14	6,78
4	Eu tenho a intenção de utilizar o *NeanderSIM* ao longo dos próximos semestres.	0	0	0	5	2	4	7	5,72
5	Eu recomendarei o uso do *NeanderSIM*.	0	0	0	0	0	4	14	6,78

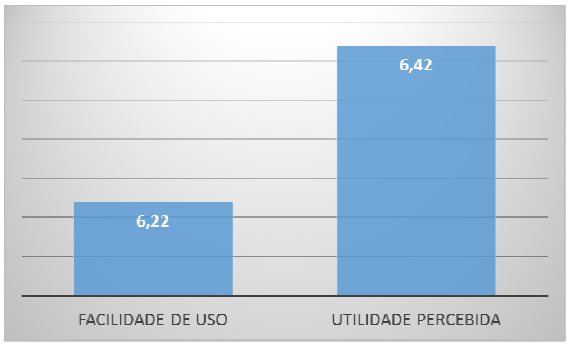

Figura 6. Médias das Notas dadas pelos Avaliadores para cada Constructo.

A nota mais baixa do constructo "Utilidade Percebida" está relacionada à questão que se refere à utilização do *NeanderSIM* ao longo dos próximos semestres pelos participantes, conforme a questão 4 da Tabela 1. Isso se deve ao fato de que a maioria dos participantes da avaliação já haviam cursado a disciplina "Arquitetura de Computadores", dessa forma, vários avaliadores marcaram a opção "Neutro". Em contrapartida, a questão 5 obteve a maior nota, juntamente com as questões 2 e 3, representando que os avaliadores têm a intenção de recomendar o uso do *NeanderSIM* para o ensino de "Arquitetura de Computadores".

5. Considerações Finais

O processo de ensino-aprendizagem dos conceitos da disciplina "Arquitetura de Computadores", apenas com a utilização de recursos didáticos pouco interativos, tais como lousa e giz, livros-textos, vídeo-aulas, *slides*, entre outros, pode ser inadequado, uma vez que essa disciplina trata de assuntos teóricos, abstratos e de característica dinâmica. A utilização de ferramentas educacionais, como o simulador *NeanderSIM*, pode facilitar a exposição dos conceitos dessa disciplina, já que não se faz mais necessário desenhar em lousa os componentes e os dados que simulam a execução de instruções em um computador. Ao concluir a análise dos dados coletados a partir da avaliação do *NeanderSIM*, notou-se que essa ferramenta obteve boa aceitação por parte dos avaliadores (alunos e professores do curso de Ciência da Computação), quanto à utilidade e facilidade de uso percebidas.

Algumas propostas de trabalhos futuros são: i) aprimorar o processo de animação do *NeanderSIM*, permitindo que o aluno controle o passo a passo da simulação, não apenas acelerando ou desacelerando a velocidade da animação; ii) criar animações para componentes internos do computador, por exemplo, apresentando como as operações aritméticas e lógicas são executadas nas portas lógicas da ULA (Unidade Lógica Aritmética); iii) permitir que o aluno possa salvar um código em linguagem de máquina e posteriormente carregá-lo no simulador; iv) permitir que o aluno possa gerar um relatório da execução de um determinado código fonte, no qual serão apresentados os passos para a execução deste código com *screenshots* do simulador capturados automaticamente (este material poderá servir para estudo em locais onde o aluno não possa executar o *NeanderSIM*); e v) replicar a avaliação, comparando o simulador desenvolvido com outras ferramentas de apoio ao ensino de arquitetura de computadores pré-existentes.

Referências

BORGES, J. A. S.; SILVA, G. P. (2006). “NeanderWin – um simulador didático para uma arquitetura do tipo Acumulador”. In: workshop sobre educação em arquitetura de computadores, 2006, Ouro Preto. SBC.

DAVIS, F.D., Bagozzi, R. P. and Warshaw P.R., “User Acceptance of Computer Technology: A Comparison of two Theoretical Models”. In: Management Science. v. 35, n. 8, p. 982-1003, 1989.

ELIAS, W. J.; SILVA, J. R. C. da; TIOLA, F. P. S. Simulador Muticiclo de Processador MPIS 32 bits para Apoio ao Estudo em Arquitetura de Computadores. Disponível em: http://seer.ufrgs.br/renote/article/view/21988/12756, 2011.

MAIA, D. W. N.; VIEIRA, M. M.; PESSOA R. F. PS-CAS MIPS: Um Simulador De Pipeline Do Processador MIPS 32 Bits Para Estudo De Arquitetura De Computadores. WORKSHOP SOBRE EDUCAÇÃO EM ARQUITETURA DE COMPUTADORES (WEAC), 2009.

MATTOS, M. M.; TAVARES, A. C.; FARIAS, J. S.; RENALDI, F. VXt: Um Ambiente Didático Para Ensino de Conceitos Básicos de Sistemas Operacionais e Arquitetura de Computadores, 2004. Disponível em: http://inf.unisul.br/~ines/workcomp/cd/.

ProcSIM. A Visual MIPS R2000 Processor Simulator – Freeware. Disponível em: http://jamesgart.com/procsim.

PUCRS, 2014. Plano de ensino da disciplina de Arquitetura de Computadores. Disponível em: http://www.inf.pucrs.br/flash/orgarq/programa.html. Acessado em: Março/2014.

REIS, F. P. ; PARREIRA JÚNIOR, P. A. ; COSTA, H. A. X. . TBC-SO/WEB: Um Software Educacional para o Ensino de Políticas de Escalonamento de Processos e de Alocação de Memória em Sistemas Operacionais. In: XX Simpósio Brasileiro de Informática na Educação (SBIE), 2009, Florianópolis/SC. Simpósio Brasileiro de Informática na Educação, 2009.

RODRIGUES, R.P.; MARTINS, C. A. P. S. Ensino e aprendizado de pipeline de modo motivante e eficiente utilizando simuladores didáticos. WORKSHOP SOBRE EDUCAÇÃO EM ARQUITETURA DE COMPUTADORES (WEAC), 2008.

SANTOS, R. P.; COSTA, H. A. X. Um Software Gráfico Educacional para o Ensino de Algoritmos em Grafos. In: INTERNATIONAL CONFERENCE IADIS (International Association for Development of the Information Society) CIAWI – CONFERÊNCIA IBERO AMERICANA WWW/INTERNET (CIAWI’2006), Murcia – Espanha. CD de Resumos e Anais da Conferência, 2006. v. 1.

SCHEMBERGER, E. E. ; ANDRADE, S. C.; ARAUJO, K. EKS-MIPS: Um Simulador para Processador MIPS. ENCONTRO NACIONAL DE DIFUSÃO TECNO LÓGICA (ENDITEC), 2010.

UFPE, 2014. Plano de ensino da disciplina de Arquitetura de Computadores. Disponível em: http://www.cin.ufpe.br/~acb/inf_hw/Infra_hw_LC1.htm. Acessado em: Março/2014.

UFRRJ, 2014. Plano de ensino da disciplina de Arquitetura de Computadores. Disponível em: http://cursos.ufrrj.br/grad/sistemas/ementas-e-bibliografias-das-disciplinas/. Acessado em: Março/2014.

USP, 2014. Plano de ensino da disciplina de Arquitetura de Computadores. Disponível em: http://wiki.icmc.usp.br/images/a/a8/SSC0510-Aula01.pdf. Acessado em: Março/2014.

UFRGS, 2014. Plano de ensino da disciplina de Arquitetura de Computadores. Disponível em: http://inf.ufrgs.br/index.php?option=com_content&view=article&id=531&Itemid=100. Acessado em: Março/2014.

UFSC, 2014. Plano de ensino da disciplina de Arquitetura de Computadores. Disponível em: http://posticsenasp.ufsc.br/files/2013/11/Descri%C3%A7%C3%A3o-das-Disciplinas-e-ementas.pdf. Acessado em: Março/2014.

ULLMANN, M. R. D. “NeanderSIM: Um Simulador Educacional de Apoio ao Ensino de Arquitetura de Coputadores”. Monografia de Graduação. Universidade Federal de Goiás. Regional Jataí. Jataí/GO, 2013.

WEBER, R. F. Fundamentos de Arquitetura de Computadores. 3. Ed. Porto Alegre: Sagra Luzzatto, 2004. 306 p. (Série Livros Didáticos do Instituto de Informática da UFRGS).