

DEG4Trees: Um Jogo Educacional Digital de Apoio ao Ensino de Estruturas de Dados

Weider Alves Barbosa¹, Isabella de Freitas Nunes², Ana Carolina Gondim Inocência², Thiago Borges de Oliveira², Paulo Afonso Parreira Júnior²

¹ Instituto de Informática, Universidade Federal de Goiás (UFG), Goiânia/GO, Brasil.

² Curso de Bacharelado em Ciência da Computação/Instituto de Ciências Exatas e Tecnológicas, Universidade Federal de Goiás/Regional Jataí, Jataí/GO, Brasil

weiderbarbosa@inf.ufg.br, isabelladefreitasnunes@gmail.com,
anacarolinaupt@ufg.br, thborges@ufg.br, paulojunior@jatai.ufg.br

***Abstract.** This paper presents a new educational game to help teaching of binary search tree and AVL tree data structures, called DEG4Trees (Digital Educational Game for Trees). The DEG4Trees is an interactive game where the student has to observe and act using the principles of these data structures to succeed. At the end of each round, the game provides a feedback to help the student and the teacher to recognize points of improvement on the training. A preliminary evaluation on this game gives us indications that it may be useful in the context of data structures learning.*

***Resumo.** Este artigo apresenta um jogo educacional digital, denominado DEG4Trees (Digital Educational Game for Trees), de apoio ao ensino de estruturas de Árvores Binárias de Busca e Árvores AVL. DEG4Trees é um jogo no qual o aluno precisa observar as propriedades das estruturas de dados e interagir com as mesmas para atingir seus objetivos. Ao final de cada rodada, DEG4Trees provê um feedback sobre a atuação do jogador, o que pode servir para que tanto o professor quanto o aluno reconheçam os pontos a serem melhorados durante o processo de ensino/aprendizagem. Como resultado de uma avaliação preliminar do jogo, percebeu-se que o mesmo pode ser útil no contexto do ensino de estruturas de dados.*

1. Introdução

Algoritmos e Estruturas de Dados (AED), ou apenas Estruturas de Dados (ED), corresponde a uma disciplina clássica e de suma importância para os cursos de Ciência da Computação e áreas afins. Nessa disciplina são apresentadas estruturas de programação utilizadas para resolver diversos tipos de problemas, como recuperação de informação, ordenação, entre outros [Baker *et al.*, 1999]. Algumas dessas estruturas estão ligadas a conceitos abstratos e a algoritmos complexos, como por exemplo, as estruturas de árvores de pesquisa. Uma árvore de pesquisa é uma estrutura na qual os dados ficam organizados de forma hierárquica e é uma solução eficiente para buscar, inserir ou remover elementos em determinado conjunto [Ziviani, 2007]. Exemplos de árvores de pesquisa são: Árvore Binária de Busca (ABB), Árvore AVL, Árvore Rubro-Negra, Árvore B, entre outras.

Sabe-se que a capacidade humana para compreender representações gráficas normalmente é maior do que a capacidade para compreender apenas representações textuais [Bryfczynski, 2013; Shabanah *et al.*, 2010]. Neste sentido, diversos trabalhos cujo enfoque está no ensino de ED com recursos computacionais gráficos e multimídias

podem ser encontrados, como por exemplo: (i) vídeos, tais como o *Sorting out sorting* [Baecker e Sherman, 1981]; (ii) ferramentas educacionais gráficas, tais como TBC/AED - Treinamento Baseado em Computador para Algoritmos e Estruturas de Dados [Santos e Costa, 2005a; Santos e Costa, 2005b], entre outros; e (iii) Jogos Educacionais Digitais (JEDs), como *Sortko* - Um jogo de apoio ao ensino de algoritmo de ordenação [Boticki *et al.*, 2012] e *Sortia* [Battistella, von Wangenheim e von Wangenheim, 2012].

Entretanto, apesar das duas primeiras alternativas (filmes e ferramentas educacionais digitais) utilizarem representações gráficas, há pouca ou nenhuma interação entre os alunos e os conceitos estudados na disciplina. Segundo [Battistella, von Wangenheim e von Wangenheim, 2012], essa falta de interação pode trazer os seguintes problemas: (i) levar o aluno a se distrair durante a execução de algum passo importante de um algoritmo, o que poderia levá-lo a não compreensão total do conceito abordado; e (ii) fazer com que o aluno se entedie e não volte a utilizar o recurso de aprendizagem.

A terceira alternativa apresentada para apoio ao ensino de ED consiste na utilização JEDs (ou *serious game* – jogos sérios), isto é, jogos implementados em um meio computacional ou eletrônico, que têm formato instrucional ou de aprendizagem, que envolvem competição e que possuem regras e restrições [Botelho, 2004]. Há diversos trabalhos na literatura com enfoque sobre a confecção de ferramentas educacionais de apoio ao ensino de estruturas de árvores, porém, segundo dados de um mapeamento sistemático da literatura, conduzido pelos autores deste trabalho [Barbosa e Parreira Júnior, 2013], há escassez de trabalhos com enfoque específico sobre o ensino de estruturas de árvores de pesquisa com o apoio de JEDs.

Sendo assim, o objetivo deste trabalho é apresentar um JED, denominado *DEG4Trees* (*Digital Educational Game “Four” Trees*), de apoio ao ensino de ED com enfoque nas estruturas de dados conhecidas como “árvores de pesquisa”, mais especificamente, Árvores Binárias de Busca (ABB) e Árvores AVL.

O restante deste artigo está organizado da seguinte forma: na Seção 2, há uma descrição dos principais trabalhos disponíveis na literatura que têm relação com o tema deste trabalho, destacando-se suas limitações e seus pontos fortes. Na Seção 3, o JED *DEG4Trees* é apresentado, destacando-se suas principais funções. Na Seção 4, os resultados obtidos por meio de avaliação realizada com alunos de graduação em computação, a respeito do *DEG4Trees*, são apresentados e discutidos. Por fim, na Seção 5, estão as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Os trabalhos analisados nesta seção foram escolhidos levando-se em consideração alguns critérios, tais como: (i) são trabalhos específicos para o ensino de ED; (ii) os trabalhos contemplam o tópico “árvores de pesquisa” ou são propostas de JEDs; e (iii) as ferramentas propostas nestes trabalhos são gratuitas e estão disponíveis para *download* na Internet.

Sortko [Boticki *et al.*, 2012] é um jogo educacional móvel de auxílio ao ensino de algoritmos de ordenação baseada na plataforma Android. Neste jogo, o aluno deve manipular uma sequência aleatória de números, conforme o algoritmo definido pelo método de ordenação, a fim de tornar essa sequência ordenada. O jogo apresenta como *feedback*, algumas mensagens de erro, quando algum passo é realizado incorretamente

pelo usuário. Além disso, textos de ajuda são apresentados ao usuário durante a execução do jogo para auxiliá-lo na ordenação da sequência. *Sortia* [Battistella, von Wangenheim e von Wangenheim, 2012] também consiste em um jogo de apoio ao ensino de métodos de ordenação, no qual o aluno pode, manualmente, manipular conjuntos de números inteiros com o intuito de aplicar conceitos relacionados aos algoritmos de ordenação.

Applet Binary Search Trees [Gogeshvili, 2007] é uma ferramenta educacional que apresenta suporte a diferentes tipos de árvores binárias, como AVL e Rubro-Negra. Ao executar tal ferramenta, disponível via *web*, o usuário seleciona o tipo de árvore desejado (AVL ou Rubro-Negra) e, a partir daí, é apresentada uma árvore preenchida com elementos escolhidos aleatoriamente. Após isto, é possível efetuar algumas operações nesta árvore, tais como inserir, buscar, remover um nó específico, remover todos os dados da árvore ou percorrer a árvore. *TBC-AED-Treinamento Baseado em Computador para Algoritmos e Estruturas de Dados* [Santos e Costa, 2005a, Santos e Costa, 2005b] é também uma ferramenta educacional que utiliza recursos de visualização para apresentar execução de algoritmos e de estruturas de dados. Com essa ferramenta, o aluno pode escolher o algoritmo ou a estrutura de dados desejada, tal como: (i) realizar busca em vetor; (ii) executar métodos de ordenação: *Selection Sort*, *Insertion Sort*, *Bubble Sort*, *Merge Sort* e *Quick Sort*; (iii) manipular estruturas de dados convencionais: lista, fila, pilha, listas simplesmente e duplamente encadeadas; e (iv) manipular árvores do tipo ABB. Após essa escolha, apresenta-se uma breve descrição do funcionamento (conceito) do algoritmo ou da estrutura de dados e o seu pseudocódigo. Além disso, é possível visualizar a execução das operações selecionadas de forma gráfica e animada.

Na Tabela 1 é apresentado um resumo das principais características das ferramentas citadas anteriormente, comparando-as por meio de alguns critérios.

Tabela 1. Análise dos Trabalhos Relacionados.

#	Crítérios	<i>Sortko</i>	<i>Sortia</i>	<i>Applet Binary Search Trees</i>	<i>TBC-AED</i>
1	Possui recursos de animação gráfica.	X	X	X	X
2	Permite execução via <i>web</i> .	--	X	X	X
3	Permite interação com os elementos da ferramenta.	X	--	--	--
4	Apresenta códigos ou algoritmos inerentes à estrutura de dados executada.	--	--	--	X
5	Apresenta explicação sobre os algoritmos.	--	--	--	X
6	Contempla o conteúdo de árvores ABB.	--	X	X	X
7	Contempla o conteúdo de árvores AVL.	--	X	X	--
8	Apresenta as operações mais comuns que podem ser encontradas em árvores (inserção, remoção e consulta).	--	X	X	X
9	Apresenta flexibilidade para inserção de novos algoritmos ou fases ao jogo.	--	--	--	--
10	Provê <i>feedback</i> ao aluno/professor quanto à correta utilização das estruturas.	X	??	??	??

Alguns desses critérios foram elencados com base nas características que se espera de uma boa ferramenta de apoio ao ensino [Von Wangenheim e Von Wangenheim 2012; Annetta 2010; Fernandes 2009], tais como os critérios 1, 3, 5, 9 e 10. O símbolo 'X' é utilizado quando a ferramenta em questão atende a um determinado critério, caso contrário é utilizado o símbolo '--'. Importante ressaltar que o critério "Provê *feedback* ao aluno/professor quanto à correta utilização das estruturas" só pode

ser analisado para ferramentas que permitem interação com seus elementos. Portanto, o símbolo ‘??’ foi utilizado para as ferramentas que não puderam ser analisadas por meio deste critério.

Como pode ser observado na Tabela 1, poucas ferramentas contemplam os critérios: (i) interação com os elementos da ferramenta; (ii) apresentação de códigos ou algoritmos inerentes à estrutura de dados executada; (iii) explicação dos algoritmos abordados; (iv) flexibilidade; e (v) *feedback*. Além dos cinco critérios acima citados, poucas são as ferramentas que contemplam os dois tipos de árvores apresentados neste trabalho, isto é, ABB e AVL.

O jogo desenvolvido neste trabalho (*DEG4Trees*) atende aos critérios da Tabela 1 da seguinte forma: (i) são contempladas as estruturas de árvore ABB e AVL; (ii) por se tratar de um JED, o aluno pode interagir com as estruturas de dados para completar os objetivos do jogo; (iii) são apresentados exemplos de códigos e algoritmos para que o aluno possa estudar no mesmo ambiente do jogo; (iv) o usuário (professor ou aluno) pode escolher o conjunto de entradas do jogo, garantindo assim, flexibilidade ao mesmo; e (v) o jogo provê *feedback* ao jogador quanto à correta utilização das estruturas disponíveis, a saber, as árvores ABB e AVL.

3. Digital Educational Game Four Trees (*DEG4Trees*)

O *DEG4Trees* possui duas versões: o *DEG4Trees-ABB* e o *DEG4Trees-AVL*. As Seções 3.1 e 3.2 apresentam os detalhes sobre o funcionamento de cada uma dessas versões do jogo, respectivamente. É importante salientar que *DEG4Trees* é um software livre, desenvolvido sobre a *engine Stencyl* (<http://www.stencyl.com/>), e está disponível para utilização e *download* em: <http://goo.gl/1CJTCK>.

3.1 *DEG4Trees-ABB*

Uma Árvore Binária de Busca (ABB) é um tipo de árvore que tem como objetivo principal dar suporte à operação de pesquisa em conjuntos de dados. Uma árvore binária é formada por um conjunto finito de nós, no qual há um nó principal, denominado raiz, e cada nó pode conter 0 (zero), 1 (uma) ou, no máximo, 2 (duas) subárvores binárias distintas [Cormen *et al.* 2002]. As principais propriedades de uma ABB, que a diferem de uma árvore binária genérica, é que todo nó nela contido, armazena um registro com uma chave que o identifica unicamente nesta estrutura e todo registro com valor de chave menor do que o valor da chave da raiz deve estar na subárvore esquerda do nó raiz e os registros com valores de chave maiores devem estar na subárvore direita. A Figura 1 apresenta três exemplos de árvores binárias, sendo que apenas a árvore da Figura 1 – B é uma ABB.

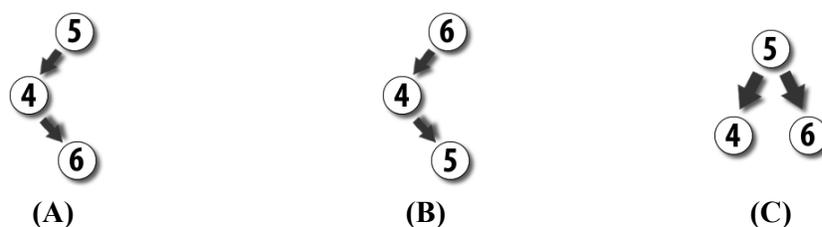


Figura 1. Exemplos de árvores binárias.

A árvore da Figura 1 – A não é uma ABB, pois o registro de chave igual a 6 está na subárvore esquerda do registro de chave igual a 5, onde deveriam estar apenas

registros com chaves menores do que 5. Para a versão *DEG4Trees-ABB*, buscou-se contemplar os conceitos de **busca**, **inserção** e **remoção** de elementos em uma ABB.

O contexto do jogo é o seguinte: o jogador deve fazer com que um submarino que se encontra no fundo do oceano e está impossibilitado de emergir, chegue à superfície em um tempo determinado. Para isso, ele terá como auxílio boias, que estão organizadas de forma a simular uma árvore binária (Figura 2). Essas boias estão, inicialmente, com “pouco ar”, sendo incapazes de fazer o submarino emergir. Para fazer com que o submarino chegue à superfície, o jogador deve inflar as boias com cilindros de ar que, neste contexto, são representados por números inteiros que caem ao mar de forma aleatória (não há relação entre o valor do número inteiro e a quantidade de ar do cilindro, o número pode ser entendido como um rótulo do cilindro de ar). Entretanto, os cilindros não podem ser utilizados de forma indiscriminada; a utilização dos mesmos precisa respeitar as propriedades de uma ABB.

Quando uma operação é realizada de forma incorreta, isto é, quando a propriedade da ABB é quebrada, a boia que o jogador selecionou continua sem ar, o jogador perde o cilindro de ar selecionado e o submarino, ao invés de subir, submerge ainda mais. Para inflar uma boia, o jogador deve “clique e arraste” o cilindro que foi jogado ao mar e soltá-lo sobre a boia referente à posição correta de inserção dessa chave, conforme as propriedades de uma ABB. A operação de “clique e arrastar” é apresentada na Figura 2, onde o número (-35) foi arrastado para raiz da árvore, as letras (A) e (B) desta figura, não fazem parte do jogo, foram colocadas para marcar o movimento do número (-35). Para atingir o objetivo do jogo, o aluno deve estar em contato com os conceitos de **consulta** e **inserção** de elementos em uma ABB, uma vez que para inserir uma chave, o aluno deve consultar a posição correta para a mesma.



Figura 2. Operação de inserção no *DEG4Trees-ABB*.

Quanto à operação de remoção, eventualmente, é escolhida aleatoriamente uma boia que armazena uma chave qualquer para se esvaziar, simulando um defeito na mesma. Quando isto ocorre, o jogador deve escolher uma boia cheia da árvore de boias para substituí-la. No entanto, a boia a ser escolhida deve estar em conformidade com o processo de **remoção** em uma ABB. Para selecionar a boia correta, o jogador deve apenas clicar sobre a mesma com o *mouse*. Cabe ressaltar que, em situações de remoção de uma boia, o jogador fica impossibilitado de manipular a estrutura até que a boia correta seja escolhida para substituir a outra com defeito. Assim, o desafio é resolver o

problema da remoção no menor tempo possível para que o jogador ainda tenha chance de fazer o submarino emergir.

O jogo *DEG4Trees-ABB* termina quando: (i) o jogador conseguiu fazer com que o submarino chegasse à superfície em tempo suficiente; ou (ii) o tempo para concluir o objetivo se esgotou. Ao término do jogo, uma mensagem de congratulação ou de apoio será exibida ao usuário. Neste ponto do jogo, o jogador será capaz de acessar o *log* do jogo, ou seja, as informações sobre as ações do jogador durante o jogo, explicitando seus acertos e erros. Esse é um recurso importante, tanto para o aluno, quanto para o professor. Por meio dele, o aluno pode descobrir quais foram seus erros e acertos ao longo do jogo e aprender com eles. O professor, por sua vez, pode utilizar o *log* gerado pelo jogo como *feedback* para descobrir em quais conceitos das estruturas de dados suportadas pelo *DEG4Trees* o aluno está com maior dificuldade.

3.2 *DEG4Trees-AVL*

Mesmo se preocupando em manter a organização hierárquica de seus elementos para facilitar a consulta dos mesmos, a *ABB* não se preocupa em garantir que a árvore seja balanceada, ou seja, que as duas subárvores de uma árvore qualquer possuam alturas iguais ou bem próximas. Altura é definida como a maior distância entre o nó raiz de uma árvore e seus nós folhas. Por exemplo, se fossem inseridos apenas elementos em ordem crescente em uma *ABB*, a estrutura ficaria degradada, com uma subárvore direita muito maior do que a subárvore esquerda. Essa situação é indesejada, pois compromete o tempo de busca, inserção e remoção de um elemento em uma árvore. Neste contexto, a estrutura de árvore *AVL* tem a preocupação de manter a árvore binária balanceada, por meio de operações de rotação, que garantem que os elementos fiquem distribuídos de forma equilibrada em suas subárvores [Ziviani, 2007].

O enfoque do *DEG4Trees-AVL* encontra-se sobre as rotações que devem ser realizadas para que a árvore *AVL*, ao sofrer algum tipo de desbalanceamento, possa ser rebalanceada. Por exemplo, a Figura 1 – C apresenta uma situação de balanceamento da árvore da Figura 1 – B, em que o neto mais à direita da subárvore esquerda do nó desbalanceado, neste caso o 6, torna-se raiz e o nó 6 torna-se filho à direita desta nova raiz; este procedimento é conhecido como “rotação dupla à direita (rdd)”. Em termos de conteúdo educacional, o *DEG4Trees-AVL* aborda todos os tipos de rotações necessários para balanceamento de uma árvore *AVL*. Devido à limitação de espaço, detalhes sobre as rotações possíveis em uma árvore *AVL* foram omitidos deste artigo; mais detalhes sobre elas podem ser encontrados em [Barbosa, 2014].

Nesta versão do jogo, o jogador é um garçom novato de um restaurante, que precisa equilibrar uma pilha de copos que se encontra sobre a raiz de uma árvore *AVL*. Os copos equilibrados podem cair quando a árvore sofre um desbalanceamento, fazendo com que a bandeja de copos se incline para a direita ou para a esquerda. Há ainda a figura de um vilão, responsável por inserir chaves na árvore de modo a torná-la desbalanceada e fazer com que os copos da bandeja caiam no chão.

Na Figura 3 é possível encontrar alguns detalhes inerentes ao funcionamento do *DEG4Trees-AVL*. Os nós da árvore *AVL* são representados pelas mesas do ambiente (Figura 3 - E). Ainda na Figura 3 - E é possível observar a ilustração da chave que é inserida de tempos em tempos na árvore pelo vilão do jogo. Como já foi dito, o objetivo do vilão é fazer com que a árvore fique desbalanceada, a fim de que os copos empilhados venham a cair no chão.

As letras existentes na Figura 3 não fazem parte do jogo e foram adicionadas visando a facilitar o entendimento do que representa cada elemento do jogo. Na Figura 3 (A), pode-se observar a pontuação atual do jogador, neste caso, (10) o sistema de pontuação explicado a seguir. A cada acerto de uma rotação a ser realizada para que a árvore se torne novamente balanceada são somados 100 (cem) pontos, se a rotação for do tipo simples, e 150 (cento e cinquenta) pontos, caso a rotação seja do tipo dupla (uma rotação dupla exige que duas rotações simples sejam executadas consecutivamente para tornar a árvore balanceada) - esta foi a forma encontrada de recompensar o jogador quando ações relativamente mais complexas são executadas.

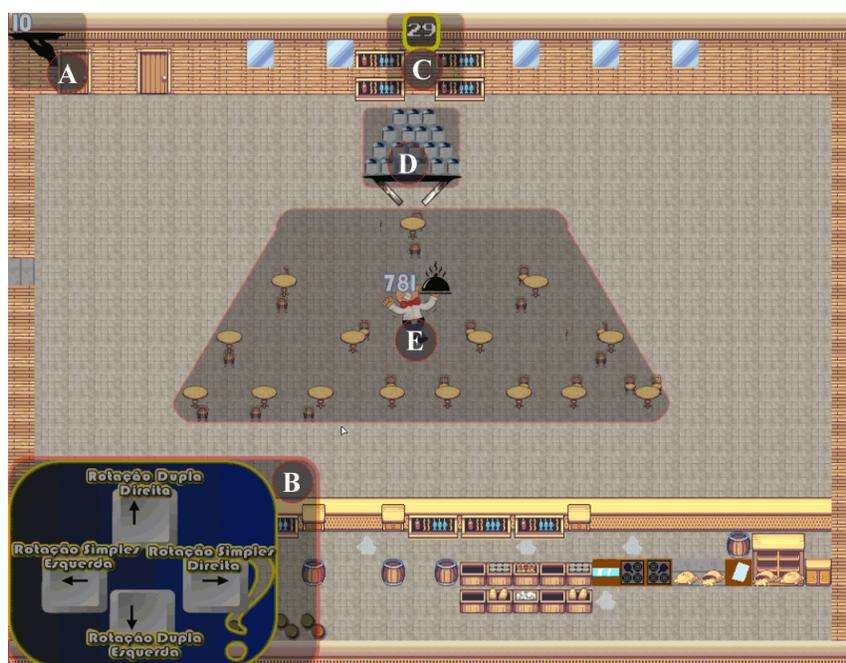


Figura 3. Simulação de uma AVL pelo DEG4Trees-AVL.

Quando o jogador erra na escolha da rotação a ser executada para rebalancear a árvore, 50 (cinquenta) pontos são decrementados da pontuação total do jogador. A cada erro do jogador, a bandeja na raiz da árvore AVL (Figura 3 - D) é desequilibrada e alguns copos podem cair. Ao final do jogo, a quantidade de copos existente sobre a bandeja é multiplicada pela quantidade de pontos obtidos pelo usuário durante o jogo. Sendo assim, quanto mais copos sobrarem, melhor. Após o término de fase no *DEG4Trees-AVL* são apresentados os pontos obtidos e o jogador é redirecionado para uma tela na qual ele pode escolher se deseja sair, jogar novamente ou se deseja analisar seu desempenho por meio dos *logs* gerados pelo jogo.

Importante ressaltar que o *DEG4Trees*, em ambas as versões, é flexível a ponto de permitir que o usuário determine qual conjunto de dados ele deseja utilizar como entrada no jogo. Esse recurso é importante, pois o professor pode, por exemplo, exercitar alguns cenários que acredite ser conveniente em um determinado momento da disciplina.

4. Avaliação

Para avaliação do *DEG4Trees*, 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, o modelo *TAM* sugere a criação de questionários, para os quais são atribuídas afirmações relacionadas à facilidade de uso e à 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.

O questionário desenvolvido possui 19 (dezenove) questões, das quais, 9 (nove) correspondem ao constructo facilidade de uso percebida e 10 (dez) ao constructo utilidade percebida (Tabela 2). O questionário foi respondido por 12 (doze) alunos de graduação do curso de Bacharelado em Ciência da Computação da Universidade Federal de Goiás (Regional Jataí).

Para cada questão foi calculada uma Nota Final (*NF*), que é uma média ponderada obtida por meio da Equação 1, onde: (i) Q_i corresponde 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 *NF* 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 avaliadores para a questão em análise. Na Tabela 2 são apresentadas as questões do questionário confeccionado neste trabalho, bem como as notas de cada questão, obtidas por meio da Equação 1.

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

De acordo com esta tabela, percebe-se que o constructo “Utilidade Percebida” obteve uma média geral alta (6,48), com baixo desvio padrão (0,31), o que provê indícios de que a maioria dos avaliadores concorda que o *DEG4Trees* possui utilidade no meio acadêmico. Este fato pode ser evidenciado observando as notas dadas para as questões relacionadas a este constructo, uma vez que todas as notas são maiores ou iguais a 6,0 (seis). A questão UP8 se destaca, pois obteve 12 (doze) notas iguais a 7,0 (sete), ou seja, todos os avaliadores “recomendam o uso do *DEG4Trees*”. Isso pode ser explicado pela novidade explorada pelo *DEG4Trees*, uma vez que há carência de JEDs para ensino de ED, principalmente, relacionados a estruturas de árvores. Destaca-se também a questão UP6, que diz respeito à versão *DEG4Trees-AVL* e que obteve a menor nota (6,0) dentre todas as questões desta categoria. Mesmo sendo uma nota alta, faz-se necessária uma análise mais profunda com o intuito de esclarecer quais situações desta versão não estão em conformidade com o que se espera de uma ferramenta para ensino deste tipo de conteúdo.

Quanto ao constructo “Facilidade de Uso Percebida”, ele também obteve uma média geral alta (6,26), com baixo desvio padrão (0,32). Destaca-se a nota recebida para a questão FU3 (6,83), que diz se o avaliador acredita que usar o *DEG4Trees* é uma boa

ideia. Acredita-se que esta questão teve boa aceitação pelos avaliadores pelo fato de ser difícil encontrar jogos voltados para apoiar o processo de ensino/aprendizado de ED.

Tabela 2. Notas obtidas para algumas questões da avaliação.

FACILIDADE DE USO									
#	Questão	Quant. de escolhas da opção:							Nota Final (NF)
		1	2	3	4	5	6	7	
FU1	Eu gostei de trabalhar com o <i>DEG4Trees</i> .	0	0	0	0	3	7	2	5,92
FU2	O acesso ao <i>DEG4Trees</i> é simples.	0	0	0	0	0	7	5	6,42
FU3	Usar o <i>DEG4Trees</i> é uma boa ideia.	0	0	0	0	0	2	10	6,83
FU4	No <i>DEG4Trees</i> eu sempre sei onde estou e como chegar aonde quero chegar.	0	0	0	0	1	8	3	6,17
FU5	Os recursos de manipulação do <i>DEG4Trees</i> estão todos claros e fáceis de achar.	0	0	0	0	0	7	5	6,42
FU6	Minha interação com <i>DEG4Trees</i> é clara e compreensível.	0	0	0	0	0	8	4	6,33
FU7	No <i>DEG4Trees</i> , é fácil encontrar a informação que desejo, com relação aos conceitos de ABB e AVL.	0	0	0	0	0	7	5	6,42
FU8	<i>DEG4Trees-ABB</i> possui visual/interface atraente.	0	0	0	0	2	7	3	6,08
FU9	<i>DEG4Trees-AVL</i> possui visual/interface atraente.	0	0	0	0	5	5	2	5,75
Média; Desvio Padrão									6,26; 0,32
UTILIDADE PERCEBIDA									
#	Questão	Quant. de escolhas da opção:							Nota Final (NF)
		1	2	3	4	5	6	7	
UP1	Utilizar o <i>DEG4Trees</i> é importante e adiciona valor ao meu estudo/trabalho.	0	0	0	0	0	8	4	6,33
UP2	O <i>DEG4Trees</i> é útil no processo de ensino/aprendizado dos conceitos sobre as árvores de pesquisa (ABB e AVL).	0	0	0	0	0	7	5	6,42
UP3	Usar o <i>DEG4Trees</i> pode aumentar meu desempenho durante os estudos ou ensino de Estruturas de Dados.	0	0	0	0	0	4	8	6,67
UP4	O <i>DEG4Trees</i> pode facilitar a realização do meu estudo/trabalho.	0	0	0	0	0	3	9	6,75
UP5	O <i>DEG4Trees</i> produz os resultados que espero de uma ferramenta de suporte ao ensino de ABB.	0	0	0	0	3	5	4	6,08
UP6	O <i>DEG4Trees</i> produz os resultados que espero de uma ferramenta de suporte ao ensino de Árvores AVL.	0	0	0	0	3	6	3	6,00
UP7	Eu pretendo integrar o <i>DEG4Trees</i> à minha rotina de trabalho/estudo.	0	0	0	0	1	3	8	6,58
UP8	Eu recomendaréi o uso do <i>DEG4Trees</i> .	0	0	0	0	0	0	12	7,00
UP9	Os conceitos das árvores de pesquisa binárias (ABB e Árvore AVL) relacionados à execução das operações foram abordados por completo no <i>DEG4Trees</i> .	0	0	0	0	1	7	4	6,25
UP10	Os conceitos das árvores de pesquisa binárias (ABB e Árvore AVL) foram abordados corretamente no <i>DEG4Trees</i> .	0	0	0	0	1	2	9	6,67
Média; Desvio Padrão									6,48; 0,31

Outro fato a ser destacado nesta tabela é a nota obtida pela questão FU9 (5,75), alusiva à aceitação do visual/interface do *DEG4Trees*. Esta questão recebeu a menor nota dentre todas as questões e por ter sido avaliada dessa forma, acredita-se que a interface do *DEG4Trees-AVL* pode não estar deixando claro para os alunos o que está acontecendo no processo do jogo, tendo em vista os valores apresentados nas questões UP6 e FU9.

5. Conclusão e Trabalhos Futuros

Neste trabalho foi apresentado o Jogo Educacional Digital (JED) denominado *DEG4Trees*, que visa apoiar o ensino de estruturas de dados conhecidas como “árvores

de pesquisa”, mais especificamente ABB e Árvore AVL. Um dos recursos inovadores do *DEG4Trees* consiste na flexibilidade disponibilizada pelo mesmo, ou seja, a possibilidade de inserir conjuntos de dados personalizados como entradas para as fases do jogo. Outra característica interessante do *DEG4Trees* é a geração automática de *feedbacks* para os jogadores, quanto à utilização das estruturas de dados contempladas pelo jogo.

Como trabalhos futuros, pretende-se incorporar no *DEG4Trees* outras árvores de busca, como Árvore Rubro-Negra e Árvore B. Outras estruturas de dados tradicionais como Listas, Pilhas e Filas também podem ser incorporadas no jogo, para ampliar a aplicabilidade do mesmo às disciplinas de ED. O fato da quantidade pessoas que avaliaram o jogo ter sido baixa, fez com que alguns pontos não pudessem ser mais bem analisados. Assim sendo, como trabalhos futuros espera-se também aumentar esse número de avaliações para que se obtenha um conjunto de dados mais significativo.

Referências

- Annetta, L. (2010) The “I’s” have it: A framework for serious educational game design. *Review of General Psychology*, v. 14, n. 2, p. 105-112.
- Baker, R. S.; Boilen, M.; Goodrich, M. T.; Tamassia, R.; Stibel, B. A. (1999) “Testers and visualizers for teaching data structures”. In: *ACM SIGCSE Bulletin*, 31(1), 261-265.
- Baecker, R.; Sherman D. (1981) “Sorting out sorting: 30 minute colour sound film”. *Dynamic Graphics Project*, University of Toronto, excerpted and reprinted in *SIGGRAPH Video Review* 7,1983.
- Barbosa, W. A. “DEG4Tree: Um Jogo Educacional Digital de Apoio ao Ensino de Estruturas de Dados”. *Monografia de Graduação. Ciência da Computação. UFG/Regional Jataí*. 2014.
- Barbosa, W. A. e Parreira Júnior, P. A. (2013). “Um Mapeamento Sistemático sobre Ferramentas de Apoio ao Ensino de Algoritmo e Estruturas de Dados”. In: *Anais do SBIE*.
- Battistella, P. E.; von Wangenheim, A.; von Wangenheim, C. G. (2012) “SORTIA-Um Jogo para Ensino de Algoritmo de Ordenação: Estudo de caso na Disciplina de Estrutura de Dados”. In: *Anais do SBIE*.
- Botelho, L. (2004) “Jogos educacionais aplicados ao *e-learning*”. Disponível em: <http://www.elearningbrasil.com.br/news/artigos/artigo_48.asp> Acesso em Abril de 2015.
- Boticki, I.; Barisic, A.; Martin, S.; Drljevic, N. (2012) “Sortko: Learning Sorting Algorithms with Mobile Devices”. In: *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE)*.
- Bryfczynski, S. *et al.* (2013) “Teaching data structures with BeSocratic”. In: *Proceedings of 44th ACM Technical Symposium on Computer Science Education*.
- Cormen, T. H. Leiserson, C. E., Stein, C.; Rivest, R. L. (2002) “Algoritmos: teoria e prática”. Rio de Janeiro : Elsevier – 6ª Reimpressão. Páginas 159 à 170-204 à 205, 349, 360.
- Davis, F. D.; Bagozzi, R. P.; Warshaw, P. R. (1989) "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models". *Management Science*, 35, 982-1003.
- Fernandes, A. M. D. R., *et al.* (2009). “Jogos eletrônicos: mapeando novas perspectiva”.
- Gogeshvili, A. (2007) “Applet Binary Search Trees”. University of LA Laguna, 2007. Disponível em: <<http://www.qmatica.com/DataStructures/Trees/BST.html>>. Acesso em Abril de 2015.
- Lawrence, R. (2004) “Teaching data structures using competitive games”. In: *IEEE Transactions*, 47(4), 459-466.
- Michail, A. (1996) “Teaching binary tree algorithms through visual programming”. In: *Visual Languages*”. *Proceedings, IEEE Symposium on. IEEE*, 1996. p. 38-45.
- Santos, R. P.; Costa, H. A. X. (2005a) “TBC-AED: Um Software Gráfico para Apresentação de Algoritmos e Estruturas de Dados aos Iniciantes em Computação e Informática”. *I Congresso de Computação do Sul do Mato Grosso - COMPSULMT. Rondonópolis, MT, Brasil*.
- Santos, R. P.; Costa, H. A. X. (2005b). “Desenvolvimento de um Software Educacional para o Ensino de Algoritmos, Estruturas de Dados e Programação”. *Projeto TBC-AED. CD de Resumos I Congresso de Extensão, Lavras, MG, v. 1, p. 113-113*. Disponível em: <<http://www.cos.ufrj.br/~rps/ic/2005.html>>. Acesso em Abril de 2015.
- Shabanah, S. S., Chen, J. X., Wechsler, H., Carr, D., & Wegman, E. (2010, April). “Designing computer games to teach algorithms”. In *7th Internat. Conf. on Information Technology: New Generations*.
- Von Wangenheim, G. C.; Von Wangenheim, A. (2012) “Ensinando Computação com Jogos.” 1. ed. Florianópolis: Bookess, 121p.
- Ziviani, N. (2007). “Projeto de Algoritmos com implementação em Java e C++.” São Paulo: Thomson, 1ª ed.