

# Ensino de Provas por Indução em Grafos utilizando uma Ferramenta Visual de Algoritmos

Frederico A. Carvalho, Flávio F. Borges, Gecirlei F. Silva,  
Thiago O. Borges, Esdras L. Bispo Jr.

Universidade Federal de Goiás

fredericoassiscarvalho@gmail.com, flavio@ufg.br, gecirlei@yahoo.com.br, thborges@gmail.com, bispojr@ufg.br

## ABSTRACT

In order to improve the teaching-learning process, new methodologies are proposed nowadays. Algorithm visualization tools (AVT) are used to view the step-by-step of algorithms in Graph Theory (GT). However, there is a lack of studies that propose an AVT as a methodological resource for teaching induction proofs in GT. This paper presents the *GraphViewer* AVT for this purpose. The adopted methodology consisted in the creation of this AVT and its use in a controlled environment. Questionnaires were applied before and after the experiment. After completing the analysis, it verified the acceptance of the tool by the participants of the experiment, in terms of usability, with a total of 96% positive evaluations.

## RESUMO

Com o intuito de aperfeiçoar o processo de ensino e aprendizagem em Computação, novas metodologias vêm sendo propostas. Como auxílio a estas, ferramentas de visualização de algoritmos (FVA) estão sendo utilizados para visualizar o passo-a-passo de sua execução em Grafos. Entretanto, há carência de trabalhos que proponham uma FVA como recurso metodológico para o ensino de provas por indução em Grafos. Este trabalho apresenta a FVA *GraphViewer* para este fim. A metodologia adotada consistiu na criação deste FVA e seu uso em um ambiente controlado. Questionários foram aplicados antes e após a realização do experimento. Após concluída a análise, concluiu-se que houve aceitação da ferramenta por parte dos participantes do experimento, em termos de usabilidade, com um total de 96% de avaliações positivas.

## Descritor de Categorias e Assuntos

#10003533 [Social and Professional Topics]: Computer Science Education.

## Termos Gerais

Human Factors

## Palavras-chave

Ensino, Computação, Grafos, Demonstração.

## 1. INTRODUÇÃO

A Educação de Computação (EC) é uma área fruto do encontro da Educação com a Ciência da Computação e outros campos de pesquisa [7][21]. Ela endereça várias questões vivenciadas por profissionais de EC, tais como (i) métodos de ensino [15], (ii) avaliação e construção de currículo [11], e (iii) animação, visualização e simulação de algoritmos [31][10].

Em relação aos métodos de ensino em EC, novas metodologias e métodos vêm sendo propostos [13]. O intuito é aperfeiçoar o

processo de ensino e aprendizagem em EC através do uso destes. Muitas destas abordagens têm como eixo norteador a aprendizagem ativa dos estudantes [9], ao invés do formato tradicional de ensino.

Como instrumento auxiliar destes métodos, ferramentas de visualização de algoritmos (FVA) vêm sendo propostas [29][10]. Estas auxiliam na visualização do passo-a-passo da execução de algoritmos por meio de algum recurso gráfico. Um dos objetivos do uso das FVAs em EC é a facilitação das atividades de abstração dos estudantes na compreensão de conceitos complexos [14].

Na disciplina de Teoria dos Grafos (TG), várias FVAs foram propostas [22][16][3]. Estas FVAs são utilizadas normalmente em ensino de algoritmos clássicos em TG (e.g. busca em largura e Bellman-Ford). Geralmente tanto o pseudocódigo, quanto o grafo, são apresentados simultaneamente durante a execução do algoritmo.

Porém uma das dificuldades encontradas por estudantes de TG está no momento de fornecer demonstrações a teoremas. Dificuldades estas que também são comuns a outros estudantes da área de exatas [18][4]. Grande parte dos alunos, por exemplo, sente dificuldades em visualizar e utilizar de forma correta a demonstração por indução [25].

Tendo em vista a carência de trabalhos que proponham uma FVA como recurso metodológico para o ensino de provas por indução em TG, é proposto a FVA *GraphViewer* para este fim. Desta forma, este trabalho apresenta a *GraphViewer* e o seu uso no ensino e aprendizagem de demonstração por indução em TG. É proposto um método de ensino e aprendizagem com intuito de auxiliar os alunos que tenham dificuldades no entendimento e compreensão tanto da técnica de demonstração por indução, quanto dos conceitos abordados na disciplina.

O restante do trabalho é dividido como se segue. Serão apresentados os trabalhos relacionados (Seção 2), alguns conceitos fundamentais (Seção 3), uma descrição sucinta da FVA *GraphViewer* (Seção 4), a metodologia (Seção 5), os experimentos e resultados (Seção 6) e, por fim, as considerações finais (Seção 7).

## 2. TRABALHOS RELACIONADOS

Foram listados alguns trabalhos relacionados em educação de TG utilizando FVAs. A FVA Rox [22] possibilita a criação e manipulação de grafos, bem como a execução de algoritmos clássicos. A FVA JAVENGA [1] permite a manipulação e edição de grafos direcionados e não-direcionados, além de permitir a inclusão de algoritmos externos para serem visualizados passo-a-passo. A FVA A-Graph [16] é bem semelhante ao Rox, com a

distinção de executar outros algoritmos, exportar os grafos e salvar a geometria e topologia destes. A FVA FURB Graphs [3] permite que o usuário possa compreender os algoritmos de busca em largura, busca em profundidade e Dijkstra de forma visual e interativa. É importante ressaltar que nenhuma destas citações referem-se ao ensino de demonstrações em TG, o qual é o foco deste trabalho.

O trabalho mais fortemente relacionado são os resultados apresentados nesta tese [20]. O autor identificou diversas fontes de dificuldades de estudantes de Computação com provas por indução. Além disto, foi construído um *e-book* interativo para auxiliar o ensino do assunto. Entretanto, pelo fato do *e-book* não apresentar visualizações de algoritmos, ele não pode ser considerado uma FVA. Embora o mesmo apresente visualizações gráficas de várias estruturas matemáticas recursivas importantes para o ensino de indução (não restritas ao ensino de TG).

### 3. FUNDAMENTOS

#### 3.1 Visualização de Algoritmos

A visualização de algoritmos (VA) ocupa um papel importante na EC [8]. A VA consiste basicamente em visualizar, através de recursos gráficos ou animações, o fluxo de execução de um algoritmo. Professores e estudantes normalmente se apropriam deste recurso durante o processo de ensino e aprendizagem.

Bons AVs “dão vida” aos algoritmos, representando graficamente seus vários estados e animando as transições entre estes [23]. Eles representam as estruturas de dados de forma natural, abstraindo os caminhos ao invés de focar em endereços de memória e chamadas de função.

É interessante salientar que a VA tem uma relação direta com o construtivismo. A teoria de aprendizagem construtivista reivindica que o conhecimento é ativamente construído pelo estudante [2]. Cada estudante constrói suas estruturas cognitivas diferentemente de outro estudante. Estas diferenças estão associadas a diversos fatores tais como o conhecimento pré-existente do estudante, estilo de aprendizagem ou até traços de personalidade.

Desta forma, a VA contribui como um ferramental auxiliar para que o estudante construa efetivamente o seu conhecimento. Durante o processo de refinamento do modelo mental, o aluno pode estabelecer relações entre os conceitos abstratos e suas instâncias concretas por meio da VA. Entretanto, é necessário garantir o uso adequado da VA para que ela cumpra apropriadamente com os propósitos pedagógicos em questão.

Existe uma iniciativa que congrega várias iniciativas em VA. O Portal AlgoViz<sup>1</sup> é uma comunidade aberta para pesquisadores, professores e usuários de FVAs. O portal fornece funcionalidades à comunidade como fóruns, relatórios de campo, classificações de ferramentas, entre outros. Cada FVA incluída no portal é descrita pelos seus criadores, apresentando como ela pode ser utilizada em sua respectiva disciplina [24].

#### 3.2 Teoria dos Grafos

A Teoria dos Grafos (TG) tem um número considerável de aplicações [5]. Campos na Ciência da Computação, Engenharias, Física e Cartografia são fortemente beneficiados do uso da TG no auxílio de soluções de seus problemas. Podemos citar, como exemplos, problemas de roteamento de pacotes em redes de computadores e problemas de coloração de mapas planos.

Apresentamos a seguir algumas definições básicas em TG [6].

**Definição 1. Grafo.** Seja  $V$  um conjunto e seja  $V^{(2)}$  o conjunto de todos os pares não-ordenados de elementos de  $V$ . Um grafo é o par  $(V, A)$  de forma que  $A \subseteq V^{(2)}$ .

**Definição 2. Vértice e Aresta.** Seja  $G = (V, A)$  um grafo. Dizemos que  $v$  é vértice de  $G$  se  $v \in V$ . Dizemos que  $\{u, v\}$  (ou simplesmente  $uv$ ) é aresta de  $G$  se  $\{u, v\} \in A$ .

**Definição 3. Grau de Vértice.** Seja  $v$  um dos vértices do grafo  $G = (V, A)$ . Denota-se por  $d(v)$  o grau de  $v$ , sendo definido como o tamanho do conjunto  $\{x \mid xv \in A\}$ .

Podemos dar como exemplos os grafos  $G = (V_G, A_G)$  e  $H = (V_H, A_H)$  apresentados na Figura 1. Os vértices de  $G$  são todos os elementos do conjunto  $V_G = \{a, b, c\}$  e os vértices de  $H$  são todos os elementos do conjunto  $V_H = \{x, y, z, t\}$ .  $ab$  é uma aresta de  $G$  e  $zt$  é uma aresta de  $H$ . O grau de  $b$  é 1 em  $G$ , i.e.,  $d_G(b) = 1$ . E o grau de  $t$  é 2 em  $H$ , i.e.,  $d_H(t) = 2$ .

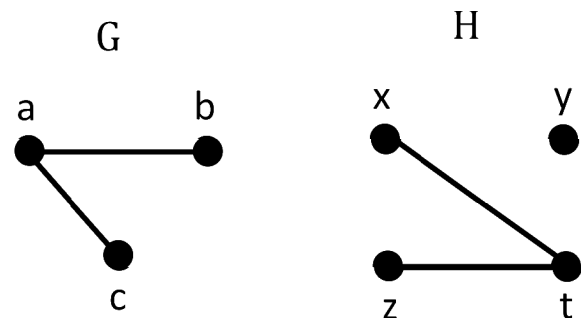


Figura 1. Dois grafos de exemplos:  $G$  e  $H$ .

Oriundas destas, apresentamos a seguir outras definições utilizadas neste trabalho [6].

**Definição 4. Grafo Conexo.** Seja o grafo  $G = (V, A)$ .  $G$  é conexo se, para todo  $u, v \in V$ , existir um caminho que liga  $u$  e  $v$ .

**Definição 5. Ponte.** Seja uma aresta  $uv$  do grafo  $G = (V, A)$  e seja o grafo  $H = (V, A \setminus \{uv\})$ . Dizemos que  $uv$  é uma ponte de  $G$  se não existir um caminho que liga os vértices  $u$  e  $v$  em  $H$ .

**Definição 6. Árvore.** Dizemos que um grafo conexo é uma árvore se todas as suas arestas são pontes.

Para exemplificar estas definições (ver Figura 1), percebe-se  $G$  como um grafo conexo (enquanto  $H$  não é). Tanto a aresta  $ac$  quanto a aresta  $xt$  são pontes em seus respectivos grafos. Entretanto, apenas  $G$  é árvore, pois  $H$  não é conexo (embora todas as suas arestas sejam pontes).

A partir das definições anteriores, alguns corolários e teoremas podem ser obtidos. Destes, dois teoremas clássicos em TG são descritos a seguir.

**Teorema 1.** Em um grafo, a soma dos graus de seus vértices é o dobro do número de suas arestas.

**Teorema 2.** Toda árvore com  $n$  vértices tem  $n-1$  arestas (para valores de  $n > 1$ ).

Com o propósito de garantir a veracidade de afirmações não-triviais (tais como os Teoremas 1 e 2), surge a necessidade de

<sup>1</sup> <http://algoviz.org/>

efetuar demonstrações. Na seção seguinte, serão apresentadas algumas técnicas de demonstrações.

### 3.3 Técnicas de Demonstração

As demonstrações matemáticas são ferramentas muito utilizadas por áreas afins da Matemática. Estas fazem parte de todo o contexto da produção de conhecimento de Matemática, pois, difere de outras áreas do conhecimento que utilizam observação e experimentação para provar suas verdades. A Matemática concebe algo como verdade, quando isso pode ser demonstrado [28].

Em TG, as demonstrações são de grande importância. Elas são utilizadas para poder provar a veracidade de teoremas, corolários e lemas da área. Existem várias técnicas de demonstrações utilizadas em TG. Dentre as quais, podemos citar (i) a demonstração direta, (ii) a demonstração por absurdo, e (iii) a demonstração por indução [26].

Serão apresentadas a seguir outras definições e teoremas sobre grafos. Estes serão utilizados como exemplos na explicitação das técnicas de demonstrações percorridas nas seções seguintes.

**Definição 7. Grafo  $k$ -regular.** Dizemos que um grafo é  $k$ -regular se todos os seus vértices têm grau  $k$ .

**Teorema 3.** Para cada número par  $n$  maior do que 2, existe um grafo 3-regular com  $n$  vértices.

**Teorema 4.** Em um grafo qualquer, o número de vértices de grau ímpar é necessariamente par.

#### 3.3.1 Demonstração Direta

Algumas afirmações matemáticas dizem respeito à existência de um tipo de objeto. Um caminho de demonstrar a veracidade desta afirmação é apresentando a construção deste objeto. Esta técnica é conhecida também como prova por construção.

Como exemplo, a demonstração direta do Teorema 3 será mostrada a seguir.

*Prova:* Seja  $n$  um número par maior que 2. Construa o grafo  $G = (V, A)$  com  $n$  vértices da seguinte forma. O conjunto de vértices de  $G$  é  $V = \{0, 1, \dots, n-1\}$ , e o conjunto de arestas de  $G$  é o conjunto

$$A = \{\{i, i+1\} \mid \text{para } 0 \leq i \leq n-2\} \cup \{\{n-1, 0\}\} \\ \cup \{\{i, i+n/2\} \mid \text{para } 0 \leq i \leq n/2-1\}.$$

Desenhe os vértices desse grafo escritos consecutivamente ao redor da circunferência de um círculo. Nesse caso as arestas descritas na linha superior de  $A$  ligam pares vizinhos de vértices ao longo do círculo. As arestas descritas na linha inferior de  $A$  ligam vértices em lados opostos do círculo. Essa figura mental mostra que todo vértice em  $G$  tem grau 3 ■

#### 3.3.2 Demonstração por Absurdo

A demonstração por absurdo é uma técnica comum de argumento para se provar um teorema. Normalmente ela é utilizada em situação que admite apenas duas hipóteses (e.g. dilemas). Ao demonstrar o absurdo de adotar uma destas hipóteses, inevitavelmente restará apenas uma possibilidade para a conclusão.

Como exemplo, a demonstração por absurdo do Teorema 4 será mostrada a seguir.

*Prova:* Vamos supor, por um momento, que, em um grafo qualquer, o número de vértices de grau ímpar não seja necessariamente par. Sendo isto verdade, existiria um grafo  $G$  que

teria um número ímpar de vértices de grau ímpar. Assim, a soma de todos os graus dos vértices de  $G$  seria ímpar. Isto é absurdo, pois a soma dos graus de todos os vértices de um grafo sempre é par (Teorema 1). Logo, pode-se afirmar que, em um grafo qualquer, o número de vértices de grau ímpar é necessariamente par ■

#### 3.3.3 Demonstração por Indução

A demonstração por indução é um eficiente instrumento para a validação de fatos referentes aos números naturais. Esta técnica costuma ser dividida em três partes: (i) a base da indução; (ii) a hipótese da indução, e (iii) o passo da indução [30].

Para ilustrar o seu funcionamento, imagine que você está subindo uma escada infinitamente alta. Como você sabe se será capaz de chegar a um degrau arbitrariamente alto? Suponha que você faça as seguintes hipóteses sobre sua capacidade de subir:

(1) Você consegue alcançar o primeiro degrau;

(2) Uma vez chegando a um degrau, você sempre é capaz de chegar ao próximo (note que esta afirmação é um condicional).

Se a proposição (1) e o condicional (2) são ambos verdadeiros, então, pela proposição (1), você consegue chegar no primeiro degrau e, portanto, pelo condicional (2), consegue chegar no segundo. Novamente, pelo condicional (2), você consegue chegar no terceiro degrau. Mais uma vez, pelo condicional (2), você consegue chegar no quarto degrau, e assim por diante.

Você pode subir tão alto quanto quiser. Ambas as hipóteses são necessárias. Se apenas a primeira proposição for verdadeira, você não teria garantia de passar do primeiro degrau. Se apenas a segunda for verdadeira, você poderia não ser capaz de começar nunca.

Ao fazer uma demonstração por indução, o estabelecimento da veracidade da proposição (1) é chamado de base da indução. O estabelecimento da veracidade de  $P(k) \rightarrow P(k+1)$  é chamado de passo indutivo. Quando supomos que  $P(k)$  é verdade para provar o passo indutivo,  $P(k)$  é chamada de hipótese da indução.

Como exemplo, a demonstração por indução do Teorema 1 será mostrada, a seguir.

*Prova por Indução:* A indução será feita sobre o número de arestas de  $G$  ( $|A_G|$ ).

*Caso Básico:* Se  $|A_G| = 0$ , então cada vértice terá grau 0 e a soma dos graus será 0 (i.e. o dobro do número de arestas). Logo, a afirmação é verdadeira para  $|A_G| = 0$ .

*Hipótese Indutiva:* Agora suponha que  $|A_G| > 0$  e que o teorema seja verdadeiro sempre que um grafo tiver  $|A_G| - 1$  arestas. Seja  $a$  uma aresta de  $G$ , e  $G'$  o grafo (com os mesmos vértices de  $G$ ) que obtemos excluindo a aresta  $a$  de  $A_G$ . Então,  $G'$  tem  $|A_G| - 1$  arestas e, portanto, por nossa hipótese indutiva, a soma dos graus dos vértices de  $G'$  é  $2(|A_G| - 1)$ .

*Passo Indutivo:* Ora a aresta  $a$  tem exatamente duas extremidades distintas. Podemos concluir que exatamente dois vértices de  $G'$  têm um grau a menos que seu correspondente em  $G$ . Assim, a soma dos graus dos vértices em  $G'$  é dois a menos que a soma dos graus dos vértices em  $G$ . Portanto, a soma dos graus dos vértices em  $G$  é

$$2|A_G| - 2 + 2 = 2|A_G|$$

Logo a soma dos graus dos vértices de um grafo é o dobro do número de suas arestas ■



### 3.4 Indução e Recursão

Uma característica importante de demonstrações por indução é a sua relação íntima com os algoritmos recursivos. É possível reescrever estes tipos de demonstração de forma a gerar um pseudocódigo recursivo correspondente.

O Algoritmo 1 é o algoritmo recursivo correspondente à demonstração apresentada na Seção 3.3.3.

---

**Algoritmo 1.** SomaDosGraus (Grafo  $g$ )

---

```
arestas  $\leftarrow$  Arestas( $g$ )  
se Tamanho(arestas) = 1  
    retorne 2  
senão  
     $a \leftarrow$  ArestaQualquer( $g$ )  
     $h \leftarrow$  GrafoSemAresta( $g, a$ )  
    retorne (SomaDosGraus( $h$ ) + 2)
```

---

### 4. FVA *GraphViewer*

A FVA *GraphViewer*<sup>2</sup> tem como objetivo auxiliar o usuário na compreensão das provas por indução dos Teoremas 1 e 2. Para isto, ele permite que usuário visualize tanto a demonstração, quanto a execução do algoritmo recursivo correspondente. Na Figura 2, é apresentada a captura de tela inicial da *GraphViewer*.



Figura 2. Captura da tela inicial da *GraphViewer*.

Na tela inicial (Figura 2), podemos iniciar a aplicação de duas formas, clicando em ambos botões de “Use agora!”. No lado direito do AVA, encontramos a aba principal, que podemos encontrar as seguintes informações sobre o trabalho como: “Sobre o projeto”, “Trabalhos publicados” e “Idealizadores”.

Após clicar no botão “Use agora!”, o usuário será direcionado para a tela conforme apresentada na Figura 3. Neste primeiro momento, o usuário deve escolher qual teorema será visualizado na execução da FVA. Suponha que o usuário escolha o Teorema 1 para prosseguir com a execução da *GraphViewer*.



Figura 3. Captura de tela da *GraphViewer* no momento da escolha do teorema a ser utilizado.

Ao escolher o Teorema 1, o usuário será direcionado para a tela conforme apresentada na Figura 4. Neste segundo momento, o usuário deve escolher o tipo de grafo e a quantidade de vértices que este possuirá. A *GraphViewer* fornece como opções cinco tipos diferentes de grafos: (i) caminho, (ii) árvore, (iii) grafo completo, (iv) circuito, e (v) grafo bipartido completo. A quantidade de vértices depende da escolha do tipo de grafo. Entretanto, o usuário dispõe, de maneira genérica, a escolher de 02 a 10 vértices. Neste exemplo, foi selecionado um grafo completo com dois vértices.



Figura 4. Captura de tela da *GraphViewer* no momento da escolha do grafo a ser utilizado.

Após clicar no botão “Próximo” (Figura 4), o usuário será direcionado para a tela conforme apresentada na Figura 5. Esta janela tem o propósito de garantir que as informações fornecidas pelo usuário são as desejadas por ele. Suponha que o usuário clique em “Gerar animação” para prosseguir com a execução da *GraphViewer*.

A Figura 6 mostra a captura da tela gerada no passo seguinte. É apresentada uma página dividida em duas partes. A primeira parte apresenta duas abas em que (i) na primeira aba, é exposto o pseudocódigo recursivo relativo à demonstração por indução do teorema, e (ii) na segunda aba (Figura 7), é exposta a demonstração por indução do respectivo teorema. A segunda parte da página está a região da tela que ocorre a animação com o grafo escolhido pelo usuário nos passos anteriores.

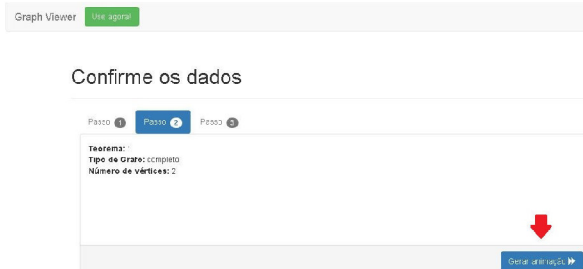


Figura 5. Captura de tela da *GraphViewer* no momento da confirmação dos dados fornecido à FVA.

<sup>2</sup> Repositório do projeto: <http://github.com/blind-review>

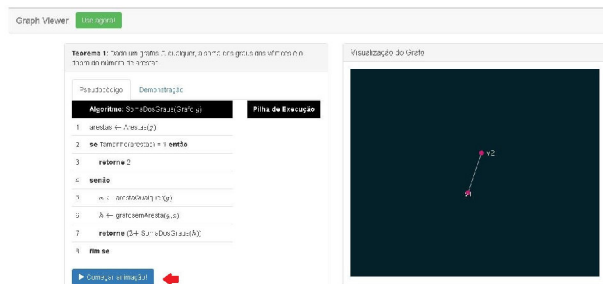


Figura 6. Captura de tela da GraphViewer no momento da visualização do pseudocódigo recursivo relativo à demonstração do Teorema 1.



Figura 7. Captura de tela da GraphViewer no momento da visualização da demonstração por indução do Teorema 1.

Ao clicar no botão “Gerar animação!” da Figura 6, a *GraphViewer* iniciará a simulação da execução do pseudocódigo com os parâmetros fornecidos pelo usuário (neste caso, um grafo completo com dois vértices). O início da animação é apresentada na Figura 8.

É possível perceber, ainda na Figura 8, os destaques em amarelo tanto no pseudocódigo, quanto na pilha de execução exibida na parte central da tela. O destaque em amarelo no pseudocódigo indica qual foi a última instrução (linha) que foi executada até então. E o destaque em amarelo na pilha de execução indica qual é a chamada de função que está sendo executada naquele momento (tendo em vista que um algoritmo recursivo faz várias chamadas de funções durante à sua execução).

Após o usuário ir clicando no botão ►►, a execução do algoritmo ocorrerá. Em cada passo, modificações serão realizadas, tanto nos destaques em amarelos, quanto na representação gráfica do grafo escolhido pelo usuário. Ao final da execução do algoritmo, a *GraphViewer* apresentará o valor de retorno em uma janela popup exibida na tela para o usuário (Figura 9).

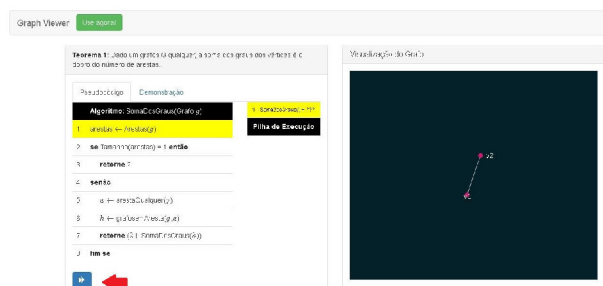


Figura 8. Captura de tela da GraphViewer no momento do início da simulação da execução do pseudocódigo da demonstração

por indução do Teorema 1.

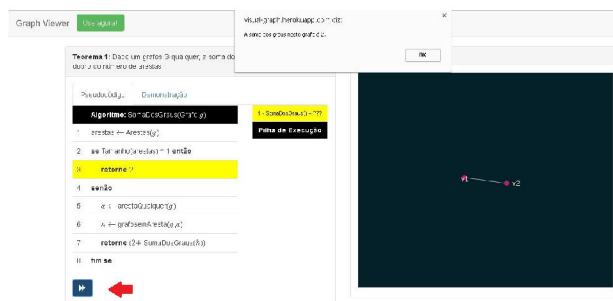


Figura 9. Captura de tela da GraphViewer no final da simulação da execução do pseudocódigo da demonstração por indução do Teorema 1.

A *GraphViewer* é um software-livre e está disponível para uso na internet<sup>3</sup>. O código-fonte pode ser obtido em repositório da plataforma GitHub<sup>2</sup>. Ele foi construído em HTML, CSS, PHP e JavaScript.

O propósito é que o docente, ao apresentar provas por indução em TG, permita que os estudantes usem a FVA simultaneamente à sua exposição. Desta forma, o estudante teria condições de materializar a demonstração através de casos concretos de grafos. Uma das vantagens desta abordagem é a possibilidade de o estudante observar, graficamente, o papel de cada elemento de uma prova indutiva através da visualização do passo-a-passo do algoritmo recursivo correspondente.

A *GraphViewer* serve como um recurso didático para auxiliar o estudante no processo de construção do seu conhecimento. O objetivo é que as visualizações das animações contribuam no processo de apropriação dos conceitos de TG e da técnica de demonstração por indução.

## 5. METODOLOGIA

A metodologia adotada consistiu na criação desta FVA e seu uso em um ambiente controlado. Este ambiente foi composto por duas turmas (A e B) de estudantes voluntários. O experimento ocorreu em dois momentos: (i) o primeiro momento, a turma A foi exposta a conceitos que envolvem prova por indução com a utilização da FVA, enquanto que a turma B foi exposta aos mesmos conceitos, porém sem a utilização da FVA criada; (ii) o segundo momento, foi semelhante ao primeiro, porém com a inversão das turmas e com a exposição de outros conceitos. Questionários foram aplicados antes, durante e após a realização do experimento.

Será apresentado a seguir o paradigma GQM (Seção 5.1) e a sua utilização no trabalho, através de suas especificações (Seção 5.2).

### 5.1 Paradigma GQM

O paradigma *Goal-Question-Metric* (GQM) [32] é utilizado na Engenharia de Software para auxiliar na medição de processos e softwares, sendo uma abordagem orientada a métricas. A vantagem da sua utilização é que ele separa as preocupações organizacionais (objetivos) das específicas do processo (questões). Isto fornece uma base para decidir quais dados deveriam ser coletados e como eles deveriam ser analisados de forma a responder as questões que se pretende resolver [27].

<sup>3</sup> <http://visual-graph.herokuapp.com/>

Os conceitos utilizados dentro do paradigma GQM podem ser definidos sucintamente como:

*objetivo* - meta a qual a pesquisa almeja alcançar;

*questões* - refinamentos dos objetivos em que áreas específicas de incertezas relacionadas são identificadas;

*métricas* - medições que precisam ser feitas para ajudar a responder as questões e confirmar, quantitativamente, se a pesquisa alcançou ou não o objetivo desejado.

Embora o GQM seja largamente utilizado na Engenharia de Software, este paradigma vem sendo utilizado e adaptado em pesquisas na área de educação [12][17].

## 5.2 Especificações do GQM

As especificações concretas para os três conceitos principais do GQM foram estabelecidas. O objetivo do projeto consiste em avaliar a utilidade e qualidade da *GraphViewer*, com ênfase no ensino e aprendizagem de demonstração por indução.

Duas questões foram elencadas como refinamentos do objetivo proposto. A primeira é (i) “a *GraphViewer* melhorou o aprendizado do alunos em relação ao não uso dele?”; e a segunda é (ii) “a *GraphViewer* é um software fácil de ser utilizado?”.

Por fim, métricas são construídas para as duas questões elencadas. Para as especificações das métricas, foram criados quatro questionários. Estes questionários foram submetidos aos participantes que participaram dos experimentos. As respostas concedidas aos questionários serviram de entrada para o cálculo das métricas.

Dois questionários (Questionários 1 e 2) serviram para identificar aspectos sobre a compreensão do estudante sobre a demonstração por indução apresentada para os Teoremas 1 e 2, respectivamente. Cada um destes questionários teve cinco questões objetivas.

O Questionário 3 serviu para identificar os aspectos socioacadêmicos dos participantes. Variáveis como gênero, idade e ano de ingresso são exemplos dos dados requisitados. O propósito deste questionário foi de correlacionar, se possível, de maneira exploratória, os resultados obtidos da aplicação dos Questionários 1 e 2 com os aspectos socioacadêmicos dos participantes do experimento.

Por fim, o Questionário 4 serviu para identificar aspectos sobre a usabilidade da *GraphViewer*. Questões sobre facilidade de aprendizado da ferramenta, eficiência de uso, frequência de ocorrência e seriedade dos erros são exemplos dos dados requisitados. Este questionário consiste de cinco perguntas objetivas utilizando uma escala do tipo *Likert* [19], com pontuações de um a cinco pontos, respectivamente com rótulos: (i) Discordo Totalmente, (ii) Discordo Parcialmente; (iii) Indiferente, (iv) Concordo parcialmente, e (v) Concordo plenamente. Todos estes questionários estão disponíveis no repositório do projeto<sup>2</sup>.

A partir das respostas obtidas dos questionários, duas métricas foram propostas. A métrica  $M_1$  é um indicador do impacto na aprendizagem dos estudantes pelo o uso da *GraphViewer* de forma que  $M_1 = DIFF_2 - DIFF_1$ , em que

$$DIFF_1 = \frac{\#PONT_A^{Q1} - \#PONT_B^{Q1}}{\max(\#PONT_A^{Q1}, \#PONT_B^{Q1})}$$

e

$$DIFF_2 = \frac{\#PONT_B^{Q2} - \#PONT_A^{Q2}}{\max(\#PONT_A^{Q2}, \#PONT_B^{Q2})}$$

sendo  $\#PONT_i^{Qx}$  a pontuação obtida no Questionário  $x$  do Grupo  $i$  (ver Seção 6). A seguinte interpretação de  $M_1$  foi adotada: se  $M_1 \geq 20\%$ , então a *GraphViewer* contribuiu positivamente no aprendizado dos participantes. Caso contrário, a contribuição da ferramenta para a aprendizagem pode estar comprometida.

A métrica  $M_2$  é um indicador para a usabilidade da *GraphViewer*.  $M_2$  é definida a partir das respostas coletadas a partir do Questionário 4 com todos os participantes.  $M_2$  é definida a seguir:

$$M_2 = \frac{\#PONT^{Q4}}{n}$$

A seguinte interpretação de  $M_2$  foi adotada: se  $M_2 \geq 3,0$ , então a *GraphViewer* é fácil de usar. Caso contrário, a ferramenta tem a sua usabilidade comprometida.

Os resultados do Questionário 3 foram utilizados primordialmente para a análise dos dados obtidos após o experimento.

## 6. EXPERIMENTOS E RESULTADOS

O experimento foi realizado no Campus Jatobá da Regional Jataí, na Universidade Federal de Goiás, em um laboratório de computação. Houve a participação de dez alunos voluntários e um professor voluntário do curso de Computação.

Os alunos foram divididos em dois grupos A e B (ambos com cinco estudantes). Todo o conteúdo ministrado no experimento foi lido pelo professor voluntário aos dois grupos. O experimento pode ser dividido em sete momentos:

*Momento 1* - aplicação do Questionário 3 aos alunos presentes;

*Momento 2* - exposição, aos dois grupos, de conceitos básicos de grafos e também de demonstração por indução;

*Momento 3* - exposição, para o Grupo A, da demonstração por indução do Teorema 1 com a utilização da *GraphViewer*; enquanto o Grupo B foi exposto ao mesmo conteúdo, porém sem a utilização da FVA criada;

*Momento 4* - aplicação do Questionário 1 para ambos os grupos;

*Momento 5* - exposição semelhante ao Momento 3, porém com a inversão dos grupos em relação ao uso da *GraphViewer* e, agora, utilizando o Teorema 2;

*Momento 6* - aplicação do Questionário 2 para ambos os grupos;

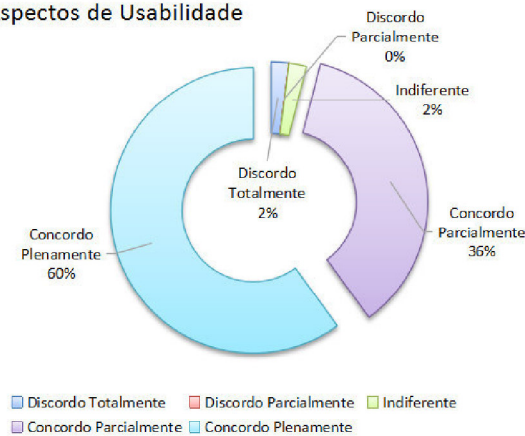
*Momento 7* - aplicação do Questionário 4 ambos os grupos.

### 6.1 Análise dos Resultados

No Momento 7, o Questionário 4 foi respondido pelos participantes do experimento. Um gráfico dos resultados deste questionário é apresentado na Figura 10. É possível verificar que a maioria dos alunos de ambos os grupos avaliaram positivamente a usabilidade da FVA. O cálculo da métrica  $M_2$  foi 4,52, atendendo satisfatoriamente com às expectativas dos autores (que era  $M_2 \geq 3,0$ ).



### Aspectos de Usabilidade

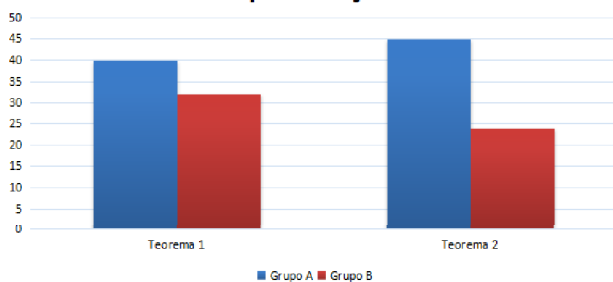


**Figura 10.** Gráfico dos resultados do Questionário de Usabilidade da GraphViewer obtidos com todos os participantes do experimento.

Nos Momentos 4 e 6, os Questionários 2 e 3 referentes à compreensão da demonstração dos Teoremas 1 e 2, respectivamente, foram respondidos pelos participantes. Um gráfico dos resultados destes questionários é apresentado na Figura 11. É possível constatar no gráfico que o Grupo A obteve melhor pontuação do que o Grupo B, em relação ao Teorema 1. Este era um resultado esperado pelos autores, pois a FVA foi utilizado pelo Grupo A, e não pelo Grupo B.

Entretanto, é possível constatar também na Figura 11 que o Grupo A obteve melhor pontuação do que o Grupo B, em relação ao Teorema 2. Este não era, de certa forma, um resultado esperado pelos autores, pois a FVA foi utilizado pelo Grupo B, e não pelo Grupo A. Desta forma, o cálculo da métrica  $M_i$  resultou no valor de -64,44%, situando bem distante do valor mínimo esperado de 20%.

### Compreensão da Demonstração por Indução



**Figura 11.** Resultados dos Questionários 2 e 3 obtidos com os participantes de cada grupo do experimento. Gráfico (b): resultados do Questionário 1 obtidos com os participantes de cada grupo do experimento em relação ao fato de trabalharem (ou não) fora do expediente acadêmico.

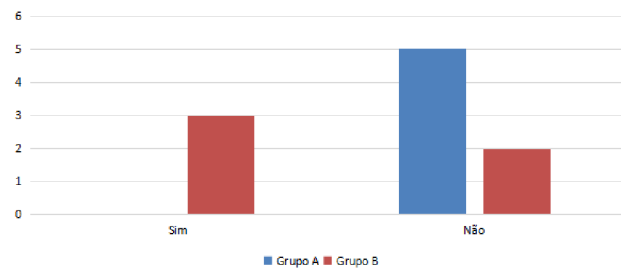
Com o propósito de compreender melhor a natureza do resultado apresentado na Figura 11, algumas conjecturas foram levantadas. Os resultados obtidos do Questionário 1 foram analisados. A média de idade do Grupo A foi de 21,6 anos, enquanto que a média do Grupo B foi de 20,2. Os autores acreditam que o fator

idade não foi determinante para justificar a natureza dos resultados da Figura 11.

Outros fatores foram analisados como a cor/raça, ano de ingresso e ano no qual o participante cursou a disciplina de TG. Nenhum destes fatores apresentaram discrepâncias entre os grupos de forma significativa. Desta forma, os autores também não creditam a estes fatores o comportamento do referido resultado.

Todavia, o fato de os participantes de cada grupo trabalharem ou não fora do expediente acadêmico apresentou uma discrepância considerável. Um gráfico dos resultados deste fator do Questionário 1 é apresentado na Figura 12. É possível constatar no gráfico que nenhum aluno do Grupo A trabalha fora do expediente acadêmico. No Grupo B, três dos cinco participantes se enquadram neste quesito. Os autores acreditam que este fator possa ter sido determinante para justificar a natureza dos resultados da Figura 11.

### Trabalha fora do expediente acadêmico?



**Figura 12.** Resultados do Questionário 1 obtidos com os participantes de cada grupo do experimento em relação ao fato de trabalharem (ou não) fora do expediente acadêmico.

## 6.2 Ameaças à Validade do Experimento

Algumas ameaças à validade do experimento foram identificadas. A primeira ameaça mais evidente diz respeito ao número de participantes. Como o experimento foi realizado com apenas dez participantes, é bem possível que os resultados obtidos não usufruam de uma diversidade mínima considerável. Com um número maior de participantes, a qualidade das afirmações aumenta, em relação à sua generalidade, devido à melhor representatividade da amostra coletada.

A segunda ameaça identificada diz respeito ao horário da realização do experimento. Como o experimento foi realizado à noite, alguns outros fatores decorrentes do fato de participantes trabalharem podem ter contribuído para o resultado da Figura 11. Podemos citar, por exemplo, o cansaço físico que, por consequência, pode levar a uma maior dificuldade de concentração destes participantes.

## 7. CONSIDERAÇÕES FINAIS

Este trabalho apresentou a FVA *GraphViewer* e o seu uso no ensino de provas por indução em Teoria dos Grafos. Foi proposto um método de ensino e aprendizagem com intuito de auxiliar os alunos que tenham dificuldades no entendimento e compreensão tanto da técnica de demonstração por indução, quanto dos conceitos abordados na disciplina.

Ao final, foi realizada uma análise quali-quantitativa dos resultados obtidos dos questionários respondidos. Após concluída a análise, foram obtidos alguns resultados: (i) não foi possível concluir com os dados levantados que a FVA criada é mais favorável ao processo de ensino e aprendizagem do que o cenário que não envolve o seu uso; entretanto (ii) houve aceitação da ferramenta por parte dos participantes do experimento, em termos de usabilidade, com um total de 96% de avaliações positivas.

A impossibilidade de uma conclusão mais incisiva sobre a FVA foi apontada devido aos resultados obtidos com variáveis socioacadêmicas. Como trabalhos futuros, os autores compreendem que estas variáveis podem ser melhor controladas, garantindo um mínimo de discrepância entre os resultados de cada grupo. Além das variáveis utilizadas neste trabalho, outras variáveis socioacadêmicas podem ser elencadas com o propósito de contrastar com os resultados obtidos pelo desempenho acadêmico dos participantes.

## 8. REFERÊNCIAS

- [1] Baloukas, T. (2012). Javenga: Java-based visualization environment for network and graph algorithms. *Computer Applications in Engineering Education*, 20(2):255–268.
- [2] Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45–73.
- [3] Bernardes, L. H. (2016). FURB Graphs: uma Ferramenta de Apoio ao Aprendizado para a Disciplina de Teoria dos Grafos. Monografia, Universidade Regional de Blumenau.
- [4] Buchbinder, O. and Zaslavsky, O. (2011). Is this a coincidence? the role of examples in fostering a need for proof. *ZDM*, 43(2):269.
- [5] Deo, N. (2017). *Graph theory with applications to engineering and computer science*. Courier Dover Publications.
- [6] Feofiloff, P. (2012). Exercícios de Teoria dos Grafos. Departamento de Ciência da Computação, IME-USP. Disponível em <https://www.ime.usp.br/~pf/grafos-exercicios/texto/ETG.pdf>. Acesso em 30 de junho de 2017.
- [7] Fincher, S. and Petre, M. (2004). *Mapping the territory*. In *Computer science education research*. CRC Press.
- [8] Fouh, E., Akbar, M. and Shaffer, C. (2012). The Role of Visualization in Computer Science Education. *Computers in the Schools: Interdisciplinary Journal of Practice, Theory, and Applied Research*, 29:1–2, 95–117.
- [9] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415.
- [10] Garcia, I., Rezende, P., and Calheiros, F. (2012). Astral: An environment for teaching data structures through algorithms animations. *Brazilian Journal of Computers in Education*, 1(1):71–80.
- [11] Gerab, F., Bueno, I., and Gerab, I. (2014). Análise das interações curriculares em um curso de ciência da computação: buscando subsídios para aprimoramento curricular. *Revista Brasileira de Informática na Educação*, 22(01).
- [12] Gladcheff, A. P., Sanches, R., and da Silva, D. M. (2002). Um instrumento de avaliação de qualidade de software educacional: como elaborá-lo. *Pensamento & Realidade*. *Revista do Programa de Estudos Pós-Graduados em Administração-FEA*. ISSN 2237-4418, 11.
- [13] Grissom, S. (2013). Introduction to Special Issue on Alternatives to Lecture in the Computer Science Classroom. *ACM Transactions on Computing Education (TOCE)*, 13(3):9.
- [14] Hazzan, O. and Hadar, I. (2005). Reducing Abstraction when learning Graph Theory. *The Journal of Computers in Mathematics and Science Teaching*, 24(3):255.
- [15] Kussmaul, C. (2012). Process oriented guided inquiry learning (POGIL) for computer science. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 373–378. ACM.
- [16] Lozada, L. A. P. (2014). A-Graph: Uma ferramenta computacional de suporte para o ensino-aprendizado da disciplina Teoria dos Grafos e seus Algoritmos. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 3, page 61.
- [17] Machado, A. S. (2016). Uso de softwares educacionais, objetos de aprendizagem e simulações no ensino de química. *Revista Química Nova na Escola*, 38(2):104–111.
- [18] Mariotti, M. A. (2006). Proof and proving in mathematics education. *Handbook of research on the psychology of mathematics education: Past, present and future*, pages 173–204.
- [19] Nemoto, T. and Beglar, D. (2014). Developing Likert-Scale Questionnaires. In *Japan Association for Language Teaching (JALT) 2013 Conference Proceedings*, pages 1–8.
- [20] Polycarpou, I. (2008). *An innovative approach to teaching structural induction for computer science*. PhD thesis, Florida International University.
- [21] Randolph, J., Sutinen, E., Julnes, G., and Lehman, S. (2008). A methodological review of computer science education research. *Journal of Information Technology Education*, 7:135–163.
- [22] Sangiorgi, U. B. (2006). Rox: uma ferramenta para o auxílio no aprendizado de Teoria dos Grafos. IV ERBASE–Escola Regional de Computação Bahia-Sergipe. *Anais do Evento: UEFS*.
- [23] Shaffer, C. A., Cooper, M. L., Alon, A. J. D., Akbar, M., Stewart, M., Ponce, S., & Edwards, S. H. (2010). Algorithm visualization: The state of the field. *ACM Transactions on Computing Education (TOCE)*, 10(3), 9.
- [24] Shaffer, C. A., Akbar, M., Alon, A. J. D., Stewart, M., & Edwards, S. H. (2011, March). Getting algorithm visualizations into the classroom. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 129–134). ACM.
- [25] Silva, B. T. d. (2015). *Indução Matemática: Discussão Teórica e uma Proposta de Ensino*. Master’s thesis, Universidade Federal do Rio Grande do Norte.
- [26] Sipser, M. (2012). Introduction: Types of Proofs. *Introduction to the Theory of Computation*. Cengage Learning.



- [27] Sommerville, I. (2010). *Software Engineering*. Pearson. 9a Edition.
- [28] Sousa, E. K. V. d. (2010). Um estudo sobre o ensino-aprendizagem das demonstrações matemáticas. Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte.
- [29] Stasko, J. T. and Hundhausen, C. D. (2004). Algorithm visualization. *Computer Science Education Research*, pages 199–228.
- [30] Stein, C., Drysdale, R. L., Bogart, K. (2013). *Indução, recursão e recorrência. Matemática Discreta para a Ciência da Computação*. Pearson.
- [31] Urquiza-Fuentes, J. and Velázquez-Iturbide, J. A. (2009). A survey of successful evaluations of program visualization and algorithm animation systems. *Trans. Comput. Educ.*, 9(2):9:1–9:21.
- [32] van Solingen, R., Basili, V., Caldiera, G., and Rombach, H. D. (2002). *Encyclopedia of Software Engineering*, chapter Goal Question Metric (GQM) Approach. John Wiley & Sons, Inc.