



## GRAPHVIEWER: UM SISTEMA DE VISUALIZAÇÃO DE ALGORITMOS NO ENSINO DE PROVAS POR INDUÇÃO EM TEORIA DOS GRAFOS

Mateus Oliveira dos Santos<sup>1</sup>

Gecirlei Francisco da Silva<sup>2</sup>

Esdras Lins Bispo Jr<sup>3</sup>

### RESUMO

A *GraphViewer* é uma ferramenta de visualização de algoritmos (FVA) com o propósito de aperfeiçoar o processo de ensino e aprendizagem da disciplina de Teoria de Grafos. As FVAs são utilizadas para visualizar o passo-a-passo da execução de algoritmos, inclusive alguns deles em Grafos. Entretanto, os autores deste trabalho apontaram uma lacuna metodológica na validação dos resultados desta FVA. Problemas em relação à validação estatística do experimento foram apontadas, como, por exemplo, a uniformidade dos grupos durante o experimento e a quantidade de participantes. Desta forma, este trabalho tem como propósito evidenciar e discutir a validação pelos usuários da FVA *GraphViewer*. Um novo experimento foi realizado com a participação de 18 estudantes (ao invés de apenas 10), adotando alguns critérios para garantir a distribuição uniforme dos grupos de controle e de teste. A utilização da FVA garantiu ganhos de aprendizagem que corresponderam a quase o dobro do ganho obtido pelo grupo de controle. Também foi constatada a aceitação dos usuários em relação à usabilidade da ferramenta, de forma a ter mais de 90% de avaliações positivas.

**Palavras-chave:** ensino, visualização, grafos, demonstração, indução.

---

<sup>1</sup> Universidade Federal de Jataí. [mateus1996oliveira@hotmail.com](mailto:mateus1996oliveira@hotmail.com).

<sup>2</sup> Universidade Federal de Jataí. [gecirlei@yahoo.com](mailto:gecirlei@yahoo.com)

<sup>3</sup> Universidade Federal de Jataí. [bispojr@ufg.br](mailto:bispojr@ufg.br).

## 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 (FINCHER; PETRE, 2004; RANDOLPH *et al.*, 2008). Ela endereça várias questões vivenciadas por profissionais de EC, tais como (i) métodos de ensino (KUSSMAUL, 2012), (ii) avaliação e construção de currículo (GERAB *et al.*, 2014), e (iii) animação, visualização e simulação de algoritmos (URQUIZA-FUENTES; VELÁZQUEZ-ITURBIDE, 2009; GARCIA *et al.*, 2012).

Em relação aos métodos de ensino em EC, novas metodologias e métodos vêm sendo propostos (GRISSOM, 2013). 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 (FREEMAN *et al.*, 2014), ao invés do formato tradicional de ensino.

Como instrumento auxiliar destes métodos, ferramentas de visualização de algoritmos (FVA) vêm sendo propostas (STASKO; HUNDHAUSEN, 2004; GARCIA *et al.*, 2012). 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 (HAZZAN; HADDAR, 2005).

Na disciplina de Teoria dos Grafos (TG), várias FVAs foram propostas (SHAFFER *et al.*, 2010; SANGIORGI, 2006; LOZADA, 2014; BERNARDES, 2016). Estas FVAs são utilizados 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 (MARIOTTI, 2006; BUCHBINDER; ZASLAVSKY, 2011). Grande parte dos alunos, por exemplo, sente dificuldades em visualizar e utilizar de forma correta a demonstração por indução (SILVA, 2015).

Tendo em vista a carência de trabalhos que proponham um SVA como recurso metodológico para o ensino de provas por indução em TG, foi proposta a FVA *GraphViewer* para este fim (CARVALHO *et al.*, 2017). Entretanto, os autores deste trabalho apontaram uma lacuna metodológica na validação dos resultados da FVA.

Problemas em relação à validação estatística foram apontadas, como, por exemplo, a uniformidade dos grupos durante o experimento e a quantidade de participantes.

Desta forma, este trabalho tem como propósito evidenciar e discutir a validação pelos usuários da FVA *GraphViewer*. Um novo experimento foi realizado com a participação de 18 estudantes (ao invés de apenas 10), adotando alguns critérios para garantir a distribuição uniforme dos grupos de controle e de teste.

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 (SANGIORGI, 2006) possibilita a criação e manipulação de grafos, bem como a execução de algoritmos clássicos. A FVA JAVENGA (BALOUKAS, 2012) 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* (LOZADA, 2014) 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* (BERNARDES, 2016) 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.

Os resultados apresentados na tese de Polycarpou (2008) é também um trabalho relacionado. 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).

E, naturalmente, o trabalho original do *GraphViewer* (CARVALHO *et al.*, 2017) é o trabalho com maior similaridade. Porém, a distinção deste trabalho reside nas

modificações no formato do experimento original. O propósito é de corrigir algumas das ameaças à validade apresentadas por Carvalho e colegas.

### 3 FUNDAMENTOS

#### 3.1 VISUALIZAÇÃO DE ALGORITMOS

A visualização de algoritmos (VA) ocupa um papel importante na EC (FOUH *et al.*, 2012). 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 (SHAFFER *et al.*, 2011). 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 (BEN-ARI, 2001). 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 os propósitos pedagógicos em questão.

Existe uma iniciativa que congrega várias iniciativas em VA. O Portal AlgoViz<sup>4</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 (SHAFFER *et al.*, 2011).

---

<sup>4</sup> <http://algoviz.org/>

### 3.2 TEORIA DOS GRAFOS

A Teoria de Grafos (TG) tem um número considerável de aplicações (DEO, 2017). 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 (FEOFILLOFF, 2012).

**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  $\{xv \mid xv \in A\}$ .

Oriundas destas, apresentamos a seguir outras definições utilizadas neste trabalho (FEOFILLOFF, 2012).

**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.

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 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 (SOUSA, 2010).

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 (SIPSER, 2013).

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 (STEIN *et al.*, 2013).

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. Os Teoremas 1 e 2, apresentados na Seção 3.2, serviram de insumo para a construção da FVA *GraphViewer*.

## 5. METODOLOGIA

A metodologia adotada consistiu na utilização da FVA *GraphViewer*<sup>5</sup> em um ambiente controlado. O experimento foi realizado no Campus Jatobá da Regional Jataí, na Universidade Federal de Goiás, em um laboratório de computação. Houve a participação de 18 alunos voluntários e um professor voluntário do curso de Computação. Foram adotados alguns critérios para garantir a distribuição uniforme dos grupos de controle e de teste (MARCONI; LAKATOS, 2003, p. 145).

Os alunos foram divididos em dois grupos A e B (ambos com nove estudantes), sendo respectivamente grupo de controle e grupo de teste. Todo o conteúdo ministrado no experimento foi lecionado pelo professor voluntário aos dois grupos simultaneamente. Entretanto, apenas o Grupo B pode utilizar a *GraphViewer* durante todo o experimento.

---

<sup>5</sup> Disponível em <http://visual-graph.herokuapp/>

Com intuito de garantir a isonomia no experimento, os dois grupos foram uniformemente distribuídos seguindo alguns critérios. Os critérios foram aplicados nesta ordem: (i) sexo biológico; (ii) vínculo empregatício (“Está trabalhando ou não?”); (iii) conhecimento anterior do conteúdo (“Cursou ou não a disciplina de Grafos?”).

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 sexo biológico, idade e ano de ingresso são exemplos dos dados requisitados.

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* (NEMOTO; BEGLAR, 2014), 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. O experimento pode ser dividido em nove momentos:

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

**Momento 2** - aplicação do Questionário 1 (pré-teste) para ambos os grupos;

**Momento 3** - exposição, aos dois grupos, de conceitos básicos de grafos e também de demonstração por indução;

**Momento 4** - exposição da demonstração por indução do Teorema 1;

**Momento 5** - aplicação do Questionário 1 (pós-teste) para ambos os grupos;

**Momento 6** - aplicação do Questionário 2 (pré-teste) para ambos os grupos;

**Momento 7** - exposição da demonstração por indução do Teorema 2;

**Momento 8** - aplicação do Questionário 2 (pós-teste) para ambos os grupos;

**Momento 9** - aplicação do Questionário 4 para o Grupo B.

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) (SOLINGER *et al.*, 2002) 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 (SOMMERVILLE, 2010).

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 (GLADCHEFF *et al.*, 2002; MACHADO, 2016).

## 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.

A partir das respostas obtidas dos questionários, duas métricas foram propostas. A métrica M1 é um indicador do impacto na aprendizagem dos estudantes pelo o uso da

*GraphViewer*. A M1 é dada pelos cálculos do ganho de aprendizagem absoluto e do ganho de aprendizagem normalizado (HAKE, 1998, p. 3). A seguinte interpretação de M1 foi adotada: se os ganhos de aprendizagem absoluto e normalizado forem maiores no grupo de teste em relação ao grupo de controle, então a *GraphViewer* contribuiu positivamente no aprendizado dos participantes. Caso contrário, a contribuição da ferramenta para a aprendizagem pode estar comprometida.

A métrica M2 é um indicador para a usabilidade da *GraphViewer*. M2 é definida a partir das respostas coletadas a partir do Questionário 4 com todos os participantes. M2 é obtida pela média da pontuação de todas as respostas de cada participante. A seguinte interpretação de M2 foi adotada: se  $M2 > 4,0$ , 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. DISCUSSÃO DOS RESULTADOS

### 6.1 ANÁLISE DOS RESULTADOS

Na Tabela 1, são apresentados os ganhos de aprendizagem alcançados pelos estudantes durante o experimento. Percebe-se que a utilização do *GraphViewer* (grupo de teste) garantiu ganhos superiores em todos os cenários. Em relação ao experimento como um todo, o ganho de aprendizagem normalizado foi quase o dobro (25,11 %) do obtido pelo grupo de controle (13,92 %).

**Tabela 1** – Ganhos de Aprendizagem Absoluto e Normalizado em relação aos Teoremas 1 e 2 a partir dos dois grupos de participantes do experimento.

	Grupo	Ganho Absoluto	Ganho Normalizado
Teorema 1	Controle	2,78 %	3,70 %
	Teste	16,67 %	22,22 %
Teorema 2	Controle	19,44 %	24,14 %
	Teste	19,44 %	28,00 %
Experimento Inteiro	Controle	11,11 %	13,92 %
	Teste	18,06 %	25,11 %

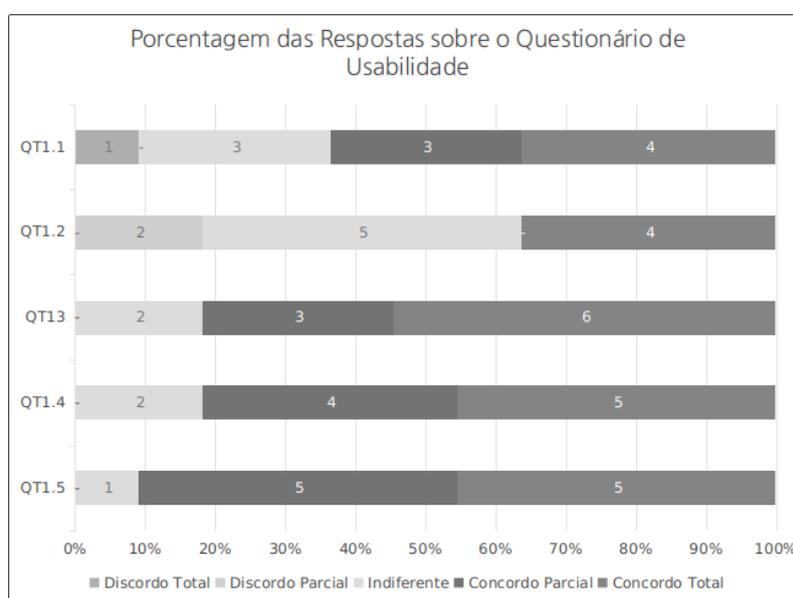
No Momento 7, o Questionário 4 foi respondido pelos participantes do experimento. Um gráfico dos resultados deste questionário é apresentado na Figura 1. É

possível verificar que a maioria dos alunos de ambos os grupos avaliaram positivamente a usabilidade da FVA. O cálculo da métrica M2 foi 4,52, atendendo satisfatoriamente com às expectativas dos autores (que era 4,0).

## 6.2 AMEAÇAS À VALIDADE DO EXPERIMENTO

Uma ameaça à validade do experimento foi identificada no que diz respeito ao número de participantes. Como o experimento foi realizado com apenas 18 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. Um próximo experimento já está previsto no andamento da pesquisa com um quantidade prevista de 40 participantes.

**Figura 1.** Gráfico dos resultados do Questionário de Usabilidade da *GraphViewer* obtidos com os participantes do grupo de teste do experimento.



## 7 CONSIDERAÇÕES FINAIS

Este trabalho teve como propósito apresentar novos resultados em relação à FVA *GraphViewer*. Este novo experimento foi feito com 18 estudantes, adotando critérios para garantir a distribuição uniforme dos grupos de controle e teste. A utilização da FVA

garantiu ganhos de aprendizagem que corresponderam a quase o dobro do obtido pelo grupo de controle. Houve aceitação dos usuários em relação à usabilidade da ferramenta, de forma a ter mais de 90% de avaliações positivas. Como trabalho futuro, espera-se estender o experimento com um número maior de participantes, garantindo os critérios adotados neste experimento.

## REFERÊNCIAS

- BALOUKAS, T. Javenga: Java-based visualization environment for network and graph algorithms. **Computer Applications in Engineering Education**. Vol. 20, n. 2, p. 255–268, 2012.
- BEN-ARI, M. Constructivism in computer science education. **Journal of Computers in Mathematics and Science Teaching**. Vol. 20, n. 1, p. 45-73, 2001.
- BERNARDES, L. H. **FURB Graphs: uma Ferramenta de Apoio ao Aprendizado para a Disciplina de Teoria dos Grafos**. Monografia, Universidade Regional de Blumenau, 2016.
- BUCHBINDER, O.; ZASLAVSKY, O. Is this a coincidence? The role of examples in fostering a need for proof. **ZDM**. Vol. 43, n. 2, p. 269.
- CARVALHO, F. A.; BORGES, F. F.; SILVA, G. F.; BORGES, T. O.; BISPO JR., E. L. Ensino de Provas por Indução em Grafos utilizando uma Ferramenta Visual de Algoritmos. **Nuevas Ideas en Informatica Educativa**. Vol. 13, p. 41-49, 2017.
- DEO, N. **Graph Theory with Applications to Engineering and Computer Science**. Courier Dover Publications, 2017.
- FEOFILOFF, P. **Exercícios de Teoria dos Grafos**. Departamento de Ciência da Computação, IME-USP, 2012. Disponível em <<https://www.ime.usp.br/~pf/grafos-exercicios/texto/ETG.pdf>>. Acesso em 30 de junho de 2017.
- FINCHER, S.; PETRE, M. Mapping the territory. In: **Computer Science Education Research**. CRC Press, 2004.
- FOUH, E.; AKBAR, M.; SHAFFER, C. The Role of Visualization in Computer Science Education. **Computers in the Schools: Interdisciplinary Journal of Practice, Theory, and Applied Research**. Vol. 29, n. 1-2, p. 95-117, 2012.
- FREEMAN, S.; EDDY, S. L.; MCDONOUGH, M.; SMITH, M. K.; OKOROAFOR, N.; JORDT, H.; WENDEROTH, M. P. Active Learning increases Student Performance in Science, Engineering, and Mathematics. **Proceedings of the National Academy of Sciences**. Vol. 111, n. 23, p. 8410–8415, 2014.
- GARCIA, I.; REZENDE, P.; CALHEIROS, F. Astral: An Environment for Teaching Data Structures through Algorithms Animations. **Brazilian Journal of Computers in Education**. Vol. 1, n. 1, p. 71–80, 2012.
- GERAB, F.; BUENO, I.; GERAB, I. Análise das interações curriculares em um curso de ciência da computação: buscando subsídios para aprimoramento curricular. **Revista Brasileira de Informática na Educação**. Vol. 22, n. 1, 2014.

- GLADCHEFF, A. P., SANCHES, R., DA SILVA, D. M. 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, n. 11, 2002.
- GRISSOM, S. Introduction to Special Issue on Alternatives to Lecture in the Computer Science Classroom. **ACM Transactions on Computing Education (TOCE)**. Vol. 13, n. 3, p. 9, 2013.
- HAKE, R. R. Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for Introductory Physics Courses. **American Journal of Physics**. Vol. 66, n. 1, p. 64–74, 1998.
- HAZZAN, O.; HADAR, I. Reducing Abstraction when learning Graph Theory. **The Journal of Computers in Mathematics and Science Teaching**. Vol. 24, n. 3, p. 255, 2005.
- KUSSMAUL, C. Process oriented guided Inquiry Learning (POGIL) for Computer Science. **Proceedings of the 43rd ACM Technical Symposium on Computer Science Education**, p. 373-378, 2012.
- LOZADA, L. A. P. A-Graph: Uma Ferramenta Computacional de Suporte para o Ensino-Aprendizado da disciplina Teoria dos Grafos e seus Algoritmos. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. Vol. 3, p. 61, 2014.
- MACHADO, A. S. Uso de softwares educacionais, objetos de aprendizagem e simulações no ensino de química. **Revista Química Nova na Escola**. Vol. 38, n. 2, p. 104–111, 2016.
- MARCONI, M. A.; LAKATOS, E. V. Fundamentos de Metodologia Científica. Atlas, 2003.
- MARIOTTI, M. A. Proof and Proving in Mathematics Education. In: **Handbook of Research on the Psychology of Mathematics Education: Past, present and future**, p. 173–204, 2006.
- NEMOTO, T.; BEGLAR, D. Developing Likert-Scale Questionnaires. **Japan Association for Language Teaching (JALT) 2013 Conference Proceedings**, p. 1–8, 2014.
- POLYCARPOU, I. **An Innovative Approach to Teaching Structural Induction for Computer Science**. Tese de Doutorado, Florida International University, 2008.
- RANDOLPH, J.; SUTINEN, E.; JULNES, G.; LEHMAN, S. A methodological review of computer science education research. **Journal of Information Technology Education**. Vol. 7, p. 135–163, 2008.
- SANGIORGI, U. B. Rox: uma ferramenta para o auxílio no aprendizado de Teoria dos Grafos. **Anais do IV ERBASE–Escola Regional de Computação Bahia-Sergipe**, 2006.
- SHAFFER, C. A.; COOPER, M. L.; ALON, A. J. D.; AKBAR, M.; STEWART, M.; PONCE, S.; EDWARDS, S. H. Algorithm visualization: The state of the field. **ACM Transactions on Computing Education (TOCE)**. Vol. 10, n. 3, p. 9, 2010.
- SHAFFER, C. A.; AKBAR, M.; ALON, A. J. D.; STEWART, M.; EDWARDS, S. H. Getting Algorithm Visualizations into the Classroom. **Proceedings of the 42nd ACM technical symposium on Computer science education**, p. 129-134, 2011.
- SILVA, B. T. **Indução Matemática**: Discussão Teórica e uma Proposta de Ensino. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte, 2015.

SIPSER, M. Introduction: Types of Proofs. In: **Introduction to the Theory of Computation**. 3ª Edição. Cengage Learning, 2013, p. 21-25.

SOMMERVILLE, I. **Software Engineering**. 9ª Edição, Pearson, 2010.

SOUSA, E. K. V. D. **Um Estudo sobre o Ensino-Aprendizagem das Demonstrações Matemáticas**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, 2010.

STASKO, J. T.; HUNDHAUSEN, C. D. Algorithm visualization. In: **Computer Science Education Research**, p. 199–228, 2004.

STEIN, C.; DRYSDALE, R. L.; BOGART, K. Indução, recursão e recorrência. In: **Matemática Discreta para a Ciência da Computação**. Pearson, 2013.

URQUIZA-FUENTES, J.; VELÁZQUEZ-ITURBIDE, J. A. A survey of successful evaluations of program visualization and algorithm animation systems. **Trans. Comput. Educ.** Vol. 9, n. 2, p. 9:1–9:21, 2009.

VAN SOLINGEN, R.; BASILI, V.; CALDIERA, G.; ROMBACH, H. D. **Goal Question Metric (GQM) Approach**. In: Encyclopedia of Software Engineering. John Wiley & Sons Inc, 2002.