

Jonathas Gonçalves Picoli

**DESENVOLVIMENTO DE UM SOFTWARE EDUCACIONAL PARA AUXÍLIO NA  
APRENDIZAGEM DE GRAFOS EM DISCIPLINAS DE COMPUTAÇÃO**

Trabalho de Conclusão de Curso apresentado à Coordenadoria do Sistemas de Informação do Instituto Federal do Espírito Santo, como requisito parcial para a obtenção do título de Graduação em Sistemas de Informação.

Prof. Rafael Vargas Mesquita

Cachoeiro de Itapemirim  
2018

Jonathas Gonçalves Picoli

**DESENVOLVIMENTO DE UM SOFTWARE EDUCACIONAL PARA AUXÍLIO NA  
APRENDIZAGEM DE GRAFOS EM DISCIPLINAS DE COMPUTAÇÃO**

Trabalho de Conclusão de Curso apresentado à Coordenação do Sistemas de Informação do Instituto Federal do Espírito Santo, como requisito parcial para a obtenção do título de Graduação em Sistemas de Informação.

Aprovado em 12 de Junho de 2018.

**COMISSÃO EXAMINADORA**

Prof. Rafael Vargas Mesquita  
Instituto Federal do Espírito Santo  
Orientador

Prof. Eros Estevão de Moura  
Instituto Federal do Espírito Santo

Prof. Flávio Izo  
Instituto Federal do Espírito Santo

## **DECLARAÇÃO DO AUTOR**

Declaro para os devidos fins de pesquisa acadêmica, didática e técnico-científica, que o presente Trabalho de Conclusão de Curso pode ser parcialmente utilizado desde que se faça referência à fonte e ao autor.

Cachoeiro de Itapemirim, 12 de Junho de 2018.

Jonathas Gonçalves Picoli

## **RESUMO**

Este trabalho apresenta o desenvolvimento de uma aplicação desktop, que permita ao professor deter ferramentas gráficas que possam auxiliá-lo a lecionar disciplinas de programação que incluam a teoria dos grafos, buscando tornar o processo mais claro e otimizando o resultado da aula. Para tal o software utiliza uma interface gráfica composta por elementos que se comportam de forma dinâmica, de acordo com a execução do algoritmo escolhido, destacando-se a exibição do código fonte ao lado do desenho do grafo, evidenciando seu funcionamento e facilitando o seu entendimento por parte dos alunos. Foram implementadas as funcionalidades de exploração para dois algoritmos de busca, sendo um de busca em largura (BFS) e outro de busca em profundidade (DFS). Através da metodologia de avaliação de softwares educacional desenvolvida por Thomas Reeves, verificou-se que a ferramenta atingiu um grau satisfatório de desempenho, tendo pontuação acima de 6 em todas as questões avaliadas.

Palavras-chave: Teoria dos Grafos. Educação. Software.

## **ABSTRACT**

This work presents the development of a desktop application that allows the teacher to hold graphical tools that can help him to teach programming disciplines that include graph theory, seeking to make the process clearer and optimizing the result of the lesson. For this the software uses a graphical interface composed by elements that behave dynamically, according to the execution of the chosen algorithm, highlighting the display of the source code next to the drawing of the graph, evidencing its operation and facilitating its understanding by the students. Exploration features were implemented for two search algorithms, one search in width (BFS) and one search in depth (DFS). Through the methodology of evaluation of educational software developed by Thomas Reeves, it was verified that the tool reached a satisfactory degree of performance, reaching more than 60 % of the score above 7 in the pedagogical part, increasing to 94 % in the questions about the interface criterion.

Keywords: Theory of graphs. Education. Software.

## LISTA DE FIGURAS

Figura 1-	Abstração do problema da pontes de Königsberg . . . . .	15
Figura 2-	Representação gráfica ilustrativa de uma rede social . . . . .	17
Figura 3-	Resultado de árvore geradora Mínima . . . . .	18
Figura 4-	Tela de resultado da aplicação AlgoDeGrafos . . . . .	19
Figura 5-	Tela do jogo WarGrafos durante uma partida . . . . .	20
Figura 6-	Tela durante a execução do algoritmo Dijkstra . . . . .	21
Figura 7-	Tela de resultado da aplicação A-Graph . . . . .	21
Figura 8-	Tela principal da aplicação TGrafos . . . . .	22
Figura 9-	Pseudocódigo do algoritmo DFS . . . . .	26
Figura 10-	Pseudocódigo do algoritmo BFS . . . . .	27
Figura 11-	Diagrama de caso de uso . . . . .	28
Figura 12-	Tela inicial . . . . .	30
Figura 13-	Tela do algoritmo BFS em execução . . . . .	31
Figura 14-	Tela do algoritmo DFS em execução . . . . .	31
Figura 15-	Tela final do algoritmo BFS . . . . .	33
Figura 16-	Tela final do algoritmo DFS . . . . .	33
Figura 17-	Média geral das notas . . . . .	36
Figura 18-	Média dos módulos do formulário . . . . .	37

## LISTA DE TABELAS

Tabela 1-	Exemplos de aplicação de grafos . . . . .	16
Tabela 2-	Comparativo entre ferramentas similares . . . . .	22

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Objetivos	12
1.1.1	<b>Geral</b>	<b>12</b>
1.1.2	<b>Específicos</b>	<b>12</b>
<b>2</b>	<b>Fundamentação Teórica</b>	<b>13</b>
2.1	Software Educacional	13
2.1.1	<b>O que é</b>	<b>13</b>
2.1.1.1	Tipos de software educacional	13
2.1.2	<b>Por que utilizar um software educacional?</b>	<b>13</b>
2.1.3	<b>Barreiras na utilização</b>	<b>14</b>
2.1.4	<b>Potencialidades</b>	<b>14</b>
2.2	Grafos	15
2.2.1	<b>Conceito</b>	<b>15</b>
2.2.2	<b>Contexto Histórico</b>	<b>15</b>
2.2.3	<b>Aplicabilidade</b>	<b>16</b>
2.2.3.1	Relações entre os pontos	16
2.2.3.2	Arvore Geradora Mínima	17
2.3	Trabalhos Relacionados	18
2.3.1	<b>Comparativo</b>	<b>22</b>
<b>3</b>	<b>Metodologia da Pesquisa</b>	<b>24</b>
3.1	Tecnologias e ferramentas	24
3.2	Escolha dos algoritmos	25
3.2.1	<b>DFS</b>	<b>25</b>
3.2.2	<b>BFS</b>	<b>26</b>
<b>4</b>	<b>Resultados</b>	<b>28</b>
4.1	Modelagem	28
4.2	Código	28
4.3	Aplicação	29
4.3.1	<b>Tela Inicial</b>	<b>29</b>
4.3.2	<b>Algoritmo DFS e BFS</b>	<b>30</b>
4.4	Validação	34
4.5	Questionário aplicado	34
4.5.1	<b>Atributos Pedagógicos</b>	<b>34</b>
4.5.2	<b>Atributos de interface gráfica</b>	<b>35</b>
4.6	Resultados obtidos	35
<b>5</b>	<b>Conclusão</b>	<b>38</b>
	Referências Bibliográficas	39



## 1 Introdução

O ensino superior é a porta de entrada dos estudantes para o mercado de trabalho. Ele ajuda a definir o quão qualificados serão os profissionais inseridos nas mais diversas áreas. Atualmente o mercado de trabalho exige cada vez mais dos profissionais, tornando a graduação requisito mínimo para diversas funções. Esta necessidade de profissionais mais capacitados incentiva as Instituições de Ensino Superior (IES), a ofertarem mais vagas.

O censo da educação superior, revelou que em 2014 haviam mais de 8 milhões de egressos na graduação (INEP, 2015). Apesar do número expressivo, este cenário apresenta um dado preocupante. Ainda segundo Inep, em 2015 existiam 140 mil vagas ociosas somente na rede federal de ensino superior. Este número além de representar prejuízos as IES, deixa clara a dificuldade em suprir as necessidades do mercado. O principal motivo para tantas vagas ociosas está na alta porcentagem de evasão escolar.

Em 2014, 49% dos alunos abandonaram o curso para o qual foram aprovados (INEP, 2015). Apesar de preocupante esta estatística pode ser ainda pior quando se trata dos cursos de tecnologia, dada o alto grau de abstração que as disciplinas de programação exigem dos alunos, principalmente as que abordam recursividade como estruturas de armazenamento e busca. O percentual médio de evasão escolar nos cursos de TI chega a 87% (BARCELOS; SILVEIRA, 2012).

Os avanços tecnológicos tornam o cotidiano dos alunos algo extremamente dinâmico, e isso gera um choque de realidade em relação a sala de aula, onde métodos rígidos e com poucas variações são utilizados para fazer com que o aluno absorva a informação. Esta realidade torna o ensino desinteressante e incentiva o abandono dos estudos.

Os alunos necessitam dominar o processo de aprendizagem para o desenvolvimento de suas competências, e não mais absorver somente o conteúdo. Faz-se necessária uma educação permanente, dinâmica e desafiadora visando o desenvolvimento de habilidades para a obtenção e utilização das informações (MORATORI, 2003).

Diante da dificuldade em manter os alunos interessados em sua formação superior, da alta porcentagem de evasão escolar nos cursos de tecnologia e cientes das necessidades de dinamizar o processo de ensino, este trabalho apresenta a proposta de desenvolvimento da aplicação EasyGrafos.

## **1.1 Objetivos**

### **1.1.1 Geral**

Desenvolver um software educacional para auxílio na aprendizagem de grafos em disciplinas de programação, tendo como objetivo principal possibilitar ao aluno a capacidade de atingir o grau de abstração necessário para compreender os conceitos e os algoritmos da teoria dos grafos, demandando menos tempo do professor e assim otimizando seu trabalho.

### **1.1.2 Específicos**

1. Ilustrar de forma gráfica a ação dos algoritmos no grafo, bem como, o status de suas variáveis.
2. Associar cada linha do código fonte do algoritmo escolhido com uma ação na interface gráfica.
3. Disponibilizar controles para possibilitar um modo debug.

## **2 Fundamentação Teórica**

### **2.1 Software Educacional**

Com o aumento do interesse no ingresso em universidades, andando em paralelo com um alto índice de evasão escolar, torna-se necessário novas formas de apresentar os conteúdos na sala de aula. Uma dessas formas pode ser exemplificada através dos softwares educacionais. Um exemplo é a plataforma virtual MOODLE, que hoje é uma ferramenta disseminada no âmbito acadêmico mundial, sendo disponibilizada em mais de 100 idiomas para milhões de usuários.

#### **2.1.1 O que é**

Várias são as definições para o termo software educacional, porém o mais adaptável a solução proposta por esse trabalho afirma que este é todo aquele programa que possa ser usado para algum objetivo educacional, pedagogicamente defensável, por professores e alunos, qualquer que seja a natureza ou finalidade para o qual tenha sido criado (TEIXEIRA, 2003).

##### **2.1.1.1 Tipos de software educacional**

Existem algumas categorias de software educacional, basicamente suas diferenças se dão no objetivo, assim de acordo com o objetivo, estas devem adotar metodologias específicas. Podemos destacar três tipos de software educacionais (GIRAFFA, 2009), sendo eles:

- Programas de reforço ou exercício - o aluno pratica e testa conhecimentos de formas dirigida e mais procedural. As versões mais atuais destes programas utilizam recursos hipermídia mantendo estas características.
- Tutoriais - seguem o padrão de ensino da sala de aula tradicional onde o conteúdo é previamente organizado numa estrutura definida pelo professor e o aluno seleciona dentre as diversas opções disponíveis o conteúdo que deseja estudar.
- Jogos educacionais e as simulações - São tarefas a serem resolvidas, onde os jogos possuem caráter competitivo e as simulações são meramente repetições de exercícios.

#### **2.1.2 Por que utilizar um software educacional?**

O computador esta em praticamente todos os ambientes que frequentamos, e isso também é válido para os alunos. O motivo disso esta na forma como eles chamam a

atenção das pessoas, devido as variadas formas de expor a informação, seja ela, propagandas, músicas, notícias e etc. Logo, faz-se necessário criar esse efeito também dentro da sala de aula, caso contrário esta será sempre a segunda opção. É preciso refletir sobre as metodologias de ensino levando em consideração a realidade vivida dessa nova sociedade (SILVA, 2014).

O computador pode ser também utilizado para enriquecer ambientes de aprendizagem e auxiliar o aprendiz no processo de construção do seu conhecimento (VALENTE, 1999). Não faltam variações na forma de utilizar tais ferramentas, já que a tecnologia avança rapidamente criando novos recursos para torna-la ainda mais atrativa, como hoje popular realidade virtual.

### **2.1.3 Barreiras na utilização**

Entretanto essas ferramentas vem apresentando alguns problemas que as impossibilitam de atingir todo o potencial, tornando assim responsabilidade do professor explorar os espaços virtuais e suas possibilidades. A matemática é uma área que possui grande variedade de aplicações que auxiliam no processo de aprendizagem, entretanto essa diversificação não implica em qualidade. Muitos programas possuem conteúdo mal formulado, problemas na execução do sistema, interfaces (telas) confusas e assim por diante (GIRAFFA, 2009).

Outro fator que deve ser observado é a dispersão. Os professores em alguns momentos não possuem ferramentas atrativas o suficiente para manter a atenção dos alunos durante a aula. Os aparelhos portáteis por exemplo oferecem dificuldade quanto a fiscalização da utilização na sala e em contrapartida, possibilita ao usuário muitas formas de distração. A falta de atenção, a desmotivação dos alunos é algo frustrante para todo o corpo o docente. A utilização dos celulares, é apontada como motivo para dispersão e falta de interesse dos alunos (SILVA, 2014).

### **2.1.4 Potencialidades**

Apesar dos problemas enfrentados os softwares educacionais possuem uma vantagem em relação a didática comum, que é a visualização em imagens e outras mídias, que são métodos importantes no auxílio da compreensão da abstração que as disciplinas de programação exigem.

O não mostrar equivale a não existir, a não acontecer. O que não se vê perde a existência. Um fato mostrado com imagem e palavra tem mais força, do que se for mostrado somente com palavra. Muitas situações importantes do cotidiano perdem força por não ter sido valorizadas pela imagem-palavra televisiva(MORAN, 2000).

## 2.2 Grafos

### 2.2.1 Conceito

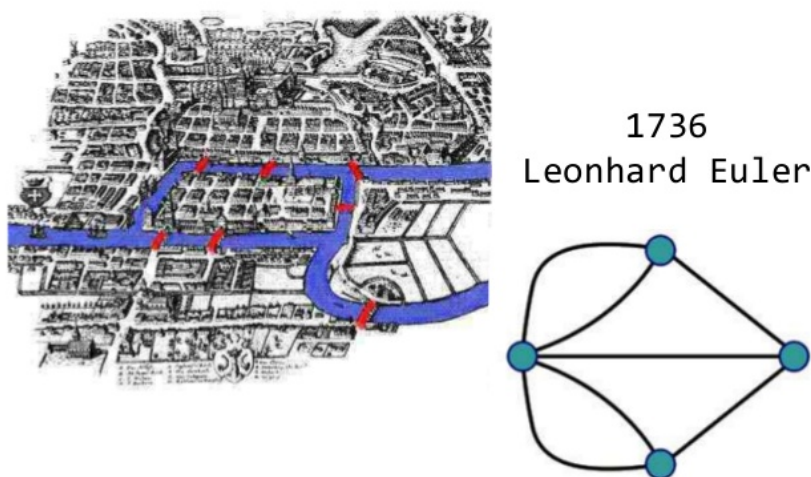
Um grafo pode ser definido como uma estrutura composta de pontos e ligações entre eles (JURKIEWICZ, 2009). Sua representação pode ser dada como sendo um grafo( $G$ ), composto por um conjunto finito de vértices  $V(G)$ , ligados por um conjunto finito de arestas  $A(G)$ .

### 2.2.2 Contexto Histórico

O primeiro estudo que se caracterizou como teoria dos grafos, ocorreu em 1736 na cidade de Königsberg, e foi realizado pelo matemático suíço Leonhard Euler. O estudo se baseava na estrutura da cidade, a mesma possuía duas ilhas e 7 pontes ligavam suas margens. O problema a ser resolvido era avaliar a possibilidade de sair de um ponto, passar por todas as 7 pontes exatamente uma vez retornando ao ponto de origem.

Para resolver o problema Euler montou um diagrama para representar a cidade, que pode ser visualizado na figura 1. Nele cada margem foi associada a um vértice e cada ponte a uma aresta. Sua conclusão foi que para realizar o percurso com as condições impostas, cada vértice deveria ter um número par de vértices ligados a ele, como haviam vértices ligados a três arestas o percurso se mostrou impossível(COSTA, 2011).

Figura 1 – Abstração do problema da pontes de Königsberg



## As pontes de Königsberg

Fonte: [http://www.obm.org.br/content/uploads/2017/01/Nivel1\\_grafos-bruno.pdf](http://www.obm.org.br/content/uploads/2017/01/Nivel1_grafos-bruno.pdf)

### 2.2.3 Aplicabilidade

O conceito de grafos pode ser empregado em diversos problemas da atualidade. Suas características o tornam modelo para situações complexas em inúmeras áreas de estudo, e assim possibilitam a aplicação de métodos da Teoria dos Grafos para analisar de forma mais precisa tais situações. A tabela 1, apresenta algumas das muitas abstrações possíveis em problemas das mais diversas áreas.

Tabela 1 – Exemplos de aplicação de grafos

<b>grafo</b>	<b>vértice</b>	<b>aresta</b>
comunicação	telefone, computador	cabo de fibra óptica
circuito	porta, processador	corrente
mecânica	articulação	haste, viga, mola
financeiro	estoque, moeda	transações
transporte	intercessão	rua
internet	rede de classe C	conexão
jogos	posição do tabuleiro	movimento legal
rede social	pessoa	amizade
rede neural	neurônio	sinapse
rede proteica	proteína	interação proteína-proteína
molécula	átomo	ligação

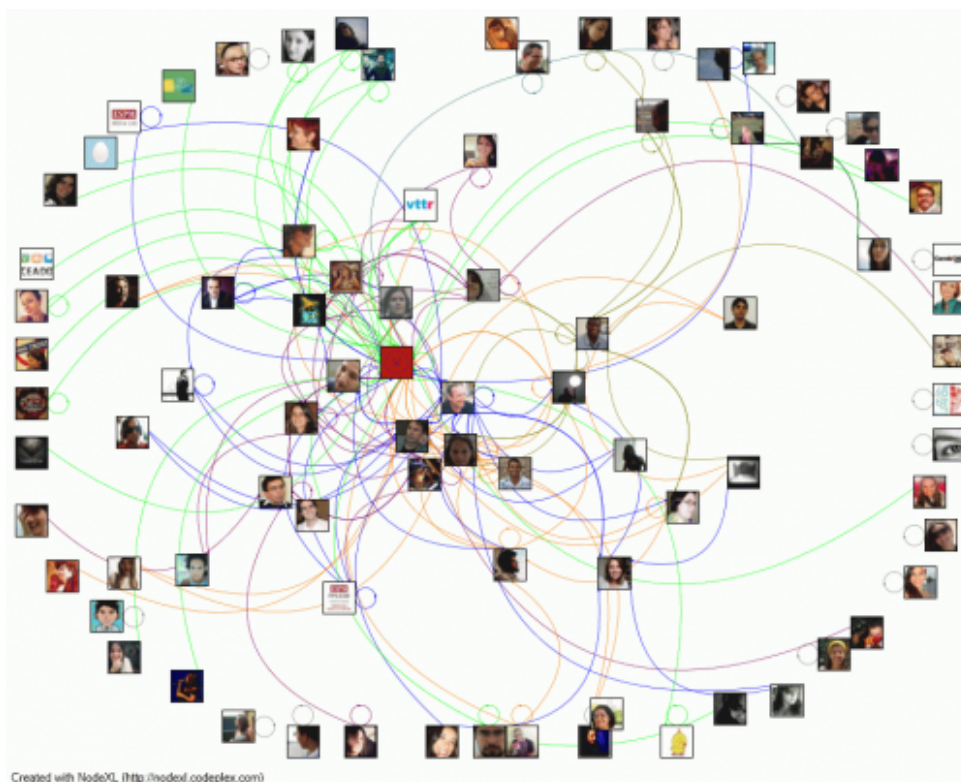
Fonte: Material da disciplina de TPA

Para ilustrar usos destes conceitos, será explanado duas categorias de problemas presentes na tabela 1, a relação entre 2 pontos, presente em todos os exemplos e a árvore geradora mínima, que pode se fazer necessária dependendo do objetivo ao explorar a estrutura.

#### 2.2.3.1 Relações entre os pontos

Atualmente este conceito pode ser claramente associado as redes sociais digitais, representadas de forma simplificada pela figura 2, onde existem usuários, que são representados pelos pontos e suas amizades representadas pelas relações. Assim novas possibilidades de análise desses grupos podem ser realizadas.

Figura 2 – Representação gráfica ilustrativa de uma rede social



Fonte: [http://www.raquelrecuero.com/assets\\_c/2013/06/compowsfinalcluster-thumb-500x403-501.png](http://www.raquelrecuero.com/assets_c/2013/06/compowsfinalcluster-thumb-500x403-501.png)

Cartwright junto com o matemático Harary ainda na década de 50 iniciou estudos sobre os comportamentos dos grupos, propondo que pessoas próximas tendem a agir de forma similar de frente a terceiros. Estes estudos possibilitaram a criação de modelos da interdependência sistemática entre pessoas diferentes dentro de um mesmo grupo (BRAGA; GOMES; RUEDIGER, 2008).

A aplicação deste conceito esta inserida nas mais variadas áreas de estudo. Na sociologia por exemplo, a teoria dos grafos é uma das bases no estudo das redes sociais, onde esta possibilita a analise das redes como um todo e não somente em parte, possibilitando a exploração de propriedades antes não expostas (RECUERO, 2004).

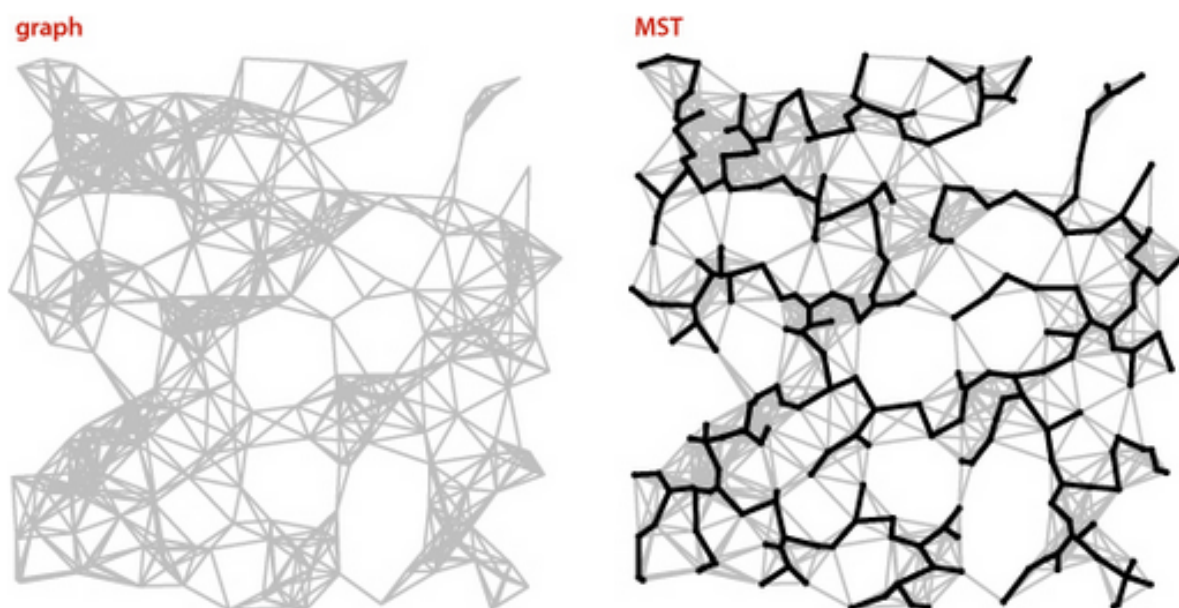
### 2.2.3.2 Árvore Geradora Mínima

O conceito de árvore geradora mínima (MST) pode ser descrito como um caminho que liga todos os nós da rede sem gerar ciclos, porém devemos observar que um mesmo grafo pode ter várias árvores geradoras mínimas, sendo a solução ideal a de menor custo, sabendo que o custo é a soma dos pesos das aresta que fazem parte da solução (ALMEIDA et al., 2006).

Para exemplificar esse conceito suponhamos o seguinte problema. A figura 3, ilustra com linhas cinza claro no primeiro quadro da imagem todas as possibilidades de caminho. Nesse momento ao aplicarmos um algoritmo MST, este nos dará qual o caminho

de menor custo, sendo este ilustrado com linhas pretas.

Figura 3 – Resultado de árvore geradora Mínima



Fonte: [https://www.ime.usp.br/pf/algoritmos\\_para\\_grafos/aulas/mst.html](https://www.ime.usp.br/pf/algoritmos_para_grafos/aulas/mst.html)

## 2.3 Trabalhos Relacionados

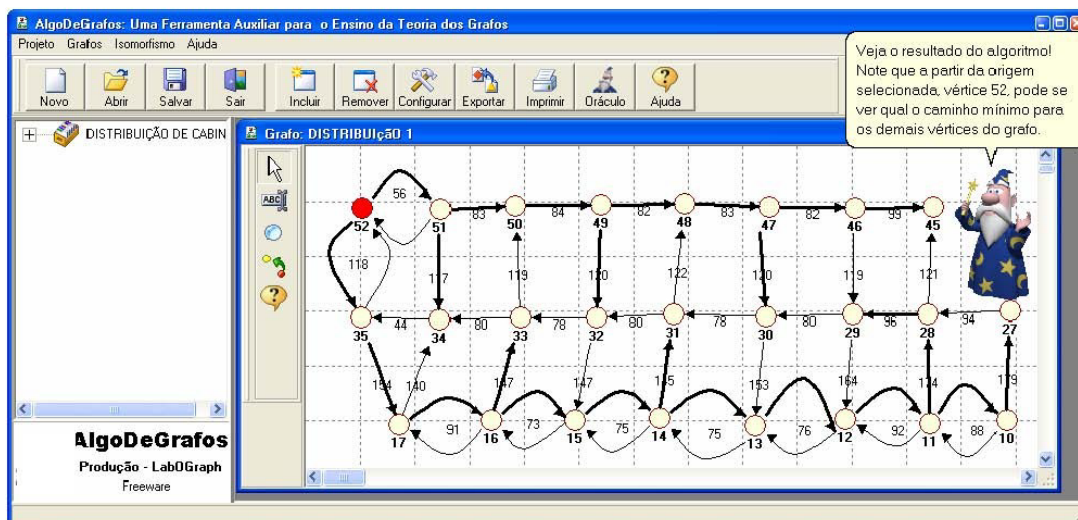
Existem hoje alguns softwares para auxiliarem no estudo da teoria dos grafos. Estes utilizam diferentes ferramentas para facilitar a compreensão da disciplina. Foram selecionadas algumas dessas aplicações afim de verificar a necessidade de desenvolvimento de uma nova aplicação para o gênero. Estas foram encontradas através de pesquisas na plataforma Google<sup>1</sup>, utilizando os termos de pesquisa "ensino e aprendizagem de grafo" e "aplicativo para aprendizagem de grafos em computação". As ferramentas deveriam possuir o mesmo propósito da ferramenta deste presente trabalho. A seguir serão citadas estas ferramentas e suas principais características.

- **AlgoDeGrafos** é uma aplicação que utiliza interface gráfica para ilustrar o resultado da execução de algoritmos para exploração de grafos, não apresentando ao usuário o código do algoritmo e não possibilitando a execução passo a passo. Há uma grande variedade de algoritmos habilitados na ferramenta e o usuário pode criar seus próprios grafos, podendo salva-los para posterior estudo. O principal objetivo da ferramenta é o auxílio no modelo EAD de educação (MELO; SILVEIRA; JURKIEWICZ, 2009). A figura 4 apresenta a tela do resultado final da execução de um algoritmo na aplicação AlgoDeGrafos.

<sup>1</sup><https://www.google.com>



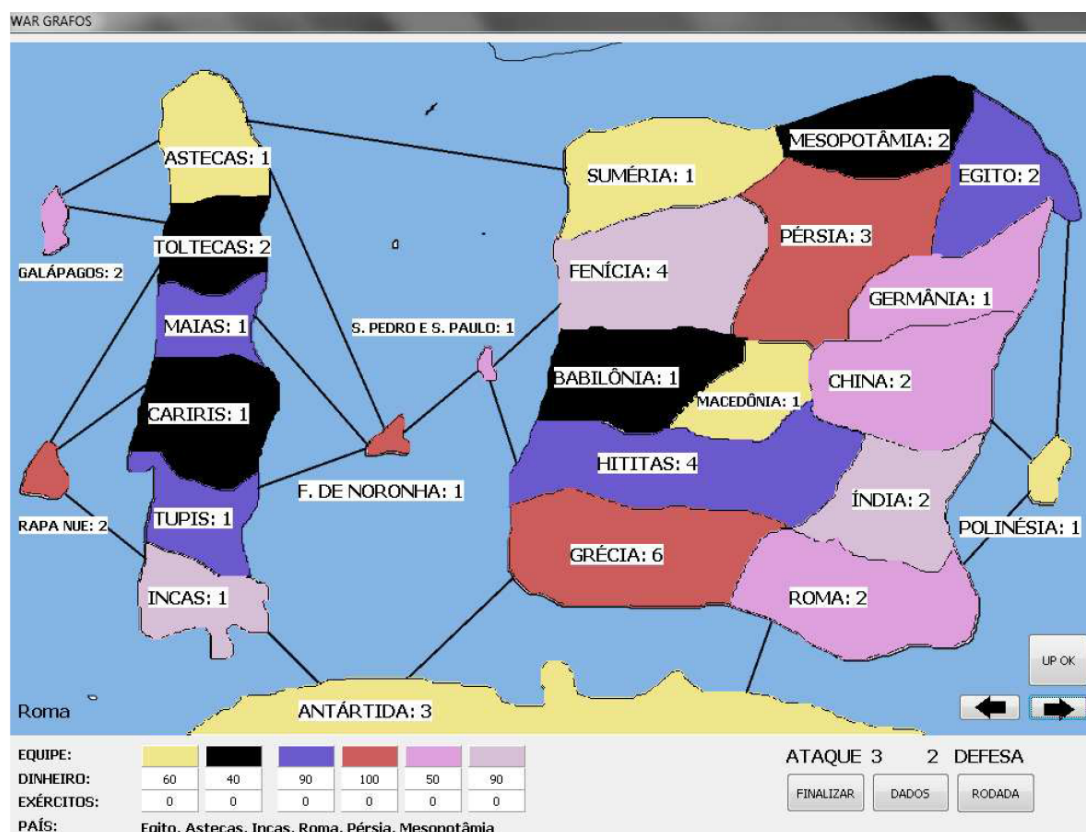
Figura 4 – Tela de resultado da aplicação AlgoDeGrafos



(MELO; SILVEIRA; JURKIEWICZ, 2009)

- **WarGrafos** é um jogo baseado no jogo de tabuleiro War lançado pela empresa Grow. Nele os alunos são divididos em equipes e cada uma recebe um objetivo, sendo que o objetivo das equipes inimigas não é conhecido, ou seja, cada equipe conhece apenas o seu. Para alcançar o objetivo e vencer a partida a equipe deve conhecer e empregar conceitos da teoria dos grafos (FIGUEIREDO; FIGUEIREDO, 2011). A figura 5 apresenta a tela durante uma partida do jogo.

Figura 5 – Tela do jogo WarGrafos durante uma partida



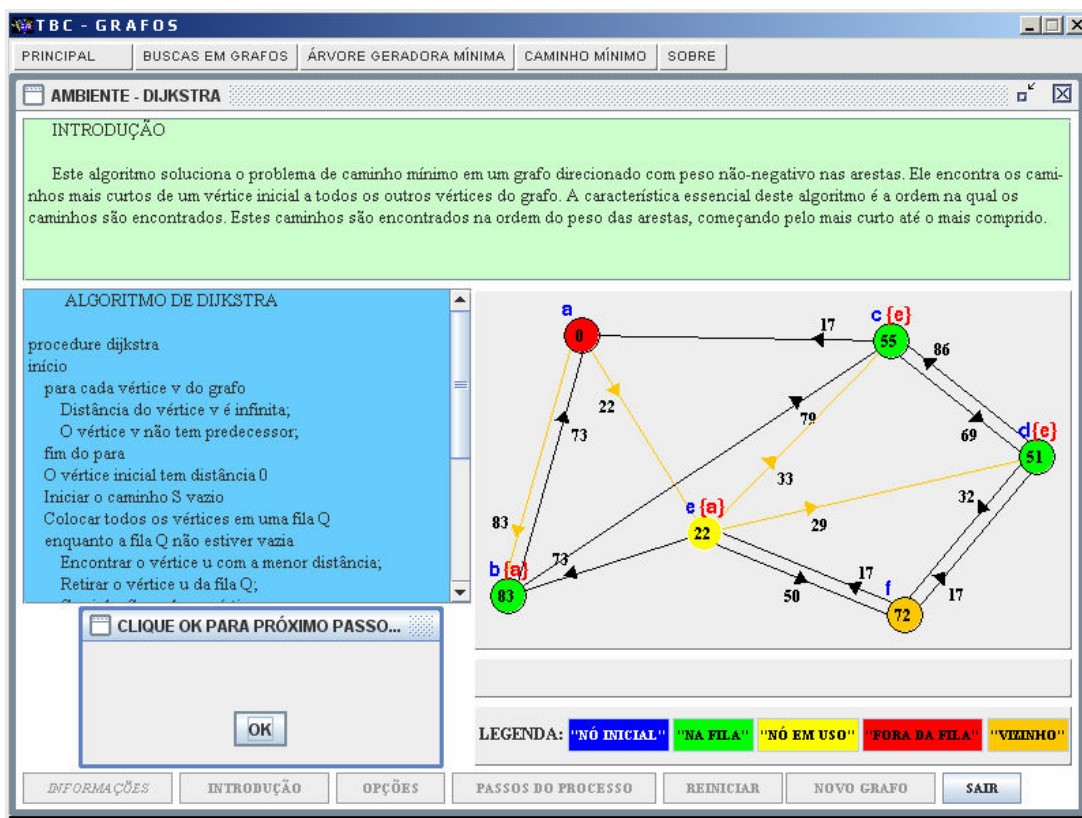
(FIGUEIREDO; FIGUEIREDO, 2011)

- **TBC-GRAFOS** é um software que visa facilitar a visualização da execução de algoritmos em estruturas de grafos. Ele se utiliza da apresentação de alguns conceitos teóricos simples, possibilidade de execução passo a passo, legendas explicativas e apresentação de um Pseudocódigo do algoritmo escolhido (SANTOS et al., 2008). Ele aborda os seguintes conceitos:

1. Busca buscas em grafos (percurso em profundidade e em largura)
2. Árvore geradora mínima (algoritmos de Kruskal e de Prim)
3. E caminho mínimo entre vértices (algoritmos de Dijkstra e de Bellman-Ford)

A figura 6 apresenta um exemplo de execução do algoritmo Dijkstra na aplicação TBC-GRAFOS

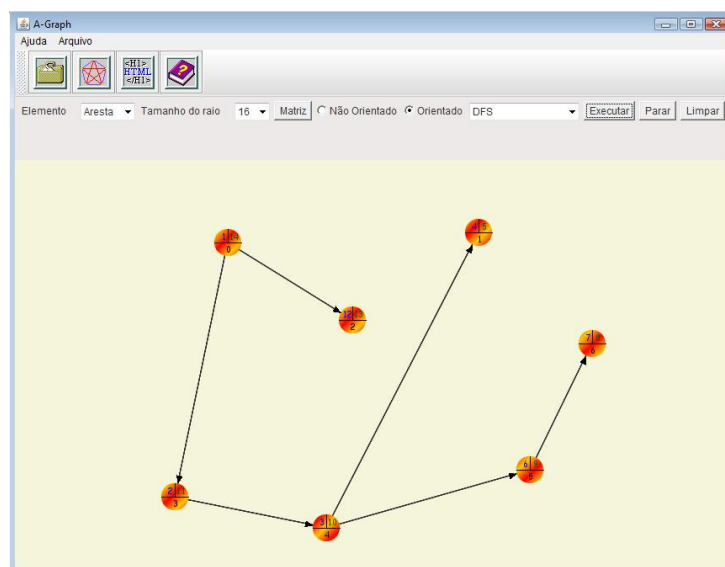
Figura 6 – Tela durante a execução do algoritmo Dijkstra



(SANTOS et al., 2008)

- **A-Graph** visa a criação de grafos utilizando interface gráfica e possibilita a execução de dois algoritmos a busca em largura e em profundidade. A aplicação não exibe o código ao usuário e não possibilita a execução passo a passo (LOZADA, 2014). A figura 7 ilustra um exemplo de execução na ferramenta.

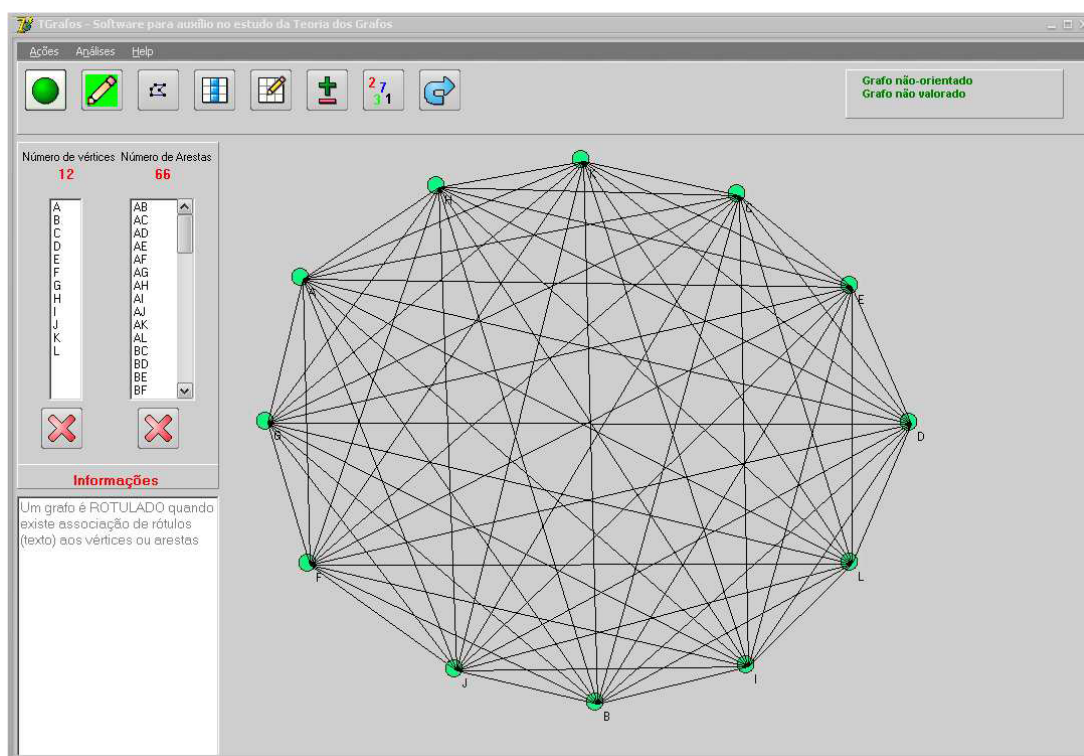
Figura 7 – Tela de resultado da aplicação A-Graph



(LOZADA, 2014)

- **TGrafos** é um software para criação e estudo de grafos. Apresenta diversos recursos em relação a teoria da disciplina e em relação a características do grafo criado. Não há a possibilidade de executar algoritmos nos grafos criados (SILVEIRA; SILVA, 2016). A figura 8 apresenta a tela principal da aplicação TGrafos.

Figura 8 – Tela principal da aplicação TGrafos



(SILVEIRA; SILVA, 2016)

### 2.3.1 Comparativo

Baseado nas principais características das aplicações citadas, foi possível realizar um comparativo, entre estas e a aplicação proposta pelo trabalho, intitulada EasyGrafos. O resultado pode ser visualizado na tabela 2.

Tabela 2 – Comparativo entre ferramentas similares

Característica/Software	WarGrafos	TBC-Grafos	AlgoDeGrafos	A-Graph	Tgrafo	EasyGrafos
Interface para manipulação e análise dos grafos		X	X	X	X	X
Criar os próprios grafos			X	X	X	X
Executar algoritmos da disciplina nos grafos		X	X	X	X	X
Execução em modo debug		X	X			X
Associação do código fonte do algoritmo em estudo ao grafo						X
Utilização de técnicas de gameificação	X					

Fonte: Próprio autor

O estudo destas ferramentas e a avaliação da tabela comparativa, nos possibilita aferir os diferenciais entre a aplicação proposta e as demais já disponíveis. A principal

delas é a associação do código fonte do algoritmo selecionado, ao desenho do grafo, que em conjunto com os controles de modo debug (disponíveis na aplicação EasyGrafos), entregam ao professor vastas possibilidades de ministrar a aula. Nenhuma das aplicações pesquisadas apresentou este recurso de código fonte. Outro diferencial é a própria execução em modo debug, que só é disponibiliza por menos da metade das aplicações pesquisadas.

### 3 Metodologia da Pesquisa

Para o desenvolvimento do objeto de aprendizagem proposto neste projeto, foram realizadas as seguintes etapas:

- Estudo sobre as principais ferramentas para o ensino de grafos na área da ciência da computação;
- Elaboração de cinco arquivos contendo dados de grafos exemplos, com diferentes características (grafos densos, grafos esparsos, etc.), para utilização na ferramenta;
- Implementação dois algoritmos clássicos que utilizam o conceito sobre teoria dos grafos: busca em largura e busca em profundidade;
- Implementação de duas telas para a exibição da execução dos dois algoritmos mencionados anteriormente;
- Documentação do código em javadoc;
- Escolha das questões do formulário de avaliação: As questões escolhidas foram retiradas da metodologia para avaliação de software educacional de Thomas Reeves. Foram escolhidas 5 questões de carácter pedagógico e 6 referentes a interface com o usuário.
- Validação o objeto de aprendizagem: nesta etapa a ferramenta será disponibilizada para a turma da disciplina de técnicas avançadas de programação do curso de sistemas de informação do Ifes Campus Cachoeiro de Itapemirim. Os alunos, após a utilização da ferramenta, responderão a um questionário de avaliação;

#### 3.1 Tecnologias e ferramentas

O software produzido utiliza como tecnologia para o desenvolvimento da aplicação a linguagem de programação Java na versão 8, utilizando-se das bibliotecas gráficas do JavaFX, para produzir as animações necessárias.

A codificação foi realizado seguindo o paradigma de orientação a objetos e a estruturação do código foi realizada seguindo o padrão de arquitetura de software MVC.

Os protótipos das telas foram desenvolvidos baseados em listas de exercícios utilizadas na disciplina de técnicas de programação avançada(TPA) e em materiais de (SEDGEWICK; WAYNE, 2011). Com isso foi possível adicionar ao protótipo elementos gráficos previamente pensados em auxiliar a aula, buscando focar os componentes

mais importantes do algoritmo em execução.

## **3.2 Escolha dos algoritmos**

Foram escolhidos dois algoritmos para serem implementados na ferramenta, o Algoritmo de busca em largura (BFS) e o algoritmo de busca em profundidade (DFS). A escolha desses algoritmos se deu baseada na ementa da disciplina de Técnicas de programação avançadas, do curso de Sistemas de Informação.

Esta disciplina abrange 4 conceitos da teoria de grafos, a busca em largura, busca em profundidade, árvore geradora mínima e caminho mínimo. Destes o DFS e o BFS são os primeiros, ou seja, são eles os utilizados para introduzir o conceito de grafos aos alunos, tornando-se importantes para a continuidade da disciplina, visto que os algoritmos restantes são mais complexos e precisam de uma base para serem compreendidos de forma satisfatória.

Os dois algoritmos escolhidos possuem o mesmo objetivo, encontrar partindo de um vértice qualquer, todos os vértices acessíveis a partir dele, entretanto cada um possuiu métodos diferente de chegar nesse resultado, como será explicado nas próximas seções.

### **3.2.1 DFS**

A principal característica desse algoritmo é sempre buscar o vértice mais profundo. Como forma de realizar esse tipo de exploração o algoritmo sempre passa a execução para alguma aresta que o liga a um vértice não visitado, assim este só para de avançar ou "afundar" no grafo quando encontra um vértice que não está ligado a vértices não explorados, ou seja, todas as arestas adjacentes a ele, o ligam a vértices já visitados, o caracterizando como vértice finalizado. Para garantir que vértices não sejam visitados mais de uma vez, sempre que a execução chega em um vértice, este é marcado como visitado e adicionado a uma pilha.

O objetivo da pilha é garantir que o primeiro vértice visitado seja o último a ser finalizado. Esta estrutura é importante pois a condição de desempilhamento no algoritmo é encontrar um vértice ligado apenas a vértices já visitados, logo, o primeiro vértice a ser finalizado deve ser o último encontrado, ela também garante que quando o primeiro vértice encontrado for finalizado, o algoritmo concluiu sua execução (SOUZA et al., 2013).

A figura 9 apresenta um pseudocódigo do algoritmo DFS, onde podemos observar que das linhas 7 a linha 11 temos um laço que verifica se cada vértice do grafo foi visitado, caso não tenha sido, visita-o.

Figura 9 – Pseudocódigo do algoritmo DFS

```

1 DFS(GRAFO(V, E))
2 PARA todo vértice i do GRAFO FAÇA
3   cor[i] ← BRANCO
4   pai[i] ← nulo
5 FIM_PARA
6 Tempo ← 0
7 PARA todo vértice i do GRAFO FAÇA
8   Se cor[i] = BRANCO ENTÃO
9     DFS-VISITA(i)
10  FIM-SE
11 FIM_PARA

1 DFS-VISITA(u)
2 cor[u] ← CINZA
3 tempo ← tempo + 1
4 inicio[u] ← tempo
5 PARA todo vértice i de Adj(u) FAÇA
6   Se cor[i] = BRANCO ENTÃO
7     pai[i] ← u
8     DFS-VISITA(i)
9   FIM-SE
10 FIM_PARA

```

Fonte: (SOUZA et al., 2013)

### 3.2.2 BFS

Diferente do DFS, o BFS explora os vértices de nível em nível. Este tipo de busca é caracterizado por visitar todos os vértices a uma distância  $k$  do vértice de origem antes de visitar os que estão a uma distância  $k + 1$ .

Aqui o conceito de vértice finalizado tem o mesmo significado do DFS, ele acontece quando todas as suas arestas já foram visitadas, nesse momento a exploração no nível  $k$  se encerra dando início a exploração no nível  $k + 1$ .

Para garantir a ordem de exploração dos vértices é utilizada uma estrutura de fila, esta garante que o primeiro vértice a ser visitado é também o primeiro a ser finalizado. A utilização da fila se dá da seguinte forma, sempre que um novo vértice é encontrado ele é adicionado na fila, quando o primeiro vértice é finalizado este é retirado da fila, dando vez para o próximo, ou seja, para o segundo vértice encontrado. A figura 10 apresenta um pseudocódigo do algoritmo BFS, que na linha 16 realiza a contagem de distância do vértice encontrado para o vértice de origem, somando a distância do vértice anterior mais 1.



Figura 10 – Pseudocódigo do algoritmo BFS

```

1 BFS (GRAFO(V, E), v)
2 PARA todo vértice i do GRAFO FAÇA
3   cor[i] ← BRANCO
4   d_arestas[i] ←  $\infty$ 
5   pai[i] ← nulo
6 FIM_PARA
7 d_arestas[v] ← 0
8 cor[v] ← CINZA
9 FILA ←  $\emptyset$ 
10 INSERE(FILA, v)
11 ENQUANTO FILA  $\neq \emptyset$  FAÇA
12   u ← REMOVE(FILA)
13   PARA todo vértice i de Adj(u) FAÇA
14     Se cor[i] = BRANCO ENTÃO
15       cor[i] ← CINZA
16       d_arestas[i] ← d_arestas[u] + 1
17       pai[i] ← u
18     INSERE(FILA, i)
19   FIM-SE
20 FIM_PARA
21 cor[u] ← PRETO
22 FIM_ENQUANTO

```

Fonte: (SOUZA et al., 2013)

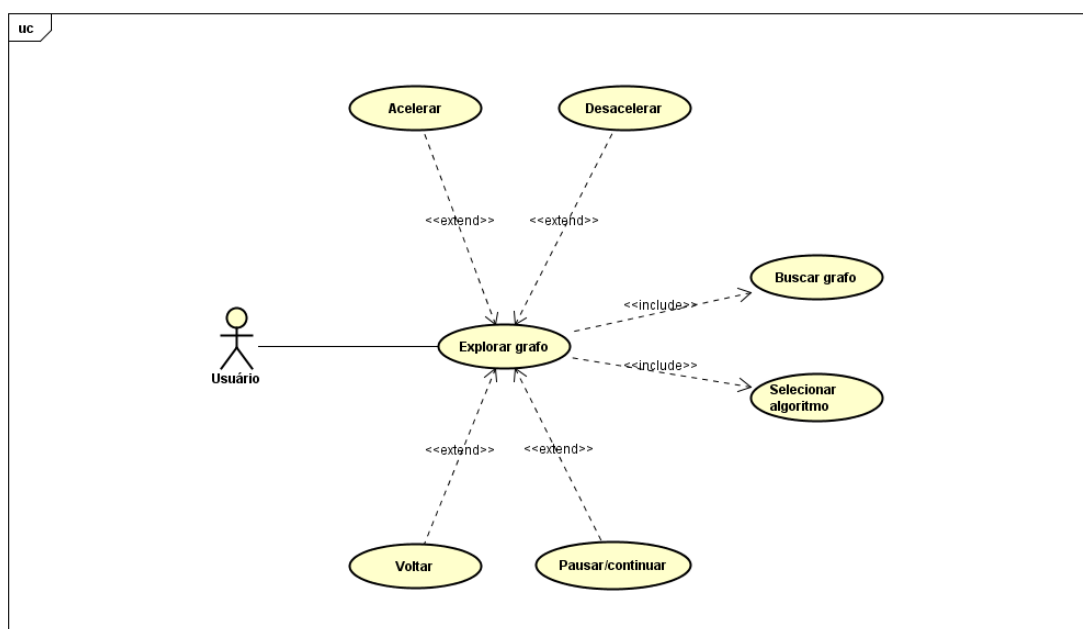
## 4 Resultados

Baseando-se nos conceitos apresentados na disciplina de TPA, e visando atender suas demandas, os seguintes artefatos foram desenvolvidos:

### 4.1 Modelagem

As funções disponibilizadas pela aplicação permitem ao usuário manusear a exploração do algoritmo de diversas formas. A figura 11 apresenta o diagrama de caso de uso da ferramenta.

Figura 11 – Diagrama de caso de uso



Fonte: Produzido pelo próprio autor

Conforme verificado na figura 11, a ação do usuário gira em torno de Explorar grafo, que pode estender outras operações acelerar e desacelerar a execução, pausar ou continuar e voltar para a tela anterior. Outro fator importante é que explorar grafo exige que um grafo e um algoritmo tenham sido escolhidos.

### 4.2 Código

A estrutura do software pode ser resumida nas pelas classes Grafo, Vértice, Aresta, AlgoritmoBFS, AlgoritmoDFS, RunnableBFS e RunnableDFS. Onde:

- A classe Grafo é a representação computacional do elemento grafo. Contendo os atributos necessários para possibilitar aos algoritmos a sua exploração, como

quantidades de vértices e arestas e como se da suas disposições, através da lista de adjacências.

- A classe vértice armazena os dados sobre os vértices, como a sua identificação e a sua posição.
- A classe aresta possui informações como principal valor os vértice de origem e destino.
- A AlgoritmoBFS consiste em uma representação do código do método de busca em grafos BFS, é efetivamente esta classe que explora os grafos.
- A AlgoritmoDFS consiste em uma representação do código do método de busca em grafos DFS, é efetivamente esta classe que explora os grafos.
- A classe RunnableBFS implementa a interface Runnable, possibilitando assim que suas instancias possam ser passadas como parâmetros para threads. Esta thread cuida das modificações na parte gráfica da aplicação, que são acionadas pela instancia da classe AlgoritmoBFS.
- A classe RunnableDFS implementa a interface Runnable, possibilitando assim que suas instancias possam ser passadas como parâmetros para threads. Esta thread cuida das modificações na parte gráfica da aplicação, que são acionadas pela instancia da classe AlgoritmoDFS.

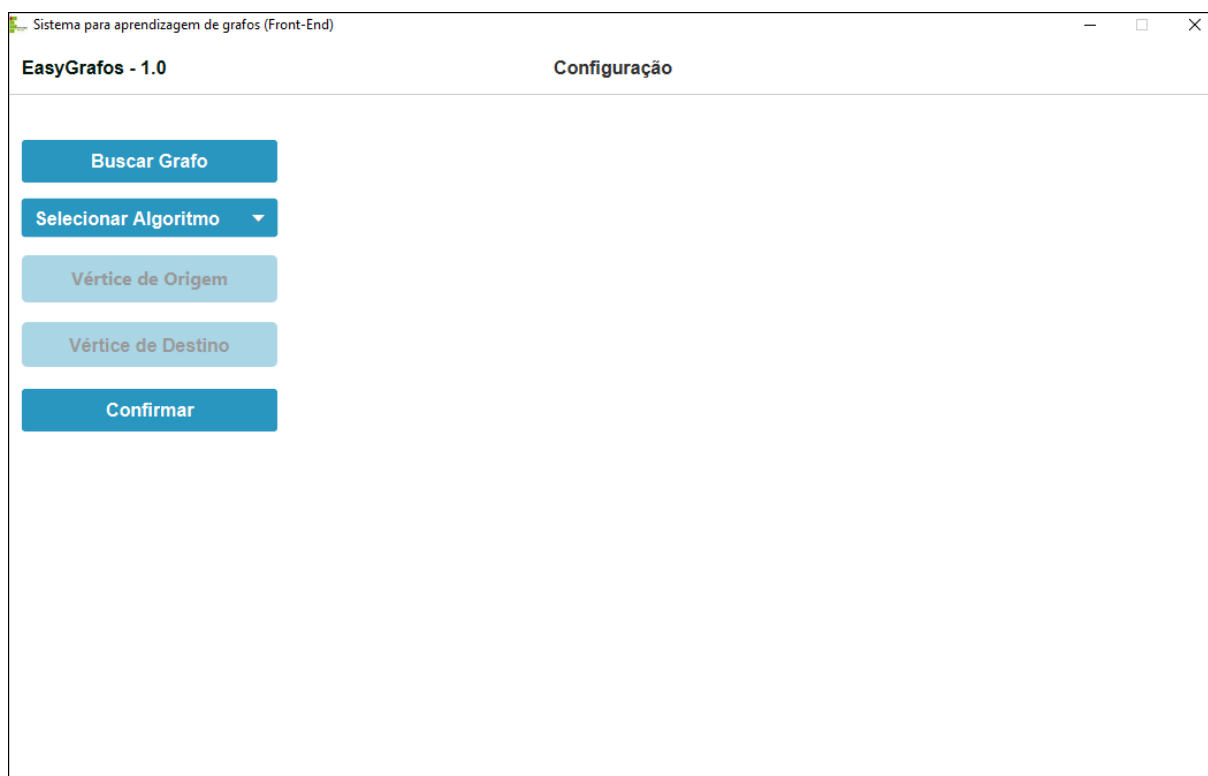
### **4.3 Aplicação**

O principal objetivo da ferramenta é preencher as lacunas deixadas por softwares similares. A maior delas é a apresentação do código fonte do algoritmo que o usuário escolheu. A ferramenta proposta irá proporcionar um modo debug de execução , onde a linha que está sendo executada ficará em destaque. A seguir serão apresentadas as telas da aplicação EasyGrafos, demonstrando os principais estados da ferramenta durante sua execução.

#### **4.3.1 Tela Inicial**

A figura 12 apresenta a primeira tela da ferramenta. Nela o usuário irá preencher os parâmetros necessários para utilização da ferramenta. Nesta etapa o usuário deve escolher um grafo para ser explorado e um algoritmo para explorá-lo. Após, os parâmetros necessários para execução do algoritmo serão solicitados para que a função confirmar possa ser acionada.

Figura 12 – Tela inicial

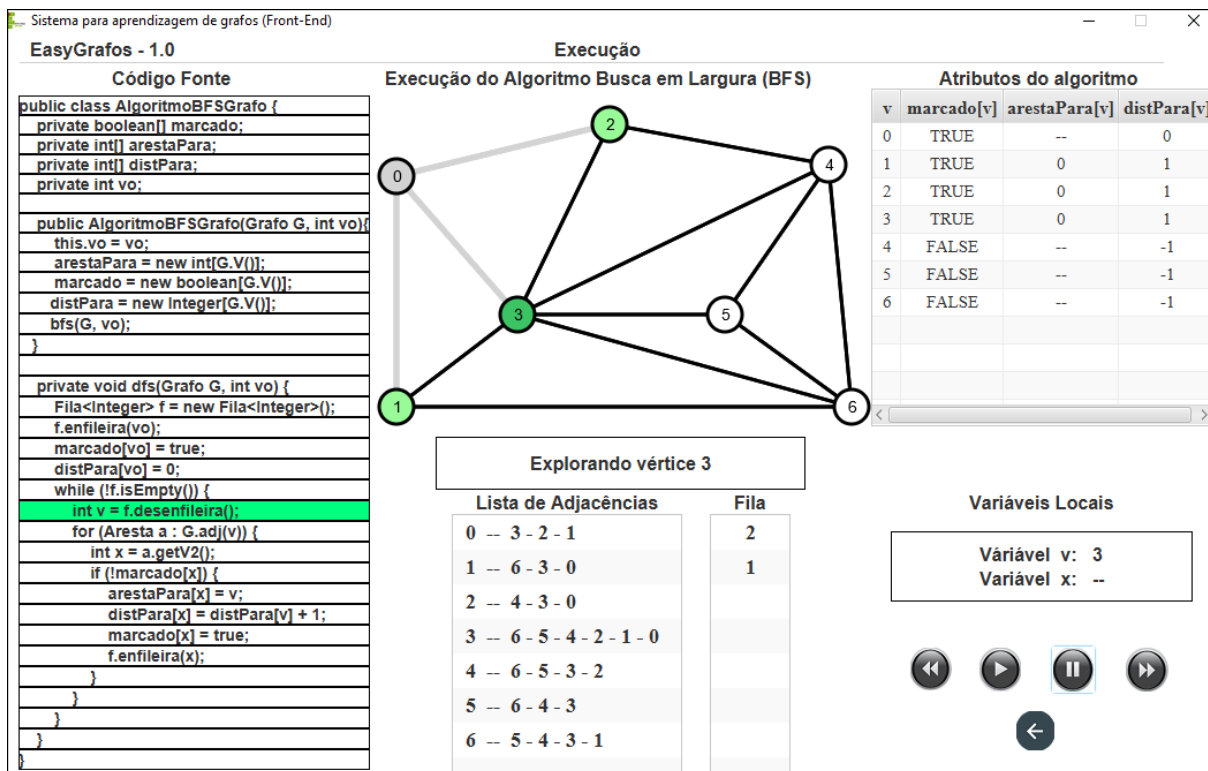


Fonte: Próprio autor

#### 4.3.2 Algoritmo DFS e BFS

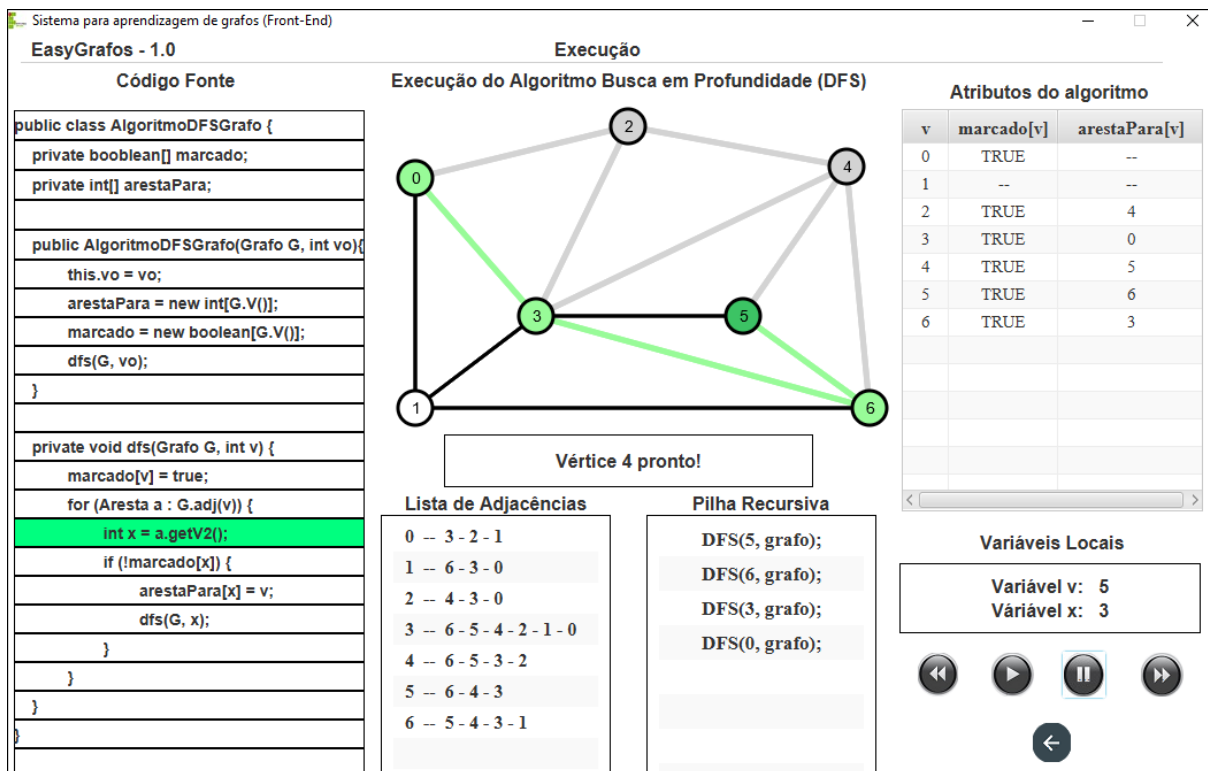
As figuras 13 e 14 apresentam respectivamente as telas da ferramenta durante a execução dos algoritmos BFS e DFS. Nelas podemos observar os elementos gráficos que foram inseridos baseados nas imagens da seção anterior, como o desenho do grafo e o valor de algumas variáveis do código. Outro aspecto que pode ser notado é que há uma linha de código colorida de verde em ambas as telas. Esta representa a instrução que está sendo executada no momento e junto a ela outras ações podem ocorrer, como mudança no desenho do grafo (referente as cores) e alteração nos valores das variáveis que estão dispostas em uma tabela do lado direito e em um retângulo abaixo desta.

Figura 13 – Tela do algoritmo BFS em execução



Fonte: Próprio autor

Figura 14 – Tela do algoritmo DFS em execução



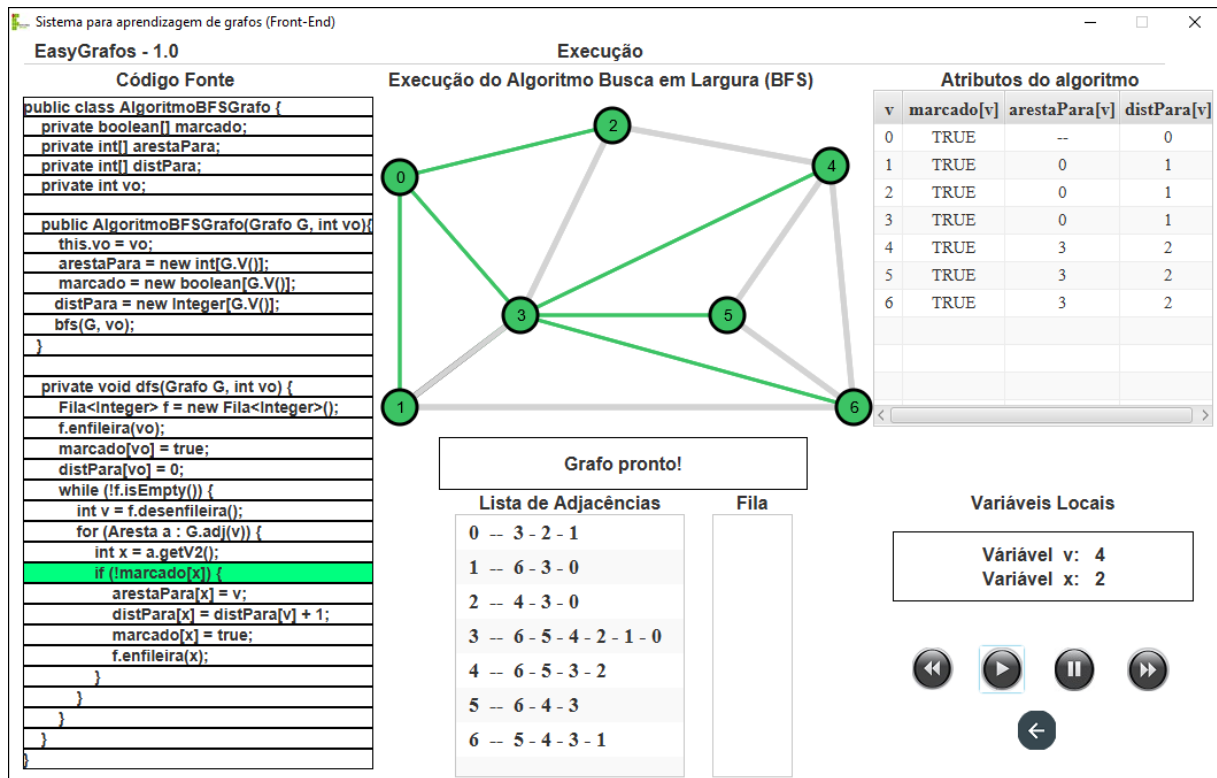
Fonte: Próprio autor

Há ainda dois pontos a serem observados. O primeiro é a coloração do grafo, que

nos dois algoritmos segue um padrão. O verde escuro significa que o algoritmo está explorando aquele elemento, seja ele um vértice ou uma aresta. O verde escuro marca os elementos já visitados mas ainda não terminados, ou seja, a execução passará por eles novamente. E por fim a cor cinza representa os elementos já terminados, aqueles que o algoritmo não irá explorar novamente. O outro ponto são os controles disponibilizados no canto inferior direito da tela. Com eles o usuário poderá parar e continuar a execução da ferramenta, acelerar e desacelerar a execução e retornar para a tela anterior. O principal objetivo neste ponto é disponibilizar ao professor formas de controlar o ritmo da execução de acordo com que ele ministra a aula.

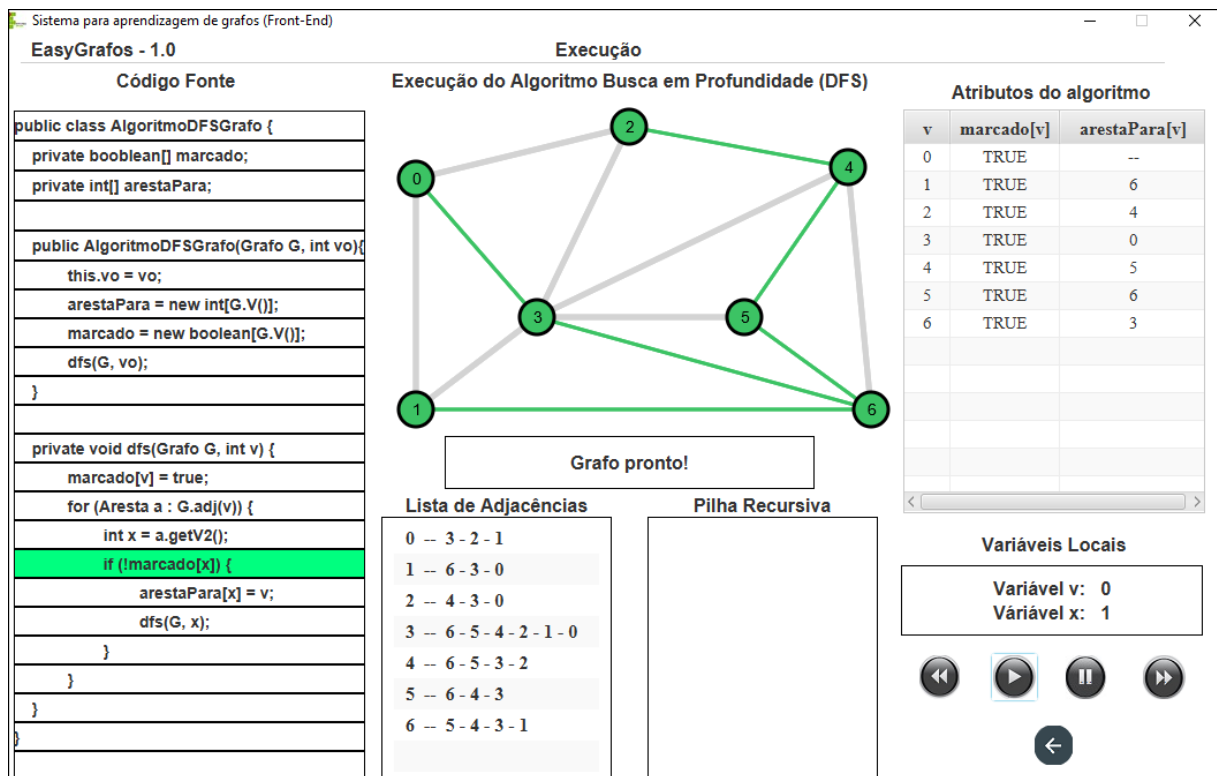
A seguir as figuras 15 e 16 ilustram respectivamente o resultado final da execução dos algoritmos BFS e DFS.

Figura 15 – Tela final do algoritmo BFS



Fonte: Próprio autor

Figura 16 – Tela final do algoritmo DFS



Fonte: Próprio autor

#### 4.4 Validação

A ferramenta EasyGrafos foi avaliada baseando-se no formulário de validação de software educacional desenvolvido por Thomas Reeves<sup>1</sup>, aplicando 5 das 14 questões pedagógicas e 6 das 10 questões de interface gráfica.

Cada questão consiste em um atributo do software que pode ser pontuado de 0 a 10. A nota é associada a duas características, uma é negativa, ou seja informa que este atributo não foi implementado satisfatoriamente, e outra positiva, informando que o software implementou de forma satisfatória este atributo. Quanto mais próxima do 0 a nota esta, mais associada a característica negativa esta a implementação deste atributo e quanto mais próxima do 10, mais associada a característica positiva esta a implementação deste atributo.

#### 4.5 Questionário aplicado

A seguir serão listadas cada uma das questões do formulário utilizado<sup>2</sup>, e qual foi o objetivo com sua aplicação.

##### 4.5.1 Atributos Pedagógicos

1. Filosofia Pedagógica:

**Instrutivista** é aquela que enfatiza a importância de metas independentes do aluno, embasada na teoria comportamentalista.

**Construtivista** enfatiza a intenção, a estratégia, a experiência do aluno, sendo ele visto como um indivíduo detentor de conhecimentos e motivações.

2. Psicologia Subjacente:

**Comportamental** significa que os fatores de aprendizado não são comportamentos que podem ser diretamente observados (comportamento desejável obtido através de estímulo-resposta).

**Cognitiva** é a psicologia que dá ênfase aos estados mentais internos, reconhece que uma ampla variedade de estratégias de aprendizagem deve ser empregada.

---

<sup>1</sup>Professor emérito em Aprendizagem, Design e Tecnologia pela Universidade de Georgia, EUA. Página pessoal disponível em: <http://treeves.coe.uga.edu/edit8350/UIRF.html>.

<sup>2</sup>[https://docs.google.com/forms/d/e/1FAIpQLSfDvTgPTK-HSh0InOSr7D9wdv2HzES6k9pMP0cUF8lcZj-bsQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfDvTgPTK-HSh0InOSr7D9wdv2HzES6k9pMP0cUF8lcZj-bsQ/viewform?usp=sf_link).



3. Objetividade:

**Precisamente focalizado** é a forma utilizada nos tutoriais e treinamentos.

**Não focalizado** é a forma empregada nas simulações virtuais e ambientes de aprendizado.

4. O papel do instrutor:

**Provedor de materiais**, o instrutor é considerado o detentor do conhecimento.

**Agente facilitador**, o instrutor é uma fonte de orientação e consulta.

5. Aprendizado Cooperativo:

**Não suportado**, quando não permite trabalho em pares ou grupos.

**Integral**, quando permite o trabalho cooperativo, fazendo com que os objetivos sejam compartilhados.

#### 4.5.2 Atributos de interface gráfica

1. Facilidade de Utilização:

**Difícil** ou **Fácil** do ponto de vista do programa ser de fácil entendimento.

2. Navegação:

**Difícil** ou **Fácil**, refere-se ao ato de ir de um tópico a outro dentro do programa.

3. Compatibilidade Espacial do Conhecimento:

**Incompatível** ou **Compatível**, verifica a compatibilidade do sistema com as expectativas e necessidades do usuário em sua tarefa (informação apresentada é compatível com o nível de conhecimento do usuário).

4. Apresentação da informação:

**Confusa** ou **Clara**, verifica se a a forma como a informação foi apresentada, possibilita sua compreensão.

5. Estética:

**Desagradável** ou **Agradável**, quanto a beleza e elegância.

6. Funcionalidade Geral:

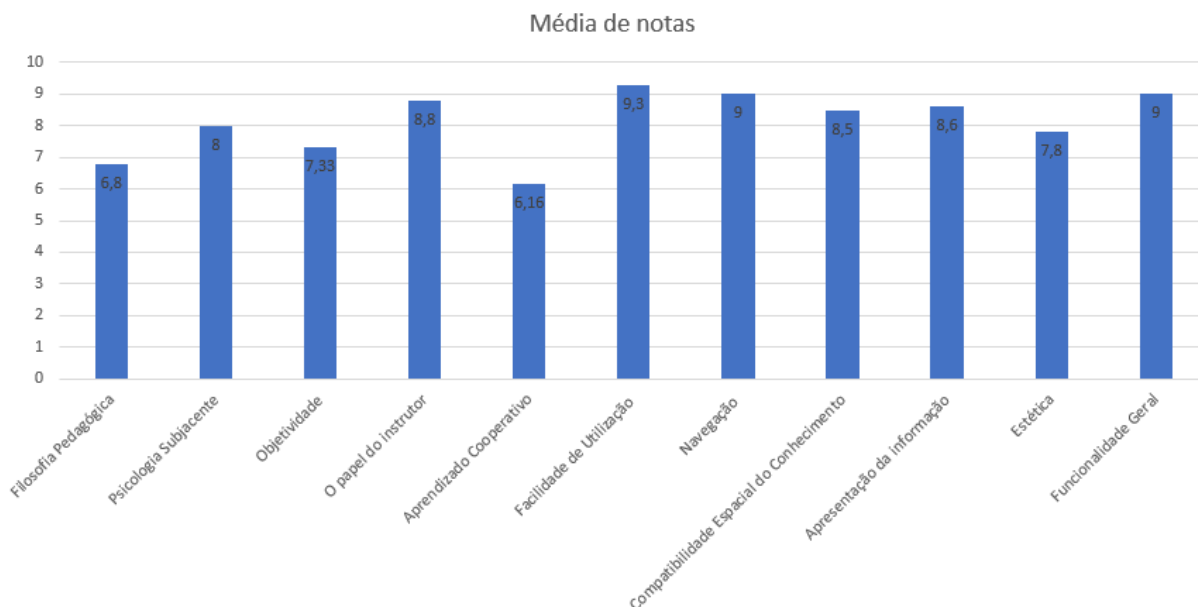
**Não funcional** ou **Altamente funcional**, quanto a utilidade percebida do programa, que é altamente relacionada com o objetivo de uso.

#### 4.6 Resultados obtidos

Um formulário com as questões descritas na seção 5.1, foi aplicado a turma do 6º período de Sistemas de Informação, do Ifes campus Cachoeiro de Itapemirim, no segundo semestre de 2018, após uma aula da disciplina de TPA, utilizando a ferramenta

EasyGrafos. A turma é composta de 6 alunos e todos responderam ao questionário. A figura 17 apresenta um gráfico com a média de cada quesito avaliado.

Figura 17 – Média geral das notas

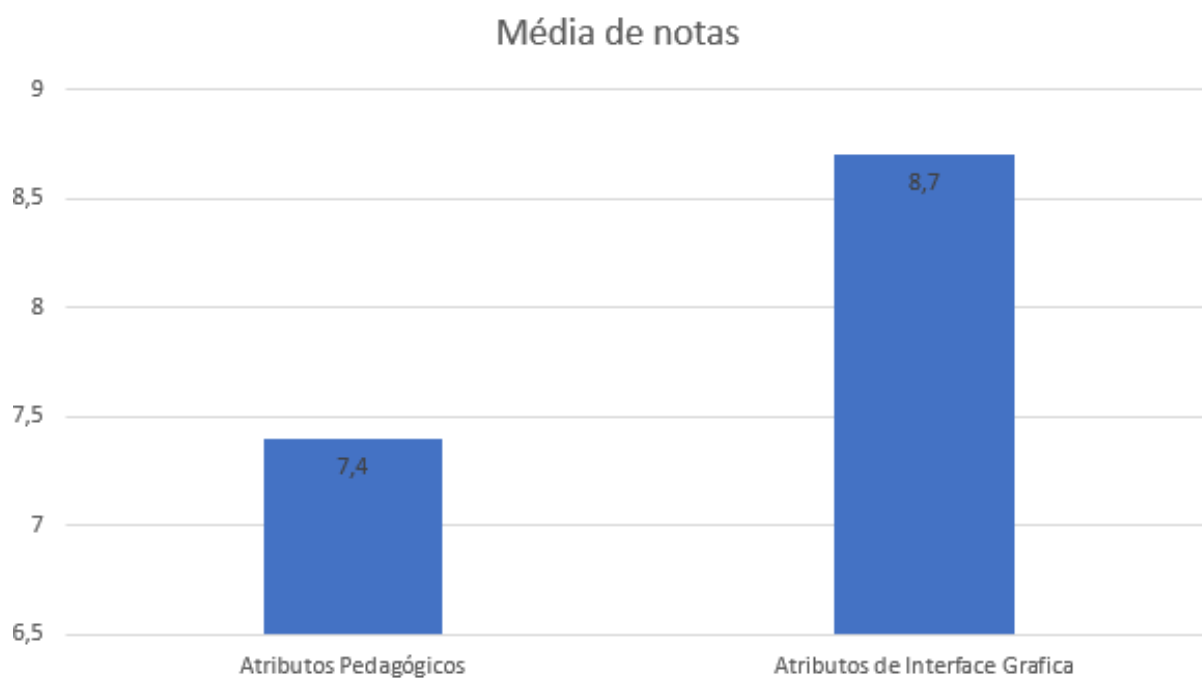


Fonte: Elaborado pelo próprio autor

Observando os resultados apresentados, pode-se verificar que nenhum quesito foi implementado de forma insatisfatória, já que, todas as médias ultrapassaram a nota 6. É necessário enfatizar que o atributo aprendizado cooperativo recebeu as notas mais baixas, o que representa uma necessidade de desenvolver mecanismos que integrem os alunos da sala e possibilite que estes interajam entre si durante os exemplos da ferramenta. Em contrapartida, o atributo facilidade de utilização recebeu média de 9,3, demonstrando que a ferramenta proporciona um ambiente de simples entendimento, oferecendo poucos ou nenhum empecilho ao aluno, quanto a dificuldade de utilização ou de compreensão de como a ferramenta funciona.

A figura 18 apresenta um gráfico comparativo entre os dois módulos do questionário, o referente aos atributos pedagógicos e o referente aos atributos de interface gráfica.

Figura 18 – Média dos módulos do formulário



Fonte: Elaborado pelo próprio autor

Analizando as duas médias podemos afirmar que tanto a parte pedagógica quanto a parte de interface da ferramenta EasyGrafos, atingiu resultados satisfatórios nas avaliações, visto que ambas ultrapassaram a nota 7 de média, destacando-se os atributos de interface gráfica que passaram de 8,5.

## 5 Conclusão

## Referências Bibliográficas

- ALMEIDA, T. A. d. et al. Computação evolutiva aplicada a resolução do problema da árvore geradora mínima com parâmetros fuzzy. [sn], 2006.
- BARCELOS, T. S.; SILVEIRA, I. F. Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica. In: *XX Workshop sobre Educação em Computação, Curitiba. Anais do XXXII CSBC*. [S.l.: s.n.], 2012. v. 2, p. 23.
- BRAGA, M. J. d. C.; GOMES, L. F. A. M.; RUEDIGER, M. A. Mundos pequenos, produção acadêmica e grafos de colaboração: um estudo de caso dos enanpads. *Revista de Administração Pública-RAP, SciELO Public Health*, v. 42, n. 1, 2008.
- COSTA, P. P. d. Teoria dos grafos e suas aplicações. Universidade Estadual Paulista (UNESP), 2011.
- FIGUEIREDO, R. T.; FIGUEIREDO, C. Wargrafos—jogo para auxílio na aprendizagem da disciplina de teoria dos grafos. *X Simpósio Brasileiro de Games e Entretenimento Digital (SBGames 2011)*, 2011.
- GIRAFFA, L. M. M. Uma odisséia no ciberespaço: O software educacional dos tutoriais aos mundos virtuais. *Brazilian Journal of Computers in Education*, v. 17, n. 01, p. 20, 2009.
- INEP. *Resumo técnico censo da educação superior 2015*. 2015. Disponível em <[http://download.inep.gov.br/educacao\\_superior/censo\\_superior/resumo\\_tecnico/resumo\\_tecnico\\_censo\\_da\\_educacao\\_superior\\_2015.pdf](http://download.inep.gov.br/educacao_superior/censo_superior/resumo_tecnico/resumo_tecnico_censo_da_educacao_superior_2015.pdf)>. Acesso em: 20 Abril 2018.
- JURKIEWICZ, S. Grafos—uma introdução. *Programa de Iniciação Científica da*, 2009.
- LOZADA, L. A. P. 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*. [S.l.: s.n.], 2014. v. 3, n. 1, p. 61.
- MELO, V. A. de; SILVEIRA, D. S. da; JURKIEWICZ, S. Teoria de grafos: Uma proposta de objeto de aprendizagem para o modelo ead. *XLI SBPO 2009 - Pesquisa Operacional na Gestão do Conhecimento*, 2009.
- MORAN, J. M. *Novas tecnologias e mediação pedagógica*. [S.l.]: Papirus Editora, 2000.
- MORATORI, P. B. Por que utilizar jogos educativos no processo de ensino aprendizagem. *UFRJ. Rio de Janeiro*, 2003.
- RECUERO, R. da C. Teoria das redes e redes sociais na internet: considerações sobre o orkut, os weblogs e os fotologs. In: *XXVII Congresso Brasileiro de Ciências da Comunicação. XXVII INTERCOM*. [S.l.: s.n.], 2004.

SANTOS, R. P. et al. O uso de ambientes gráficos para ensino e aprendizagem de estruturas de dados e de algoritmos em grafos. In: *Anais do XVI Workshop sobre Educação em Computação, XXVIII Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2008. p. 157–166.

SEDGEWICK, R.; WAYNE, K. *Algorithms*. [S.l.]: Addison-Wesley Professional, 2011.

SILVA, K. M. F. d. S. O uso de novas tecnologias em sala de aula: um estudo de caso. 2014.

SILVEIRA, E. B. de A.; SILVA, M. O. da. Desenvolvimento de um aplicativo educacional para o estudo de teoria dos grafos. 2016.

SOUZA, A. L. et al. Teoria dos grafos e aplicações. Universidade Federal do Amazonas, 2013.

TEIXEIRA, A. C. Software educacional: o difícil começo. *RENOTE*, v. 1, n. 1, 2003.

VALENTE, J. A. Análise dos diferentes tipos de softwares usados na educação. *O computador na sociedade do conhecimento*, Gráfica da UNICAMP Campinas e SP, p. 71, 1999.