

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

Jonathas Gonçalves Picoli - Ifes, jonathasgoncalvespicoli@gmail.com

Thiago Mendes - Ifes, thiagommendes16@gmail.com

Rafael Vargas Mesquita dos Santos - Ifes, rafaelv@ifes.edu.br

Justificativa

- ▶ Necessidade de possuir ensino superior;
- ▶ Evasão Escolar;
- ▶ As dinâmicas oferecidas pela tecnologia;

Objetivos

- ▶ Desenvolver um software educacional, para auxílio na aprendizagem de grafos em disciplinas de computação

Trabalhos Relacionados

Figura 1. Comparativo entre ferramentas relacionadas

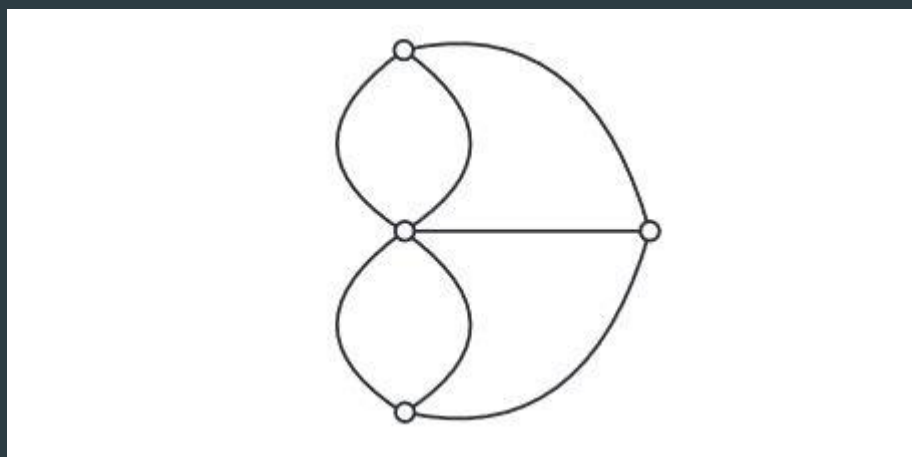
Característica/Software	WarGrafos	TBC-Grafos	AlgoDeGrafos	A-Graph	Tgrafo	EasyGrafos
Interface para manipulação e análise dos grafos		X	X	X	X	X
Criar os próprios grafos			X	X	X	X
Executar algoritmos da disciplina nos grafos		X	X	X	X	X
Execução em modo debug		X	X			X
Associação do código fonte do algoritmo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

Fonte: Próprio autor

Contexto Histórico

- ▶ Ano de 1736
- ▶ Leonhard Euler e a pontes Königsberg

Figura 2. Diagrama de Euler



Aplicabilidade

- ▶ Caminho mínimo entre dois pontos
- ▶ Relações entre os pontos

Metodologia da Pesquisa

- ▶ Linguagem de programação Java
- ▶ Listas de exercícios da disciplina de TPA

Conceitos Abordados

Figura 3. Conceitos de grafos abordados pela ferramenta

Grafos direcionados		Grafos não direcionados	
Busca	Árvore Geradora mínima	Busca	Caminho mínimo
Busca em Largura(BFS)	AGM – Kruskal	Busca em Largura(BFS)	CM – Dijkstra
Busca em Profundidade(DFS)	AGM – Prim	Busca em Profundidade(DFS)	CM – Belman-Ford

Fonte: Próprio autor

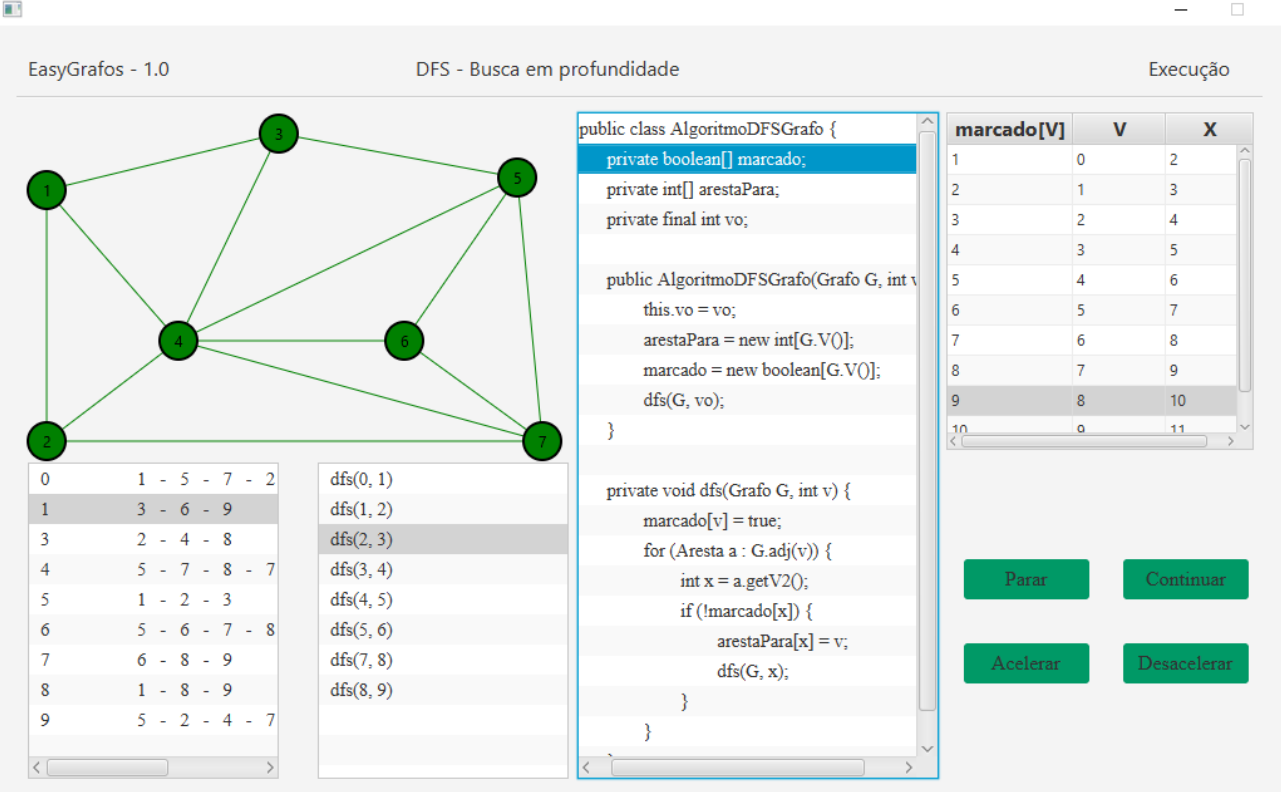
Protótipo

- ▶ Preencher as lacunas deixadas por softwares similares
- ▶ Listas de exercícios da disciplina de TPA

Protótipo

Figura 4. Tela do modo de execução

EasyGrafos - 1.0 DFS - Busca em profundidade Execução



0 1 - 5 - 7 - 2
1 3 - 6 - 9
3 2 - 4 - 8
4 5 - 7 - 8 - 7
5 1 - 2 - 3
6 5 - 6 - 7 - 8
7 6 - 8 - 9
8 1 - 8 - 9
9 5 - 2 - 4 - 7

dfs(0, 1)
dfs(1, 2)
dfs(2, 3)
dfs(3, 4)
dfs(4, 5)
dfs(5, 6)
dfs(7, 8)
dfs(8, 9)

```
public class AlgoritmoDFSGrafo {  
    private boolean[] marcado;  
    private int[] arestaPara;  
    private final int vo;  
  
    public AlgoritmoDFSGrafo(Grafo G, int v)  
    {  
        this.vo = vo;  
        arestaPara = new int[G.V()];  
        marcado = new boolean[G.V()];  
        dfs(G, vo);  
    }  
  
    private void dfs(Grafo G, int v) {  
        marcado[v] = true;  
        for (Aresta a : G.adj(v)) {  
            int x = a.getV2();  
            if (!marcado[x]) {  
                arestaPara[x] = v;  
                dfs(G, x);  
            }  
        }  
    }  
}
```

marcado[V]	V	X
1	0	2
2	1	3
3	2	4
4	3	5
5	4	6
6	5	7
7	6	8
8	7	9
9	8	10
10	9	11

Parar Continuar
Acelerar Desacelerar

Fonte: Próprio autor

Utilização

- ▶ Selecionar Grafo
- ▶ Selecionar algoritmo
- ▶ Preencher os parâmetros
- ▶ Iniciar

Conclusão

- ▶ O protótipo proposto foi finalizado
- ▶ Outras áreas da computação

Dúvidas ?