

# **INSTITUTO FEDERAL**

## Minas Gerais

### Campus Ouro Branco

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS  
CAMPUS OURO BRANCO  
PROJETO E ANÁLISE DE ALGORITMOS

João Gabriel Alves Junior

**Trabalho II 2023/02**

Implementação do algoritmo de Kruskal

Ouro Branco  
2023

João Gabriel Alves Junior

**Trabalho II 2023/02**

Ouro Branco  
2023

## 0.1 Tecnologias utilizadas

- 1) React: Uma biblioteca JavaScript amplamente utilizada para criar interfaces de usuário dinâmicas e interativas. Ela nos permite construir componentes reutilizáveis e gerenciar eficientemente o estado da aplicação.
- 2) Vite: Um framework de build que oferece uma experiência de desenvolvimento incrivelmente rápida, com inicialização de projeto quase instantânea e hot module replacement.
- 3) D3.js: Uma biblioteca poderosa para a manipulação e visualização de dados, especialmente adequada para a criação de gráficos e diagramas interativos.

### 0.1.1 O projeto como um todo permite:

- Carregar um grafo a partir de um arquivo de texto: O usuário pode fornecer um arquivo contendo a representação de um grafo em formato de matriz de adjacência.
- Visualizar o grafo: O grafo carregado é exibido visualmente na tela, permitindo a análise da sua estrutura e conexões.
- Gerar a árvore de abrangência mínima (MST): Com um clique no botão “Gerar MST”, o algoritmo de Kruskal é aplicado para encontrar a MST do grafo, que é então destacada na visualização.

## 0.2 Algoritmo de Kruskal

O algoritmo de Kruskal é um método eficiente para encontrar a MST de um grafo não direcionado e ponderado. Ele funciona da seguinte maneira:

- Ordena as arestas do grafo em ordem crescente de peso.
- Percorre as arestas ordenadas, adicionando-as à MST se não criarem ciclos.
- Repete o passo 2 até que todas as arestas tenham sido consideradas ou a MST esteja completa.

### 0.2.1 Aplicação do algoritmo no projeto

A implementação do algoritmo de Kruskal neste projeto utiliza estruturas de dados como mapas e conjuntos para representar o grafo e a MST, e emprega técnicas de ordenação e busca para encontrar a MST de forma eficiente.

## 0.2.2 Código de Kruskal :

Figura 1 – Código de Kruskal em JS

```
/**
 * Função para calcular a Árvore de Abrangência Mínima (MST) usando o algoritmo de Kruskal.
 * @param {Array} nodes - Lista de nós no grafo.
 * @param {Array} links - Lista de arestas no grafo.
 * @returns {Array} - Lista de arestas que compõem a MST.
 */
function kruskalMST(nodes, links) {
  // Cria uma nova lista de arestas para a MST
  const mstlinks = [];

  // Cria uma "floresta" (um conjunto de árvores), onde cada vértice no grafo é uma árvore separada
  const forest = new Map(nodes.map(node => [node.id, node]));

  // Cria um mapa para armazenar o "nome" da árvore de cada nó
  const name = new Map(nodes.map(node => [node.id, node.id]));

  // Função para verificar se dois nós estão na mesma árvore
  const connected = (a, b) => name.get(a.id) === name.get(b.id);

  // Função para unir duas árvores
  const union = (a, b) => {
    const aName = name.get(a.id);
    const bName = name.get(b.id);

    // Une as árvores
    for (const node of forest.values()) {
      if (name.get(node.id) === bName) {
        name.set(node.id, aName);
      }
    }
  };

  // Ordena as arestas por peso (ascendente)
  links.sort((a, b) => a.value - b.value);

  // Adiciona cada aresta à MST (se não criar um ciclo)
  for (const link of links) {
    if (!connected(link.source, link.target)) {
      mstlinks.push(link);
      union(link.source, link.target);
    }
  }

  return mstlinks;
}
```

O código inteiro pode ser achado no reposotirio do GITHUB: <https://github.com/Gabrieljr42/TreeVisualizer>

### 0.3 Demonstração

#### 0.3.1 Exemplo :

Figura 2 – Antes da geração de MST

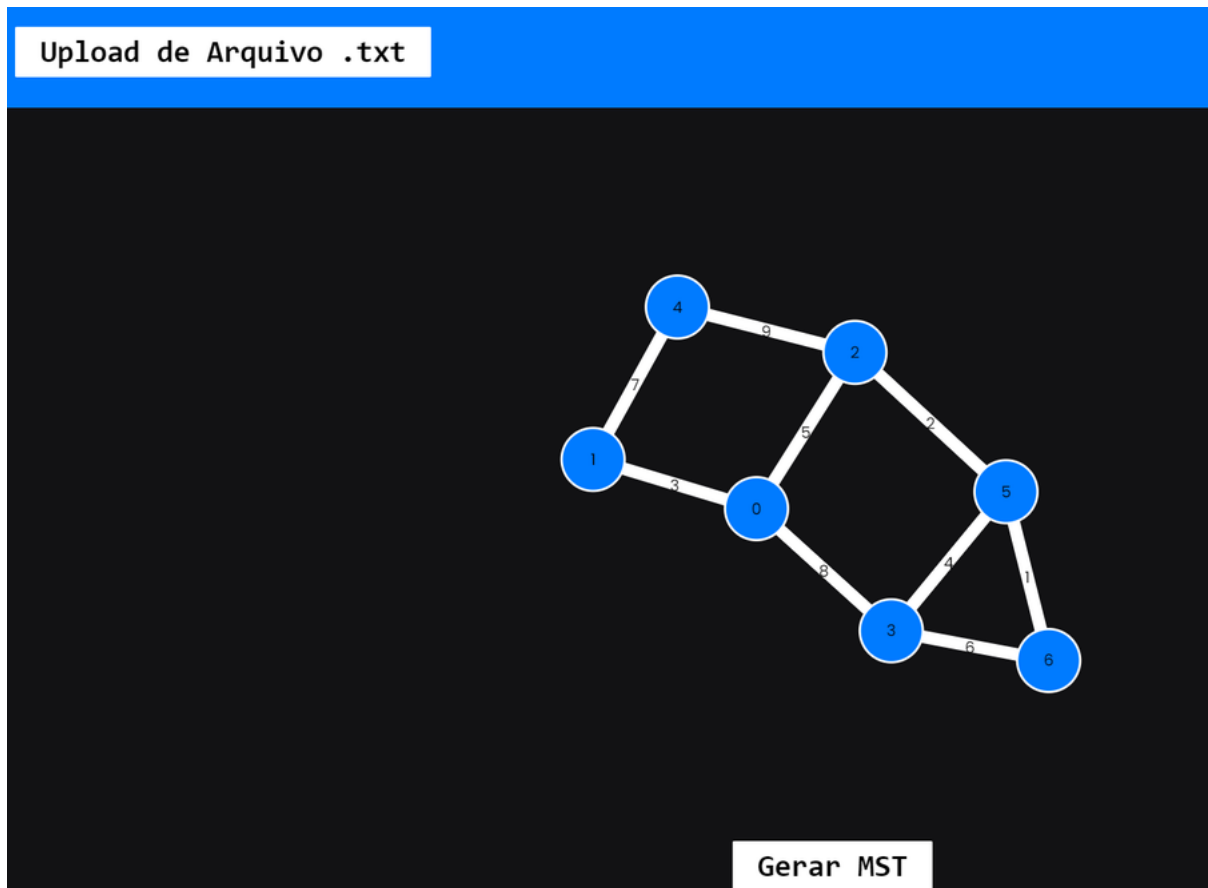
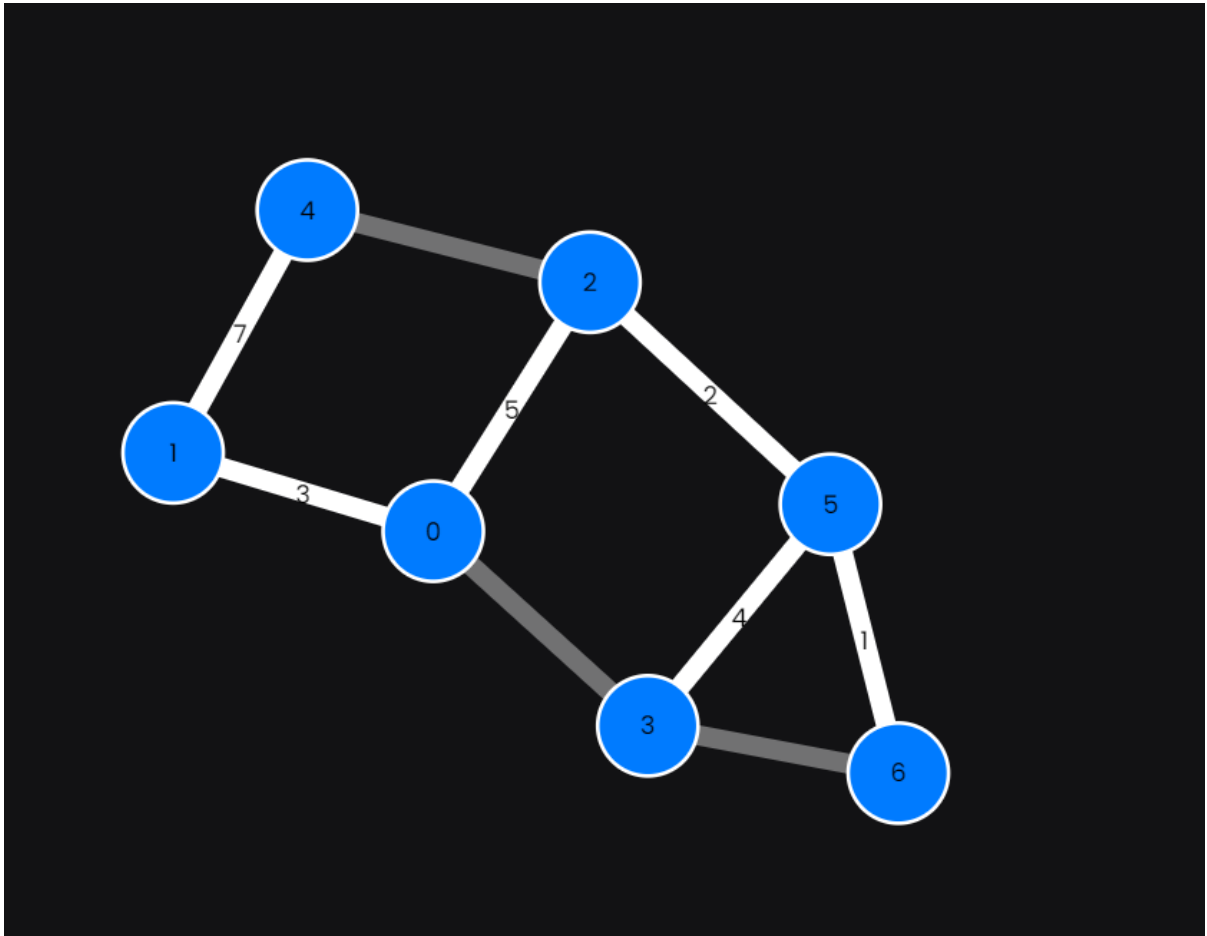


Figura 3 – Após a geração da MST



O site para geração de mais exemplos pode ser encontrado aqui : <https://tree-visualizer-three.vercel.app>

Um vídeo mostrando a utilização do site está em : [https://youtu.be/cU-gz9BU\\_0M](https://youtu.be/cU-gz9BU_0M)