# Relatório de funcionamento do Algoritmo

**Douglas José Tertuliano do Santos – 11513928**

**Matheus Pires Fernandes - 11518681**

**Descrição do código:**

**- Função Busca()**

Esta função recebe um grafo como argumento e ela marca todos os vértices como “Não visitados” e colori-los de branco. Esta função também percorre o grafo e verifica se cada vértice já foi visitado. Caso não tinham sido visitados, é chamado a função BuscaProf().

**- Função BuscaProf()**

Esta função recebe o vértice analisado no momento como argumento e visita todos os adjacentes do mesmo. Este vértice é adicionado na pilha de nós. Então é verificado se este tem adjacentes.

Para cada nó adjacente a este é verificado se ele possui o atributo “NãoVisitado”, pois caso já tenha sido analisado seus adjacentes, não necessita analisa-lo. Caso ele realmente não tenha sido visitado, marca a aresta (a aresta entre o nó passado como argumento e o nó adjacente em questão) como visitada.

Após isso, é chamado a BuscaProf() para o vértice adjacente ao vértice inicialmente analisado.

Quando já tiver sido analisado todos os vértices adjacentes ao vértice inicialmente analisado, ele é desempilhado e também marcado com a cor Preta, para mostrar que já foi completamente analisado.

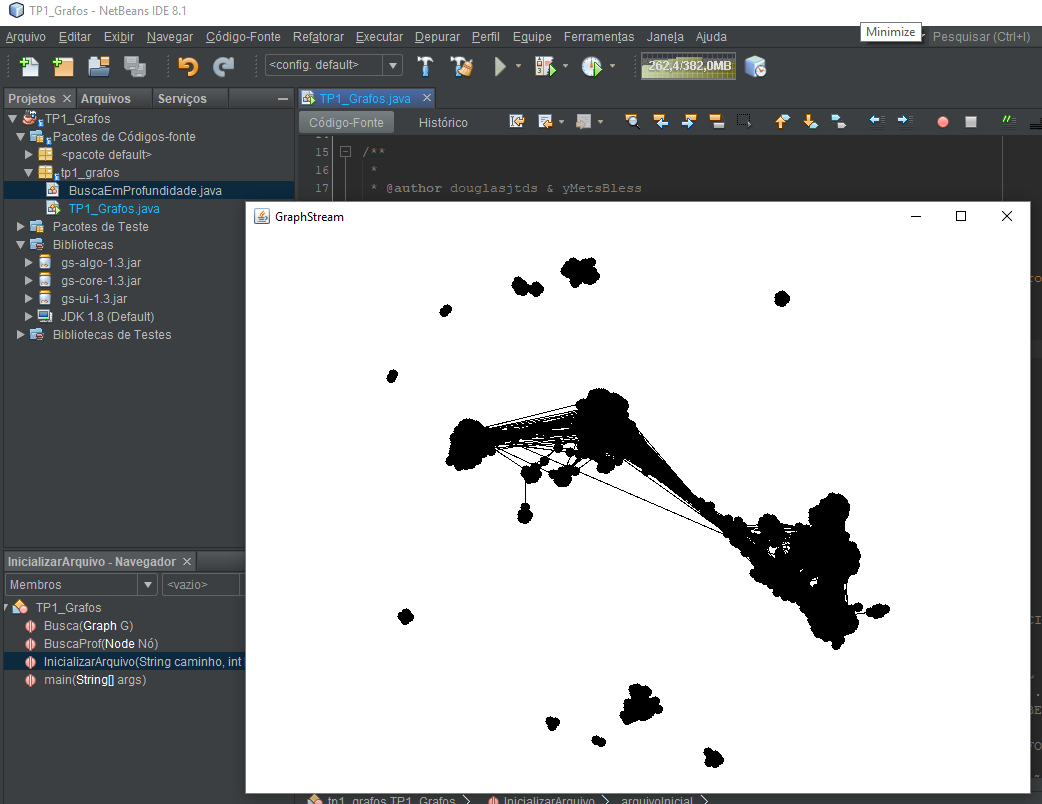
**- Função MaiorComponente()**

Esta função recebe por parâmetro o grafo, o nó de origem e o nó de destino. Cria duas variáveis booleanas para ajudar a conferir se precisa repetir a origem e destino. Depois temos um foreach para percorrer os nós repetidos e se o o nó de origem tiver sido repetido a variável para isso recebe true; a mesma coisa é feita para a variável booleana para o nó de destino. Se o nó de origem ainda não tiver sido repetido, ele adiciona ao grafo da função e depois adiciona à lista de nós repetidos; a mesma coisa é feita para o nó de destino.

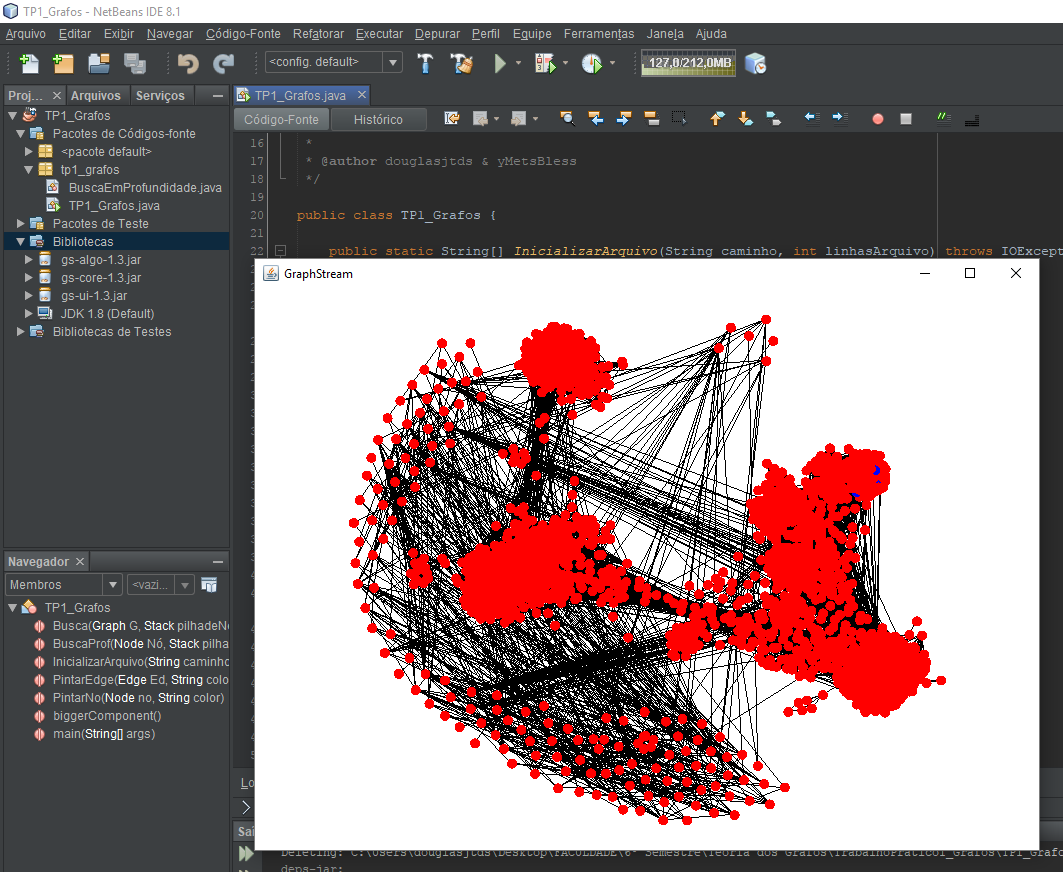
**- Função BuscaLargura()**

**Imagens dos grafos gerados:**

**- Grafo inicial:**



- **Somente o maior componente**



**- Arvore gerada pelo Busca em Profundidade**

**- Arvore gerada pelo Busca em Largura**