

# Visual Graphs Manual

Bruno Bastos

# Sumário

**Introdução**

**A Interface**

**Criando os nós**

**Movimentação**

**Mais nós**

**As Conexões**

**Criando as arestas**

**Os pesos das arestas**

**Algoritmos de busca**

**Outras Opções**

# Introdução

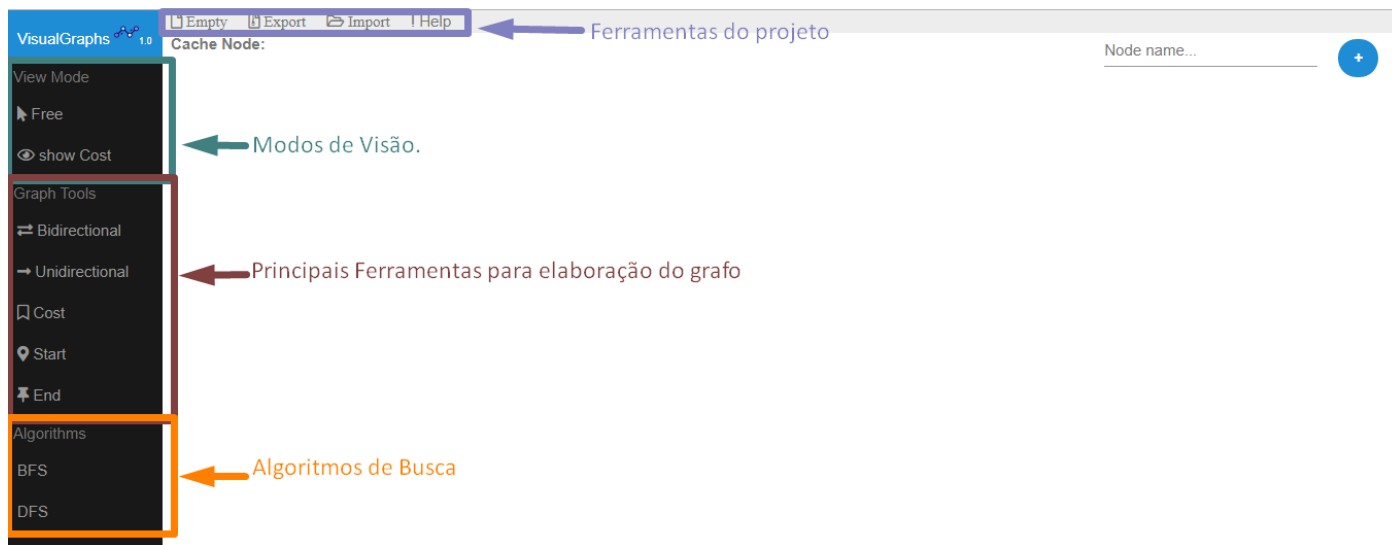
**V**isual Graphs é uma aplicação que permite a criação de grafos de forma dinâmica e simples.

## O que é um grafo?

Grafo é um objeto de estudo da teoria dos grafos, que são estruturas matemáticas usadas para modelar relacionamentos entre objetos.

# A Interface

Você pode ficar um pouco perdido quando usar o **visualGraphs** pela primeira vez, abaixo segue uma imagem que descreve melhor a localização das ferramentas.

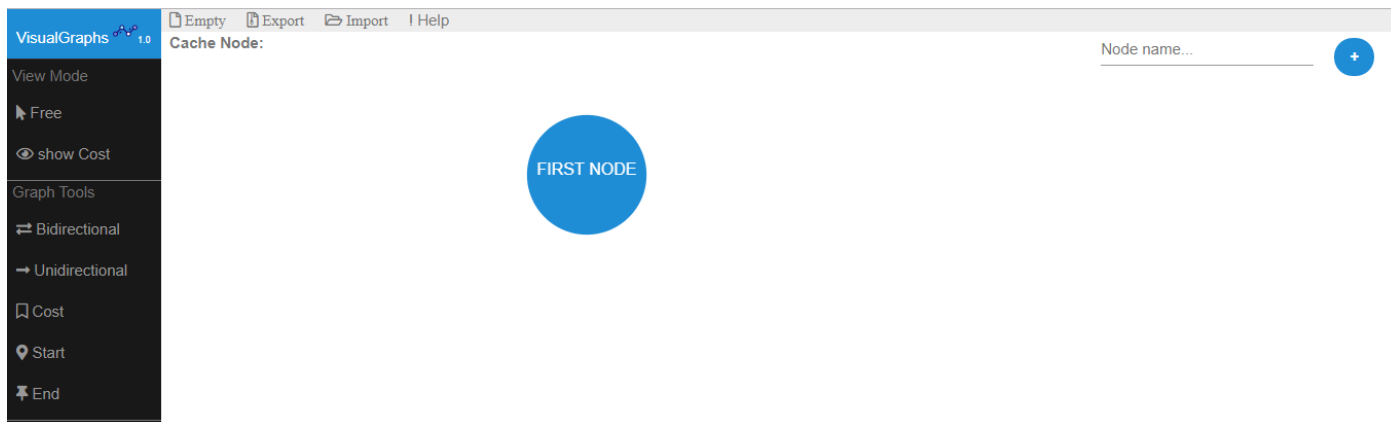


Após criarmos o nosso primeiro nó, iremos abordar cada item disponível em nossa aplicação.

# Criando os Nós

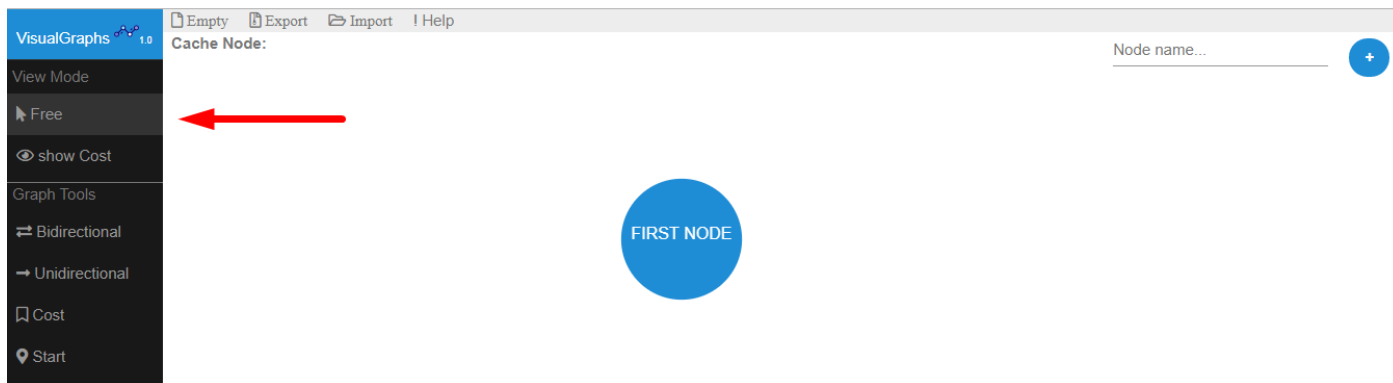
A criação de um nó é bem simples, o **VisualGraphs** contém um único **input**(campo de digitação), na qual você pode encontra-lo no canto superior direito da tela, e em seguida preenche-lo com algum valor e clicar no botão de adicionar(ou pressionar **enter**).

Digite **First Node** no campo input e em seguida clique no botão de adicionar. Deverá aparecer um nó em sua tela, conforme a imagem abaixo.



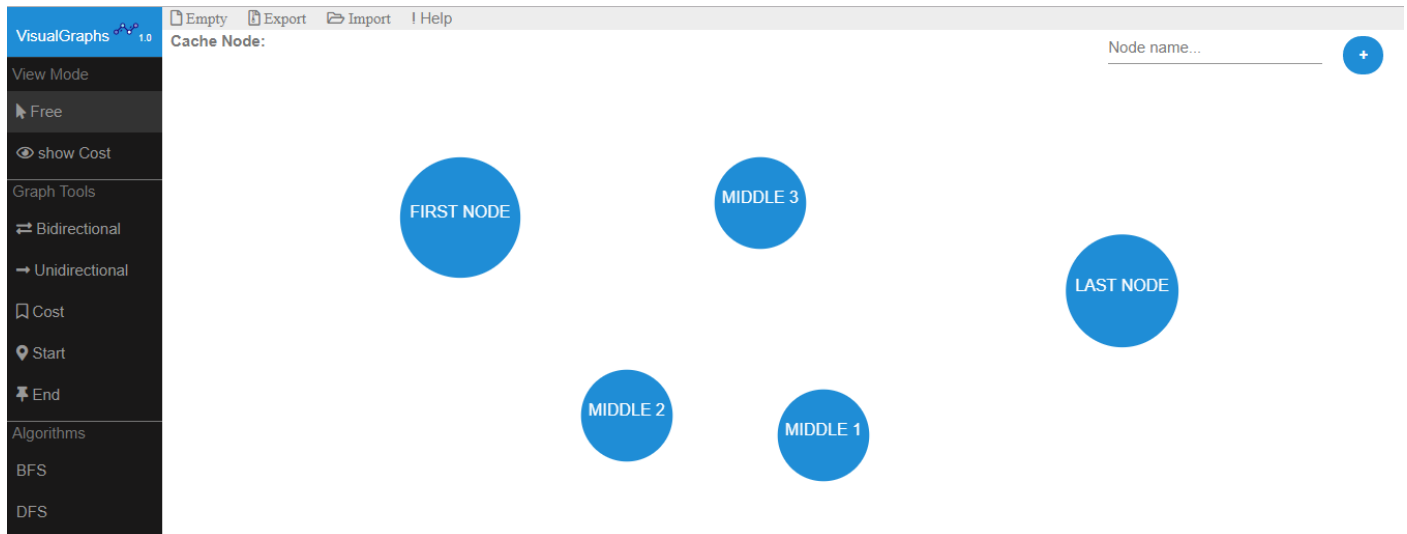
# Movimentação

O nó é inserido em sua tela em uma posição aleatória, e com isso, você pode querer arrasta-lo para uma determinada posição. Se você clicar no nó, não irá conseguir move-lo, para liberar essa funcionalidade, é necessário que você clique em **Free**, no bloco **Modos de Visão**. Agora clique no nó e arraste-o para onde desejar.



# Mais nós...

Para realizarmos as operações com nosso grafo, é necessário criar mais nós, portanto iremos criar mais 4 nós, Siga o exemplo abaixo.



# As conexões

As conexões entre nós podem ser duas, **bidirecional** ou **unidirecional**. A conexão **bidirecional**, diz que **A** se comunica com **B** e **B** se comunica com **A**. na conexão **unidirecional** o **A** se comunica com **B**, porém **B** não se comunica com **A**.

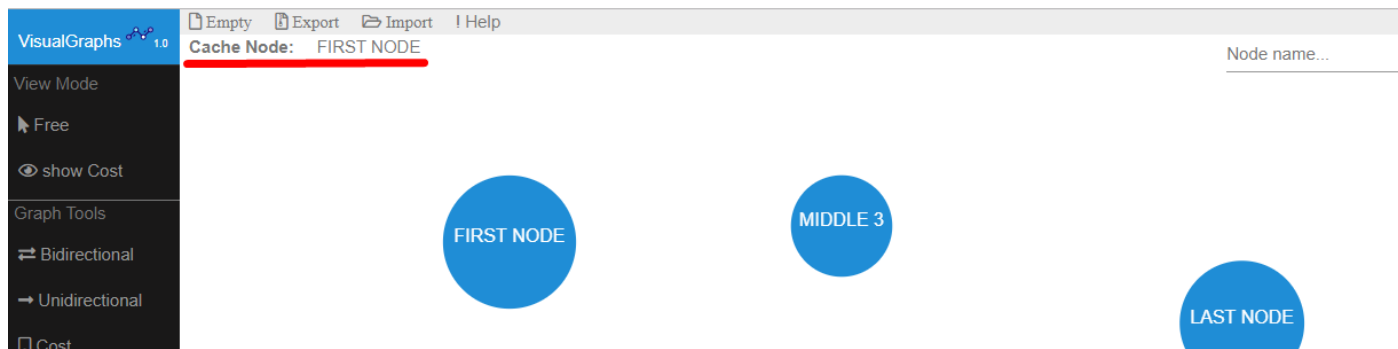
## IMPORTANTE!

É importante ressaltar que essa é uma definição bem simplificada sobre os grafos e todo o seu conceito, para um melhor entendimento, recomendo uma busca do conhecimento sobre os grafos através da **literatura**.



# Criando as arestas

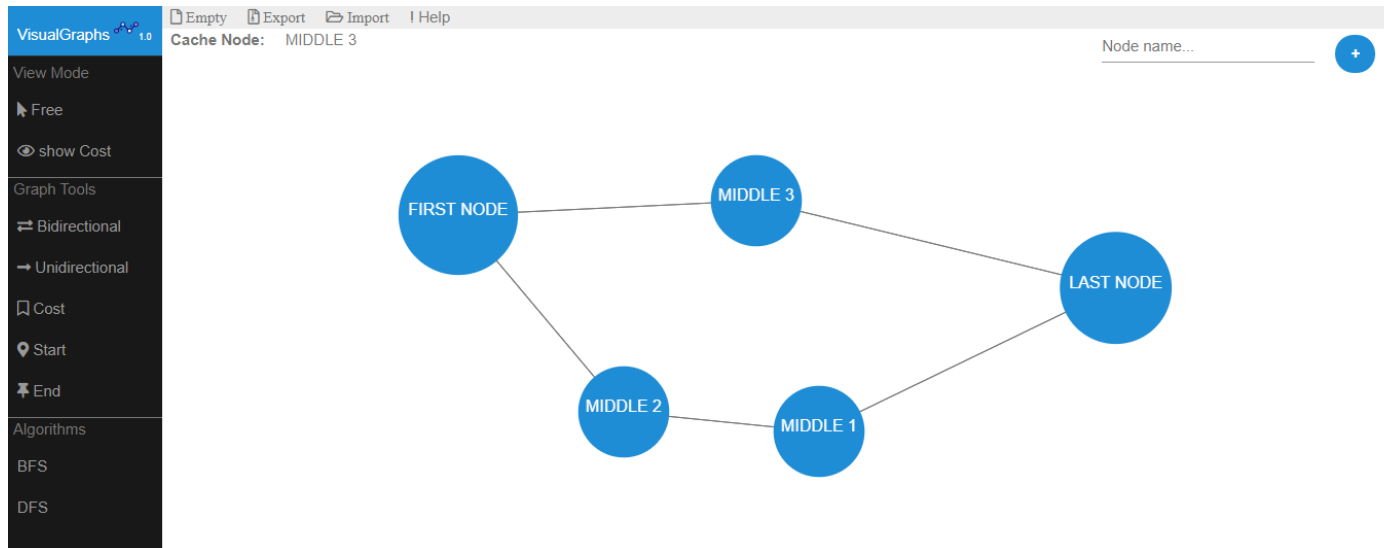
Para um fácil entendimento, iremos criar **apenas** as arestas **bidirecionais**. Para criar a aresta selecione a opção **bidirecional** e depois clique no nó **First node**. É importante observar que próximo a nossa barra de menu, existe **Cache Node**, que de modo simples, é o nó que estamos trabalhando no momento. Clicando então no **First node**, você deve ter algo parecido com isso...



Em seguida, clique em **middle 3**, e observe que foi criada uma linha entre os nós, e que agora o cache node é o **middle 3**. Antes de prosseguirmos, é preciso entender que o que aconteceu basicamente foram os seguintes passos:

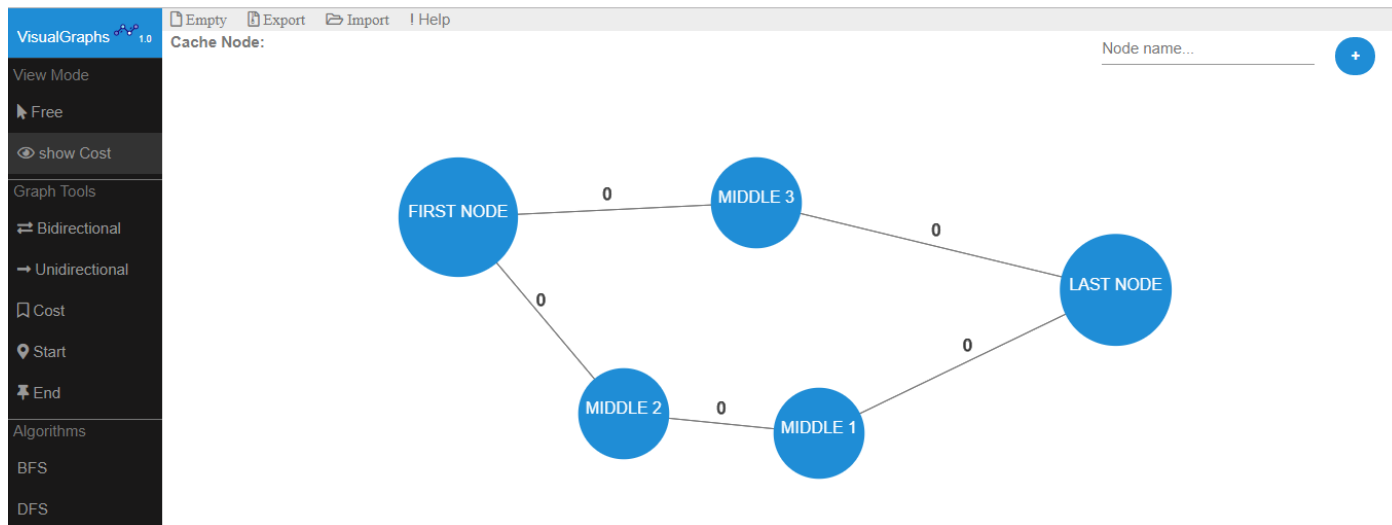
1. Foi selecionado a opção bidirecional
2. Foi selecionado um nó, que ficou em nosso cache
3. Foi selecionado um segundo nó, e esse ganhou uma relação bidirecional com o primeiro nó.
4. O segundo nó se tornou o nó cache
5. Foi desenhado uma linha entre os nós.

Agora, **crie** a **seu gosto** diversas relações entre os nós, para você familiarizar com a ferramenta, deixarei um modelo abaixo caso esteja sem idéia.



# Os pesos das arestas

A distância entre os nós têm algum valor numérico, isto é, um peso. Por **default**, toda conexão criada, é inicializada com zero. Para ver o valor atual das arestas, selecione a opção **show cost** no **modos de visão**, E você deve ter algo conforme a imagem abaixo:

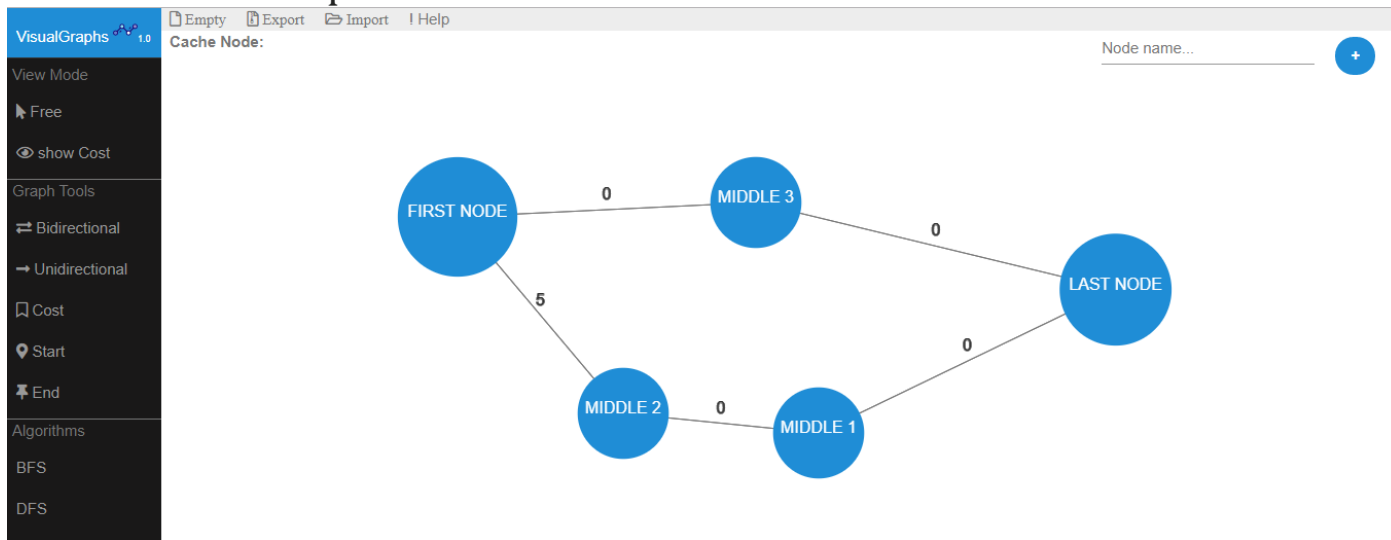


Você também pode alterar esses valores, selecionando a opção **cost** em **graph Tools**, e depois selecionando o nó em que deseja alterar, prosseguido da

seleção do nó destino(identico ao passo de criar uma conexão). Segue abaixo uma observações.

*Nas relações unidirecionais, você precisa selecionar o nó A que tem conexão com o nó B, pois se selecionar o nó B e depois A, irá aparecer uma mensagem dizendo B não tem relação com A.*

Para melhor demonstrar alteração do custo, iremos alterar o valor do **first node** e **middle 2** para 5.



# Algoritmos de busca

Atualmente o **VisualGraphs** contém dois algoritmos de busca, o **BFS** e o **DFS**.

Não iremos detalhar sobre esses algoritmos, caso tenha curiosidade, você pode pesquisar por **busca em largura**( Breadth-First Search) para saber sobre o **BFS**, e **busca em profundidade**(Depth-First Search ) para saber sobre o **DFS**.

Para executar esses algoritmos, primeiramente é preciso definir o nó inicial e o nó final, para isso, há duas opções no **graphs tools**, selecionar **start** e em seguida selecione um nó inicial e após definir o nó inicial, selecione **end** e em seguida o nó que será o nó final.

Em seguida selecione o algoritmo que deseja visualizar. Para demonstração, iremos selecionar o **BFS**. Observe que o melhor caminho foi pintado de verde, conforme a imagem abaixo.

VisualGraphs 1.0

View Mode

Free

show Cost

Graph Tools

Bidirectional

Unidirectional

Cost

Start

End

Algorithms

BFS

DFS

EmptyExportImportHelp

Cache Node:

Node name...

```
graph LR; FN((FIRST NODE)) ---|0| M3((MIDDLE 3)); FN ---|5| M2((MIDDLE 2)); M3 ---|0| LN((LAST NODE)); M2 ---|0| M1((MIDDLE 1)); M1 ---|0| LN;
```

# Outras Opções

**Terminei meu projeto, quero salvar, e agora?**

Pensando nisso, o visual graphs tem um sistema de **export** e **import**, na qual voce pode exportar seu projeto em json e depois importa-lo novamente.

**Quer começar um novo grafo?**

Então selecione a opção **Empty** para limpar sua workspace.