

**ALGORITMIA AVANÇADA**

SPRINT C

TURMA 3NA

GRUPO 74

| 1140858 Carlos Filipe Borges Moutinho

| 1171602 Rui Manuel Castro Marinho

| 1181882 Rafael Martins Soares

| 1181892 Sara Santos Teixeira Silva

| 1181895 Fábio Alves da Silva

Índice

[Introdução 3](#_Toc93855762)

[Desenvolvimento 4](#_Toc93855763)

[Criação de uma rede à parte com os utilizadores que podem ser alcançados até N ligações a partir de um dado utilizador 4](#_Toc93855764)

[Gerador aleatório das forças 6](#_Toc93855765)

[Adaptação do A\* ao problema da determinação do caminho mais forte (máximo de N ligações) 7](#_Toc93855766)

[Implementar a estimativa no algoritmo a\* 10](#_Toc93855767)

[Adaptação do Best First ao problema da determinação do caminho mais forte (máximo de N ligações) 13](#_Toc93855768)

[Adaptação do Primeiro em Profundidade para gerar a melhor solução (já implementado no Sprint anterior) para o máximo de N ligações 15](#_Toc93855769)

[Comparação dos 3 métodos com vários exemplos, comparando tempos de geração da solução e valor da solução gerada 16](#_Toc93855770)

[Implementação da função multicritério que contemple forças de ligação e diferença entre likes e dislikes 19](#_Toc93855771)

[Adaptação dos 3 métodos (Primeiro em Profundidade, Best First e A\*) para considerar a função multicritério do ponto anterior 20](#_Toc93855772)

[Primeiro em profundidade 20](#_Toc93855773)

[Best First 21](#_Toc93855774)

[A\* 22](#_Toc93855775)

[Comparação dos 3 métodos com vários exemplos e usando a função multicritério 23](#_Toc93855776)

[Conclusão 26](#_Toc93855777)

# Introdução

O presente trabalho foi desenvolvido no âmbito da disciplina de Algoritmia Avançada (ALGAV), onde foi pedido o desenvolvimento do planeamento de contatos numa rede social. Para a mesma, foi utilizada a linguagem de programação PROLOG e utilizados métodos de pesquisa lecionados durante o semestre.

Neste relatório será efetuada o estudo das seguintes UC’s:

* Criação de uma rede à parte com os utilizadores que podem ser alcançados até N ligações a partir de um dado utilizador
* Gerador aleatório de forças
* Adaptação do A\* ao problema da determinação do caminho mais forte (máximo de N ligações)
* Implementar a estimativa
* Adaptação do Best First ao problema da determinação do caminho mais forte (máximo de N ligações)
* Adaptação do Primeiro em Profundidade para gerar a melhor solução (já implementado no Sprint anterior) para o máximo de N ligações
* Comparação dos 3 métodos com vários exemplos, comparando tempos de geração da solução e valor da solução gerada
* Implementação da função multicritério que contemple forças de ligação e diferença entre likes e dislikes
* Adaptação dos 3 métodos (Primeiro em Profundidade, Best First e A\*) para considerar a função multicritério do ponto anterior
* Comparação dos 3 métodos com vários exemplos e usando a função multicritério

# Desenvolvimento

## Criação de uma rede à parte com os utilizadores que podem ser alcançados até N ligações a partir de um dado utilizador

De forma a responder ao requerimentos do sprint C criamos uma nova rede. Abaixo encontra-se 3 imagens recriando um grafo com indicação das forças de ligação, forças de relação e conjunto de tags respetivamente.

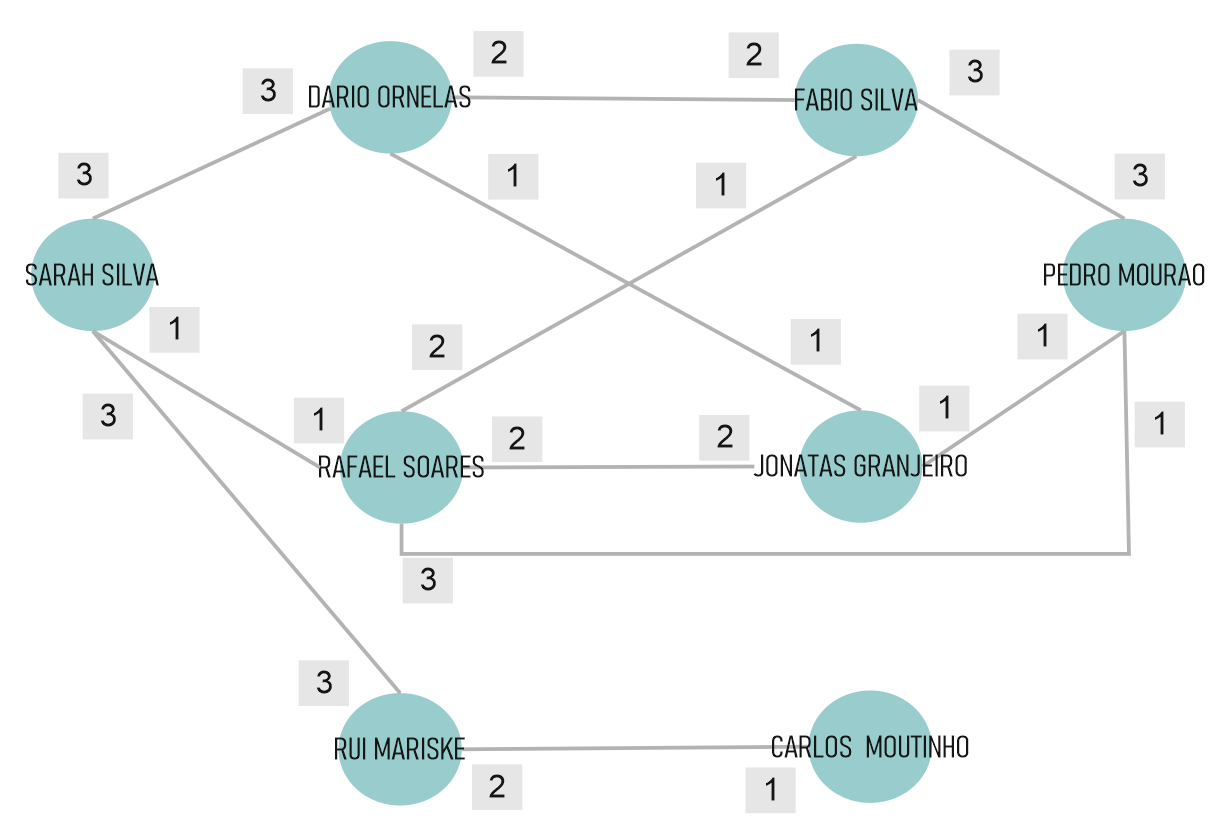


Figura 1 – Grafo com forças de ligação

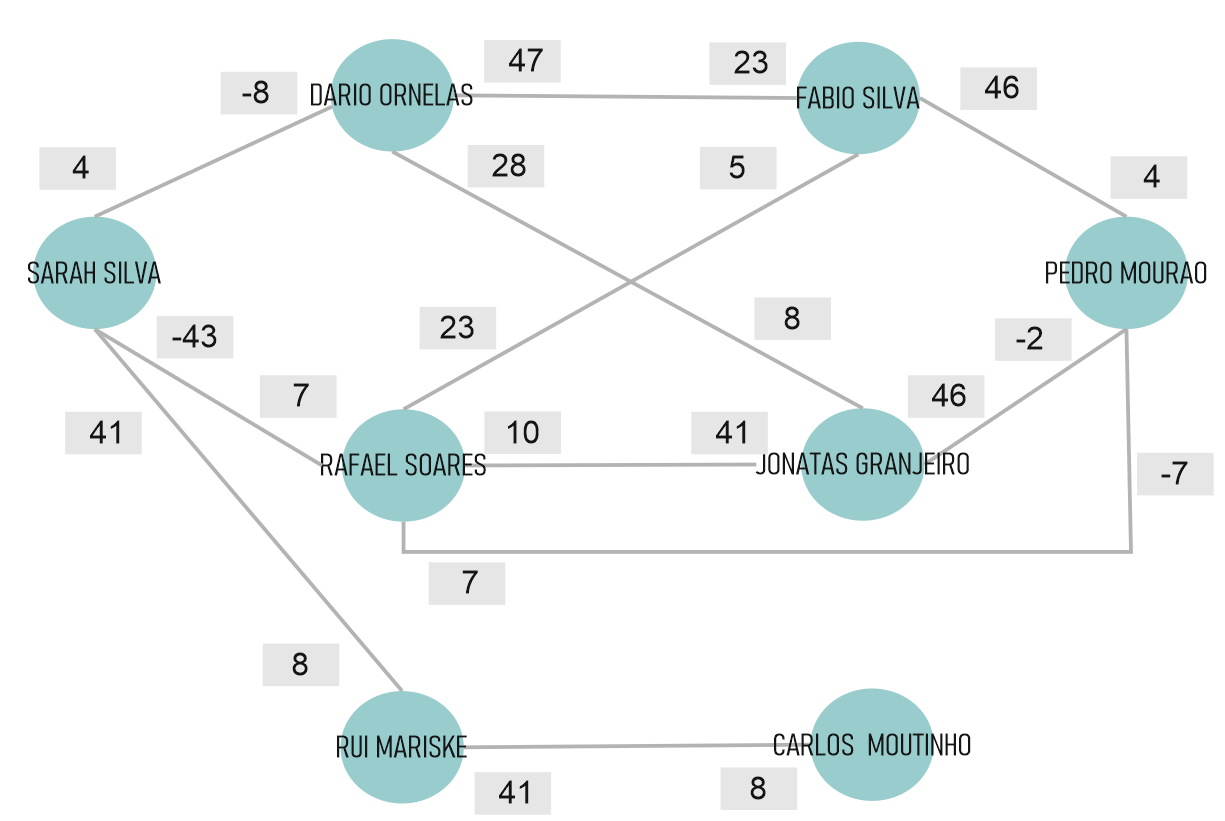


Figura 2 – Grafo com forças de relação.

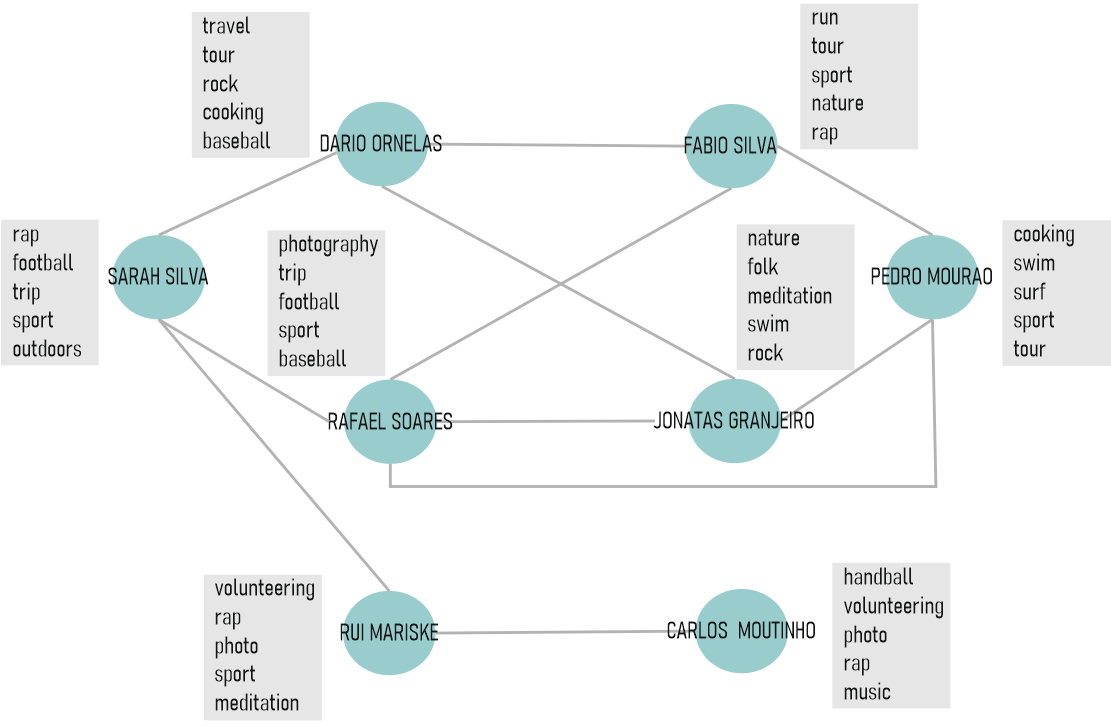


Figura 3 – Grafo com indicação das tags de cada utilizador.

## Gerador aleatório das forças

Uma imagem com texto

Descrição gerada automaticamente

**Exemplo:**

Ligação em estudo: *Carlos Moutinho – Rui Mariske*

Base de conhecimento:



Uma imagem com texto

Descrição gerada automaticamente

Como visto na imagem acima, quando pretendemos verificar qual a força da ligação entre ambos os utilizadores obtemos uma ligação=1.

Ao gerarmos novas forças de ligação através do método generate\_ForcasLigacao, estamos a mudar o valor das forças de ligação em toda a rede. Para comprovar a eficiência do algoritmo, voltamos a chamar o predicado ‘ligação’ para verificar que efetivamente o valor da ligação exemplo mudou.

## Adaptação do A\* ao problema da determinação do caminho mais forte (máximo de N ligações)

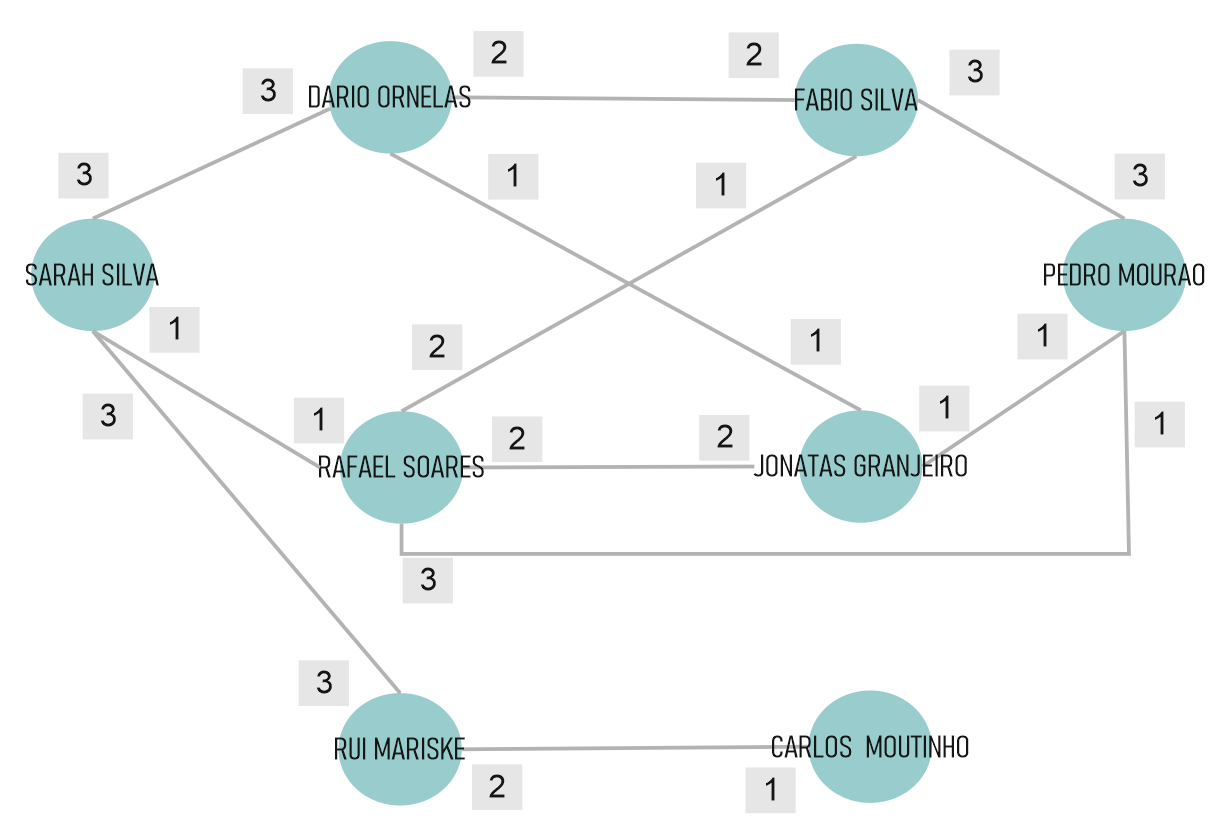
Neste código apenas são considerados as forças de ligação e um número de máximo de ligações.

No que diz respeito à estimativa, foi colocada a 0.

Uma imagem com texto

Descrição gerada automaticamente

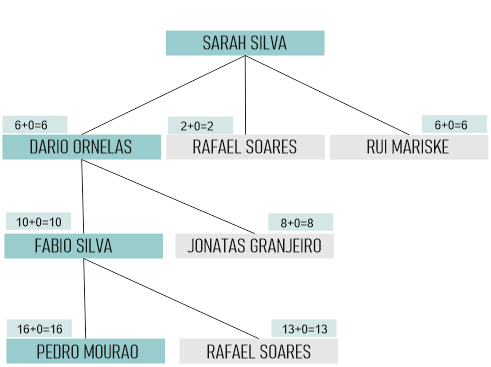
Para fins de exemplo utilizamos o seguinte grafo:



Teste com máximo de 3 ligações (2 nós intermédios):

Uma imagem com texto

Descrição gerada automaticamente



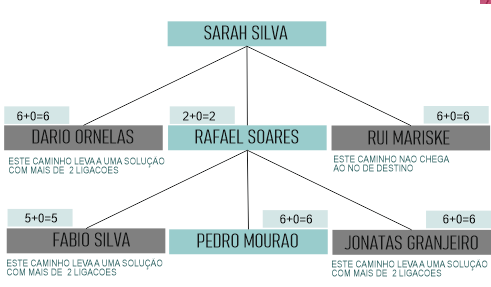
O resultado e a traçagem acima apresentada validam ambos o resultado obtido e o cálculo do custo.

Esta solução apresenta 2 nós intermédios e, de forma a validarmos que o nosso predicado funcione no que diz respeito ao controlo do número de ligações iremos testar a mesma condição indicando como número máximo de ligações o valor 2 e 1.

Teste com máximo de 2 ligações (1 nó intermédio):

Uma imagem com texto

Descrição gerada automaticamente



Teste com máximo de 1 ligações (0 nós intermédios):

Uma imagem com texto

Descrição gerada automaticamente

Tal como expectável, nenhuma solução é disponibilizada uma vez que não existe nenhuma ligação direta entre o utilizador Orig e Dest.

## Implementar a estimativa no algoritmo a\*

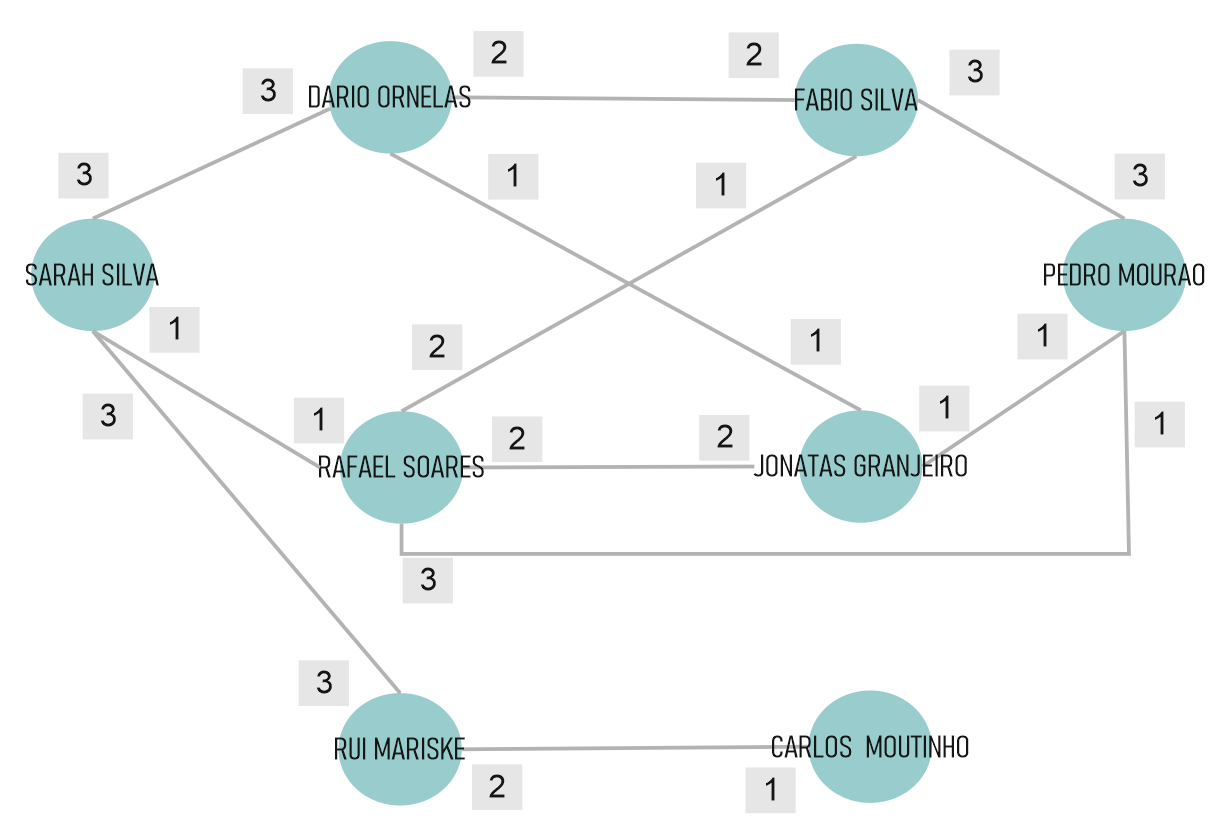
Neste exercício implementamos uma estimativa no algoritmo A\*.

Uma imagem com texto

Descrição gerada automaticamente

Para o cálculo da estimativa , obtivemos uma lista ordenada de todas as forças de ligação presentes no grafo (método: get\_ForcasLigacao). A estimativa é calculada multiplicando a força máxima do grafo com o número de níveis ainda a percorrer (diferença entre o nível máximo imposto e o nível atual). A força máxima do grafo usada no cálculo varia de nível para nível. A lista de todas as forças ordenadas, permite obter as forças de ligação usadas por cada nível , isto é, numa primeira ligação a estimativa é calculada usando o primeiro valor da lista de forças; numa segundo iteração, será usada o segundo valor da lista e por aí a diante.

Para fins de exemplo utilizamos o seguinte grafo:



Teste com máximo de 3 ligações (2 nós intermédios):

Uma imagem com texto

Descrição gerada automaticamente

Teste com máximo de 2 ligações (1 nó intermédios):

Uma imagem com texto

Descrição gerada automaticamente

Neste exercício, sendo exigido um máximo de 2 ligações o caminho Sarah-Dario-Fábio-Pedro não é possível uma vez que possui 2 nós intermédios. O segundo caminho com maior estimativa é Sarah-Rui-Carlos e neste caso também não é possível pelo facto do Carlos não ter mais ligações. Por fim, temos o caminho Sarah-Rafael-Pedro que preenche todos os requisitos e é dado como resposta ao problema.

Teste com máximo de 1 ligações (0 nó intermédios):

Uma imagem com texto

Descrição gerada automaticamente

Neste teste é pedido um caminho com 1 ligação apenas, o que implica uma ligação direta entre o userOrig e o UserDest. Não havendo esse tipo de ligação no nosso grafo, o resultado é false.

## Adaptação do Best First ao problema da determinação do caminho mais forte (máximo de N ligações)

Neste exercício implementamos um máximo de ligações no algoritmo ao BestFirst.



De forma a controlar o número de ligações controlamos o número de nós na lista do caminho. Consideramos o número de ligações atuais como sendo o número de nós presentes na lista menos 1 (referente ao userOrig) – exemplo: na lista [S-T-E-A] consideramos o nível=3 (3 ligações). Esse valor é então comparado ao nível máximo inserido pelo utilizador.

Teste com máximo de 3 ligações (2 nós intermédios):

Uma imagem com texto, interior, captura de ecrã

Descrição gerada automaticamente

Teste com máximo de 2 ligações (1 nós intermédios):

Uma imagem com texto, interior, captura de ecrã

Descrição gerada automaticamente

Teste com máximo de 1 ligações (sem nós intermédios):

Uma imagem com texto, interior, captura de ecrã

Descrição gerada automaticamente

## Adaptação do Primeiro em Profundidade para gerar a melhor solução (já implementado no Sprint anterior) para o máximo de N ligações

Neste exercício implementamos um máximo de ligações no algoritmo ao DFS.



De forma a validar um número máximo de ligações, a cada caminho verificamos o tamanho da lista e subtraímos 1 (relativo ao userOrig). Efetuando essa subtração estamos a auferir o número de ligações do caminho em estudo. Este valor é depois comparado com o valor máximo de ligações imposto pelo utilizador.

## Comparação dos 3 métodos com vários exemplos, comparando tempos de geração da solução e valor da solução gerada

**NÍVEL 2**

|  |  |  |
| --- | --- | --- |
|  | **TEMPO** | **FORÇA DE LIGAÇÃO** |
| **A\*** | 0.00 | 6 |
| **BestFirst** | 0.00 | 6 |
| **Primeiro em Profundidade** | 0.000999 | 6 |

Uma imagem com texto

Descrição gerada automaticamente

**NÍVEL 3**

|  |  |  |
| --- | --- | --- |
|  | **TEMPO** | **FORÇA DE LIGAÇÃO** |
| **A\*** | 0.001004 | 16 |
| **BestFirst** | 0.00 | 16 |
| **Primeiro em Profundidade** | 0.000971 | 16 |

Uma imagem com texto

Descrição gerada automaticamente

**NÍVEL 4**

|  |  |  |
| --- | --- | --- |
|  | **TEMPO** | **FORÇA DE LIGAÇÃO** |
| **A\*** | 0.00 | 16 |
| **BestFirst** | 0.00 | 16 |
| **Primeiro em Profundidade** | 0.0006070 | 17 |

Uma imagem com texto

Descrição gerada automaticamente

Na rede desenhada para o efeito, o BestFirst parece ser mais rápido na determinação de um caminho. O primeiro em profundidade tem a vantagem de verificar todos os caminhos e retornar o mais forte, no entanto, os recursos consumidos na procura da solução dependem do tamanho do problema. Se neste exemplo o primeiro em profundidade permite com certeza obter o melhor resultado, numa situação real poderá não ser realista a sua utilização. Num enquadramento real, por exemplo no caso de uma rede social, a melhor escolha assentaria no A\*. Ainda que não garanta o caminho mais forte apresentará sempre o caminho com melhor estimativa.

## Implementação da função multicritério que contemple forças de ligação e diferença entre likes e dislikes

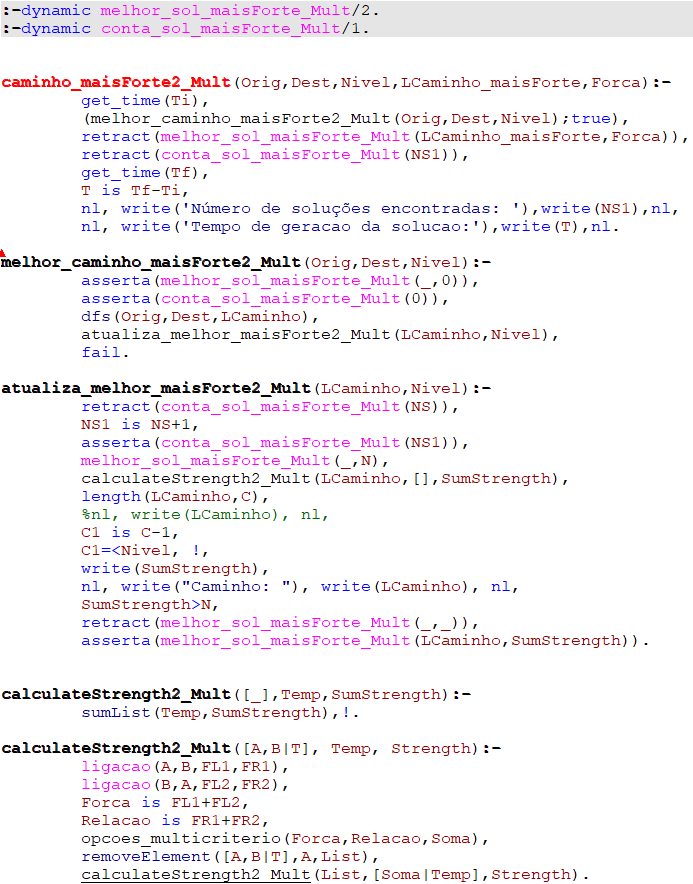
Uma imagem com texto

Descrição gerada automaticamente

De forma a limitarmos os valores apresentados pela força de relação e tomar em conta os valores da força de ligação criamos uma regra com as várias possibilidades que pode o fator multicritério apresentar (opções\_multicriterio). Uma vez o fator multicritério calculado, determinamos a estimativa multiplicando o fator multicritério pela diferença entre o número de ligações máximas impostas com o nível atual.

## Adaptação dos 3 métodos (Primeiro em Profundidade, Best First e A\*) para considerar a função multicritério do ponto anterior

### Primeiro em profundidade



### Best First



### A\*



## Comparação dos 3 métodos com vários exemplos e usando a função multicritério

**NÍVEL 2**

|  |  |  |
| --- | --- | --- |
|  | **TEMPO** | **FORÇA DE LIGAÇÃO** |
| **A\*** | 0.02129888 | 6 |
| **BestFirst** | 0.02681207 | 6 |
| **Primeiro em Profundidade** | 0.00229811 | 6 |

Uma imagem com texto

Descrição gerada automaticamente

**NÍVEL 3**

|  |  |  |
| --- | --- | --- |
|  | **TEMPO** | **FORÇA DE LIGAÇÃO** |
| **A\*** | 0.02404618 | 6 |
| **BestFirst** | 0.02600383 | 16 |
| **Primeiro em Profundidade** | 0.0 | 16 |

Uma imagem com texto

Descrição gerada automaticamente

**NÍVEL 4**

|  |  |  |
| --- | --- | --- |
|  | **TEMPO** | **FORÇA DE LIGAÇÃO** |
| **A\*** | 0.01822710 | 6 |
| **BestFirst** | 0.02463603 | 16 |
| **Primeiro em Profundidade** | 0.00102901 | 16 |

Uma imagem com texto

Descrição gerada automaticamente

Como mencionado anteriormente, o uso de Primeiro em profundidade permite obter a melhor solução, no entanto, quando enfrentamos uma rede de grande tamanho este algoritmo torna-se inviável. O Best first, como métodos de pesquisa informado, apresenta uma melhor eficiência comparativamente ao anterior, mas não tão eficaz como o A\*.

O A\*, devido ao cálculo da estimativa, permite otimizar o resultado. No entanto, uma solução otimizada não é sinónima de melhor solução. Como exemplo, abaixo está o resultado do A\* para uma rede até o 3º nível. Estando este algoritmo orientado para devolver um caminho com maior estimativa, este devolveu um caminho com uma estimativa de 200 e força de ligação igual a 6. Ora, mais à frente verifica-se a presença de um caminho com uma maior força de ligação (16) mas como possui uma estimativa mais baixa não é apresentada.

Uma imagem com texto

Descrição gerada automaticamente

# Conclusão

Este sprint permitiu estudar e assimilar o funcionamento de diversos métodos de pesquisa.

A pesquisa em profundidade apresenta a vantagem de ter poucos requisitos em termos de memória, no entanto não garante que a primeira solução encontrada seja a melhor solução. Este fator faz com que este seja eficiente numa amostra pequena, no entanto, numa situação real, tal como seria uma rede social, o seu uso é totalmente inviável.

O Best First é um método de pesquisa dito informado que se assemelha bastante com o primeiro em profundidade. A grande diferença situa-se no facto da decisão sobre o nó a explorar é efetuado com base numa decisão local (por exemplo a força da ligação). Este algoritmo não garante que a melhor solução seja encontrada.

O A\* é o algoritmo ideal para redes de maior dimensão uma vez que permite construir uma solução com base numa estimativa. A estimativa é calculada usando a soma do custo acumulado desde o nó de origem até o nó atual e a estimativa do custo até o nó final. Ainda que não garanta a melhor solução, gera uma solução otimizada independentemente do tamanho da rede em estudo.