

# **Little Fifa Graph Analyzer**

## Análise da relação entre jogadores para formação de um time consistente

Rafael Rampim Soratto, Lucas Gabriel da Silva

*Universidade Tecnológica Federal do Paraná - UTFPR Campus Campo Mourão*

12 de julho de 2019

## **1 Introdução**

O seguinte trabalho tem o intuito de criar um grau de separação entre dois jogadores profissionais quaisquer de acordo com os times em que ambos já treinaram. A aplicação principal é para um time verificar se os novos jogadores a serem contratados irão se habituar bem com o time: se o jogador possuir um grau de distância alto em relação ao resto do time então isso significa que a contratação pode não ser vantajosa. Por outro lado se o jogador for “próximo” do seu elenco do time então a chance dele se habituar no time deve ser maior.

### **1.1 Ferramentas**

Como metodologia para tal experimento será utilizada a estrutura de grafos (conjunto de vértices e arestas) para representar os jogadores e seu grau de separação. Os jogadores são representados como um vértice do grafo, e a relação entre os jogadores pode ser uma aresta.

## **2 Estrutura**

O trabalho foi desenvolvido na linguagem python e possui algumas bibliotecas essenciais para seu funcionamento:

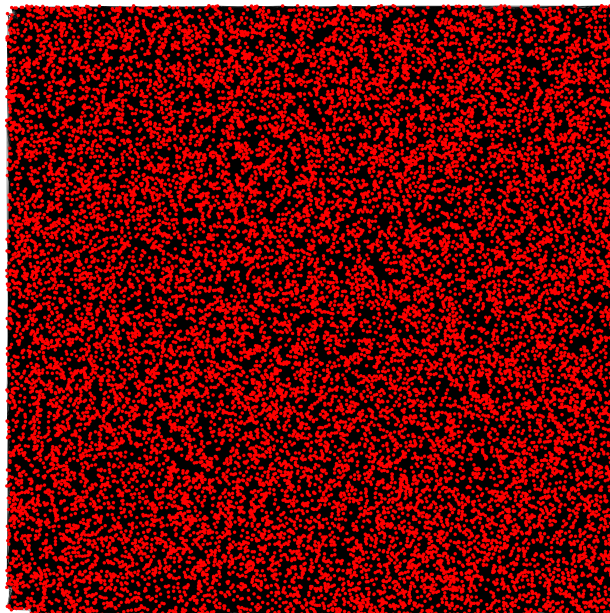
- Para o grafo  
panda;  
networkx;

Para o funcionamento do experimento proposto será extraída uma base de dados contendo informações essenciais dos jogares para inserir no grafo. Estes dados são armazenados no formato “.csv” disponível em <https://github.com/ewenme/squads/tree/master/data>. Tal base de dados foi convertida para o formato de grafo utilizado para o algoritmo (.gml) para então gerar os vértices e arestas. Desta maneira a biblioteca **networkx** possui o comando para gerar o grafo à partir de um arquivo .gml:

```
G = nx.read_gml('grafo.gml')
```

Como a base de dados é grande foram gerados 22 mil vértices (jogadores) e 760 mil arestas que representam a relação entre jogadores que atuaram no mesmo time. O grafo pode ser visualizado na figura 1:

Figura 1: Visualização do grafo



## 3 Metodologia

Com o grafo criado é possível utilizar o algoritmo de caminho mínimo inicialmente criado por Dijkstra. Este algoritmo encontra a distância mínima de um vértice fonte para os outros vértices do grafo que são conexos a ele por algum caminho, o objetivo é minimizar o caminho entre dois vértices. No trabalho será minimizada a distância entre dois jogadores.

### 3.1 Selecionando jogadores na base de dados

Para selecionar os jogadores basta utilizar o seguinte script em python:

```
print('Digite dois jogadores que nunca jogaram juntos: ')
jogador1 = 'null'
# enquanto o usuário não sair
while(jogador1 != 'exit'):
    # recebe um jogador
    jogador1 = input('Digite o nome do primeiro jogador: ')
    if(jogador1 != 'exit'):
        if(G.has_node(jogador1)):
            print('JOGADOR ENCONTRADO NO GRAFO!')
            # procura o segundo
            jogador2 = input('Digite o segundo jogador: ')
            if(jogador2 != 'exit'):
                if(G.has_node(jogador2)):
                    print('JOGADOR ENCONTRADO NO GRAFO!\n')
                    # Caminho mínimo
                    caminhoMinimo(jogador1, jogador2, G)
                else:
                    print('JOGADOR 2 NAO ENCONTRADO ')
            else:
                print('JOGADOR 1 NAO ENCONTRADO')
```

Portanto a função **has\_node** recebe um jogador e retorna ele se ele existir no grafo.

## 3.2 Caminho Mínimo

De acordo com [1] algoritmo de caminho mínimo foi criado pelo cientista da computação Edsger Dijkstra em 1956 e publicado em 1959. Neste algoritmo encontra-se o caminho mínimo entre um vértice fonte e todos outros vértices do grafo, com complexidade de vértices mais arestas.

Com a biblioteca **networkx** é possível encontrar o caminho mínimo entre dois vértices (jogadores):

```
def caminhoMinimo(player1, player2, grafo):
    try:
        caminho = nx.shortest_path(grafo, player1, player2)
        print('A distância mínima é ', len(caminho)-1)
        print('O caminho percorrido foi: ')
        for node in caminho:
            print(node, '\n')
    except:
        # se os jogadores não possuem um caminho entre si
        print('OS JOGADORES NÃO POSSUEM NENHUMA LIGAÇÃO\n')
```

O funcionamento do algoritmo de caminho mínimo é bem simples:

1. Inicializa todos os vértices em um conjunto Q;

Todos vértices começam com a distância  $D[V]$  infinita e predecessor nulo, exceto o primeiro vértice que possui distância 0 por ser a raiz.

2. Enquanto o conjunto Q não for vazio remove-se a menor distância e para todos seus adjacentes:

Se  $(D[V] > D[U] + \text{aresta}U - V)$

Então:  $D[V] = D[U] + \text{aresta}U - V$  e

$\text{Predecessor}[V] = U$

## 4 Testes

O primeiro teste realizado foi para verificar qual a distância do jogador Cristiano Ronaldo e Cafú dado que eles nunca jogaram juntos antes, porém eles possuem um grau de separação próximo pois por serem bons jogadores já

possuíram muitos companheiros de equipe que são responsáveis por diminuir o grau de separação entre ambos:

Figura 2: Grau de separação entre Cristiano Ronaldo e Cafú

```
BEM-VINDO A O FIFINHA-GRAFO
```

```
CARREGANDO O GRAFO...
```

```
GRAFO CARREGADO!
```

```
OBS: PARA SAIR DIGITE exit
```

```
Digite o nome de dois jogadores que não atuaram juntos durante a carreira:
```

```
Digite o nome do primeiro jogador: Cristiano Ronaldo
```

```
JOGADOR ENCONTRADO NO GRAFO!
```

```
Digite o nome do segundo jogador: Cafú
```

```
JOGADOR ENCONTRADO NO GRAFO!
```

```
A distância mínima é 2
```

```
O caminho percorrido foi:
```

```
Cristiano Ronaldo
```

```
Kaká
```

```
Cafú
```

```
Digite o nome do primeiro jogador:
```

de acordo com a Figura 2 é possível visualizar que o grau de separação é 2 pois ambos jogadores já jogaram com Kaká provando que os 3 jogadores possuem alto nível de futebol sendo que ambos já foram melhor do mundo pelo menos uma vez.

## 5 Conclusões

O intuito do trabalho é utilizar uma ferramenta de grafos chamada de caminho mínimo para analisar a relação entre os jogadores para verificar sua proximidade que reflete diretamente no entrosamento dentro de campo. Alguns testes foram realizados é conclui-se que para um jogador se entrosar

bem no seu elenco ele precisa ter uma distância de no máximo 2 dos outros jogadores do time, isso no caso ideal. Se a distância for 1 significa que os dois jogadores já jogaram juntos o que pode ser vantajoso também para equipe.

## Referências

- [1] Caminho de Custo Mínimo, Online, Disponível em: <http://www.inf.ufsc.br/grafos/temas/custo-minimo/dijkstra.html>, Acesso: 11-07-2019.