

Relatório Acadêmico – Projeto de Ciência de Redes

Estrutura de Dados

COMPONENTE CURRICULAR: Estrutura de Dados

PROFESSOR: Cleyton Rodrigues

CURSO: Engenharia da Computação

ALUNO: Bruno Eduardo de Vasconcelos Diniz

Projeto de Ciência de Redes

1. Objetivo

Este projeto teve como finalidade aplicar os conceitos de grafos no contexto da Ciência das Redes, uma área interdisciplinar que estuda sistemas formados por elementos conectados. Para isso, foi construído um grafo de interações entre os **jogadores do elenco atual do FC Barcelona**. A proposta consistiu em representar e analisar os níveis de relacionamento e características comuns entre os jogadores, utilizando grafos e seus algoritmos clássicos.

2. Justificativa e Modelagem

A base de dados principal foi construída diretamente no código Python, contendo os seguintes atributos para cada jogador (nó): nome, posicao, nacionalidade, altura, idade, nota_performance e e_capitao. A segunda base, personagens.csv, foi criada para mapear cada nó à sua respectiva imagem .png.

- **Nós (Vértices):** Cada jogador relevante do elenco atual do FC Barcelona. Arestas: conexões de aliança ou inimizade entre os personagens.
- **Arestas (Conexões):** As arestas representam o nível de "conexão" ou similaridade entre Os grafos foram construídos separadamente para cada saga. Essa conexão é definida por uma combinação de fatores:
 - **tipo_conexao:** Se compartilham a mesma_posicao ou a mesma_nacionalidade.
 - **Pesos:** A intensidade de cada conexão é inversamente proporcional a um "custo" calculado. Um peso menor indica uma conexão mais forte. Este custo é influenciado pela proximidade de:

- **Altura:** Jogadores com alturas mais próximas têm uma conexão mais forte.
- **Idade:** Jogadores com idades mais próximas têm uma conexão mais forte.
- **Nota de Performance:** Jogadores com notas de performance mais próximas têm uma conexão mais forte.

Essa escolha foi devida a complexidade inerente a um time de futebol, onde a interconexão e as características dos jogadores são cruciais para o desempenho coletivo. A vasta gama de atributos individuais e a dinâmica de um elenco serviram como excelente inspiração para representar essas interações por meio de grafos, permitindo uma análise estrutural e relacional do time. O capitão, **Robert Lewandowski**, foi definido como o nó principal para análises de busca e centralidade, representando um ponto de referência fundamental na rede.

3. Estrutura e Dados de Implementação

Foi utilizada a biblioteca **NetworkX** para a modelagem e manipulação da rede. O grafo foi construído programaticamente em Python, adicionando cada jogador como um nó com seus respectivos atributos. As arestas foram criadas entre todos os pares de jogadores, com seus pesos calculados dinamicamente com base nos critérios definidos (posição, nacionalidade, altura, idade, nota de performance). A representação interna do grafo é baseada em uma **lista de adjacência**, otimizada pela implementação do NetworkX.

Segue trecho da lógica implementada para criar o grafo via NetworkX e calcular os pesos:

```
def calcular_peso_aresta(jogador1, jogador2):
    # Peso base inicial: quanto maior, mais 'distante' os jogadores estão por padrão.
    peso = 10.0

    # 1. Mesma Posição: Conexão muito forte
    if jogador1["posicao"] == jogador2["posicao"]:
        peso -= 4.0 # Reduz significativamente o peso (menor peso = conexão mais forte)

    # 2. Mesma Nacionalidade: Conexão forte
    if jogador1["nacionalidade"] == jogador2["nacionalidade"]:
        peso -= 3.0 # Reduz o peso
```

```

G = nx.Graph()

# Adicionar nós (jogadores) com seus atributos
for jogador in jogadores_data:
    G.add_node(jogador["nome"], **jogador) # Adiciona todos os atributos do dicionário como atributos do nó

# Adicionar arestas ponderadas
for i in range(len(jogadores_data)):
    for j in range(i + 1, len(jogadores_data)): # Evita duplicatas e conexões de um nó consigo mesmo
        jogador1 = jogadores_data[i]
        jogador2 = jogadores_data[j]

        # Garantir que não estamos conectando o mesmo jogador
        if jogador1["nome"] != jogador2["nome"]:
            peso = calcular_peso_aresta(jogador1, jogador2)
            G.add_edge(jogador1["nome"], jogador2["nome"], weight=peso)

```

<https://github.com/Brunovdiniz/Barcelona-EE.git>

4. Algoritmo e Métricas Aplicadas

Algoritmos utilizados no projeto:

- **Busca em Largura (BFS):** Foi aplicada a partir do nó do capitão (Robert Lewandowski) para identificar os jogadores diretamente conectados a ele (Nível 1) e aqueles conectados em até dois passos (Nível 2). Isso permite visualizar a "proximidade" não ponderada entre os jogadores e o capitão, revelando subgrupos imediatos ou a extensão da rede a partir de um ponto central.
- **Algoritmo de Dijkstra:** Criado para encontrar os caminhos mais "fortes" (ou seja, de menor custo/peso acumulado) do capitão para outros jogadores chave do time (como goleiro, meio-campista e defensor). Este algoritmo evidencia as conexões mais eficientes ou com maior similaridade de atributos, permitindo identificar os "caminhos de entrosamento" mais otimizados dentro da rede. A lógica do peso inversamente proporcional (quanto mais forte a ligação, menor o custo) foi crucial para essa interpretação.

Métricas da Ciência das Redes:

- **Grau dos Nós:** Calculado para cada jogador, indicando o número total de conexões que ele possui na rede. Jogadores com alto grau são considerados "hubs" de conexão.
- **Centralidade de Grau:** Uma versão normalizada do grau, indicando a proporção de conexões que um jogador tem em relação ao máximo possível.
- **Centralidade de Intermediação (Betweenness Centrality):** Mede a frequência com que um jogador aparece nos caminhos mais curtos entre outros pares de jogadores na rede. Jogadores com alta centralidade de intermediação atuam como "pontes" essenciais para o fluxo de "informação" ou interação.
- **Centralidade de Proximidade (Closeness Centrality):** Mede o quão perto um jogador está de todos os outros na rede, considerando os pesos das arestas.

- Jogadores com alta proximidade podem interagir ou influenciar o restante da rede mais rapidamente.
- **Componentes Conectados:** Verifica se a rede forma um único grupo interconectado ou se há subgrupos isolados. No caso do FC Barcelona, a rede demonstrou ser totalmente conectada, indicando que todos os jogadores estão interligados por alguma combinação de critérios.
- **Diâmetro da Rede:** Representa o caminho mais longo entre quaisquer dois jogadores na rede (em termos de número de arestas, não ponderado). Ele fornece uma medida da "extensão" máxima da rede.

Pode-se perceber que os algoritmos de busca e as métricas foram desenvolvidos para percorrer e analisar o grafo gerado pela estrutura do **nx.Graph()** da biblioteca NetworkX, utilizando seus recursos para extrair insights significativos sobre a rede de jogadores.

Além disso, pode perceber que os algoritmos de busca foram desenvolvidos para percorrer o grafo gerado pela estrutura do **nx.Graph()**, a biblioteca do Networkx.

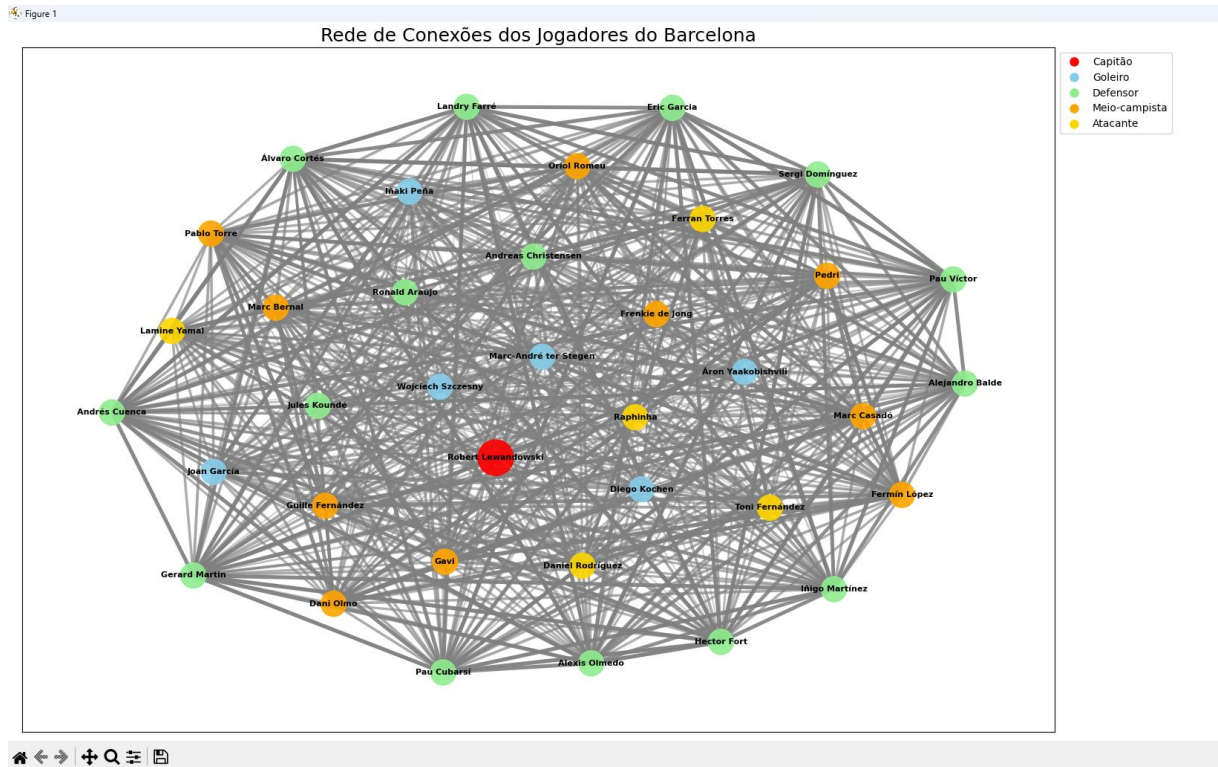
5. Visualizações

Para facilitar a visualização da rede, foi utilizada a biblioteca **Matplotlib** em conjunto com o NetworkX.

A figura gerada exibe o grafo dos jogadores do Barcelona:

- Os **nós** (jogadores) são representados por círculos coloridos:
 - **Capitão (Robert Lewandowski):** Em destaque na cor vermelha e com um tamanho maior.
 - **Goleiros:** Em azul claro.
 - **Defensores:** Em verde claro.
 - **Meio-campistas:** Em laranja.
 - **Atacantes:** Em dourado.
- As **arestas** (conexões) são representadas por linhas entre os jogadores. A espessura e a opacidade das linhas variam conforme o peso da conexão:

- **Linhas mais grossas e mais opacas:** Indicam uma conexão mais forte (menor custo) entre os jogadores, refletindo maior similaridade em altura, idade, nota de performance, ou o compartilhamento de posição/nacionalidade.
- **Linhas mais finas e mais transparentes:** Indicam uma conexão mais fraca (maior custo).



Análise da Visualização: A visualização permite identificar padrões estruturais importantes, como subgrupos de jogadores (por exemplo, clusters de defensores ou meio-campistas com muitas conexões internas). O destaque do capitão, **Robert Lewandowski**, e a variação visual das arestas, ajudam a interpretar rapidamente quem são os jogadores mais conectados ou quais pares têm uma "sinergia" maior, conforme nossos critérios.

6. Análise Crítica e Aprendizados

Ao realizar o desenvolvimento deste projeto, foi possível observar a importância da modelagem de sistemas complexos por meio de grafos. A definição clara dos nós (jogadores) e das arestas (tipos e força de conexão) permitiu uma representação significativa das relações dentro de um time de futebol.

A aplicação dos algoritmos de busca (BFS e Dijkstra) e o cálculo das métricas de centralidade foram fundamentais para obter insights sobre a estrutura da rede:

- O **BFS** demonstrou a "proximidade" direta dos jogadores ao capitão, mostrando uma rede bastante compacta onde a maioria dos jogadores está a apenas um ou dois passos dele.
- O **Dijkstra** revelou os caminhos de "maior entrosamento" (menor custo) entre o capitão e jogadores em diferentes posições, indicando quais conexões são otimizadas pelos critérios de similaridade definidos.
- As **métricas de centralidade** foram essenciais para identificar os jogadores mais influentes ou conectados, não apenas pelo número de conexões (grau), mas também pela sua importância como "pontes" (intermediação) ou pela sua capacidade de se comunicar rapidamente com o resto do time (proximidade). A alta centralidade de grau para a maioria dos nós e o diâmetro da rede em 1 indicam uma rede altamente densa e interconectada, o que é esperado em um time onde todos os jogadores se conectam por múltiplos fatores.

Em suma, este projeto proporcionou um melhor entendimento dos princípios de Ciência de Redes e da aplicação prática de grafos na análise de um sistema real, como um time de futebol, demonstrando como a representação gráfica e as métricas podem revelar padrões e estruturas que seriam difíceis de identificar apenas com dados brutos.

Segue o link do github: <https://github.com/Brunovdiniz/Barcelona-EE.git>