

Relatório de Projeto: Knowledge Graphs

Aluno: Emmanoel Barbosa

Disciplina: Estrutura de Dados **Data:** 11 de dezembro

Tema: Modelagem de Transferências de Futebol usando Grafos de Conhecimento

1. Descrição da Aplicação

O objetivo deste projeto foi desenvolver um **Knowledge Graph (Grafo de Conhecimento)** para representar e analisar dados do mercado de transferências no futebol brasileiro. O estudo de caso foca especificamente no fluxo de saída de jogadores do **Santa Cruz Futebol Clube** após a temporada de 2016 para o ano de 2017.

A aplicação utiliza a teoria dos grafos para transformar dados tabulares de transferências em uma rede semântica. Isso permite visualizar não apenas quem saiu, mas para onde foram e qual a relação posicional (atacante, goleiro, etc.) dentro dessa rede. Em *Sports Analytics*, essa abordagem é crucial para entender o "ciclo de vida" de elencos e o mapeamento de mercado dos clubes.

2. A Base de Conhecimento Criada

A base de conhecimento foi estruturada definindo entidades (nós) e semânticas de conexão (arestas) que representam a realidade do mercado esportivo.

2.1. Estrutura dos Nós (Entidades)

Os nós foram categorizados através de um atributo **type** para diferenciar as entidades no grafo:

- **Clube Origem:** O nó centralizador (neste cenário, o "Santa Cruz").
- **Jogador:** As entidades dinâmicas que sofrem a ação de transferência (ex: Grafite, Keno, Tiago Cardoso).
- **Clube Destino:** As entidades receptoras dos atletas (ex: Athletico-PR, Palmeiras, Náutico).

2.2. Propriedades e Metadados

Cada nó do tipo "Jogador" foi enriquecido com propriedades adicionais (**props**), armazenando dados fundamentais como a **Posição** (Goleiro, Zagueiro, Atacante). Isso permite, futuramente, consultas refinadas como "*Quais defensores saíram do clube?*".

2.3. Relacionamentos (Arestas)

As arestas são direcionadas e possuem rótulos semânticos:

- **[jogou]:** Conecta **Santa Cruz** → **Jogador**. Representa o vínculo anterior.
- **[foi para]:** Conecta **Jogador** → **Clube Destino**. Representa a transação de mercado.

3. Implementação do Código

A implementação foi realizada na linguagem **Python**, escolhida pela sua robustez em Ciência de Dados. Foram utilizadas as seguintes bibliotecas:

- **NetworkX**: Utilizada como o motor principal de grafos (**DiGraph**). Foi responsável por gerenciar a estrutura de dados (listas de adjacência) e calcular o layout dos nós.
- **Matplotlib**: Utilizada para a renderização gráfica da rede.

Destaques da Implementação

O código foi estruturado em uma classe **KnowledgeGraph**, promovendo a orientação a objetos:

1. **Método `add`**: Garante a unicidade dos nós e atribuição correta de metadados.
2. **Método `addRelations`**: Estabelece as conexões direcionadas.
3. **Método `exibir_visualizacao_limpa`**: Implementa um algoritmo de layout em camadas (**`shell_layout`**). Esta escolha foi estratégica para organizar visualmente o fluxo:
 - *Centro*: Santa Cruz.
 - *Camada Média*: Jogadores.
 - *Camada Externa*: Clubes de Destino.

4. Potenciais de Aplicação

A utilização de Knowledge Graphs neste domínio apresenta potenciais significativos:

1. **Análise de Scouting**: Clubes podem usar essa estrutura para identificar "rotas de transferência" frequentes. Ex: Identificar que o *Clube X* frequentemente contrata *Laterais* do *Clube Y*.
2. **Sistemas de Recomendação**: Com um grafo populado com dados históricos, algoritmos de *Link Prediction* poderiam sugerir prováveis destinos para um jogador baseado em transferências passadas de atletas com perfil similar.
3. **Integração de Dados Heterogêneos**: O grafo permite conectar dados díspares facilmente. Poderíamos adicionar nós de "Agentes/Empresários" ou "Treinadores", revelando conexões ocultas no mercado (ex: um treinador que sempre leva os mesmos 3 jogadores para seus novos times).

5. Limitações Encontradas

Durante o desenvolvimento e análise, foram identificadas as seguintes limitações:

1. **Visualização Estática**: A biblioteca **matplotlib**, embora excelente para gráficos científicos, possui limitações para grafos densos. A imagem gerada é estática, impedindo a exploração interativa (zoom, clique para detalhes) que seria necessária em uma base de dados maior.
2. **Temporalidade**: O modelo atual é um "snapshot" (foto) de uma janela de transferência específica. Para uma análise contínua, o grafo precisaria suportar propriedades temporais nas arestas (ex: **`jogou: {inicio: 2016, fim: 2016}`**), aumentando a complexidade da modelagem.
3. **Escalabilidade de Inserção**: A inserção manual dos dados (hardcoded) não é escalável. Para produção, seria necessário um pipeline ETL (*Extract, Transform, Load*) conectando o grafo a uma API de futebol ou base de dados SQL.

6. Conclusão

O projeto cumpriu o objetivo de aplicar conceitos de Knowledge Graphs em um cenário real. A estrutura criada permitiu organizar a informação do desmanche do Santa Cruz de 2016 de forma clara e lógica. Diferente de uma tabela simples, o grafo evidencia visualmente a dispersão dos atletas, provando ser uma ferramenta valiosa para a inteligência de dados esportiva.