

Relatório de Projeto: Knowledge Graphs

Aluno: Emmanoel Barbosa

Disciplina: Estrutura de Dados

Data: 11 de dezembro de 2025

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 bibliotecas **NetworkX** (motor principal de grafos e gerenciamento de listas de adjacência) e **Matplotlib** (renderização gráfica).

Destaques da Implementação

O código foi estruturado em uma classe **KnowledgeGraph**, promovendo a orientação a objetos. As principais funcionalidades implementadas foram:

1. **Método `add`:** Garante a unicidade dos nós e atribuição correta de metadados.
2. **Método `addRelations`:** Estabelece as conexões direcionadas.
3. **Método `removeNode` (Nova Funcionalidade):** Implementa a manutenção do grafo. O método remove o nó especificado e varre a lista de adjacência para eliminar quaisquer conexões órfãs que apontavam para o nó removido, garantindo a integridade da estrutura.
4. **Método `exibir_visualizacao`:** 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:** 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, como nós de "Agentes" ou "Treinadores", revelando conexões ocultas no mercado.

5. Limitações Encontradas

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

1. **Visualização Estática:** A biblioteca matplotlib possui limitações para grafos densos, gerando imagens estáticas que impedem a exploração interativa (zoom, clique) necessária em bases maiores.
2. **Temporalidade:** O modelo atual é um "snapshot" de uma janela específica. Para análise contínua, o grafo precisaria suportar propriedades temporais nas arestas.
3. **Escalabilidade de Inserção:** A inserção manual dos dados (*hardcoded*) não é escalável. Para produção, seria necessário um pipeline ETL conectando o grafo a uma API de futebol ou banco 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.