Estuardo Menéndez (18072)
Lorena Beltrán (18629)
Roberto Castillo (18546)

Proyecto #2
Grupo: Booleano

Curso: Lógica Matemática
Instructor: Paulo Mejía

Segundo Semestre 2020
Fecha de Entrega: 21 de septiembre del 2020

Descripción

Los deportes son una de las áreas en donde los grafos son utilizados más ampliamente. Cualquier aficionado sueña con predecir cuales serán los resultados de un partido en particular, pero esto no es siempre fácil de predecir. Utilizando datos históricos se pueden tener predicciones que, si bien no son precisas en todos los casos, dan una idea general bastante buena de los posibles resultados. En el presente proyecto se trabaja con las estadísticas de los equipos de la NBA para predecir los resultados de los playoffs del año 2020, así como proveer al usuario de información relevante sobre sus equipos favoritos o de interes.
El programa, desarrollado en Neo4j (versión 4.1.0) hace uso de 3 tipos de nodos, los cuales son:

1. Equipo
    a) Oeste
    b) Este
2. Partido

En este caso, los partidos tienen asignado un año y tipo (primer partido, semifinal de conferencia, final de conferencia, gran final) mientras que el equipo tiene nombre y acrónimo. La relación que se trabajó fue la de ganar, la cual representa, para un partido en particular, cuantas veces el equipo ganó.

Script de instrucciones

Adicional al código con la creación de los equipos y partidos, se hicieron las siguientes instrucciones para analizar los datos dados e intentar predecir los resultados del partido que se está jugando actualmente (L.A. Lakers contra Denver Nuggets). El codigo para la creación de nodos puede ser enccontrado en el siguiente link: `https://github.com/EstuardoMenendez/ProyectoLogica2`

```
//Codigo para mostrar la historia de un equipo en particular en los playoffs.
MATCH (t:Team {Name: "Golden State"})-[w:WIN]->(:Playoff)<-[l:WIN]-()
RETURN t,w,l

//Codigo para mostrar una tabla con las estadisticas generales de los equipos
MATCH (t:Team)-[w:WIN]->(:Playoff)<-[l:WIN]-()
RETURN t.Name AS TEAM, SUM(w.Win) AS TOTAL_WIN, SUM(l.Win) AS TOTAL_LOSS,
(toFloat(SUM(w.Win)) / (toFloat(SUM(w.Win))+ toFloat(SUM(l.Win)))) AS WIN_PERCENTAGE
ORDER BY SUM(w.Win) DESC

//Codigo para mostrar el grafo del camino mas corto entre dos equipos
MATCH (t1:Team {Name: "L.A. Lakers"}),(t2:Team {Name:"Denver"}),
    p = AllshortestPaths((t1)-[*..14]-(t2))
RETURN p

//Codigo para sacar la ventaja que tendra un equipo sobre considerando el camino mas corto
MATCH p= AllShortestPaths((t1:Team {Name: "L.A. Lakers"})-[:WIN*0..14]-(t2:Team {Name:"Denver"}))
WITH [r IN relationships(p)| r.Win] AS RArray, LENGTH(p)-1  AS s
RETURN AVG(REDUCE(x = 0, a IN [i IN range(0,s) WHERE i % 2 = 0 | RArray[i] ] | x + a)) //total win
- AVG(REDUCE(x = 0, a IN [i IN range(0,s) WHERE i % 2 <> 0 | RArray[i] ] | x + a)) //total loss
AS NET_WIN
```

Consultas

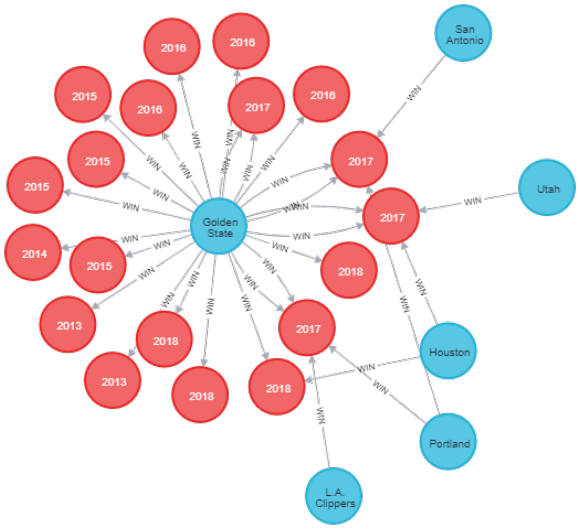

Figura 1: Grafo de todos los nodos que se relacionan con el equipo Golden State

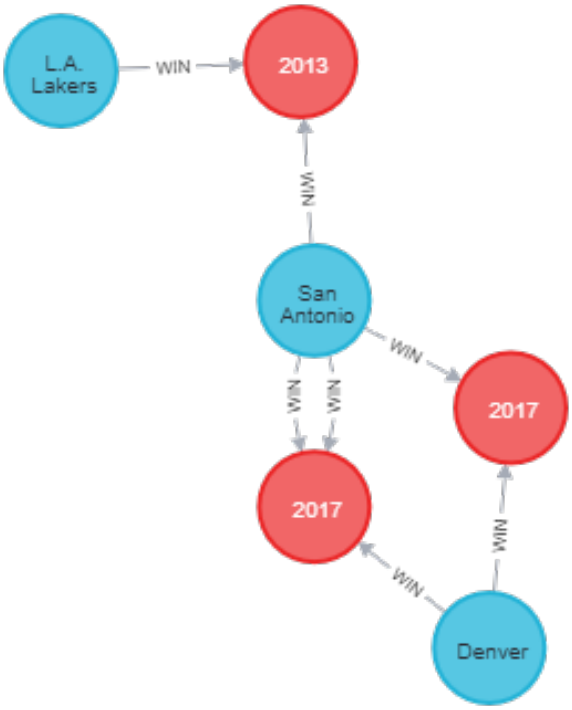

Figura 2: Grafo de camino más corto entre Denver y Lakers

Referencias

[1] Rossen, K. (2007). *Discrete Mathematics and its applications, seventh edition.*New York, U.S.: McGraw-Hill Companies, Inc.