



**Prueba Practica Sistemas Expertos Basados en casos.**

**Nombre: Luis Orellana P.**

**Objetivo:**

- Consolidar los conocimientos adquiridos en clase de los sistemas expertos basados en casos.

**Enunciado:**

Se desea generar un sistema de recomendación de películas, por tal motivo se va a utilizar una base de datos orientada a grafos para lograr esto se describe los pasos a seguir:

- 1) Con estos datos aplicar el algoritmo de KNN y Similitud de Coseno para la recomendación de películas, seguir el siguiente tutorial: <https://www.markhneedham.com/blog/2018/09/28/neo4j-graph-algorithms-cosine-game-of-thrones/> o <https://vladbatushkov.medium.com/one-month-graph-challenge-flags-5d30aec366a0>.
- 2) Finalmente realizar alguna interfaz para poder acceder a la recomendación e ingreso de datos y resultados de los procesos en python.

Generar el Informe en PDF y subir los scripts al repositorio Git para su evaluación.

**Scripts para la creación de de la creación de las banderas con sus colores**

```
CREATE (red:Color { name: "Red" })
CREATE (white:Color { name: "White" })
CREATE (blue:Color { name: "Blue" })
CREATE (green:Color { name: "Green" })
CREATE (yellow:Color { name: "Yellow" })
CREATE (black:Color { name: "Black" })
CREATE (f1:Flag { name: "Belarus" })
CREATE (f1)-[:CONTAINS { weight: 60 }]->(red)
CREATE (f1)-[:CONTAINS { weight: 30 }]->(green)
CREATE (f1)-[:CONTAINS { weight: 10 }]->(white)
CREATE (f2:Flag { name: "Russia" })
CREATE (f2)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f2)-[:CONTAINS { weight: 33 }]->(blue)
CREATE (f2)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f3:Flag { name: "Ukrain" })
CREATE (f3)-[:CONTAINS { weight: 50 }]->(yellow)
CREATE (f3)-[:CONTAINS { weight: 50 }]->(blue)
CREATE (f4:Flag { name: "Finland" })
CREATE (f4)-[:CONTAINS { weight: 80 }]->(white)
CREATE (f4)-[:CONTAINS { weight: 20 }]->(blue)
CREATE (f5:Flag { name: "Sweden" })
CREATE (f5)-[:CONTAINS { weight: 20 }]->(yellow)
CREATE (f5)-[:CONTAINS { weight: 80 }]->(blue)
CREATE (f6:Flag { name: "Norway" })
CREATE (f6)-[:CONTAINS { weight: 70 }]->(red)
CREATE (f6)-[:CONTAINS { weight: 20 }]->(white)
CREATE (f6)-[:CONTAINS { weight: 10 }]->(blue)
```



**Prueba Practica Sistemas Expertos Basados en casos.**

```
CREATE (f7:Flag { name: "Denmark" })
CREATE (f7)-[:CONTAINS { weight: 80 }]->(red)
CREATE (f7)-[:CONTAINS { weight: 20 }]->(white)
CREATE (f8:Flag { name: "Estonia" })
CREATE (f8)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f8)-[:CONTAINS { weight: 33 }]->(blue)
CREATE (f8)-[:CONTAINS { weight: 33 }]->(black)
CREATE (f9:Flag { name: "Latvia" })
CREATE (f9)-[:CONTAINS { weight: 66 }]->(red)
CREATE (f9)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f10:Flag { name: "Lithuania" })
CREATE (f10)-[:CONTAINS { weight: 33 }]->(yellow)
CREATE (f10)-[:CONTAINS { weight: 33 }]->(green)
CREATE (f10)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f11:Flag { name: "Poland" })
CREATE (f11)-[:CONTAINS { weight: 50 }]->(red)
CREATE (f11)-[:CONTAINS { weight: 50 }]->(white)
CREATE (f12:Flag { name: "Germany" })
CREATE (f12)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f12)-[:CONTAINS { weight: 33 }]->(black)
CREATE (f12)-[:CONTAINS { weight: 33 }]->(yellow)
CREATE (f13:Flag { name: "Belgium" })
CREATE (f13)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f13)-[:CONTAINS { weight: 33 }]->(black)
CREATE (f13)-[:CONTAINS { weight: 33 }]->(yellow)
CREATE (f14:Flag { name: "Czechia" })
CREATE (f14)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f14)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f14)-[:CONTAINS { weight: 33 }]->(blue)
CREATE (f15:Flag { name: "Hungary" })
CREATE (f15)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f15)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f15)-[:CONTAINS { weight: 33 }]->(green)
CREATE (f16:Flag { name: "Romaina" })
CREATE (f16)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f16)-[:CONTAINS { weight: 33 }]->(yellow)
CREATE (f16)-[:CONTAINS { weight: 33 }]->(blue)
CREATE (f17:Flag { name: "Austria" })
CREATE (f17)-[:CONTAINS { weight: 66 }]->(red)
CREATE (f17)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f18:Flag { name: "Italy" })
CREATE (f18)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f18)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f18)-[:CONTAINS { weight: 33 }]->(green)
CREATE (f19:Flag { name: "Switzerland" })
CREATE (f19)-[:CONTAINS { weight: 90 }]->(red)
CREATE (f19)-[:CONTAINS { weight: 10 }]->(white)
```



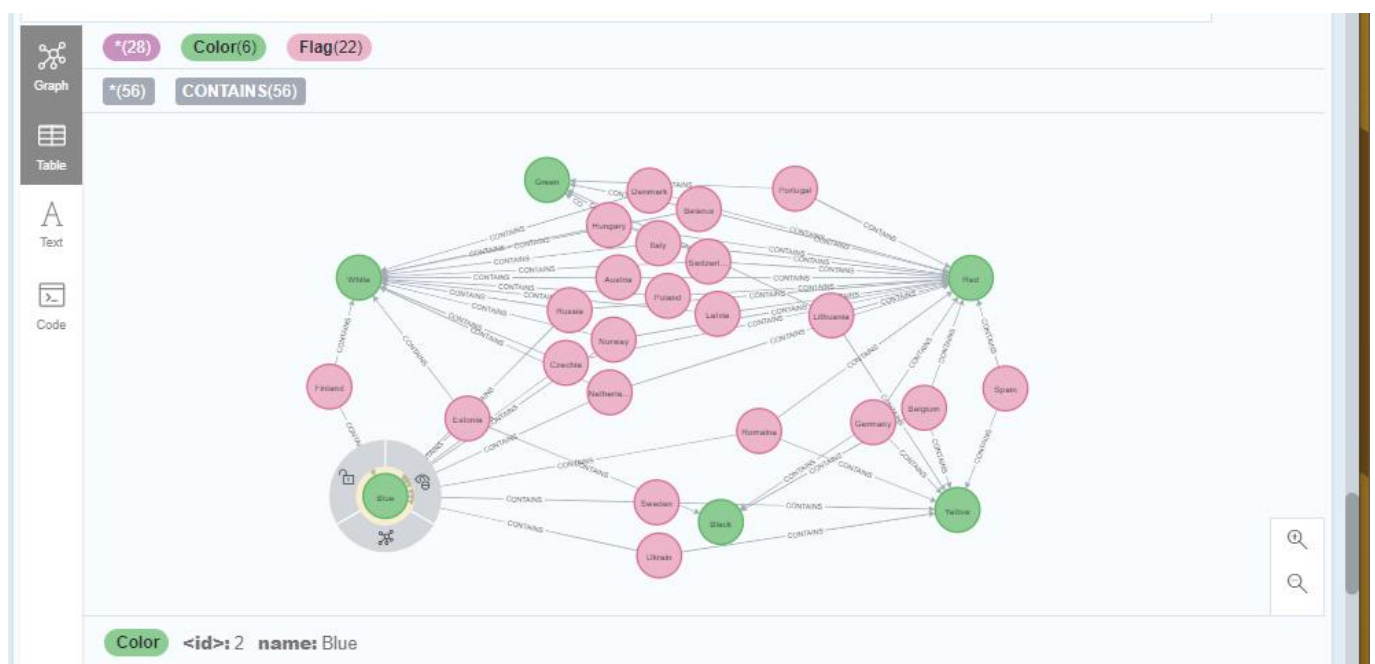
### Prueba Practica Sistemas Expertos Basados en casos.

```
CREATE (f20:Flag { name: "Spain" })
CREATE (f20)-[:CONTAINS { weight: 60 }]->(red)
CREATE (f20)-[:CONTAINS { weight: 40 }]->(yellow)
CREATE (f21:Flag { name: "Portugal" })
CREATE (f21)-[:CONTAINS { weight: 60 }]->(red)
CREATE (f21)-[:CONTAINS { weight: 40 }]->(green)
CREATE (f22:Flag { name: "Netherlands" })
CREATE (f22)-[:CONTAINS { weight: 33 }]->(red)
CREATE (f22)-[:CONTAINS { weight: 33 }]->(white)
CREATE (f22)-[:CONTAINS { weight: 33 }]->(blue)
```

Una témenos creado los 28 nodos en la base de datos Neo4j



Aquí tenemos el grafico con sus respectivas relaciones.





**Prueba Practica Sistemas Expertos Basados en casos.**

En primer lugar hagamos un pequeño filtrado y encontremos el color más utilizado:

```
MATCH (:Flag)-[c1:CONTAINS]->(:Color)
WITH sum(c1.weight) as total
MATCH (:Flag)-[c2:CONTAINS]->(cl:Color)
WITH cl.name as colorName, sum(c2.weight) as colorUsed, total
RETURN colorName, colorUsed * 100 / total as percentage
ORDER BY percentage DESC
LIMIT 10
```

**Como Salida temenos**

"colorName"	"percentage"
"Red"	41
"White"	20
"Blue"	14
"Yellow"	11
"Green"	7
"Black"	4

Y finalmente busquemos las banderas más similares de todos los países según el peso del color:

```
MATCH (item:`Flag`), (category:`Color`)
OPTIONAL MATCH (item:`Flag`)-[rel:`CONTAINS`]->(category:`Color`)
WITH {item:id(item), weights: collect(coalesce(rel.`weight`, gds.util.NaN()))} as userData
WITH collect(userData) as data
WITH $config AS config, data
WITH config { .*, data: data} as config

CALL gds.alpha.similarity.cosine.stream(config)

YIELD item1, item2, similarity
```



**Prueba Practica Sistemas Expertos Basados en casos.**

```
RETURN gds.util.asNode(item1) AS from, gds.util.asNode(item2) AS to, similarity  
ORDER BY similarity DESC  
LIMIT toInteger($limit)
```

Como salida témenos las banderas que tiene una coincidencia del 1.

"from"	"to"	"similarity"
{"name": "Belarus"}	{"name": "Germany"}	1.0
{"name": "Belarus"}	{"name": "Estonia"}	1.0
{"name": "Russia"}	{"name": "Ukraine"}	1.0
{"name": "Russia"}	{"name": "Estonia"}	1.0
{"name": "Russia"}	{"name": "Sweden"}	1.0
{"name": "Ukraine"}	{"name": "Russia"}	1.0
{"name": "Ukraine"}	{"name": "Norway"}	1.0
{"name": "Ukraine"}	{"name": "Finland"}	1.0
{"name": "Finland"}	{"name": "Belarus"}	1.0
{"name": "Finland"}	{"name": "Sweden"}	1.0
{"name": "Finland"}	{"name": "Ukraine"}	1.0
{"name": "Sweden"}	{"name": "Russia"}	1.0
{"name": "Sweden"}	{"name": "Norway"}	1.0
{"name": "Sweden"}	{"name": "Finland"}	1.0
{"name": "Belarus"}	{"name": "Finland"}	1.0

Usando la herramienta de GDS



## Prueba Practica Sistemas Expertos Basados en casos.

1. Configure 2. Results 3. Code

Table

Flag

From Labels	From Properties	To Labels	To Properties	Similarity
Flag	Belarus	Flag	Germany	1
Flag	Belarus	Flag	Estonia	1
Flag	Russia	Flag	Ukrain	1
Flag	Russia	Flag	Estonia	1
Flag	Russia	Flag	Sweden	1
Flag	Ukrain	Flag	Russia	1
Flag	Ukrain	Flag	Norway	1
Flag	Ukrain	Flag	Finland	1
Flag	Finland	Flag	Belarus	1

Help us improve NEuler

