

# **Aplicaciones de Datos en Redes Complejas**

Poma Ludeña, Rodrigo Renato

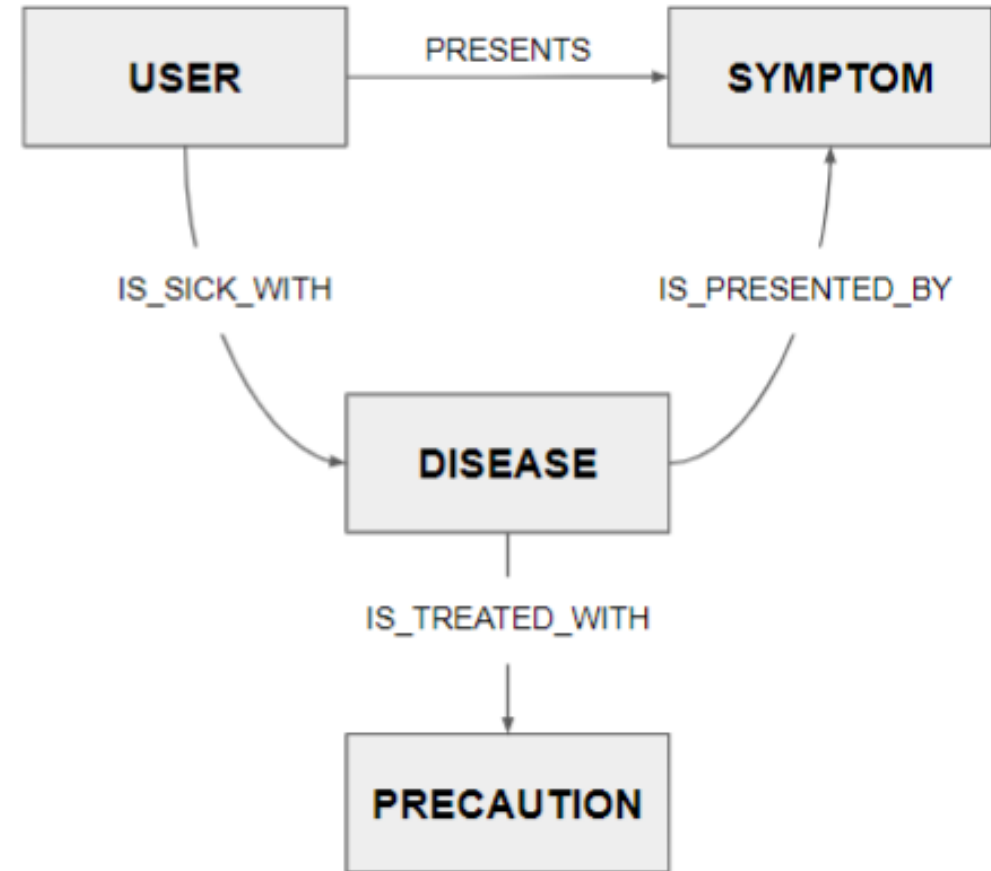
<https://github.com/RodrigoRenatoPomaLudena>

# **DATASET:** Síntomas y Enfermedades de Usuarios de una Aplicación Médica

**USER CASE:** Diagnóstico de Enfermedades en Base a Síntomas

# Descripción de la Información

Conjunto de datos relacionado a la atención médica mediante plataformas digitales. Tiene información sobre los usuarios que han hecho uso de la aplicación, sus síntomas, las enfermedades diagnosticadas y las precauciones a tomar. De este conjunto de datos, se puede construir un modelo predictivo, que utilice los síntomas descritos por el usuario y pueda indicar que enfermedad presenta.



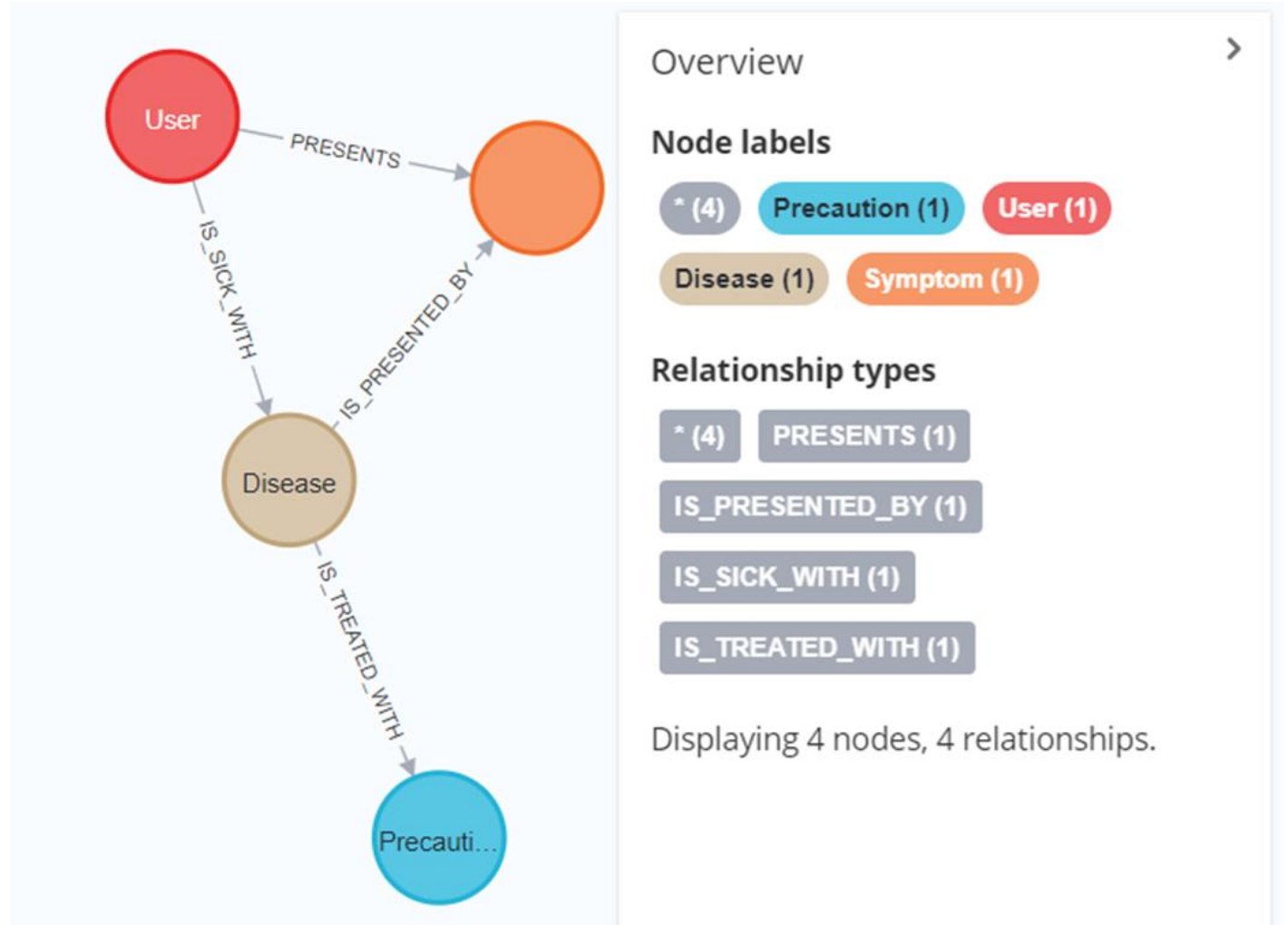
# Metadatos de la Base de Datos de Grafos

## Nodos:

- Disease: 41
- Precaution: 96
- Symptom: 132
- User: 4920

## Relaciones:

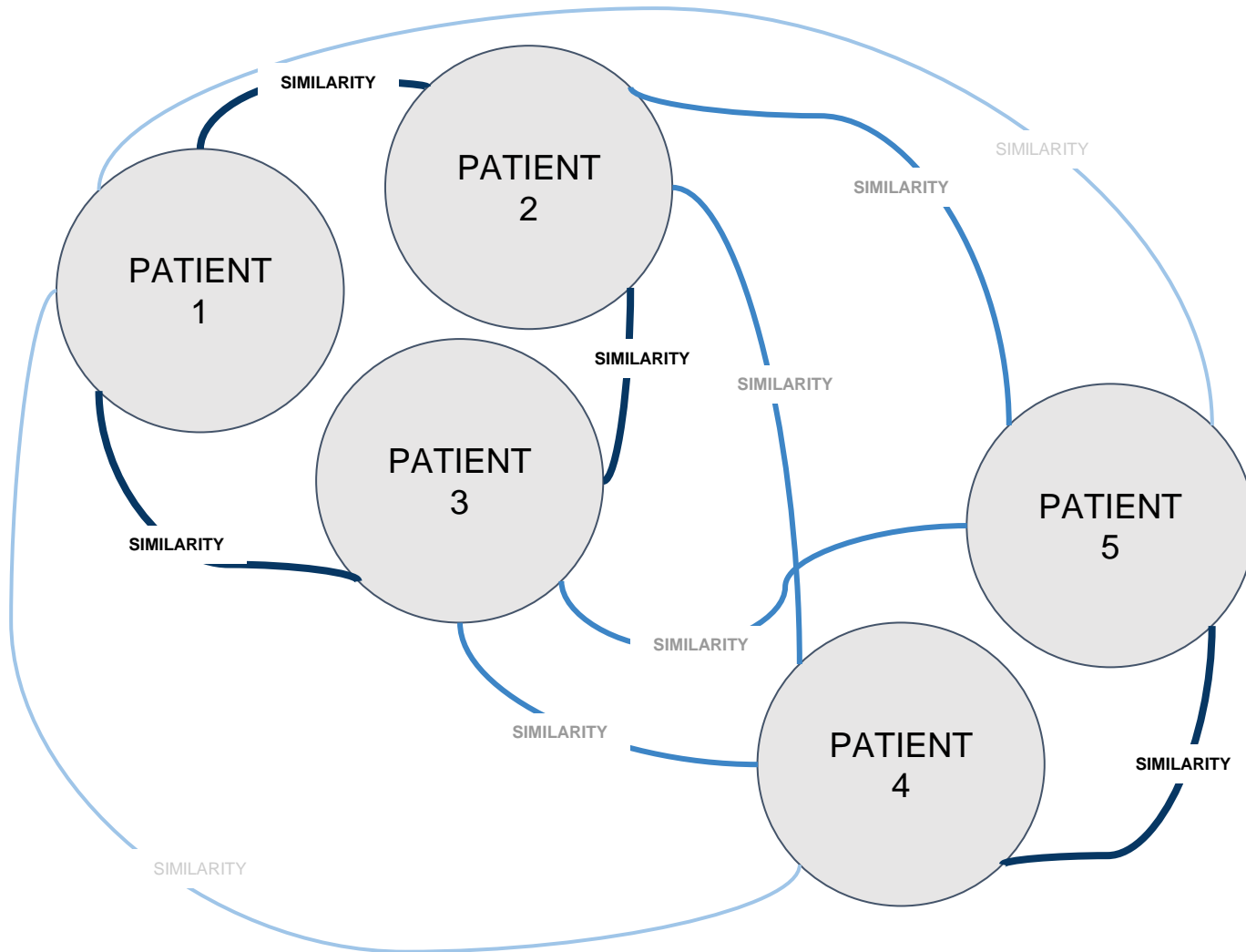
- PRESENTS: 36276
- IS\_SICK\_WITH: 4553
- IS\_TREATED\_WITH: 150
- IS\_PRESENTED\_BY: 298



# Propuesta: K-Nearest Neighbors

**DATASET:** Síntomas y Enfermedades de Usuarios de una Aplicación Médica

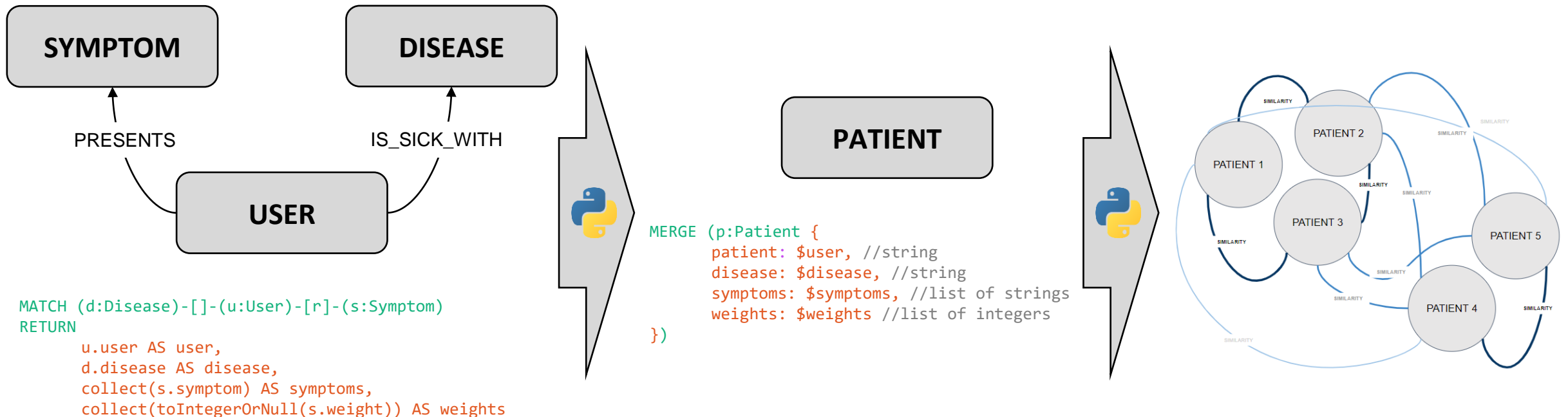
# K-Nearest Neighbors



El algoritmo K-Nearest Neighbors (KNN) puede ser aplicado en pacientes con listas de síntomas para determinar la similitud entre ellos. Para calcular la similitud entre los nodos, se puede utilizar una métrica de distancia especializada, como la distancia de Jaccard o la similitud del coseno. El resultado de la similitud también se puede expresar en forma de puntaje, donde un puntaje más alto indica una mayor similitud entre los pacientes.

# Proceso de elaboración del modelo

Para aplicar el modelo se construye un nodo PATIENT, el cual contiene información de los nodos SYMPTOM, USER y DISEASE. Con el nodo PATIENT se podrá aplicar el algoritmo KNN utilizando el paquete GDS de NEO4J.



# Comprobación del modelo

```
1 widget_seleccion = widgets.SelectMultiple(  
2     options=options_list_top_10,  
3     value=[options_list_top_10[0]],  
4     description='Selecciona opciones:'  
5 )  
6  
7 widget_seleccion.layout.height = '300px'  
8 display(widget_seleccion)  
9
```

✓ 0.0s

Selecciona ...

fatigue  
vomiting  
high fever  
loss of appetite  
nausea  
headache  
abdominal pain  
yellowish skin  
yellowing of eyes  
chills  
skin rash  
malaise  
chest pain  
joint pain  
itching  
sweating  
dark urine  
cough

```
1 symptoms = [x for x in widget_seleccion.value]  
2 print("Evaluating your symptoms: ", symptoms)  
3 result = diagnose(driver=driver, symptoms=symptoms)  
4 result
```

✓ 0.6s

Evaluating your symptoms: ['fatigue', 'loss of appetite', 'headache']

```
[{'disease': 'hepatitis b', 'similarity': 1.0},  
{ 'disease': 'migraine', 'similarity': 1.0},  
{ 'disease': 'allergy', 'similarity': 1.0},  
{ 'disease': 'typhoid', 'similarity': 1.0},  
{ 'disease': 'chronic cholestasis', 'similarity': 1.0},  
{ 'disease': 'alcoholic hepatitis', 'similarity': 1.0},  
{ 'disease': '(vertigo) paroxysmal positional vertigo', 'similarity': 1.0},  
{ 'disease': 'heart attack', 'similarity': 1.0},  
{ 'disease': 'gastroenteritis', 'similarity': 1.0},  
{ 'disease': 'paralysis (brain hemorrhage)', 'similarity': 1.0},  
{ 'disease': 'common cold', 'similarity': 1.0},  
{ 'disease': 'hypoglycemia', 'similarity': 1.0},  
{ 'disease': 'gerd', 'similarity': 1.0},  
{ 'disease': 'pneumonia', 'similarity': 1.0},  
{ 'disease': 'chicken pox', 'similarity': 1.0},  
{ 'disease': 'peptic ulcer disease', 'similarity': 1.0},  
{ 'disease': 'hyperthyroidism', 'similarity': 1.0},  
{ 'disease': 'hepatitis c', 'similarity': 1.0},  
{ 'disease': 'varicose veins', 'similarity': 1.0},  
{ 'disease': 'bronchial asthma', 'similarity': 1.0}]
```