

**Universidad del Valle de Guatemala**

**Proyecto: UniMatch - Recomendaciones Inteligentes para tu Vida  
Universitaria**

Curso: Algoriitmos y Estructura de Datos

Proyecto 2: Sistema de Recomendaciones Usando Grafos

Fase 1 Entrega Final - Diseño del Algoritmo de Recomendación

Integrantes:

Diego Guevara, Julio Orellana, JuanJose Rivas

## 1. Área del Proyecto y Algoritmos Existentes ante problemas similares.

Sistema de recomendación personalizado para actividades extracurriculares en estudiantes universitarios.

Algoritmos Existentes para Este tipo de problema:

- **Personalized PageRank (PPR)**

Calcula la importancia de los nodos en función de su cercanía a un nodo específico (por ejemplo, un usuario). Es útil para obtener recomendaciones personalizadas explorando conexiones como intereses y participaciones previas.

- **Node Similarity (Similitud de nodos)**

Mide qué tan similares son dos nodos en función de sus conexiones compartidas. Se usa para recomendar actividades realizadas por usuarios con patrones similares (filtrado colaborativo).

- **K-Shortest Paths**

Encuentra los caminos más cortos entre usuarios y actividades, considerando relaciones intermedias. Ayuda a descubrir conexiones no obvias entre intereses y posibles recomendaciones.

- **Label Propagation**

Identifica comunidades dentro del grafo propagando etiquetas entre nodos conectados. Útil para agrupar usuarios o actividades similares y ofrecer recomendaciones dentro del mismo grupo.

- **Graph Embedding (Node2Vec, DeepWalk, GraphSAGE)**

Convierte los nodos del grafo en vectores numéricos que conservan la estructura del grafo. Estos vectores pueden usarse para entrenar modelos de aprendizaje automático que predicen las actividades más relevantes para cada usuario.

- **Filtrado basado en contenido (Content-Based Filtering)**

Utiliza información sobre las categorías de las actividades y los intereses del usuario para sugerir actividades similares a las que ya ha mostrado interés.

## 2. Justificación del Uso de Grafos

El uso de grafos permite modelar relaciones complejas entre usuarios, intereses y actividades, lo cual sería difícil con bases de datos relacionales. Neo4j permite consultas como 'recomiéndame actividades que disfrutaron personas con intereses similares', lo cual sería costoso computacionalmente con SQL tradicional.

## 3. Algoritmo de Recomendación Detallado

El sistema usará un enfoque híbrido de filtrado colaborativo y basado en contenido sobre grafos:

- Se analizarán intereses y participación previas.
- Se calculará la similitud entre usuarios por intereses compartidos.
- Se priorizarán actividades con alta participación y cercanas geográficamente.
- Se utilizará un score ponderado por afinidad, ubicación y popularidad.

#### 4. Pseudocódigo del Algoritmo

func recomendarActividades(usuario):

    intereses = obtenerIntereses(usuario)

    similares = usuariosConInteresesSimilares(intereses)

    actividades = actividadesDeUsuarios(similares)

    actividadesFiltradas = filtrarPorUbicacionYHorario(actividades, usuario)

    return ordenarPorScore(actividadesFiltradas)

#### 5. Design Thinking

- Empatía: Se realizó una encuesta a 20 estudiantes en la cual se obtuvo como resultado que el 70% no participa en actividades recreativas por falta de información.
- Definición: El problema es la desconexión entre actividades disponibles e intereses personales.
- Ideación: Se propusieron filtros de intereses, horarios, geolocalización, popularidad y recomendaciones de amigos.
- Prototipo: Se diseñará un prototipo interactivo en la Fase 2.
- Testing: Se planea realizar pruebas con usuarios reales tras la implementación inicial.

#### 6. Modelo de Base de Datos en Neo4j

- Nodos: Usuario, Actividad, Categoría

- Relaciones:

(Usuario)-[:INTERESADO\_EN]->(Categoría)

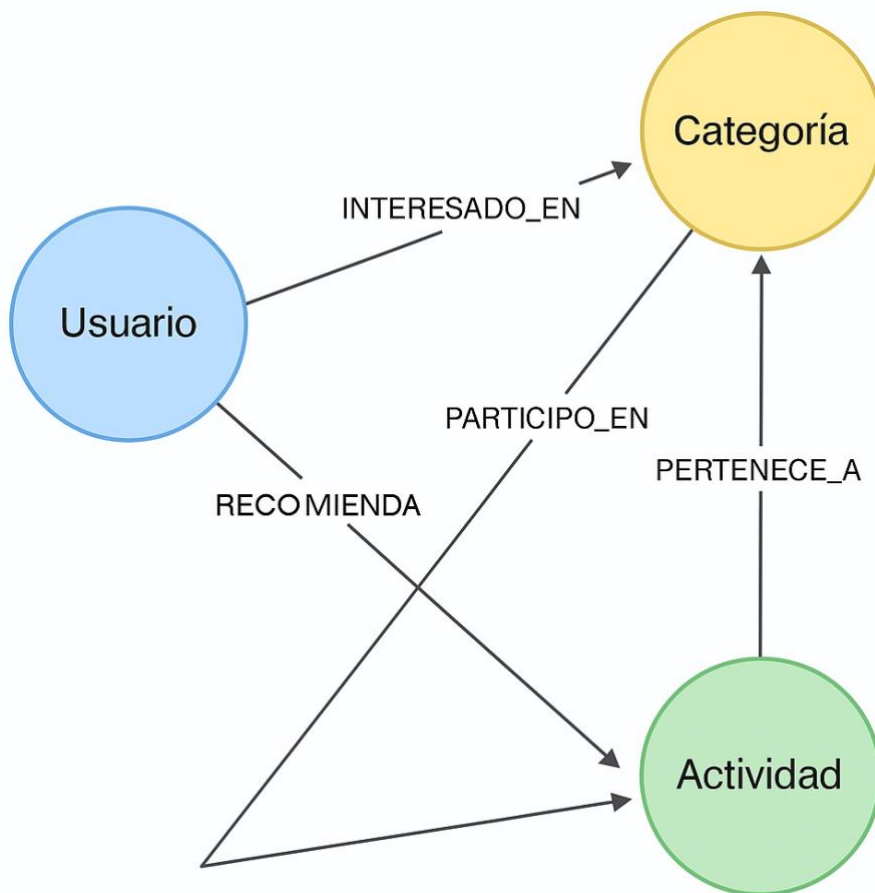
(Actividad)-[:PERTENECE\_A]->(Categoría)

(Usuario)-[:PARTICIPO\_EN]->(Actividad)

(Usuario)-[:RECOMIENDA]->(Actividad)

## 7. Explicacion de la Base de Datos Inicial

Un nodo Usuario se conecta a Categoría mediante INTERESADO\_EN, y a Actividad por PARTICIPO\_EN. Las Actividades se agrupan por Categoría, lo que permite recomendaciones según comunidad e intereses compartidos.



## 8. Plan de Testing

- Se crearán perfiles de prueba.
- Se simularán consultas de recomendación para verificar precisión y eficiencia.
- Se recolectará retroalimentación de estudiantes para mejorar el modelo.

## 9. Repositorio GitHub

Repositorio: <https://github.com/Julio-orellana/Proyecto-2-DSA>

## Referencias

Jannach, D., Adomavicius, G., & Tuzhilin, A. (2011). Recommender systems: An introduction. Cambridge University Press.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2021). A survey of recommender systems based on graph neural networks. ACM Computing Surveys (CSUR), 54(5), 1–38.

<https://doi.org/10.1145/3458445>

Robinson, I., Webber, J., & Eifrem, E. (2015). Graph databases (2nd ed.). O'Reilly Media.

<https://neo4j.com/graph-databases-book/>

Neo4j. (n.d.). Neo4j Java Developer Manual.

<https://neo4j.com/docs/java-reference/current/>

Neo4j. (n.d.). Neo4j Documentation.

<https://neo4j.com/docs/>

Neo4j Examples. (n.d.). GitHub – neo4j-examples.

<https://github.com/neo4j-examples>