



Aplicación con una B.D. NoSQL

Julio Domínguez Arjona
Francisco Javier Ortiz Valverde



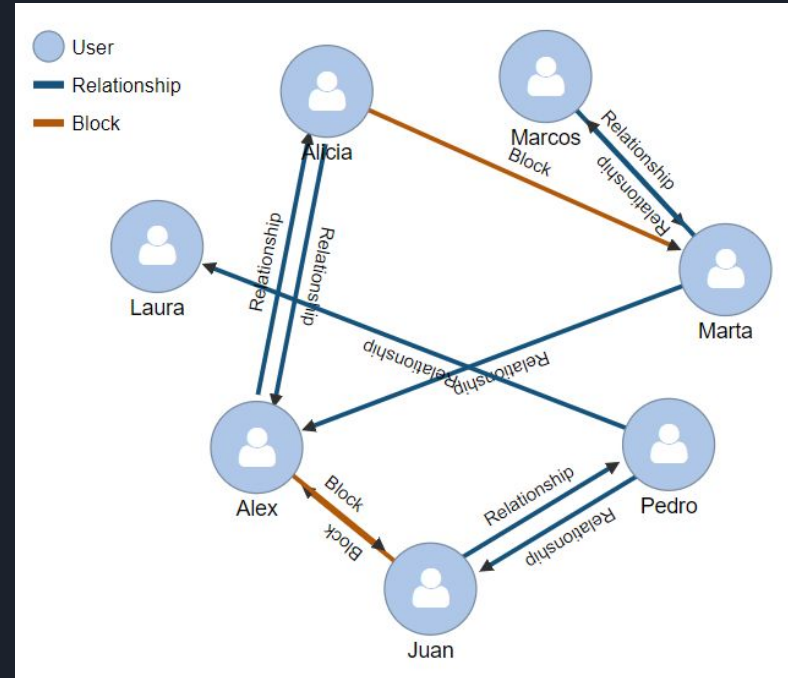
Índice

1. Concepto
2. Estructura de la aplicación
3. Relaciones entre usuarios
4. Operaciones del cliente
5. El algoritmo

Concepto:

Un gestor de amistades basado en grafos

- Gestión de amistades al estilo *red social*
- Emplear **nodos** para usuarios
- Establecer las relaciones mediante **aristas**



Estructura de la aplicación

Aplicación cliente *Java*

Aplicación de consola con la interfaz de gestión diseñada para el usuario final.



Base de datos de grafos *OrientDB*

Nodo *User*

Representa a un usuario y almacena:

- @rid
- name
- email
- password

Arista *Relationship*

Define una relación entre dos usuarios.

Arista *Block*

Representa el deseo de un usuario de evitar a otro.

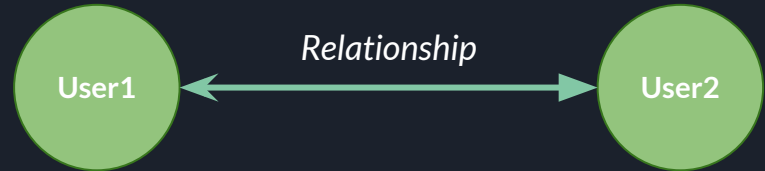


Relaciones entre usuarios

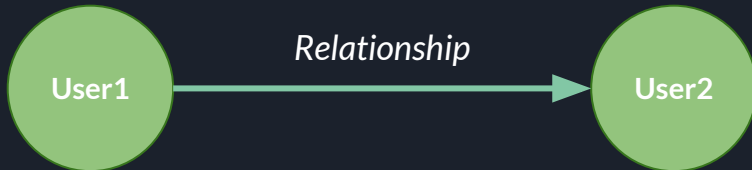
Usuarios **sin relación** entre ellos



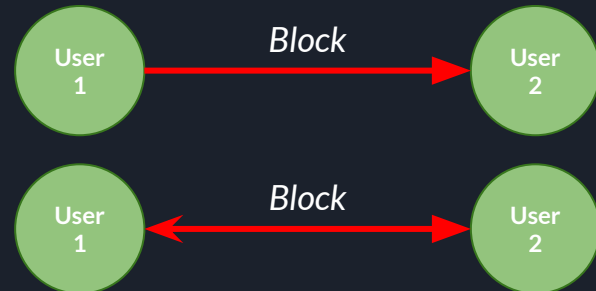
Amistad entre dos usuarios



Petición de amistad
De User1 para User2



Bloqueos entre usuarios





Operaciones del cliente

- Registro
- Cerrar sesión

- Ver lista de amigos
- Ver solicitudes de amistad entrantes
- Enviar solicitudes de amistad

- Ver usuarios bloqueados
- Bloquear usuarios
- Desbloquear usuarios
- Buscar a otros usuarios
- Eliminar cuenta

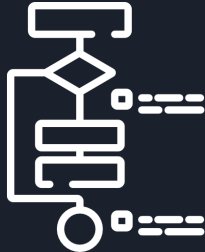
El algoritmo: Graph Convolutional Networks (GCN)

Objetivo: Sugerecias de amistad

Algoritmo clásico

“El amigo de mi amigo es mi posible amigo”

- Se ofrece como sugerecia la lista de amigos de amigos, ordenados aleatoriamente
- Bajo nivel de variabilidad en cuanto a sugerecias



Machine Learning

“¿Pertenece a algún grupo en especial?”

- Permite la división por proximidad
- Mejora las sugerecias y variabilidad



¿Por qué?: Graph Convolutional Networks (GCN)

Problema:

- Datos de la ubicación del usuario irrelevantes
- Aristas no ponderadas en distancia

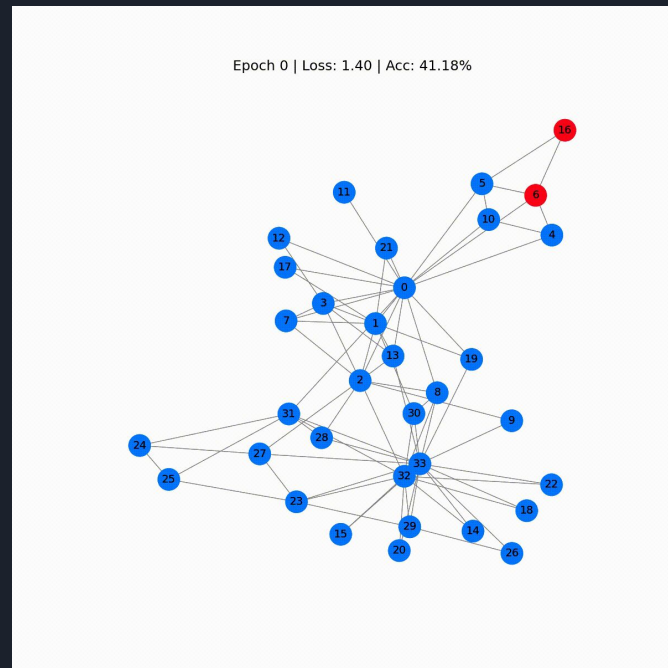


Necesidades:

- Principal:
 - Trabajar con grafos no ponderados
 - No considerar la posición de los nodos
 - Clasificación de nodos en sub-grupos
- Secundario:
 - Eficiencia

El algoritmo: Graph Convolutional Networks (GCN)

- Similar a K-Means
- Inicia el algoritmo con **n centroides** y se amplía si es necesario
- Genera un grafo subdividido por grupos *altamente cohesionados*.



Conclusiones



Elección de BD: *OrientDB* mostró ser eficaz para organizar las relaciones en forma de grafos.



Escalabilidad: El sistema podría ser escalado a un escenario real más complejo.



Facilidad de uso y flexibilidad: Aplicación intuitiva y sencilla para la gestión de amistades.



Clara visualización: La representación mediante grafos facilitó la comprensión de las interacciones sociales entre usuarios.



Modelo para fuente de información: El modelo de datos creado demostró ser perfecto para el uso de algoritmos de minería de datos sobre la información almacenada.

Fin

¿Dudas?