

Proyecto 2: Investigación

Introducción

Para el proyecto número 2 de la clase de algoritmos y estructuras de datos se nos pide diseñar un sistema que de sugerencias personalizadas a los usuarios utilizando grafos. Como es requisito para el proyecto se explorará el uso de grafos como modelo para representar relaciones complejas entre entidades. Esta investigación se centrará en analizar algoritmos avanzados de grafos y en identificar bases de datos eficientes para almacenar y consultar estas estructuras, todo ello con el objetivo de diseñar e implementar un sistema de recomendaciones eficiente y funcional para el proyecto en cuestión.

Bases de datos

El componente de bases de datos juega un papel fundamental para el desarrollo del proyecto de sistemas de recomendación basados en grafos. Una base de datos eficiente y bien diseñada es crucial para almacenar y gestionar la información de manera óptima, permitiendo consultas rápidas y precisas en las estructuras de grafo que representan las relaciones entre entidades. Algunas de las bases de datos orientadas a grafos más utilizadas son:

- Neo4j: Es una de las bases de datos orientadas a grafos más populares y maduras. Utiliza un modelo de grafo de propiedad de nodo-arista y está diseñada para ser eficiente en la consulta y manipulación de relaciones entre datos. Es ampliamente utilizada en aplicaciones que requieren análisis de redes, recomendaciones y gestión de datos altamente interconectados.
- Amazon Neptune: Es un servicio de base de datos en la nube que admite modelos de grafo. Está diseñado para ser altamente escalable y duradero, lo que lo hace adecuado para aplicaciones que requieren almacenar grandes cantidades de datos relacionales y ejecutar consultas complejas sobre ellos.

- Dgraph: Es una base de datos de grafo distribuida y de código abierto que se enfoca en ser rápida y eficiente en la consulta de datos altamente conectados. Utiliza un lenguaje de consulta propio llamado GraphQL+ y es adecuada para aplicaciones que necesitan modelar y consultar relaciones complejas entre entidades.

Algoritmos para recorrer grafos

Debido a la naturaleza del proyecto es evidente que el uso de grafos es esencial para cumplir el diseño del programa. Debido a esto es importante tener en claro que algoritmos utilizar para recorrer los grafos de manera que el programa sea capaz de proveer al usuario con resultados relevantes para su búsqueda, así como también cortos tiempos de carga. Debido a esto se investigó información clave de los siguientes grafos para tener en claro cual de ellos es el que mejor se adapta a la visión del equipo para este proyecto.

- Búsqueda en profundidad (DFS - Depth-First Search): Este algoritmo comienza en un nodo raíz y explora cada uno de sus nodos vecinos antes de retroceder. Es recursivo por naturaleza y utiliza una pila (o la recursión misma) para llevar un registro de los nodos visitados y pendientes de exploración. DFS es útil para encontrar componentes conectados, detectar ciclos y recorrer grafos no ponderados.
- Búsqueda en amplitud (BFS - Breadth-First Search): A diferencia de DFS, BFS explora todos los nodos vecinos de un nodo antes de pasar al siguiente nivel de nodos. Utiliza una cola para mantener un registro de los nodos a visitar y garantiza que los nodos más cercanos al nodo inicial se visiten antes que los más distantes. BFS es útil para encontrar el camino más corto en grafos no ponderados y para determinar la accesibilidad entre nodos.
- Algoritmo de Dijkstra: Este algoritmo encuentra el camino más corto desde un nodo inicial a todos los demás nodos en un grafo ponderado y dirigido (o no dirigido) con pesos no negativos. Utiliza una cola de prioridad para seleccionar el siguiente nodo a visitar en función de la distancia más corta conocida hasta el momento. Dijkstra es

muy útil en problemas de rutas, como la planificación de caminos en redes de transporte.

Conclusión

En conclusión, la investigación sobre bases de datos y algoritmos de grafos revela la importancia de entender cómo representar, almacenar y manipular datos interconectados de manera eficiente y efectiva.