Base de Datos I

Ibarra Padilla Sebastian
Leyva Silva Andrés Jovany
Martínez Ruiz Josué Ignacio

Historia

¿Qué es Neo4j?
Es un Sistema de Gestión de Bases de Datos de Grafos (SMBDG) nativo, diseñado para almacenar, gestionar y consultar datos altamente conectados de manera eficiente.

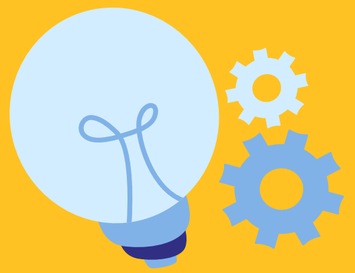
Los fundadores **Emil Eifrem**, Johan Svensson y Peter Neubauer, encontraron problemas de rendimiento con los RDBMS y comenzaron a construir el primer prototipo de Neo4j en el año 2000.

La Empresa Neo4j se estableció el 23 de enero de 2007, surgió del reconocimiento de las limitaciones en los sistemas de bases de datos tradicionales cuando se trata de relaciones complejas.


Emil


Johan


Peter



Características distintivas

- **Ecosistema y Herramientas**: Cuenta con un ecosistema robusto que incluye herramientas de visualización y conectores para Big Data y ciencia de datos.

- **Rendimiento de Recorrido de Grafo**: Mantiene un rendimiento **constante** y lineal para consultas profundas gracias a su arquitectura de adyacencia sin índices.

- **Casos de Uso Estratégicos**: Es la solución óptima para problemas donde el valor de negocio reside en las **relaciones**, como la detección de fraude, los motores de recomendación y el análisis de redes.

- **Fiabilidad Empresarial:** Admite transacciones ACID y permite configuraciones de múltiples nodos, garantizando consistencia, alta disponibilidad y escalabilidad en las operaciones de lectura.

Funciones

Neo4j implementa funciones avanzadas de bases de datos optimizadas para **grafos**:

01 **Almacenamiento y Recuperación de Datos**

- Almacena datos como nodos y relaciones en archivos optimizados.
- Recorrer relaciones es **rápido y constante**, sin importar el tamaño de los datos.

02 **Soporte de Transacciones ACID**

- Garantiza que las operaciones sean atómicas y consistentes.
- Usa un registro de escritura anticipada (Write-Ahead Log) para recuperación ante fallos.

03 **Control de Concurrencia**

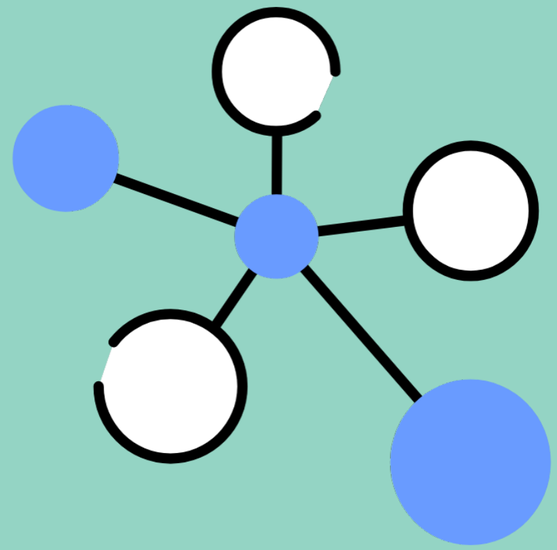
- Usa bloqueos optimistas: las transacciones se verifican al final, evitando cuellos de botella.
- Ideal para entornos con muchas lecturas y menos escrituras.

04 **Autorización y Seguridad**

- Modelo de roles (RBAC): permisos granulares sobre nodos y relaciones.
- Ejemplo: Restringir acceso a datos sensibles por etiquetas (ej: Payment).

05 **Integridad de Datos**

- Elimina relaciones huérfanas automáticamente al borrar nodos.
- Soporta restricciones de unicidad (ej: email único para usuarios).



Descripción de la implementación

01. Almacenamiento y Recuperación de Datos

- **Mecanismos/tecnologías:** Usa índices y archivos organizados para guardar y encontrar datos rápido (árboles B+, grafos, hash).
- **Impacto**: Las búsquedas son rápidas aunque la base de datos sea muy grande.

02. Soporte de Transacciones ACID

- **Mecanismos/tecnologías:** Guarda un registro de cada cambio (WAL) y asegura que las operaciones se completen bien o no se hagan.
- **Impacto**: Los datos nunca quedan a medias o dañados, siempre son confiables.

03. Control de Concurrencia

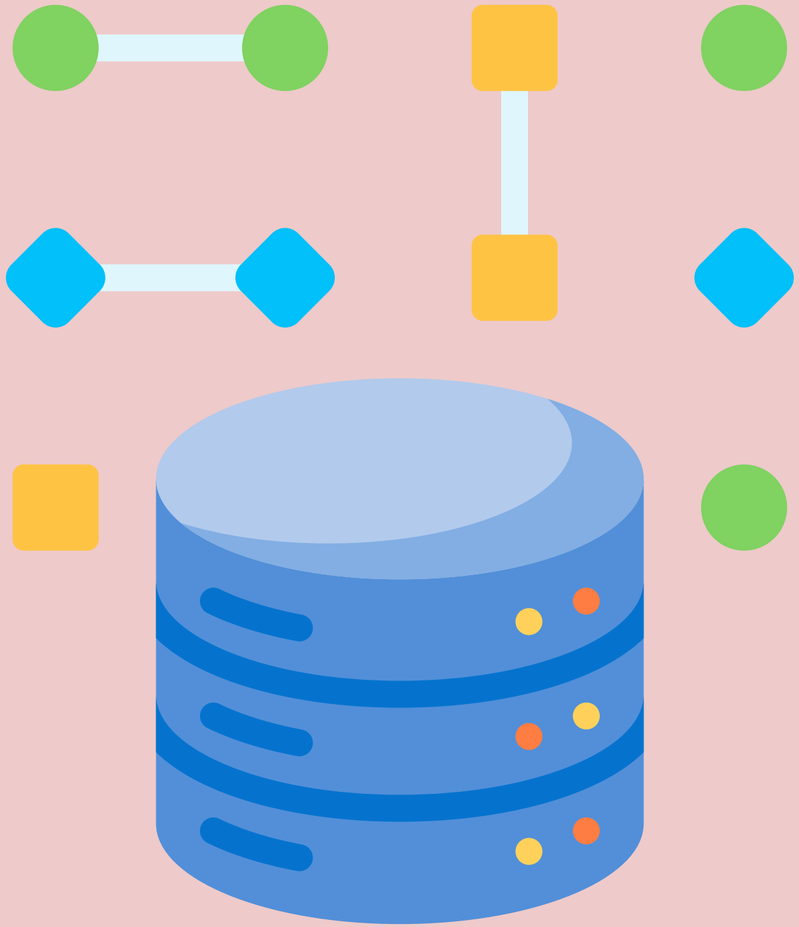
- **Mecanismos/tecnologías:** Usa métodos como bloqueos o versiones de datos para que varias personas trabajen al mismo tiempo sin chocar.
- **Impacto**: Consultas más **rápidas y constantes** → mejor experiencia de usuario y alto rendimiento.

04. Autorización y Seguridad

- **Mecanismos/tecnologías:** Aplica roles, permisos y reglas para controlar quién accede a qué datos (cifrado de datos, autenticación).
- **Impacto**: Solo las personas autorizadas ven la información, aumentando la seguridad.

05. Integridad de Datos

- **Mecanismos/tecnologías:** Aplica reglas como “no duplicar correos” y elimina relaciones inválidas automáticamente.
- **Impacto**: Datos coherentes y sin duplicados, información confiable y menor riesgo de errores.



Manejo de DDL / DML

¿Cómo se utilizan?
En Neo4j, el lenguaje **Cypher** integra tanto las funciones de DDL (Definición de Datos) como de DML (Manipulación de Datos), permitiendo gestionar tanto la estructura como los datos del grafo de manera flexible y eficiente.

Neo4j es esquema-opcional, pero el **DDL** define índices y restricciones para mejorar el rendimiento y la integridad:

Crear Índice (Para búsquedas rápidas):
CREATE INDEX FOR (p:Product) ON (p.name);
-Acelera consultas que filtran por name en nodos con etiqueta Product.

Crear Restricción de Unicidad (Evita duplicados):
CREATE CONSTRAINT FOR (u:User) REQUIRE u.email IS UNIQUE;
- Garantiza que no existan dos usuarios con el mismo email.

Por otro lado, el **DML** se usa para crear, actualizar, consultar y eliminar nodos y relaciones:

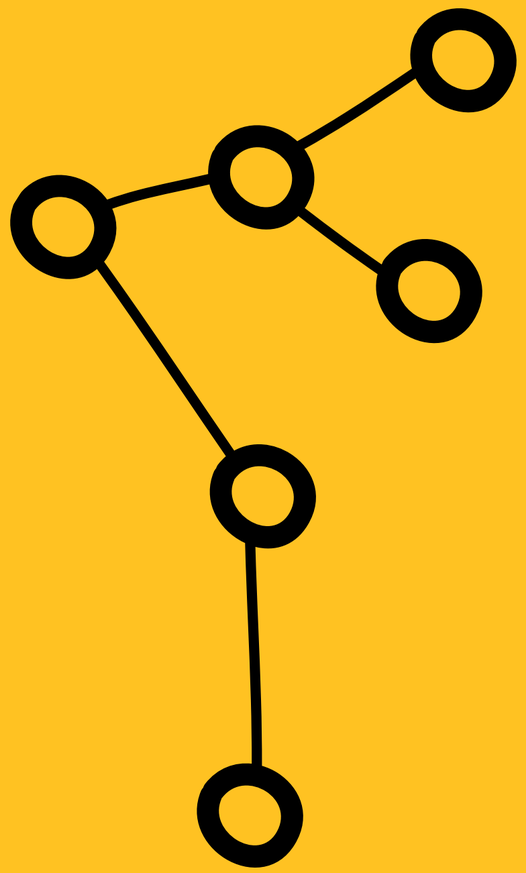
Crear nodos:
CREATE (p:Product {name: 'Laptop', price: 1200});

Crear relaciones:
MATCH (u:User {name: 'Juan'}), (p:Product {name: 'Laptop'})
CREATE (u)-[:BOUGHT]->(p);

Actualizar propiedades:
MATCH (p:Product {name: 'Laptop'})
SET p.price = 1100;

Eliminar nodos o relaciones:
MATCH (p:Product {name: 'Laptop'})
DELETE p;

Consultar datos:
Copiar código
MATCH (u:User)-[:COMPRÓ]->(p:Product)
RETURN u.name, p.name;



Ejemplos de funcionamiento

En su lenguaje nativo **CYPHER** por ejemplo, para obtener películas dirigidas por un director:

```text
MATCH (director:Person)-[:DIRECTED]->(movie:Movie)
WHERE director.name = 'Steven Spielberg'
RETURN movie.title
```

Ejemplo de código en Python:

```python
from neo4j import GraphDatabase

driver = GraphDatabase.driver("bolt://localhost:7687", auth=
("neo4j", "password"))

def create_person(tx, name):
    tx.run("CREATE (a:Person {name: $name})", name=name)

with driver.session() as session:
    session.write_transaction(create_person, "Alice")
```

Ejemplo de BD

```
$ match r=(p:Person)-[a:ACTED_IN]->(m:Movie) return r limit 5
```

return r: Devuelve ese patrón encontrado.

limit 5: Limita los resultados a 5 patrones para evitar demasiada información.

Explicación de la imagen

Aparece un nodo central de color amarillo (película "The Matrix") y cinco nodos verdes (personas) conectados a la película mediante relaciones con etiqueta "ACTED_IN".

Los nodos verdes representan actores que participaron en la película "The Matrix".
Cada línea representa la relación de actuación entre la persona y ese film

La linea “match r=(p:Person)-[a:ACTED_IN]->(m:Movie)” Busca **patrones** en el grafo donde un nodo con etiqueta "Person" (persona) está **conectado** mediante la relación "ACTED_IN" (actuó en) a un nodo con etiqueta "Movie" (película). Esta relación **indica que la persona actuó en esa película**.

Compatibilidad del Sistema Operativo

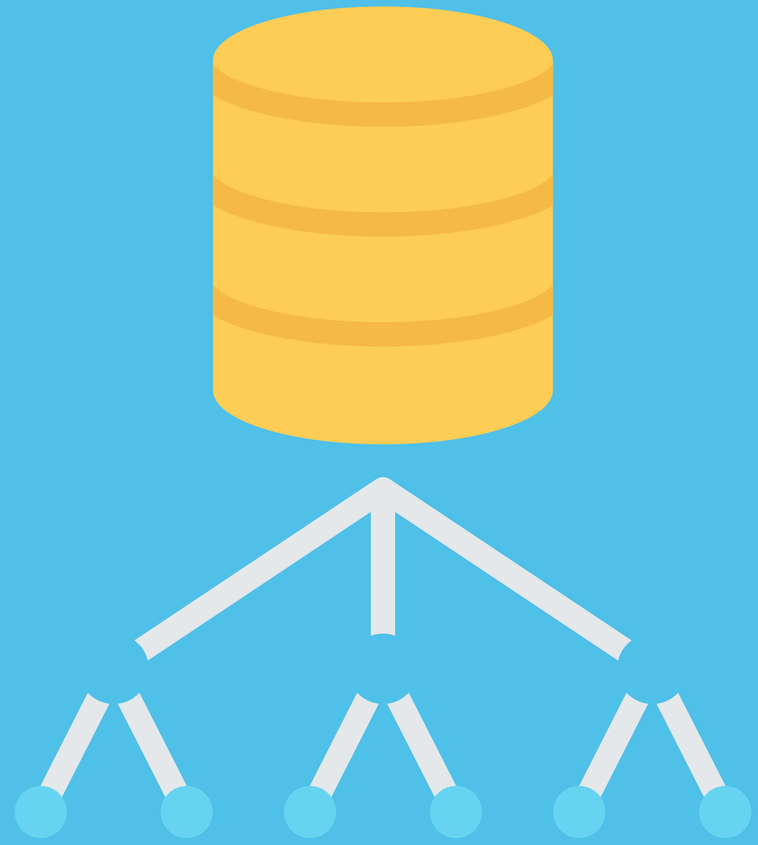
Neo4j es altamente compatible gracias a que está desarrollado en **Java**, lo que le permite ejecutarse en cualquier sistema operativo que tenga una Máquina Virtual Java (JVM) instalada.

Plataformas compatibles

Neo4j es compatible con sistemas con arquitecturas **x86_64** y **ARM** en plataformas físicas, virtuales o en contenedores.

Requisitos de acceso

Máquina virtual Java (**JVM**) preinstalada, acceso a la base de datos de tu elección (ya sea localmente o a través de **Neo4j AuraDB** en la nube) y familiaridad con **Cypher**.



Compatibilidad del Sistema Operativo

Sistema Operativo	Uso Principal	Requisitos/Limitaciones
Windows (10/11, Server 2016+)	Usado en escritorio algunos servidores, ideal para desarrollo	Necesita Java instalado; recomendable tener buena RAM para bases grandes
Linux (Ubuntu, Debian, RedHat, CentOS, SUSE)	Servidores y producción **(el más común)**	Requiere Java; funciona mejor en servidores con suficiente memoria y espacio en disco
macOS	Usado en escritorio y algunos servidores, ideal para desarrollo	Menos usado en producción; requiere Java y buena memoria
Docker (multiplataforma)	Instalación flexible en cualquier sistema	Neo4j corre dentro de un contenedor; solo necesita que el SO soporte Docker

Requisitos de Hardware

Para uso personal y desarrollo de software:

Sistema Operativo	Uso Principal
UPC	Intel x86-x64 Core i3 mínimo, se recomienda Core i7. AMD x86-x64, Mac ARM.
Memoria	Mínimo 2 GB, se recomienda 16 GB o más.
Almacena miento	SATA mínimo de 10 GB, SSD con SATA Express o NVMe recomendado.

Para entornos de nube:

Sistema Operativo	Uso Principal
UPC	Mínimo 2vCPU, se recomiendan 16+.
Memoria	Mínimo 2 GB. Los requisitos reales dependen de las cargas de trabajo.
Almacena miento	Almacenamiento en bloque mínimo de 10 GB; se recomienda SSD NVMe conectado.

Requisitos de hardware

Para entornos locales basados en servidor:

Sistema Operativo	Uso Principal
UPC	Intel/AMD x86-x64. ARM64.
Memoria	Mínimo 8 GB. Los requisitos reales dependen de las cargas de trabajo.
Almacena miento	RAID/SAN o SSD con más de 5000 IOPS. Se recomienda SSD NVMe.

Aclaración

Todos los casos anteriores van en proporción con la carga de trabajo, son mas que nada sugerencias. El tamaño del **almacenamiento** depende del tamaño de las bases de datos.

En el caso de la memoria se recomienda usar instancias con **memoria** adecuada al tamaño del gráfico en uso.

Soporte de
Lenguaje de Programación

En Neo4j se utilizan distintos lenguajes de programación para interactuar con la base de datos y manipular sus grafos de manera efectiva, los principales son:

Java

- Es el lenguaje base en que está implementado Neo4j.
- Permite usar la API embebida para control directo y eficiente del motor de base de datos

Python

- Usado mediante bibliotecas como neo4j-driver o py2neo.
- Permite ejecutar consultas Cypher y manipular resultados fácilmente para aplicaciones basadas en Python.

Aplicaciones del mundo real

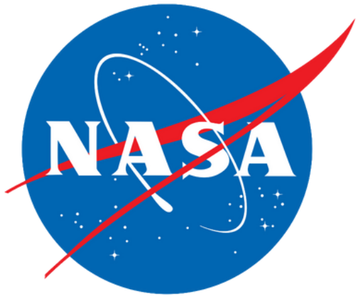
Walmart

- Usan Neo4j para análisis de datos relacionados y mejorar su sistema de **recomendaciones** y **detección de fraudes** en el comercio electrónico y logística.
- Beneficio: optimización en la gestión de productos y aumento en la seguridad de transacciones.

NETFLIX

- Utiliza Neo4j para **recomendaciones** personalizadas basadas en análisis de grafos de preferencias y comportamientos de usuarios.
- Beneficio: mejor retención y satisfacción del cliente por recomendaciones precisas y personalizadas.

Aplicaciones del mundo real



- Usa Neo4j para conectar datos de conocimientos, habilidades, tareas y tecnologías con roles y capacitación.
- **Beneficio**: optimiza la gestión del talento y mejora la asignación de personal en proyectos espaciales.

atpco

- Empresa de transporte aéreo que usa Neo4j para análisis de **rutas** y precios, facilitando ofertas personalizadas.
- **Beneficio**: innovación en productos y mejora en la experiencia del cliente.

Enlaces

Garavaglia, H. (2022, 24 enero). *Neo4j, una interesante alternativa para los datos.* Medium. https://medium.com/we-are-shifters/neo4j-una-interesante-alternativa-para-los-datos-f6e333cca0a6

NEO4J - Jortilles. (2019, 14 junio). Jortilles. https://blog.jortilles.com/neo4j/

Communications. (2018, 13 septiembre). Qué es Neo4j y para qué sirve una base de datos orientada a grafos. *BBVA NOTICIAS*. https://www.bbva.com/es/que-es-neo4j-y-para-que-sirve-una-base-de-datos-orientada-a-grafos/

https://www.intersystems.com/es/recursos/graph-database-vs-base-relacional-cual-se-adapta-mejor/

https://go.neo4j.com/rs/710-RRC-335/images/Neo4j-case-study-ATPCO-ES.pdf

Enlaces

Howard, R. (2025, 25 abril). *How NASA Finds Critical Data through a Knowledge Graph*. Graph Database & Analytics. https://neo4j.com/blog/knowledge-graph/nasa-critical-data-knowledge-graph/

https://www.enmilocalfunciona.io/conoces-neo4j-o-sabes-de-que-va/

https://xurxodeveloper.blogspot.com/2014/04/neo4j-instalacion-y-neo4j-browser.html

https://mx.gauthmath.com/solution/1826459015824402/Actividad-13-Instrucciones-Mediante-una-b-squeda-en-Internet-contesta-las-siguie

Gracias

Por su atención