INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

MARIEL SOARES NICIOLI

ESTUDO DE CASO UTILIZANDO NEO4J

CAMPOS DO JORDÃO

2024

RESUMO

Este estudo acadêmico se concentra no Neo4j como Sistema de Gerenciamento de Banco de Dados (SGBD) orientado a grafos, inserido na categoria NoSQL. O Neo4j proporciona uma abordagem inovadora ao permitir a modelagem de dados como grafos, onde entidades são representadas como nós e suas interações como arestas, ideal para a análise de redes complexas de relacionamentos. É também discutido as considerações iniciais sobre o estudo, detalhado aprofundamente sobre os modelos não relacionais e o Neo4j, tipo de modelo não relacional dentro do leque de tipos do NoSQL. Por fim, o exemplo prático, desenvolvido que é caracterizado como um sistemas de gerenciamento de inventário para sebos, no qual irá facilitar a organização meticulosa de produtos diversos como livros, CDs e vinis. Utilizando a linguagem de consulta Cypher, o Neo4j habilita operações sofisticadas que aprimoram o controle de estoque, oferecendo visões detalhadas sobre vendas, comportamento do consumidor e tendências de mercado. Essa capacidade não apenas otimiza a eficiência operacional, mas também enriquece a experiência do cliente ao possibilitar recomendações personalizadas e estratégias de marketing direcionadas com base em padrões de compra e preferências individuais. Assim, o Neo4j não só se destaca por sua flexibilidade e escalabilidade na gestão de dados dinâmicos, mas também por seu papel crucial na inovação e na melhoria contínua de processos em contextos comerciais e acadêmicos.

Palavras-Chave: NoSQL; modelos de dados NoSQL; Neo4j; sistema; gerenciamento de inventário; banco de dados orientado à grafos.

ABSTRACT

This academic study focuses on Neo4j as a graph-oriented Database Management System (DBMS) within the NoSQL category. Neo4j provides an innovative approach by allowing data modeling as graphs, where entities are represented as nodes and their interactions as edges, ideal for analyzing complex networks of relationships. It also discusses initial considerations about the study, delving deeply into non-relational models and Neo4j as a specific type within the spectrum of NoSQL types. Finally, it details a practical example developed as an inventory management system for used bookstores, enabling meticulous organization of various products such as books, CDs, and vinyl records. Using the Cypher query language, Neo4j enables sophisticated operations that enhance inventory control, offering detailed insights into sales, consumer behavior, and market trends. This capability not only optimizes operational efficiency but also enriches the customer experience by enabling personalized recommendations and targeted marketing strategies based on purchasing patterns and individual preferences. Thus, Neo4j stands out not only for its flexibility and scalability in managing dynamic data but also for its crucial role in innovation and continuous improvement of processes in both commercial and academic contexts.

Keywords: NoSQL; NoSQL data models; Neo4j; inventory management system; graph database.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

NoSQL *Not Only SQL*

JSON *JavaScript Object Notation*

BSON *Binary JSON*

XML *Extensible Markup Language*

SGBD Sistema de Gerenciamento de Banco de Dados

SUMÁRIO

1 INTRODUÇÃO

No cenário digital em constante evolução, a gestão da informação assume um papel crucial para o sucesso de empresas e organizações. Nesse contexto, os sistemas de gerenciamento de banco de dados se consolidam como ferramentas essenciais para o armazenamento, recuperação e análise de dados relevantes. Entre os modelos tradicionais, os bancos de dados relacionais, baseados no modelo relacional proposto por Edgar F. Codd, dominaram por décadas, por oferecerem uma estrutura organizada baseada em tabelas, onde os dados são armazenados em linhas e colunas e inter-relacionados por meio de chaves primárias e estrangeiras. No entanto, com o advento de novas necessidades e desafios tecnológicos, como o aumento exponencial de dados não estruturados e semiestruturados, a demanda por soluções mais flexíveis levou ao desenvolvimento e ascensão dos bancos de dados não relacionais, também conhecidos como NoSQL (Not Only SQL).

Os bancos de dados não relacionais diferem fundamentalmente dos relacionais em sua abordagem ao armazenamento e gerenciamento de dados. Em vez de utilizar uma estrutura tabular rígida, os bancos de dados NoSQL oferecem modelos de dados variados que se adaptam melhor a diferentes tipos de dados e cenários de uso. Eles são projetados para fornecer escalabilidade horizontal, alta disponibilidade e desempenho otimizado para grandes volumes de dados, características essenciais para atender às necessidades de aplicações modernas, como redes sociais, e-commerce, análise de big data, entre outras.

Existem diversos modelos de dados dentro da categoria de bancos de dados não relacionais, cada um com suas particularidades e casos de uso específicos:

- Document-oriented/Modelo em Documentos: Armazenam dados no formato de documentos, geralmente utilizando JSON, BSON ou XML. Exemplos incluem MongoDB e CouchDB.
- Key-Value/Modelo Chave-valor: Funcionam como um dicionário, onde cada chave está associada a um valor, proporcionando uma forma simples e eficiente de armazenamento e recuperação de dados. Exemplos incluem Amazon DynamoDB e Redis.

● Wide-column/Modelo em Colunas:: Organizam dados em colunas e famílias de colunas, oferecendo alta eficiência para operações de leitura e escrita em grandes volumes de dados. Exemplos incluem Apache Cassandra e HBase.

● Graph/Modelo em Grafos: Modelam dados como grafos, com nós, arestas e propriedades, facilitando a representação e consulta de relacionamentos complexos. Os nós geralmente armazenam dados sobre indivíduos, lugares e coisas(como substantivos), enquanto as arestas armazenam as relações entre os nós. Exemplos incluem Neo4j e Amazon Neptune.

A escolha do modelo NoSQL ideal depende de diversos fatores, como a natureza dos dados, os padrões de acesso, a escalabilidade desejada e os requisitos de performance da aplicação. Em resumo, a diversidade de modelos NoSQL reflete a necessidade de soluções flexíveis e adaptáveis para lidar com a crescente variedade e volume de dados que caracterizam o mundo atual.

Portanto, neste trabalho será explorado em detalhes o conceito de bancos de dados não relacionais, com foco particular no Neo4j. Serão discutidas considerações iniciais, descrever o modelo não relacional, o modelo escolhido, neste caso sendo o Neo4j, suas características, e, por fim, exemplos práticos de sua aplicação.

1.1 Objetivos

No cenário atual, a crescente complexidade e volume dos dados gerados pelas aplicações modernas exigem soluções de armazenamento e gerenciamento que ultrapassem as limitações dos tradicionais bancos de dados relacionais. Bancos de dados não relacionais, ou NoSQL, emergiram como alternativas viáveis, oferecendo escalabilidade, flexibilidade de esquema e desempenho otimizado para diferentes tipos de dados. Dentro dessa categoria, o Neo4j, um banco de dados de grafos, destaca-se por sua capacidade de representar e consultar relacionamentos complexos de maneira eficiente. Este estudo visa investigar a aplicabilidade e os benefícios do uso do Neo4j através de um estudo de caso prático.

1.2 Justificativa

A escolha do Neo4j para este estudo de caso é fundamentada em suas características distintivas como um banco de dados de grafos, oferecendo

vantagens significativas em comparação com outros modelos de bancos de dados não relacionais. Entre os bancos de dados NoSQL, o Neo4j se destaca pela sua capacidade de modelar e consultar estruturas de dados complexas de forma intuitiva e eficiente. Sua abordagem baseada em grafos permite representar entidades como nós e seus relacionamentos como arestas, proporcionando uma representação natural para dados interconectados, como redes sociais, sistemas de recomendação e análise de redes.

A facilidade de acesso e utilização do Neo4j também contribui significativamente para sua escolha neste estudo. Através da plataforma Neo4j Sandbox, o ambiente de desenvolvimento e teste fica acessível de forma rápida e prática, permitindo aos pesquisadores e desenvolvedores explorar e experimentar com o banco de dados de grafos sem necessidade de infraestrutura complexa ou configurações demoradas.

1.3 Aspectos Metodológicos

Os aspectos metodológicos no estudo de caso sobre banco de dados não relacionais, como o Neo4j, envolvem uma abordagem prática e aplicada que visa explorar as capacidades específicas desta tecnologia. Inicialmente, foi realizada uma revisão bibliográfica abrangente para compreender os fundamentos teóricos dos bancos de dados não relacionais, incluindo suas categorias (Document-oriented databases, Key-value databases, Wide-column stores e graph databases) e as características distintivas de cada uma. Esse levantamento bibliográfico foi essencial para fundamentar a escolha do Neo4j como foco principal deste estudo, devido às suas capacidades específicas de modelagem e consulta de dados grafos.

Em seguida, utilizando aplicação prática, foi delimitado no estudo de caso, para simular um cenário realista, um sistema de gerenciamento de inventário sebos, onde os dados de inventário incluem informações detalhadas sobre livros, como título, autor, categoria, condição do livro e relacionamentos entre livros similares ou pertencentes a coleções específicas. A escolha deste cenário se justifica pela necessidade de um sistema flexível e escalável que possa lidar com a complexidade e a variabilidade dos dados típicos de um sebo.

Por fim, os resultados obtidos foram interpretados e comparados com abordagens tradicionais de bancos de dados relacionais e outros modelos NoSQL. Essa análise permitiu identificar as vantagens do Neo4j em termos de flexibilidade, escalabilidade e desempenho na gestão de inventários complexos. A metodologia adotada abrange desde a seleção e delimitação do objeto de estudo até a análise dos resultados obtidos

1.4 Aporte Teórico

No âmbito deste estudo de caso que explora o banco de dados não relacional Neo4j, foi realizada uma pesquisa bibliográfica. Dentre os diversos tipos de bancos de dados não relacionais, os bancos de dados de grafos, como o Neo4j, destacam-se por sua capacidade de modelar relações complexas de forma intuitiva e eficiente. Ao contrário dos modelos relacionais baseados em tabelas, onde a estrutura dos dados é rigidamente definida, os grafos permitem representar entidades e suas interações através de nós e arestas, proporcionando uma visualização clara e direta das relações entre os dados (Robinson et al., 2013). Também destaca-se a evolução dos sistemas de gerenciamento de banco de dados, desde os tradicionais modelos relacionais até os emergentes modelos NoSQL, que se destacam pela flexibilidade e capacidade de lidar com volumes massivos de dados não estruturados e semiestruturados.

Por fim, outra a literatura acadêmica recorrida foi a de Claudino et al. (2015), que exploram, de forma comparativa, os conceitos entre os modelos relacionais e NOSQL em contextos específicos.

2. PROJETO PROPOSTO

Nesta seção, serão apresentadas detalhadamente a metodologia utilizada neste trabalho, os motivos para sua escolha e suas etapas. Será explicado como os documentos referentes ao sistema proposto foram elaborados e descreveremos os demais artefatos associados a este projeto.

2.1 Considerações Iniciais

As demandas crescentes por gestão de informações, especialmente online, têm impulsionado o uso de sistemas NoSQL (CURÉ et al., 2012).

Contextualizando, a utilização de sistemas NoSQL se torna crucial especialmente quando as aplicações necessitam manter operações de alto desempenho sobre grandes volumes de dados (MPINDA; BUNGAMA; MASCHIETTO, 2015). Essa necessidade é evidente em aplicações com dados distribuídos globalmente, acessados simultaneamente por um vasto número de usuários. Diante desse contexto, equipes de desenvolvimento frequentemente optam por iniciar novos projetos de banco de dados diretamente com sistemas NoSQL ou, caso já tenham infraestrutura em bancos de dados relacionais, consideram a conversão para melhor atender às demandas de escalabilidade e performance.

O termo NoSQL não se refere a um modelo de dados específico, mas sim a uma categoria que engloba diversos modelos distintos dos tradicionais relacionais. Suas características gerais incluem serem geralmente de código aberto, distribuídos e capazes de escalar horizontalmente, além de oferecerem flexibilidade ou dispensarem esquemas estritos (HAN et al., 2011). Esses sistemas foram concebidos para resolver desafios específicos na manipulação de dados, especialmente para suportar esquemas flexíveis ou a ausência de esquemas, permitindo o armazenamento de dados semiestruturados ou não estruturados. Eles lidam com grandes volumes de dados que devem estar prontamente disponíveis para aplicações em tempo real (INDRAWAN-SANTIAGO, 2012). Em muitos casos, esses desafios foram ou são específicos de determinadas organizações.

Assim, por buscarem solucionar problemáticas específicas e pontuais, as

implementações de bancos de dados NoSQL acabam divergindo entre sim, resultando em diferentes estruturas de sistemas. Estas estruturas são associadas aos modelos categorizados como Chave-Valor, Colunas, Documentos e Grafos.

Este trabalho está estruturado de forma a proporcionar uma visão aprofundada sobre o NoSQL, destacando o Neo4j, por ser o modelo selecionado, e a implementação, ao fim, de exemplo de um projeto utilizando esse SGBD.

2.2 Modelos NoSQL

- Modelo em Chave-Valor: é o que apresenta representação mais simples. Sua estrutura constitui-se de uma lista de pares de elementos compostos por uma chave e um valor.
- Modelo em Colunas: conceitualmente, é o que mais se assemelha ao Modelo Relacional, pois também são organizados por meio de linhas e colunas. No entanto, ao contrário dos bancos de dados SQL tradicionais, os armazenamentos de colunas amplas são flexíveis, onde diferentes linhas podem ter conjuntos diferentes de colunas. Esses bancos de dados podem empregar técnicas de compactação de colunas para reduzir o espaço de armazenamento e melhorar o desempenho.
- Modelo em Documentos: assim como o Modelo em Chave-valor, o em Documentos também faz uso de constitui-se entre pares de chaves e valores. Porém, nesse último modelo, os dados não são dispostos em uma única estrutura de dados, mas sim agrupados em documentos. Os valores em um banco de dados de documentos podem ser de diversos tipos, como strings, números e arrays. Este modelo de dados é flexível e adequado para dados semi-estruturados e não estruturados.
- Modelo em Grafos: Dentre os modelos classificados como NoSQL, o Modelo em Grafo é o que mais se diferencia dos demais. Enquanto as outras abordagens focam no armazenamento dos dados, este modelo destaca-se principalmente pelos relacionamentos. Essa abordagem possuem três tipos de elementos: nós, arestas e propriedades. Normalmente, os nós contêm dados sobre pessoas, lugares e objetos, enquanto as arestas registram as relações entre esses nós.

Conceito	Relacional	Chave-valor (Redis)	Colunas (Cassandra)	Documentos (MongoDB)	Grafo (Neo4j)
Entidade	Tabela ou conjunto de tabelas	Não existe equivalente	Família de colunas	Coleção de documentos	Rótulo
Instância	Tupla	Um valor ou coleção de valores	Itens de família de colunas	Documento	Nó
Atributo					
Simples	Atributo	Valor	Atributo	Campo	Propriedade
Composto	Dividido em atributos simples ou em tabelas interdependentes	Conjunto de valores	Aninhamento de atributos	Campos incorporados	Conjunto de propriedades ou novo nó
Multivalorado	Dividido em atributos ou em tabelas interdependentes	Conjunto de valores	Aninhamento de atributos	Campos incorporados	Conjunto de propriedades
Nulo	Possui (obrigatório)	Não possui	Não possui	Não possui	Não possui
Restrições					
Identificador de entidade	PK	Chave	Chave primária (não garante unicidade)	*Object identifier* (_id)	Id
Referência	FK	Não possui	Não possui	Possui (DBRefs)	Possui (aresta)
Relacionamentos (1:1, 1:N, M:N)	Referência por FK	Não possui	Aninhamento de atributos	Campos incorporados ou DBRef	Por meio das arestas
Integridade Referencial	Possui	Não possui	Não possui	Não possui	Possui

Figura 1 – Mapeamento de conceitos entre os modelos (Freitas, Myller & Souza, Damires & Salgado, Ana Carolina, 2015)

2.3 Descrição do Neo4j

Neo4j é um banco de dados especializado em grafos, o que significa que ele adota um modelo de grafos completo, desde o armazenamento até a manipulação dos dados, seguindo a lógica de um quadro branco. Em contraste com abstrações de grafos sobre outras tecnologias, Neo4j oferece além do modelo de grafos, transações ACID, suporte a clusters e recuperação automática em tempo real.

O Neo4j é disponibilizado como um serviço de nuvem gerenciado através do AuraDB. No entanto, também é possível operar Neo4j independentemente, optando pela Community Edition ou pela Enterprise Edition. A última inclui todos os recursos da versão Community, além de funcionalidades empresariais adicionais, como backups, clustering e capacidade de recuperação em caso de falhas. Escrito em Java e Scala, o código-fonte do Neo4j está acessível no GitHub.

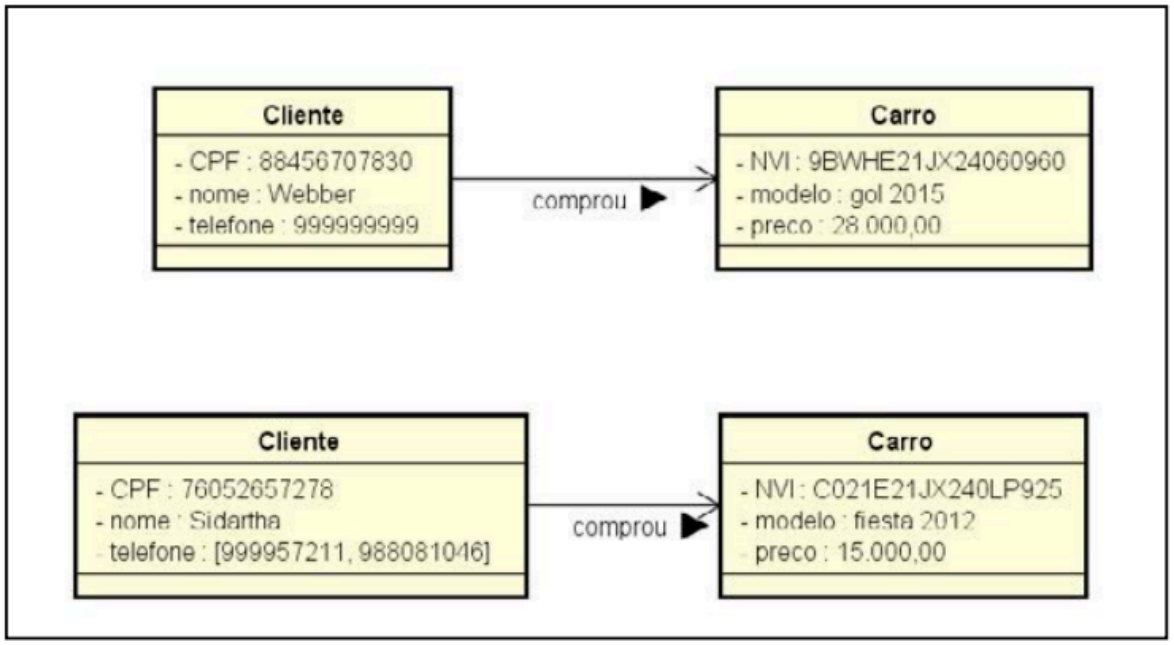

Figura 2 – Exemplo de um modelo em grafo (Freitas, Myller & Souza, Damires & Salgado, Ana Carolina, 2015)

2.4 Modelo de Dados Utilizado pelo Neo4j

O Neo4j adota um modelo de dados único baseado em grafos, que se diferencia dos modelos utilizados por bancos de dados tradicionais. No ambiente do Neo4j, as informações são estruturadas em nós, arestas e propriedades. Os nós representam entidades no contexto do problema, como pessoas, produtos ou locais, conectados por arestas que denotam os relacionamentos entre essas entidades.

Cada nó pode ser descrito por várias propriedades, que são pares de chave-valor que oferecem detalhes específicos sobre o nó. Por exemplo, um nó representando uma pessoa pode incluir propriedades como "nome", "idade" e "cidade de residência". As arestas também podem ter suas próprias propriedades, o que enriquece os relacionamentos entre nós com informações adicionais relevantes.

Além disso, o Neo4j permite a criação de índices para melhorar a velocidade das consultas aos nós com base em propriedades específicas, e oferece constraints para garantir a consistência dos dados dentro do grafo.

Através dessas interações, é possível compreender de forma detalhada as expectativas, fluxos operacionais e particularidades do cenário em que o sistema será implementado, tornando-se fundamental para garantir uma base sólida para o desenvolvimento de um sistema de gestão de ingresso eficiente e alinhado os requisitos reais e necessidades do usuário.

3. RESULTADOS OBTIDOS

Nesta seção, será apresentado o projeto exemplo desenvolvido utilizando o SGBD selecionado.

3.1 Desenvolvimento do Gerenciamento de Inventário

Para aplicar e praticar, então, o que foi desenvolvido, foi-se projetado um sistema de gerenciamento de inventário, desenvolvido com o Neo4j, para lojas de sebo que comercializam tanto como livros, CDs, vinis e outros itens culturais, para atender às necessidades dinâmicas deste estabelecimento e cenário.

A escolha do Neo4j também é motivada para esse sistema específico por sua habilidade de gerenciar de forma eficiente essas relações complexas. Em um contexto onde os produtos frequentemente possuem conexões por temas culturais, históricos ou artísticos, a capacidade de explorar e acessar essas conexões de maneira ágil e intuitiva é crucial para melhorar a experiência de busca e descoberta, tanto para os funcionários quanto para os clientes da loja. O suporte análises avançadas e recomendações personalizadas, do Neo4j, com base nos padrões de compra dos clientes e nas características dos produtos é outra vantagem e motivação para a seleção deste SGBD. O sistema pode sugerir itens adicionais que complementem interesses específicos, melhorando a experiência de compra e aumentando as vendas. Essa capacidade de personalização e inteligência na recomendação de produtos é um diferencial significativo em um mercado competitivo.

```
// Adicionar livros
CREATE (:Livro {titulo: 'Dom Casmurro', autor: 'Machado de Assis', ano: 1899});
CREATE (:Livro {titulo: 'A Hora da Estrela', autor: 'Clarice Lispector', ano: 1977});
CREATE (:Livro {titulo: 'A Tempestade', autor: 'William Shakespeare', ano: 1611});
CREATE (:Livro {titulo: 'A Redoma de Vidro', autor: 'Sylvia Plath', ano: 1963});
CREATE (:Livro {titulo: 'Vidas Secas', autor: 'Graciliano Ramos', ano: 1938});

// Adicionar  CDs
CREATE (:CD {artista: 'Simon & Garfunkel', album: 'Bookends', ano: 1968});
CREATE (:CD {artista: 'Bon Iver', album: 'Bon Iver', ano: 2011});

// Adicionar  vinis
CREATE (:Vinil {artista: 'Borislav Slavov', album: 'Baldurs Gate 3 (Original Game Soundtrack)', ano: 2023});
CREATE (:Vinil {artista: 'Fleetwood Mac', album: 'Rumours', ano: 1977});

```

Figura 3 – Consulta de dados utilizando o Neo4j

4. CONCLUSÃO

Para este estudo de caso sobre NoSQL, utilizando Neo4j como exemplo, a aplicação prática demonstrou benefícios significativos na gestão de um sebo que comercializa livros, CDs e vinis. Utilizando Neo4j, foi possível modelar graficamente o relacionamento entre os diferentes tipos de itens disponíveis, como livros, CDs e vinis, e suas propriedades específicas como título, autor, artista e ano de lançamento. Este modelo gráfico facilitou a visualização e a busca de informações inter-relacionadas de maneira intuitiva, algo que seria complexo com bancos de dados relacionais tradicionais.

Além da eficiência operacional, a implementação do Neo4j possibilitou uma melhor experiência para os usuários finais, tanto para os clientes na busca por produtos específicos quanto para os funcionários na gestão diária do inventário. A capacidade de integrar dados de forma dinâmica e a representação visual clara das conexões entre os nós contribuíram significativamente para a precisão das operações de vendas e para a identificação de tendências de mercado.

No entanto, para aprimorar ainda mais o projeto, algumas melhorias podem ser consideradas. Uma delas seria a implementação de algoritmos de recomendação baseados em grafos, os quais poderiam sugerir produtos complementares ou relacionados com base nos padrões de compra identificados. Isso não apenas aumentaria as oportunidades de venda cruzada, mas também melhoraria a fidelização dos clientes ao atender suas preferências de maneira mais proativa. Outro aspecto crucial para considerar é a segurança e a escalabilidade do sistema Neo4j. Garantir que o banco de dados possa lidar com o crescimento contínuo do inventário e o aumento das transações é fundamental para manter a estabilidade operacional. Implementar práticas robustas de monitoramento de desempenho e manutenção regular do banco de dados ajudaria a mitigar potenciais problemas de performance no futuro.

Em suma ,o estudo de caso demonstrou de maneira clara os benefícios estratégicos e operacionais de adotar o Neo4j como uma solução de banco de dados para gerenciamento de inventário em ambientes comerciais. Com as

melhorias sugeridas, o poderá alcançar novos níveis de eficiência e eficácia e, em uma realidade positivista, poderá até se consolidar como uma plataforma inovadora e adaptável às necessidades dinâmicas do mercado de varejo de produtos culturais.

REFERÊNCIAS

A. LIVROS:

Robinson et al. **Graph databases.** 2. ed. O'Reilly Media, 2015

B. MONOGRAFIAS, DISSERTAÇÕES, TESES:

Freitas, Myller & Souza, Damires & Salgado, Ana Carolina. (2015). **Mapeamentos conceituais entre os modelos relacional e NoSQL**: Uma abordagem comparativa. Revista Principia - Divulgação Científica e Tecnológica do IFPB. 1. 37.

CURÉ et al. **On the potential integration of an ontology-based data access approach in NoSQL stores**. In: INTERNATIONAL CONFERENCE ON EMERGING INTELLIGENT DATA AND WEB TECHNOLOGIES (EIDWT), 3., 2012, Bucareste. Proceedings… Bucareste: IEEE, 2012. p.166-173

MPINDA et al. **From relational database to columnoriented NoSQL database**: migration process. International Journal of Engineering Research & Technology, v. 4, n. 5, p. 399-403, 2015.

Han, Jing et al. "**Survey on NoSQL database**." 2011 6th International Conference on Pervasive Computing and Applications (2011): 363-366.

INDRAWAN-SANTIAGO, M. **Database research: are we at a crossroad? Reflection on NoSQL**. In: INTERNATIONAL CONFERENCE ON NETWORKBASED INFORMATION SYSTEMS (NBiS), 15., 2012, Melbourne, Australia. Proceedings… Melbourne: IEEE, 2012. p. 45-51

C. ONLINE:.

MongoDB. **What Is NoSQL? NoSQL Database Explained | MongoDB.** Disponível em: <https://www.mongodb.com/resources/basics/databases/nosql-explained/>. Acesso em: 20 jun 2024.

Neo4j. **What is a graph database? - Getting Started.** Disponível em: <https://neo4j.com/docs/getting-started/get-started-with-neo4j/graph-database/#property-graph)/>. Acesso em: 20 jun 2024.