

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

HEITOR APARECIDO DA SILVA

**ESTUDO SOBRE A VIABILIDADE DO USO DO BANCO DE DADOS NEO4J
PARA A INTERPOLAÇÃO DE DADOS NO CONTEXTO DE AGRICULTURA DE
PRECISÃO**

MEDIANEIRA

2025

HEITOR APARECIDO DA SILVA

**ESTUDO SOBRE A VIABILIDADE DO USO DO BANCO DE DADOS NEO4J
PARA A INTERPOLAÇÃO DE DADOS NO CONTEXTO DE AGRICULTURA DE
PRECISÃO**

**Study on the feasibility of using the Neo4j database for data interpolation in
the context of precision agriculture**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação
do Curso de Bacharelado em Ciência da
Computação da Universidade Tecnológica
Federal do Paraná.

Orientador: Prof^a. Dr^a. Kelyn Schenatto

MEDIANEIRA

2025



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

HEITOR APARECIDO DA SILVA

**ESTUDO SOBRE A VIABILIDADE DO USO DO BANCO DE DADOS NEO4J
PARA A INTERPOLAÇÃO DE DADOS NO CONTEXTO DE AGRICULTURA DE
PRECISÃO**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação
do Curso de Bacharelado em Ciência da
Computação da Universidade Tecnológica
Federal do Paraná.

Data de aprovação: 27/junho/2025

Alan Gavioli

Doutor

Universidade Tecnológica Federal do Paraná - Campus Medianeira

Alessandra Bortoletto Garbelotti Hoffmann

Doutora

Universidade Tecnológica Federal do Paraná - Campus Medianeira

Kelyn Schenatto

Doutora

Universidade Tecnológica Federal do Paraná - Campus Medianeira

MEDIANEIRA

2025

AGRADECIMENTOS

Gostaria de agradecer a todos que, em algum momento, fizeram parte desta minha jornada. Como citá-los nominalmente seria inviável, inicio agradecendo à UTFPR e ao corpo docente do curso de Ciência da Computação do câmpus Medianeira, pelos ensinamentos e apoio ao longo desta trajetória.

Meu sincero agradecimento à minha orientadora, Dra. Kelyn Schennato, por me socorrer nos momentos de desespero, quando parecia que nada daria certo. Agradeço também ao Dr. Claudio Bazzi, por ter me auxiliado nas configurações do projeto no PostgreSQL e por ter dedicado parte do seu tempo para isso.

Gostaria ainda de agradecer aos amigos que fizeram parte desta jornada, em especial ao meu colega de apartamento, Matheus, por ter me aturado durante esses longos anos.

Por fim, mas de forma especial, agradeço à minha família – meu pai, minha mãe e minha irmã – pois, sem eles, eu definitivamente não estaria aqui. A vocês, o meu mais sincero obrigado.

RESUMO

A geração de mapas temáticos por meio da interpolação de dados geoespaciais é uma prática essencial na Agricultura de Precisão, pois permite estimar atributos do solo em locais não amostrados a partir de dados coletados em pontos vizinhos. Esses mapas auxiliam na tomada de decisões agronômicas, permitindo otimizar e aumentar a produtividade. Este trabalho teve como objetivo verificar a viabilidade da implementação dos métodos de interpolação pelo inverso da distância, inverso da distância ao quadrado e médias móveis como *procedures* no banco de dados Neo4j, visando sua aplicação nesse contexto. A motivação para o estudo reside no fato de que, embora o PostgreSQL, especialmente com a extensão PostGIS, seja amplamente reconhecido por sua robustez no tratamento de dados geoespaciais, pode apresentar limitações em aplicações que exigem grande escalabilidade, consultas complexas ou manipulação intensiva de relacionamentos entre dados. Nesse cenário, bancos orientados a grafos, como o Neo4j, vêm se destacando pela flexibilidade e eficiência em operações que envolvem conexões complexas. Os métodos foram implementados e executados em diversas configurações sobre dois conjuntos de dados distintos, com os resultados comparados a uma implementação equivalente no PostgreSQL. Foram coletados os tempos de execução e gerados mapas temáticos para avaliar os níveis de concordância entre os mapas. A análise indicou que o Neo4j apresentou desempenho superior, com reduções significativas no tempo de execução, especialmente em cenários de alta densidade de pontos, atingindo até 17 vezes mais rapidez na configuração mais extrema. Assim, conclui-se que a implementação dos métodos no Neo4j é não apenas viável, mas também promissora, contribuindo para o avanço do uso de bancos orientados a grafos em aplicações geoespaciais.

Palavras-chave: comparação de sgbd; bancos de dados orientados a grafos; postgis; sistemas de informação geográfica.

ABSTRACT

The generation of thematic maps through geospatial data interpolation is an essential practice in Precision Agriculture, as it enables the estimation of soil attributes in unsampled locations based on data collected from neighboring points. These maps support agronomic decision-making by optimizing input use and increasing productivity. This study aimed to verify the feasibility of implementing interpolation methods — inverse distance, inverse distance squared, and moving averages — as *procedures* in the Neo4j database, targeting their application in this context. The motivation for the study lies in the fact that, although PostgreSQL, especially with the PostGIS extension, is widely recognized for its robustness in handling geospatial data, it may present limitations in applications requiring high scalability, complex queries, or intensive manipulation of relationships between data. In this scenario, graph-oriented databases such as Neo4j have stood out for their flexibility and efficiency in operations involving complex connections. The methods were implemented and executed in various configurations across two distinct datasets, with the results compared to an equivalent implementation in the PostgreSQL environment. Execution times were collected and thematic maps were generated to assess the levels of agreement between the environments. The analysis indicated that Neo4j demonstrated superior performance, with significant reductions in execution time, especially in scenarios with high point density, achieving up to 17 times faster execution in the most extreme configuration. Thus, it is concluded that implementing these methods in Neo4j is not only feasible but also promising, contributing to the advancement of the use of graph-oriented databases in geospatial applications.

Keywords: dbms comparison; graph databases; postgis; geographic information systems (gis).

LISTA DE FIGURAS

Figura 1 – Modelo Entidade Relacionamento HR	15
Figura 2 – Tipos de modelos de armazenamento de dados em bancos Not Only SQL (NoSQL): (a) Bancos baseados em documentos; (b) Bancos orientados a chave-valor; (c) Bancos baseado em colunas; (d) Banco orientado a grafos.	16
Figura 3 – Estrutura de grafos no Neo4j.	18
Figura 4 – Exemplo de query Cypher utilizando o tipo Point.	18
Figura 5 – Sintaxe de expressões, queries e cláusulas do Cypher.	19
Figura 6 – Etapas do Estudo: Interpolação de Dados em PostgreSQL e Neo4j . . .	25
Figura 7 – Pontos de coleta de amostras para análises físico-químicas do solo na área 1.	27
Figura 8 – Pontos de coleta de amostras para análises físico-químicas do solo na área 2.	28
Figura 9 – Pontos interpolados ID 10x10 Neo4j x PostgreSQL na área 1.	37
Figura 10 – Mapas gerados da área 1 com grade 10x10.	38
Figura 11 – Mapas gerados da área 2 com grade 10x10.	40
Figura 12 – Relação entre a densidade de pontos na grade de interpolação e o índice de concordância.	42
Figura 13 – Comparação dos pontos interpolados fora do contorno nas áreas 1 e 2, utilizando o método ID no PostgreSQL.	42
Figura 14 – Relação entre os métodos de interpolação no Neo4j e PostgreSQL e tempo de execução.	46

LISTA DE TABELAS

Tabela 1 – Interpretação do índice Kappa	30
Tabela 2 – Relação tamanho da grade x concordância entre os mapas - Inverso da distância (ID) Área 1	37
Tabela 3 – Concordância entre os mapas de amostras interpoladas na área 1 - grade 10x10	39
Tabela 4 – Concordância entre os mapas de amostras interpoladas na área 2 — grade 10x10	41
Tabela 5 – Concordância entre os mapas gerados R x Neo4j na área 1 — grade 10x10	43
Tabela 6 – Concordância entre os mapas gerados R x PostgreSQL na área 1 — grade 10x10	43
Tabela 7 – Concordância entre os mapas gerados R x Neo4j na área 2 — grade 10x10	44
Tabela 8 – Concordância entre os mapas gerados R x PostgreSQL na área 2 — grade 10x10	44
Tabela 9 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 2x2	45
Tabela 10 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 5x5	45
Tabela 11 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 10x10	45
Tabela 12 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 15x15	45
Tabela 13 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 20x20	46

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Classe SourcePoint	32
Listagem 2 – Classe ResultadoInterpolacaoBulk	32
Listagem 3 – Assinatura do método interpolarIDWBulk	33
Listagem 4 – Principais passos do método interpolarIDWBulk	34
Listagem 5 – Implementação do método calcularIDW	35
Listagem 6 – Assinatura do método interpolarMMBulk	35
Listagem 7 – Implementação do método calcularMediaMovel	36

LISTA DE ABREVIATURAS E SIGLAS

Siglas

AP	Agricultura de Precisão
ID	Inverso da distância
IDQ	Inverso da distância ao quadrado
IDW	Inverso da distância elevado a uma potência
MM	Médias móveis
NoSQL	Not Only SQL
PL/SQL	Procedural Language/Structured Query Language
SGBD	Sistemas Gerenciadores de Banco de Dados
SQL	Structured Query Language

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos	11
1.1.1	Objetivo geral	12
1.1.2	Objetivos específicos	12
1.2	Justificativa	12
1.3	Estrutura do trabalho	13
2	REFERENCIAL TEÓRICO	14
2.1	Banco de dados	14
2.1.1	Modelo Relacional	14
2.1.2	Modelo Não Relacional	15
2.2	Sistema Gerenciador de Banco de Dados Neo4j	16
2.2.1	Tipo de Dados Point no Neo4j	18
2.2.2	Cypher	19
2.3	Agricultura de Precisão	20
2.3.1	Dependência espacial dos dados	20
2.3.2	Mapas temáticos	20
2.3.3	Métodos de Interpolação de Dados	21
2.4	Trabalhos Relacionados	22
3	MATERIAIS E MÉTODOS	25
3.1	Ambiente e ferramentas utilizadas	25
3.2	Modelagem de dados no Neo4j	26
3.3	Implementação dos métodos de interpolação ID, IDQ e MM no Neo4j	26
3.4	Interpolação dos dados físicos e químicos	26
3.5	Interpolação pelos métodos ID, IDQ e MM no PostgreSQL	28
3.6	Análise de desempenho	29
3.7	Geração dos mapas temáticos	29
3.8	Comparação dos mapas	29
4	RESULTADOS E DISCUSSÃO	31
4.1	Implementação dos Métodos no Neo4j	31
4.2	Implementação da classe InterpolacaoIDWBulk	33

4.3	Implementação da classe InterpolacaoMMBulk	35
4.4	Interpolação dos dados	36
4.5	Comparação entre os mapas gerados	38
4.6	Interpolação com R	43
4.7	Análise de tempo de consulta	44
5	CONCLUSÃO	47
	REFERÊNCIAS	48

1 INTRODUÇÃO

A agricultura sempre foi um dos pilares fundamentais da sociedade, sendo responsável por garantir a segurança alimentar e impulsionar a economia em diversas regiões do mundo. Com o avanço das tecnologias aplicadas ao campo, surgiu o conceito de Agricultura de Precisão (AP), um conjunto de técnicas e ferramentas voltadas para a gestão localizada da produção agrícola, visando aumentar a produtividade e otimizar o uso de recursos naturais e insumos.

Segundo Bassoi *et al.* (2019), a AP busca, além de elevar a eficiência produtiva, reduzir o desperdício de recursos como água, fertilizantes e defensivos agrícolas. Entre as principais ferramentas utilizadas nesse contexto, destaca-se a interpolação de dados geoespaciais, técnica que permite a geração de mapas temáticos para representar características físicas e químicas do solo, auxiliando diretamente na tomada de decisão pelos produtores.

A interpolação, nesse cenário, consiste na estimativa de atributos em locais não amostrados a partir de dados coletados em pontos vizinhos. A aplicação dessa técnica resulta na geração de grandes volumes de dados espaciais, o que exige soluções eficazes para seu armazenamento e processamento. Tradicionalmente, Sistemas Gerenciadores de Banco de Dados (SGBD) relacionais têm sido utilizados para esse fim, devido à sua capacidade de garantir integridade e consistência nas informações. No entanto, em cenários que envolvem dados volumosos ou com estrutura complexa, como ocorre na AP, esses bancos podem apresentar limitações de desempenho.

Nesse sentido, os SGBDs NoSQL surgem como alternativa, sendo classificados em categorias como orientados a documentos, chave-valor, colunas e grafos. Os bancos orientados a grafos, em particular, oferecem uma abordagem flexível para o armazenamento de dados altamente conectados, representando entidades e seus relacionamentos por meio de nós e arestas. Essa estrutura facilita consultas complexas e pode proporcionar ganhos de desempenho em diversas aplicações.

Este trabalho concentra-se na utilização do banco de dados orientado a grafos Neo4j como plataforma para o desenvolvimento e execução de métodos de interpolação no contexto da Agricultura de Precisão. O Neo4j foi escolhido por sua popularidade e desempenho em aplicações baseadas em grafos, além da possibilidade de criação de procedimentos personalizados para manipulação de dados geoespaciais, mesmo que seu suporte nativo ainda seja limitado nesse aspecto.

1.1 Objetivos

Esta seção tem como objetivo apresentar as metas principais a serem alcançadas com a elaboração deste trabalho.

1.1.1 Objetivo geral

Analisar a viabilidade da utilização do SGBD orientado a grafos Neo4j na interpolação de dados geoespaciais provenientes de pontos de coleta de atributos do solo e das plantas no contexto da Agricultura de Precisão, visando apoiar a tomada de decisões por meio da comparação de desempenho com soluções implementadas no SGBD relacional PostgreSQL.

1.1.2 Objetivos específicos

- Implementar os métodos de interpolação "Inverso da Distância Elevado a uma Potência" e "Média Móvel", avaliando sua precisão e aplicabilidade no contexto da Agricultura de Precisão;
- Desenvolver *procedures* personalizadas no Neo4j para o processamento eficiente de dados geoespaciais, com o intuito de reduzir o tempo de execução das operações de interpolação;
- Comparar o desempenho dos métodos implementados no Neo4j com os métodos já existentes no PostgreSQL, com base em métricas como tempo de execução, acurácia dos resultados e índice de concordância entre os mapas gerados.

1.2 Justificativa

Embora o PostgreSQL, especialmente com o uso da extensão PostGIS, seja amplamente reconhecido por sua robustez no tratamento de dados geoespaciais, seu desempenho pode ser limitado em aplicações que exigem grande escalabilidade, consultas complexas ou manipulação intensiva de relacionamentos entre dados. Nesse contexto, bancos de dados orientados a grafos, como o Neo4j, vêm ganhando destaque por oferecerem uma estrutura mais flexível e eficiente para determinadas operações, especialmente aquelas que envolvem conexões complexas entre entidades.

Apesar de o Neo4j ainda apresentar limitações em relação ao suporte nativo a geometrias espaciais, sua crescente adoção e capacidade de personalização, por meio do desenvolvimento de *procedures* em Java e da utilização de bibliotecas complementares como a JTS, justificam a investigação de sua aplicabilidade em cenários como a interpolação de dados para Agricultura de Precisão.

Este estudo busca, portanto, contribuir para o avanço do conhecimento sobre o uso de bancos de dados orientados a grafos em aplicações geoespaciais, avaliando se o Neo4j pode oferecer vantagens em termos de desempenho e flexibilidade em comparação a soluções tradicionais baseadas em bancos relacionais. Os resultados obtidos poderão servir como base

para futuras pesquisas e aplicações no desenvolvimento de sistemas de informação geográfica mais eficientes.

1.3 Estrutura do trabalho

O trabalho está organizado em cinco capítulos principais. O primeiro capítulo, a Introdução, contextualiza o tema e apresenta a proposta desenvolvida.

O segundo capítulo explora os principais conceitos relacionados ao estudo, além de destacar trabalhos que serviram de base para este.

O terceiro capítulo detalha a metodologia aplicada no desenvolvimento do trabalho, apresentada no capítulo 3, na qual são descritas as ferramentas e tecnologias utilizadas, bem como a forma como foram empregadas para o desenvolvimento e a realização das análises.

No quarto capítulo, são apresentados os resultados obtidos a partir das implementações realizadas, acompanhados da análise dos dados.

Por fim, no quinto capítulo, são apresentadas as conclusões do estudo, bem como sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados os principais conceitos e fundamentos teóricos que embasam o desenvolvimento deste trabalho, incluindo tópicos relacionados à AP, interpolação de dados e SGBDs orientados a grafos.

2.1 Banco de dados

Segundo Date (2004), um banco de dados pode ser definido como um sistema informatizado de manutenção de registros, que permite ao usuário gravar novos registros e recuperar aqueles já armazenados.

De acordo com Su (2022), os SGBDs desempenham um papel fundamental no desenvolvimento de software, permitindo o armazenamento, a recuperação e a organização dos dados, além de suportar a geração de arquivos necessários para sua manipulação. Essas funcionalidades são essenciais para o funcionamento eficiente das aplicações.

Os SGBDs podem ser classificados em dois grandes grupos: relacionais e não relacionais. Os SGBDs relacionais foram os primeiros a surgir, caracterizando-se por regras mais rígidas e pelo armazenamento dos dados em estruturas tabulares. Por outro lado, os SGBDs não relacionais surgiram como uma alternativa ao modelo relacional, buscando implementar estruturas de armazenamento mais específicas para determinadas situações. Nesse modelo, os dados deixam de ser organizados em tabelas e passam a ser armazenados em diferentes tipos de estruturas, cada uma com suas próprias vantagens e desvantagens.

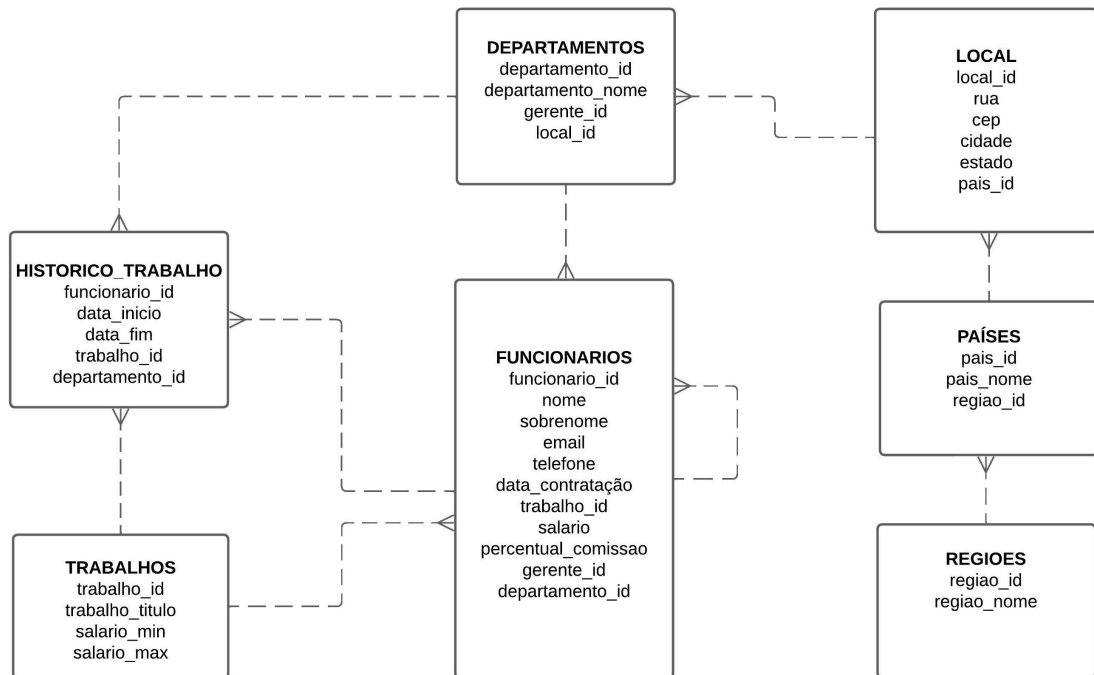
2.1.1 Modelo Relacional

Bancos de dados relacionais são os mais tradicionalmente empregados em sistemas de informação. Segundo Macário e Baldo (2005), o modelo relacional foi proposto por Edgar Codd em 1970, com o objetivo de descrever os dados de forma mais natural, eliminando a necessidade de estruturas adicionais para representá-los.

Um banco de dados relacional é composto por tabelas, que são restringidas por regras de integridade relacional. "A restrição de integridade é uma condição especificada no esquema do banco de dados, com o objetivo de restringir a entrada de dados não desejáveis a uma instância do banco de dados"(CAMARGO *et al.*, 2001). Embora essas restrições limitem o modelo em termos de dinamicidade e escalabilidade, elas garantem uma maior integridade dos dados. Isso ocorre porque o modelo relacional implementa regras que impedem a alteração de dados que possam comprometer a integridade relacional com outros elementos do banco.

A Figura 1 apresenta um exemplo de modelagem de um banco de dados relacional para um sistema de recursos humanos, evidenciando a disposição das informações em tabelas e os relacionamentos entre elas.

Figura 1 – Modelo Entidade Relacionamento HR



Fonte: Adaptado de Oracle (2024).

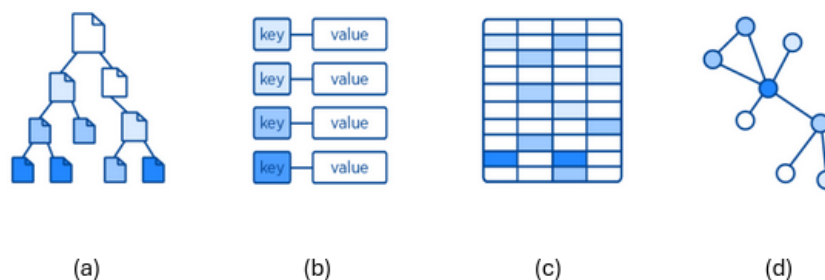
2.1.2 Modelo Não Relacional

De acordo com Čerešňák e Kvet (2019), os SGBDs NoSQL surgiram como uma resposta às limitações dos SGBDs relacionais, especialmente diante do crescimento de dados não estruturados. Além disso, foram desenvolvidos para atender às demandas de desempenho, escalabilidade e flexibilidade exigidas por aplicações modernas. Dentre os tipos de bancos de dados NoSQL existentes, destacam-se os quatro principais:

- **Banco de dados orientado a documentos:** Segundo Truică *et al.* (2021), os SGBDs orientados a documentos armazenam dados em estruturas similares a XML e JSON. A flexibilidade desses formatos facilita a manipulação dos dados, especialmente quando comparados à estrutura rígida de tabelas em SGBDs relacionais. Dentre os SGBDs orientados a documentos o MongoDB é um dos mais populares, sendo considerado um dos principais SGBDs NoSQL (Figura 2a).

- **Banco de dados orientado a chave-valor:** Segundo González-Aparicio *et al.* (2018), um SGBD orientado a chave-valor consiste em pares formados por um identificador (chave) e um valor correspondente. Os valores podem ser *strings*, listas ou conjuntos. As consultas são realizadas com base nas chaves, limitando-se a operações simples de criação, leitura, atualização e deleção (CRUD). Uma das principais aplicações desse tipo de SGBD é como banco secundário, visando principalmente melhorar a performance e a responsividade da aplicação. Um dos principais SGBDs desse tipo é o Redis, um SGBD em memória amplamente utilizado para cacheamento de dados mais acessados (Figura 2b).
- **Banco de dados orientado a colunas:** Segundo Mehmood, Culmone e Mostarda (2017), ao contrário dos SGBDs relacionais, que utilizam linhas para recuperar informações, os SGBDs orientados a colunas armazenam e acessam os dados por meio de colunas indexadas com identificadores únicos. Essa forma de armazenamento torna esse tipo de SGBD especialmente indicado para aplicações que realizam consultas extensivas à base de dados (Figura 2c).
- **Banco de dados orientado a grafos:** Segundo Monteiro, Sá e Bernardino (2023), esse tipo de SGBD fornece recursos que facilitam a compreensão do grafo, permitindo que os nós contêm um ou mais rótulos para auxiliar na identificação. Além disso, os relacionamentos entre nós podem possuir propriedades, e o mesmo nó pode ser relacionado a diversos outros elementos no grafo (Figura 2d).

Figura 2 – Tipos de modelos de armazenamento de dados em bancos NoSQL: (a) Bancos baseados em documentos; (b) Bancos orientados a chave-valor; (c) Bancos baseado em colunas; (d) Banco orientado a grafos.



Fonte: Adaptado de Tecnologia (2022).

2.2 Sistema Gerenciador de Banco de Dados Neo4j

O Neo4j é um SGBD orientado a grafos, de código aberto, desenvolvido em Java, amplamente utilizado em diversas aplicações. Entre suas aplicações mais conhecidas, destaca-se

a análise de relacionamentos em redes sociais. Graças à sua estrutura baseada em grafos, o Neo4j facilita a identificação e a consulta rápida de conexões, como as relações de amizade, nesse contexto.

Segundo o próprio site do Neo4j (2025), o desenvolvimento do SGBD Neo4j teve início no começo dos anos 2000, quando seus três fundadores enfrentavam dificuldades com bancos de dados relacionais em cenários específicos.

A ferramenta evoluiu ao longo dos anos até que, em 2007, seu código-fonte foi disponibilizado sob a licença GPL (General Public License), permitindo à comunidade o uso, a modificação e a redistribuição do software. Pouco tempo depois, em 2010, foi lançada a versão 1.0 do banco de dados orientado a grafos Neo4j.

Atualmente, no momento da realização deste trabalho, o Neo4j é utilizado por milhares de empresas, incluindo LinkedIn, Microsoft, Adobe, eBay, Cisco, entre outras.

O Neo4j utiliza o *Cypher* como linguagem de consulta. Embora o *Cypher* tenha algumas limitações em comparação com o Procedural Language/Structured Query Language (PL/SQL), o Neo4j permite a criação de *procedures* personalizadas por meio de código Java, possibilitando o desenvolvimento de funções e procedimentos para manipulação direta dos dados no SGBD.

Segundo a documentação oficial de Neo4j (2024), o Neo4j armazena dados em nós, relacionamentos e propriedades, em vez de tabelas ou documentos, permitindo organizar as informações de forma semelhante a um esboço de ideias em um quadro. Além de proporcionar uma visualização simples e intuitiva dos dados, o Neo4j também oferece alta escalabilidade. De acordo com Mathew e Kumar (2014), o Neo4j é extremamente escalável, sendo capaz de lidar com bilhões de nós e relacionamentos por meio de técnicas de otimização de desempenho, o que possibilita consultas eficientes mesmo com grandes volumes de dados.

A imagem 3 ilustra um exemplo de modelagem de dados utilizando o banco de dados orientado a grafos Neo4j, demonstrando como as relações entre diferentes entidades podem ser representadas de forma visual e intuitiva.

No grafo, podemos observar a presença de nós que representam diferentes tipos de entidades. Os nós incluem pessoas (representadas por ícones de rostos humanos), restaurantes (nós azuis claros com a propriedade *name*), tipo de culinária (nó azul escuro com a propriedade *cuisine*) e localização (nó vermelho com a propriedade *location*: New York).

Esses nós estão conectados por arestas que representam as relações entre eles. As relações visíveis no grafo incluem: *É_AMIGO_DE*, que conecta duas pessoas entre si; *GOSTA_DE*, que conecta uma pessoa a um restaurante que ela aprecia; *SERVE*, que conecta um restaurante ao tipo de culinária oferecida; e *LOCALIZADO_EM*, que indica a cidade onde o restaurante ou tipo de culinária está situado.

Figura 3 – Estrutura de grafos no Neo4j.



Fonte: Adaptado de Rondón (2022).

2.2.1 Tipo de Dados Point no Neo4j

A partir da versão 3.4, o Neo4j introduziu o tipo *Point*, juntamente com uma biblioteca de funções espaciais.

O tipo de dado Point é usado para representar coordenadas geográficas ou cartesianas. Esse tipo é especialmente útil em aplicações que envolvem geolocalização, como mapas, localização de usuários, pontos de interesse, rotas, entre outros.

O tipo Point permite armazenar localizações espaciais diretamente em propriedades de nós ou relacionamentos.

Na figura 4 é possível observar a estrutura de uma consulta utilizando do tipo de dado Point.

Figura 4 – Exemplo de query Cypher utilizando o tipo Point.

```
MATCH (l:Local)
WHERE distance(l.localizacao, point({latitude: 40.78, longitude: -73.97})) < 1000
RETURN l
```

Fonte: Autoria própria (2025).

Apesar de oferecer suporte nativo a coordenadas geográficas por meio do tipo de dado Point e funções espaciais básicas, o Neo4j apresenta algumas limitações importantes no que se refere ao trabalho com dados espaciais mais complexos. Esse suporte é suficiente para operações simples, como o cálculo de distância entre pontos ou a filtragem de nós dentro de um

determinado raio geográfico. No entanto, para aplicações mais robustas envolvendo geolocalização, essas funcionalidades podem se mostrar insuficientes.

Segundo Sun e Sarwat (2019), uma das principais limitações do Neo4j é a ausência de suporte nativo a geometrias espaciais complexas, como polígonos (representando áreas), linhas (representando rotas) ou multi-geometrias. Isso significa que operações como verificação de interseção entre áreas geográficas, inclusão de um ponto em uma região poligonal ou geração de buffers espaciais não são possíveis diretamente com os recursos atuais da plataforma.

2.2.2 Cypher

Segundo Francis *et al.* (2018), Cypher é uma linguagem de consulta projetada inicialmente para o Neo4j e posteriormente adotada por outros produtos de banco de dados. Ela oferece diversos operadores e limitadores de escopo para filtrar informações em consultas, muitos dos quais são similares aos encontrados no Structured Query Language (SQL), como ilustrado na Figura 5.

De acordo com Green *et al.* (2019), embora o Cypher seja influenciado pelo SQL, ele se diferencia por permitir que operações no banco sejam intercaladas com cláusulas de consulta. Além disso, segue uma sequência em que o resultado de uma consulta anterior serve como entrada para a consulta subsequente.

Figura 5 – Sintaxe de expressões, queries e cláusulas do Cypher.

EXPRESSÕES	
$expr ::= v \mid a \mid f(expr_list) \quad v \in \mathcal{V}, a \in \mathcal{A}, f \in \mathcal{F}$	valores/variáveis
$ expr.k \mid \{ \} \mid \{ prop_list \}$	maps
$ [] \mid [expr_list] \mid expr \text{ IN } expr \mid expr [expr] \mid expr [expr..] \mid expr [.. expr] \mid expr [expr..expr]$	listas
$ expr \text{ STARTS WITH } expr \mid expr \text{ ENDS WITH } expr \mid expr \text{ CONTAINS } expr$	strings
$ expr \text{ OR } expr \mid expr \text{ AND } expr \mid expr \text{ XOR } expr \mid \text{ NOT } expr \mid expr \text{ IS NULL } \mid expr \text{ IS NOT NULL }$	lógica
$ expr < expr \mid expr <= expr \mid expr >= expr \mid expr > expr \mid expr = expr \mid expr <> expr$	desigualdades
$expr_list ::= expr \mid expr, expr_list$	listas de expressões
CONSULTAS	
$query ::= query^\circ \mid query \text{ UNION } query \mid query \text{ UNION ALL } query$	uniões
$query^\circ ::= \text{ RETURN } ret \mid clause \ query^\circ$	sequência de cláusulas
$ret ::= * \mid expr [AS a] \mid ret, expr [AS a]$	listas de retorno
CLÁUSULAS	
$clause ::= [OPTIONAL] \text{ MATCH } pattern_tuple [WHERE expr]$	cláusulas correspondentes
$ WITH ret [WHERE expr] \mid UNWIND expr AS a \quad a \in \mathcal{A}$	cláusulas relacionais
$pattern_tuple ::= pattern \mid pattern, pattern_tuple$	tupla de padrões

Fonte: Adaptado de Francis *et al.* (2018).

2.3 Agricultura de Precisão

De acordo com Ezenne *et al.* (2019), a AP é uma técnica de manejo voltada para a aplicação consciente de recursos nos cultivos, como água, fertilizantes e pesticidas. Dessa forma, a AP tem como principais objetivos evitar o desperdício de insumos e aumentar a produtividade.

Segundo Rosa *et al.* (2022), os dados utilizados para análises no contexto da produção vegetal podem ser originados de informações georreferenciadas de colheita, dados meteorológicos, ações de manejo realizadas nas fases de pré e pós-cultivo, além de propriedades do solo.

Para alcançar esses objetivos, a AP utiliza diversas ferramentas para a coleta de dados e a tomada de decisões, sendo o GPS uma das primeiras e mais importantes. Essa tecnologia é utilizada para a demarcação de pontos de coleta e talhões, gerando dados geográficos associados a diversas variáveis, como dados químicos, físicos e temporais relacionados ao ponto de coleta.

2.3.1 Dependência espacial dos dados

De acordo com Reimov *et al.* (2023), dados espaciais no contexto da agricultura consistem em coordenadas geográficas associadas às propriedades mensuradas na respectiva localização.

Para a aplicação de métodos de interpolação, é necessário pressupor a dependência espacial dos dados, considerando a heterogeneidade das propriedades físico-químicas da área medida. Essa característica possibilita estimar os valores em pontos vizinhos com base nos pontos coletados.

Segundo Meena, Muthusamy e Behera (2024), a heterogeneidade dos dados dentro do campo amostral refere-se à variação das variáveis em diferentes pontos medidos, o que permite a suposição de valores para pontos não medidos por meio de métodos geoestatísticos, como a interpolação de dados.

2.3.2 Mapas temáticos

A cartografia temática é uma ferramenta essencial para a representação e análise do espaço geográfico, sendo particularmente relevante na agricultura, onde o uso eficiente dos recursos naturais depende da compreensão espacial de fenômenos ambientais, sociais e econômicos. No contexto da agricultura de precisão, os mapas temáticos permitem a visualização espacial de variáveis que influenciam diretamente a produção rural, como tipos de solo, precipitação, cobertura vegetal, uso da terra e produtividade das culturas.

Segundo Shiratsuchi *et al.* (2007), a geração de mapas multitemáticos auxilia na tomada de decisão nas culturas de grãos, considerando fatores como fertilidade do solo, nutrição de plantas, presença de plantas daninhas e estande de plantas. A utilização de técnicas como a análise de componentes principais (ACP) permite correlacionar a produtividade das culturas com esses atributos, resultando em mapas que orientam práticas agrícolas mais eficientes.

Além disso, o desenvolvimento de aplicações web para criação de mapas temáticos tem facilitado o acesso a essas ferramentas por produtores e técnicos agrícolas. Borges *et al.* (2020) desenvolveram um aplicativo capaz de gerar mapas temáticos utilizando métodos de interpolação como inversa da distância, média móvel e vizinho mais próximo, permitindo identificar áreas com comportamentos semelhantes em relação às variáveis do solo e das plantas.

Os mapas de solos são fundamentais para identificar os tipos, classes e características físico-químicas dos solos, auxiliando na escolha das culturas e nas práticas de manejo adequadas. Já os mapas de uso e cobertura da terra permitem monitorar a expansão agrícola, o desmatamento e as transformações no uso do solo ao longo do tempo. Os mapas de precipitação e de umidade do solo são essenciais para o planejamento de sistemas de irrigação e para a avaliação da vulnerabilidade das áreas agrícolas à seca.

A integração dessas informações em sistemas de informação geográfica (SIG) e a utilização de ferramentas de agricultura de precisão contribuem para um manejo mais sustentável e eficiente das propriedades rurais, promovendo o aumento da produtividade e a conservação dos recursos naturais.

2.3.3 Métodos de Interpolação de Dados

Métodos de interpolação no contexto da AP são técnicas geoestatísticas destinadas a estimar valores em pontos não medidos de propriedades específicas do solo, cultivo, clima, entre outros, com base em informações obtidas nos pontos vizinhos. Esses métodos contribuem para aumentar a precisão da modelagem de dados aplicada ao cultivo analisado.

Segundo Řezník *et al.* (2020), existem diversos métodos de interpolação aplicáveis a diferentes áreas do conhecimento, entretanto, apenas alguns são utilizados na agricultura para a predição de pontos não medidos, como a Krigagem e o Método do Inverso da Distância.

De acordo com o trabalho de Koutsos, Menexes e Eleftherohorinos (2022), a aplicação de métodos de interpolação no contexto da AP pode aumentar significativamente a confiabilidade dos dados, ao reduzir valores extremos e lacunas, o que resulta em uma diminuição de até 30% no coeficiente de variação dos dados.

Dentre os métodos de interpolação amplamente utilizados na AP, este trabalho contempla a interpolação pelo inverso da distância elevado à potência e a interpolação por médias móveis. Segundo Wang *et al.* (2021), o método de interpolação pelo inverso da distância elevado à potência estima os valores não mensurados atribuindo pesos aos pontos próximos, que são inversamente proporcionais à distância entre o ponto conhecido e o ponto a ser interpolado.

Dessa forma, pontos mais próximos exercem maior influência. Já o método de interpolação por médias móveis, de acordo com Maleika (2015), calcula os valores interpolados a partir da média dos pontos próximos, desconsiderando aqueles mais distantes.

De acordo com Burrough, McDonnell e Lloyd (2015), a interpolação pelo inverso da distância elevada à potência (IDW) é um método determinístico que não requer pressupostos estatísticos rigorosos, sendo especialmente útil em áreas com alta densidade de pontos amostrados. No entanto, sua precisão pode ser comprometida em regiões com distribuição irregular dos dados ou em terrenos com forte variabilidade espacial, uma vez que não considera a tendência global dos dados nem a estrutura de dependência espacial entre os pontos.

Por outro lado, segundo Yamamoto e Landim (2013), a interpolação por médias móveis, tende a suavizar excessivamente os dados, podendo mascarar variações locais importantes que são fundamentais para decisões agronômicas específicas, como a aplicação localizada de insumos ou a identificação de zonas de manejo.

2.4 Trabalhos Relacionados

No trabalho de Idziaszek *et al.* (2017), foi desenvolvida uma interface web que possibilita, além da definição de zonas de manejo, a anexação de fotos aéreas da topografia dessas áreas. A inclusão dessas imagens acrescenta um componente visual adicional ao mapa, enriquecendo as análises e oferecendo ao usuário uma maneira mais clara de identificar as áreas de plantio. Isso também contribui para a tomada de decisões mais informadas, uma vez que a visualização detalhada das características topográficas facilita a identificação de variações no solo e na vegetação. A escolha do Neo4j para este projeto se deu principalmente pela alta escalabilidade proporcionada por um banco de dados orientado a grafos. Além do Neo4j, outras ferramentas foram utilizadas em conjunto, destacando-se ASP.NET, HTML, CSS e Google Maps.

No estudo de Zheng *et al.* (2017), foi realizada uma comparação experimental entre três SGBDs (PostGIS, MongoDB e Neo4j) utilizando dados espaço-temporais sob operações de escrita e leitura. Os resultados das análises mostraram que o Neo4j apresentou o maior tempo de execução nas operações de escrita entre os três SGBDs avaliados, no entanto, em operações de leitura o Neo4j demonstrou um desempenho superior, uma vez que o tempo de leitura é relativamente pouco afetado pelo aumento do volume de dados, aproximando-se de uma complexidade de tempo $O(\log n)$.

No trabalho de Perçuku, Minkovska e Stoyanova (2017), os autores modelaram um banco de dados Neo4j para o monitoramento e processamento de *Big Data* em subestações de redes de transmissão de energia. Dado que a rede elétrica é composta por diversas fontes de geração e fluxos de energia em múltiplas direções, as características de direcionalidade e escalabilidade do Neo4j mostraram-se vantajosas para o processamento desses dados, especialmente em cenários de análise de *Big Data*, onde há um volume elevado de dados em um curto intervalo de tempo. Além disso, a velocidade de consulta do Neo4j é uma vantagem significativa

em comparação com bancos de dados relacionais, nos quais cada consulta requer um *lookup* em outra tabela. No Neo4j, os relacionamentos entre entidades são definidos no momento da criação, o que proporciona maior desempenho ao banco de dados. Por esses motivos, os autores concluem que a implementação do Neo4j como banco de dados possibilitou uma boa performance à operação, além de facilitar a evolução do sistema, dada a sua escalabilidade.

No estudo de Karim, Boulmakoul e Zeitouni (2021), os autores propuseram um modelo baseado em grafos para representar e analisar as trajetórias de pedestres por meio de dados coletados via GPS, com o objetivo de identificar possíveis riscos, como acidentes e congestionamentos. O Neo4j foi utilizado como banco de dados para armazenar as trajetórias dos pedestres, enquanto o MongoDB foi empregado no módulo de análise de trajetórias. Além disso, o Apache Kafka foi utilizado para comunicação entre o Neo4j e o serviço de coleta de dados geoespaciais. Apesar das dificuldades enfrentadas na coleta de dados, devido à alta densidade em ambientes urbanos e à baixa precisão frequente em ambientes internos, o estudo demonstra o potencial do uso de bancos de dados orientados a grafos em conjunto com outras tecnologias para a análise da mobilidade urbana e segurança de pedestres.

No trabalho de Rosa *et al.* (2022), foi apresentado o AgroGraph, uma ferramenta web para a plotagem de pontos amostrais e interpolação de dados utilizando o método geoestatístico de krigagem. O sistema foi desenvolvido com Neo4j na camada de dados, scripts em R para os métodos estatísticos e PHP na camada de aplicação, além de HTML, CSS e Javascript na camada de cliente. A API do Google Maps foi utilizada para manipulação dos mapas. O Neo4j foi escolhido como SGBD devido à sua facilidade em manter e manipular dados de medições realizadas em diferentes datas, possibilitando o gerenciamento eficiente de dados históricos.

No trabalho de Cunjin *et al.* (2022), foi modelado um SGBD Neo4j para a representação das trajetórias dos flutuadores autônomos Argo, que têm suas rotas determinadas pelas correntes oceânicas e são utilizados na coleta de dados oceanográficos em diversas profundidades. Segundo os autores, a implementação desse modelo no Neo4j possibilitou uma representação mais detalhada e precisa das trajetórias, além de melhorias significativas na eficiência das consultas e no desempenho geral do sistema. Os tempos de resposta das consultas foram significativamente menores em comparação ao modelo relacional, mesmo com o aumento do volume de dados. Isso se deve, em parte, ao fato de que há muitos desses flutuadores em operação, e eles podem permanecer ativos por vários anos, gerando grandes volumes de informações.

No trabalho de Kuhn, Kaminski e Franke (2022), os autores propuseram o uso do Neo4j em conjunto com o serviço de mensageria Apache Kafka na manufatura inteligente, com o objetivo de integrar dados estáticos e dinâmicos da linha de produção, a fim de rastrear e monitorar as operações. A arquitetura foi implementada em uma linha de produção automotiva, especificamente na fabricação de sistemas elétricos e eletrônicos. No modelo implementado, os dados foram divididos entre estáticos e dinâmicos, sendo os dados dinâmicos relacionados ao comportamento das peças e artefatos monitorados. Os autores concluíram que o uso dessa arquitetura e do modelo de dados foi altamente eficaz, pois permitiu uma maior rastreabilidade

dos dados dinâmicos associados às peças. Além disso, o uso de *procedures* do Neo4j, como o *spanningTree*, possibilitou o rastreamento de mudanças de estado em tempo real.

Em sua pesquisa, Rajšp e Fister (2023), utilizaram o SGBD gráfico Neo4j para geração de um conjunto de rotas de ciclismo na Eslovênia, integrando-o com outras ferramentas para o detalhamento das rotas, como o MongoDB, Overpass API e OpenElevation API. O modelo forneceu informações como distância, tipo de estrada e elevação. Esses dados foram disponibilizados com o objetivo de apoiar o desenvolvimento e a otimização de algoritmos, bem como a criação de aplicações voltadas à geração de rotas para o treinamento de ciclistas. A escolha do Neo4j se deu, principalmente, pela direcionalidade dos dados, uma vez que, em rotas para ciclistas, a direção impacta diretamente nas elevações de terreno encontradas.

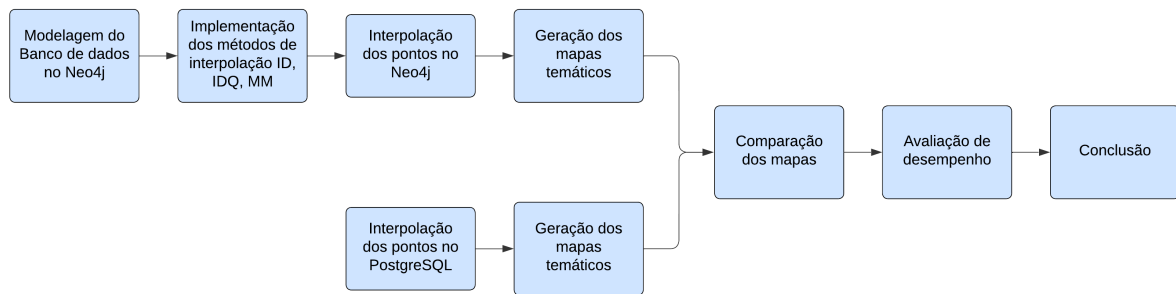
No trabalho de Germinian e S.T. (2023), foi realizado um estudo para o processamento de dados em um SGBD relacional (PostgreSQL) utilizando a extensão PostGIS para manipulação de dados geoespaciais, os quais estavam armazenados em um SGBD NoSQL Neo4j. O autor propôs a transformação de consultas Cypher (linguagem utilizada pelo Neo4j) em consultas SQL. Para isso, foi utilizado o conceito de Árvores de Sintaxe Abstrata (AST) para tornar as consultas SQL mais genéricas, permitindo a construção de consultas Cypher. Com exceção de alguns operadores SQL, como IN, MINUS e INTERSECTS, que se mostraram inviáveis para a transformação, a solução proposta foi considerada satisfatória. No entanto, o estudo não avaliou o desempenho das operações, focando apenas no alcance do objetivo final.

No estudo de Monteiro, Sá e Bernardino (2023), foram comparados quatro dos principais SGBDs orientados a grafos disponíveis no mercado: Neo4J, JanusGraph, Nebula Graph e TigerGraph. O objetivo foi identificar as vantagens e desvantagens de cada banco, baseando-se em métricas como tempo de execução de consultas, tempo de carregamento de nós e uso de recursos computacionais (memória e CPU). Os autores utilizaram a ferramenta LDBC SNB para realizar a análise comparativa entre os bancos avaliados. As conclusões do estudo indicaram que o Neo4j ficou em primeiro lugar em todas as análises realizadas. Além disso, os autores destacaram a capacidade do Neo4j de lidar com grandes volumes de dados.

3 MATERIAIS E MÉTODOS

Este capítulo descreve os materiais e métodos utilizados para o desenvolvimento do trabalho, bem como as etapas necessárias para sua conclusão. O fluxograma apresentado na Figura 6 foi desenvolvido para facilitar a compreensão das etapas descritas neste capítulo.

Figura 6 – Etapas do Estudo: Interpolação de Dados em PostgreSQL e Neo4j



Fonte: Autoria própria (2025).

3.1 Ambiente e ferramentas utilizadas

Os testes foram realizados em um computador com as seguintes configurações:

- Processador: Intel Core i5-10400F, com clock máximo de 4,3 GHz;
- Memória RAM: 16 GB;
- Armazenamento: SSD M.2 de 480 GB.

Como ambiente de desenvolvimento para as *procedures* do Neo4j, utilizou-se o IntelliJ IDEA. A montagem e execução das consultas na linguagem Cypher, bem como o desenvolvimento dos scripts em R, foram realizados no Visual Studio Code.

A execução das *procedures* implementadas no PostgreSQL foi feita por meio das ferramentas PGAdmin e DBeaver, utilizadas em conjunto para a administração e os testes no banco de dados.

Por fim, a geração dos mapas temáticos apresentados foi realizada com o uso da ferramenta AgDataBox-Map.

3.2 Modelagem de dados no Neo4j

Como o Neo4j é um banco de dados orientado a grafos, a modelagem de dados foi baseada na representação das informações em nós, relacionamentos e propriedades, em vez de tabelas e linhas, como ocorre em bancos de dados relacionais. Nesta etapa, é essencial compreender o domínio do problema, identificar os nós e os relacionamentos, além de considerar como as consultas serão realizadas.

3.3 Implementação dos métodos de interpolação ID, IDQ e MM no Neo4j

Novos conjuntos de dados foram gerados por cada método de interpolação utilizado, sendo eles: interpolação pelo inverso da distância e interpolação pelo inverso da distância elevada ao quadrado, ambos representados pela Equação 1, diferindo apenas pelo valor do expoente. Além disso, foi implementado o método de interpolação por média móvel (Equação 2).

$$Z = \frac{\sum_{i=1}^n \frac{Z_i}{d_i^p}}{\sum_{i=1}^n \frac{1}{d_i^p}} \quad (1)$$

onde:

- Z - é o ponto a ser interpolado;
- Z_i - é o valor conhecido no ponto i ;
- d_i - é a distância entre o ponto a ser interpolado e o dado amostrado;
- p - é a potência a qual a distância será elevada.

$$Z(x) = \frac{1}{n} \sum_{i=1}^n Z_i \quad (2)$$

onde:

- $Z(x)$ é o ponto a ser interpolado;
- n é o número de pontos vizinhos considerados;
- Z_i é o valor da variável medida no i -ésimo ponto.

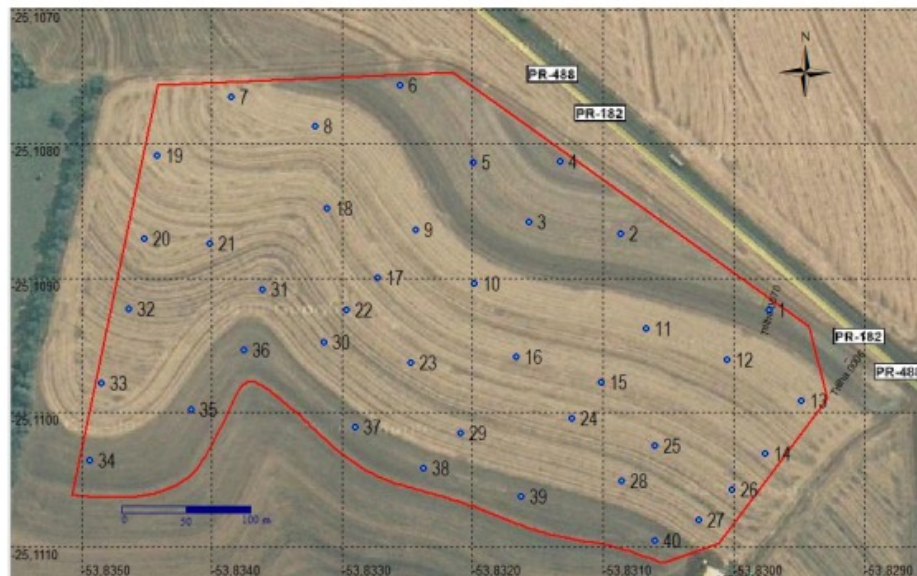
3.4 Interpolação dos dados físicos e químicos

Os dados utilizados nos testes de interpolação consistem em atributos físicos e químicos coletados do solo e do cultivo analisado, incluindo informações como concentração de nutrientes

e produtividade em pontos específicos. Esses dados foram obtidos em duas áreas distintas, localizadas nos municípios de Céu Azul-PR (Figura 7) e Serranópolis do Iguaçu-PR (Figura 8).

Para este estudo, foi utilizado o atributo zinco como base para a interpolação, escolhido de forma arbitrária, uma vez que a troca por outro atributo não comprometeria a validade das análises comparativas entre os métodos implementados.

Figura 7 – Pontos de coleta de amostras para análises físico-químicas do solo na área 1.



Fonte: Schenatto (2014).

Segundo Ferguson e Hergert (2009), a quantidade recomendada de pontos de coleta para o mapeamento das propriedades do solo é de aproximadamente 2,47 pontos por hectare, o que corrobora a escolha dos conjuntos de dados utilizados. Na área 1 (Figura 7), foram coletados 40 pontos amostrais, resultando em uma densidade de 2,58 pontos por hectare. Na área 2 (Figura 8), foram coletados 42 pontos amostrais, totalizando uma densidade média de 4,24 pontos por hectare.

Figura 8 – Pontos de coleta de amostras para análises físico-químicas do solo na área 2.



Fonte: Schenatto (2014).

Os dados coletados nas duas áreas foram modelados no Neo4j para representar as informações dos pontos de coleta, que serviram de base para os métodos de interpolação aplicados neste estudo.

3.5 Interpolação pelos métodos ID, IDQ e MM no PostgreSQL

Os métodos de interpolação implementados no banco de dados PostgreSQL utilizados neste trabalho foram desenvolvidos previamente no estudo realizado por Bazzi *et al.* (2019). Nesse estudo, foi elaborado um software para a definição de zonas de manejo que, entre suas funcionalidades, inclui métodos de interpolação.

As comparações descritas na seção 3.8 foram realizadas com base nas implementações apresentadas naquele estudo.

3.6 Análise de desempenho

Inicialmente, as análises de desempenho dos métodos interpoladores implementados nos dois ambientes (Neo4j e PostgreSQL) foram focadas na latência observada durante a execução. Como esses métodos realizam várias operações matemáticas em um grande conjunto de pontos, o tempo de execução tende a ser significativamente impactado.

3.7 Geração dos mapas temáticos

Após a aplicação dos métodos de interpolação, foram gerados mapas temáticos correspondentes a cada método utilizado. Esses mapas são frequentemente empregados como ferramentas para melhorar a visualização dos dados, facilitando a identificação de padrões na área analisada. Os mapas gerados foram utilizados como base para a comparação entre os métodos de interpolação implementados no PostgreSQL e no Neo4j, conforme descrito na seção 3.8.

Além das implementações realizadas diretamente nos SGBDs Neo4j e PostgreSQL, este trabalho também contempla a realização de interpolações utilizando a linguagem de programação R. Essa etapa adicional visa fornecer uma referência externa e estatisticamente robusta para a avaliação da acurácia e da qualidade dos resultados gerados pelos bancos de dados.

O R foi escolhido devido à sua ampla utilização em análises estatísticas e geoespaciais, além da disponibilidade de pacotes especializados, como `gstat`, `sp`, `sf` e `raster`, que permitem a execução eficiente de métodos de interpolação como Inverso da distância elevado a uma potência (IDW), Krigagem, entre outros. As interpolações geradas no R foram utilizadas como baseline comparativo, permitindo analisar a proximidade dos resultados obtidos via bancos de dados (Neo4j e PostgreSQL) em termos de distribuição espacial e valores interpolados.

3.8 Comparação dos mapas

Os mapas gerados pelos métodos implementados no Neo4j foram comparados com os gerados por meio dos métodos de interpolação já implementados em ambiente PostgreSQL. A comparação entre os mapas produzidos pelo PostgreSQL e pelo Neo4j foi realizada para avaliar o índice de concordância dos dados interpolados pelos métodos ID, Inverso da distância ao quadrado (IDQ) e Médias móveis (MM). Para essa análise, foi utilizado o índice de concordância Kappa descrito na Equação 3.

$$K = \frac{\{n \sum_{i=1}^r X_{ii} - \sum_{i=1}^r (X_{i+} * X_{+i})\}}{\{n^2 - \sum_{i=1}^r (X_{i+} * X_{+i})\}} \quad (3)$$

onde:

- K - é o índice de concordância Kappa;
- n - é o número de pontos amostrais;
- r - é o número de classes da matriz de erro;
- X_{ii} - é o número de combinações na diagonal;
- X_{i+} - é o total de observações na linha i ;
- X_{+i} - é o total de observações na coluna i .

O índice de concordância Kappa varia de 0 a 1, sendo que valores mais próximos de 1 indicam um maior nível de concordância, assim como pode ser observado na tabela 1.

Tabela 1 – Interpretação do índice Kappa

Valor de Kappa	Interpretação
< 0,00	Pobre
0,00-0,20	Leve
0,21-0,40	Razoável
0,41-0,60	Moderada
0,61-0,80	Substancial
0,81-1,00	Quase Perfeita

Fonte: Adaptado de Landis e Koch (1977).

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta discussões sobre os resultados obtidos durante o desenvolvimento das *procedures* no Neo4j, assim como a comparação com os métodos já desenvolvidos em ambiente PostgreSQL.

4.1 Implementação dos Métodos no Neo4j

Foram implementados no SGBD Neo4j utilizando a linguagem Java dois métodos de interpolação, sendo eles: Interpolação por IDW e Interpolação por MM. Ambos realizam a interpolação de múltiplos pontos dentro de um polígono pré-definido, utilizando como critério a distância entre os pontos amostrados e os locais onde se deseja estimar os valores.

Foi criada uma classe `InterpolacaoMMBulk` que implementa o método de interpolação por médias móveis, e uma classe `InterpolacaoIDWBulk` que é capaz de lidar com qualquer valor de peso na interpolação pelo inverso da distância — incluindo, neste trabalho, os casos do inverso da distância e do inverso da distância ao quadrado.

Como o Neo4j, até a versão utilizada neste estudo, não oferece suporte a funções geométricas mais complexas, como linhas e polígonos, foi adotada a biblioteca Java Topology Suite (JTS) — uma biblioteca Java voltada à manipulação de geometrias — para tratar os pontos de contorno na forma de polígonos. Essa solução facilitou a implementação dos métodos de interpolação abordados.

Para dar suporte ao desenvolvimento dessas classes, foram implementadas estruturas auxiliares, como `SourcePoint` e `ResultadoInterpolacaoBulk`, descritas nas Listagens 1 e 2, respectivamente.

A classe `SourcePoint` foi desenvolvida com o objetivo de facilitar o manuseio dos pontos amostrais no código. Já a classe `ResultadoInterpolacaoBulk` tem como finalidade formatar os resultados da interpolação, incluindo as coordenadas e os valores calculados.

Listagem 1 – Classe SourcePoint

```

1 public class SourcePoint {
2     private final org.neo4j.graphdb.spatial.Point location;
3     private final double valor;
4
5     public SourcePoint(org.neo4j.graphdb.spatial.Point location ,
6                         double valor) {
7         this.location = location;
8         this.valor = valor;
9     }
10
11     public Point getLocation() {
12         return location;
13     }
14
15     public double getValor() {
16         return valor;
17     }
18 }

```

Fonte: Autoria própria (2025).

Listagem 2 – Classe ResultadoInterpolacaoBulk

```

1 public class ResultadoInterpolacaoBulk {
2     public double latitude;
3     public double longitude;
4     public double valor;
5
6     public ResultadoInterpolacaoBulk(double latitude , double longitude ,
7                                       double valor) {
8         this.latitude = latitude;
9         this.longitude = longitude;
10        this.valor = valor;
11    }
12
13    public double getLatitude() {
14        return latitude;
15    }
16
17    public double getLongitude() {
18        return longitude;
19    }
20
21    public double getValor() {
22        return valor;
23    }
24 }

```

Fonte: Autoria própria (2025).

4.2 Implementação da classe InterpolacaoIDWBulk

O método `interpolaIDWBulk`, da classe `InterpolacaoIDWBulk`, recebe como parâmetros:

- um polígono no formato `GeoJSON`, que define os pontos do contorno da área a ser interpolada;
- o atributo a ser interpolado;
- a distância, em metros, entre os pontos da grade — quanto menor essa distância, maior será a densidade de pontos interpolados;
- a quantidade de pontos vizinhos a serem considerados — adotando-se, neste trabalho, 8 vizinhos para o método ID e 10 para o método de médias móveis;
- o peso da interpolação pelo inverso da distância — sendo utilizados os valores $p = 1$ e $p = 2$, correspondentes ao ID e ao IDQ, respectivamente.

A assinatura do método pode ser observada na Listagem 3.

Listagem 3 – Assinatura do método `interpolarIDWBulk`

```

1  @Procedure(name = "customprocedures.idwbulk", mode = Mode.READ)
2  public Stream<ResultadoInterpolacaoBulk> interpolarIDWBulk(
3      @Name("polygon") Map<String, Object> geoJsonPolygon,
4      @Name("atributo") String atributo,
5      @Name("metros") double metros,
6      @Name(value = "k", defaultValue = "8") long k,
7      @Name(value = "p", defaultValue = "2") double p) {
8  }
```

Fonte: Autoria própria (2025).

Na Listagem 4, são apresentadas as principais etapas da implementação do método `interpolarIDWBulk`.

- **Passo 1:** o polígono que define o contorno da área, fornecido no formato `GeoJSON`, é convertido para o tipo `Polygon` da biblioteca JTS. Isso facilita sua manipulação e permite o uso direto em métodos de cálculo de interseção entre os pontos da grade e o contorno da área.
- **Passo 2:** é gerada a grade de coordenadas que servirá como base para interpolação, respeitando os limites do contorno da área.
- **Passo 3:** todos os pontos amostrais contidos na área de interesse são carregados, para que possam ser utilizados no cálculo dos valores interpolados.

- **Passo 4:** são realizadas iterações sobre os pontos da grade, aplicando-se o método IDW por meio de `parallelStream`, com o objetivo de executar os cálculos de forma paralela, aumentando a eficiência da interpolação.

Listagem 4 – Principais passos do método `interpolarIDWBulk`

```

1 // 1. Converter GeoJSON para Polígono (JTS)
2 Polygon polygon = parseGeoJsonToJtsPolygon(geoJsonPolygon);
3
4 // 2. Gerar grade de pontos dentro do polígono
5 List<Coordinate> grade = gerarGrade(polygon, metros);
6
7 // 3. Carregar todos os pontos fonte com o atributo
8 // (dentro do bounding box do polígono)
9 List<SourcePoint> pontosFonte = carregarPontosFonte(atributo,
10 polygon.getEnvelopeInternal());
11
12 // 4. Calcular IDW para cada ponto da grade
13 return grade.parallelStream()
14     .map(pontoGrade -> {
15         Point pontoJts = GEOMETRY_FACTORY.createPoint(pontoGrade);
16         if (!polygon.contains(pontoJts)) return null;
17
18         List<SourcePoint> vizinhos =
19             encontrarVizinhosProximos(pontoGrade, pontosFonte, k);
20
21         // Passar o pontoGrade para calcularIDW
22         double valor = calcularIDW(pontoGrade, vizinhos, p);
23         return new ResultadoInterpolacaoBulk(pontoGrade.y,
24             pontoGrade.x, valor);
25     })
26     .filter(Objects::nonNull);

```

Fonte: Autoria própria (2025).

A parte mais específica deste método é o `calcularIDW`, uma vez que ambas as *procedures* implementadas compartilham diversos métodos auxiliares para a realização das interpolações. A implementação do método pode ser consultada na Listagem 5.

Listagem 5 – Implementação do método calcularIDW

```

1  private double calcularIDW (Coordinate pontoGrade,
2      List<SourcePoint> vizinhos, double p) {
3      double somaPonderada = 0.0;
4      double somaPesos = 0.0;
5
6      for (SourcePoint vizinho : vizinhos) {
7          double distancia = calcularDistancia (pontoGrade,
8              vizinho.getLocation());
9          if (distancia == 0) return vizinho.getValor();
10         double peso = 1 / Math.pow(distancia, p);
11         somaPonderada += vizinho.getValor() * peso;
12         somaPesos += peso;
13     }
14
15     return somaPesos == 0 ? Double.NaN : somaPonderada / somaPesos;
16 }

```

Fonte: Autoria própria (2025).

4.3 Implementação da classe InterpolacaoMMBulk

A assinatura do método `interpolarmmbulk` é semelhante à do método `interpolardwbulk`, com a exceção de que, neste caso, não há o parâmetro p , uma vez que a interpolação por médias móveis não atribui pesos à distância. A assinatura completa do método pode ser observada na Listagem 6.

Listagem 6 – Assinatura do método interpolarmmbulk

```

1  @Procedure(name = "customprocedures.mmbulk", mode = Mode.READ)
2  public Stream<ResultadoInterpolacaoBulk> interpolarmmbulk(
3      @Name("polygon") Map<String, Object> geoJsonPolygon,
4      @Name("atributo") String atributo,
5      @Name("metros") double metros,
6      @Name(value = "k", defaultValue = "10") long k) {
7  }

```

Fonte: Autoria própria (2025).

Os passos seguidos pelo método `interpolarmmbulk` são praticamente os mesmos do método `interpolardwbulk`, diferenciando-se apenas na etapa de processamento paralelo. Nesse caso, o método utilizado é o `calcularMediaMovel`, o qual pode ser conferido na Listagem 7.

Listagem 7 – Implementação do método calcularMediaMovel

```

1  private double calcularMediaMovel(List<SourcePoint> vizinhos) {
2      if (vizinhos.isEmpty()) return Double.NaN;
3
4      double soma = 0.0;
5      for (SourcePoint vizinho : vizinhos) {
6          soma += vizinho.getValor();
7      }
8      return soma / vizinhos.size();
9  }

```

Fonte: Autoria própria (2025).

4.4 Interpolação dos dados

Foram realizadas, ao todo, 60 interpolações para os testes dos métodos, sendo 30 no ambiente PostgreSQL, utilizando os métodos já implementados, e 30 no ambiente Neo4j, com base na implementação proposta por este estudo.

As interpolações foram distribuídas entre as duas áreas apresentadas na Seção 3.4: localizadas em Céu Azul (área 1) e Serranópolis do Iguaçu (área 2). Além disso, cada ambiente contemplou os três métodos de interpolação – ID, IDQ e MM –, totalizando 15 interpolações por área em cada ambiente. As execuções foram realizadas com diferentes tamanhos de grade: 20x20, 15x15, 10x10, 5x5 e 2x2 metros, para avaliar o comportamento dos métodos quando se aumenta o número de pontos.

Ao gerar os pontos interpolados nas implementações em PostgreSQL e Neo4j, observou-se que a quantidade de pontos produzidos no Neo4j era ligeiramente maior.

Ao analisar os mapas gerados, verificou-se que os métodos implementados em PostgreSQL consideram como válidos os pontos que tangenciam o contorno da área, tratando-os como internos. Por outro lado, a implementação em Neo4j inclui apenas os pontos cuja maior parte está efetivamente dentro da área delimitada, conforme ilustrado na Figura 9.

Figura 9 – Pontos interpolados ID 10x10 Neo4j x PostgreSQL na área 1.



Fonte: Autoria própria (2025).

Devido a essa diferença dos pontos próximos ao contorno, nos mapas com grades maiores em que há menor densidade de pontos a discrepância entre os mapas acabou sendo mais significativa.

Se a implementação em PostgreSQL considera como interno um ponto que o Neo4j descarta, e os pontos interpolados estão mais espaçados, isso impacta negativamente na concordância entre os mapas gerados. Como consequência, observam-se valores mais baixos para o índice Kappa, como observado na Tabela 2.

Tabela 2 – Relação tamanho da grade x concordância entre os mapas - ID Área 1

Mapa	Kappa
ID 2x2 Céu Azul	0,94
ID 5x5 Céu Azul	0,88
ID 10x10 Céu Azul	0,79
ID 20x20 Céu Azul	0,78

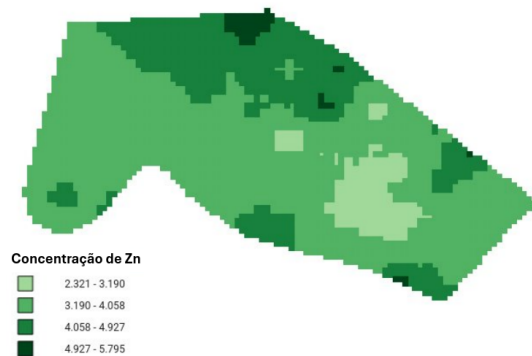
Fonte: Autoria própria (2025).

O mapa interpolado com grade 2x2 apresentou o maior índice de concordância, com um valor de Kappa igual a 0,94, indicando forte similaridade entre os mapas comparados. À medida que a grade se torna mais espaçada, observa-se uma queda progressiva nos índices de concordância, com o valor de Kappa reduzindo de 0,94 (2x2) para 0,78 (20x20).

4.5 Comparação entre os mapas gerados

Entre os mapas gerados a partir da interpolação dos dados, foram seleccionados seis de cada área – três produzidos com pontos interpolados no Neo4j e três no PostgreSQL – para a análise de similaridade entre as implementações dos métodos, utilizando o índice Kappa de concordância. Os mapas das áreas 1 e 2 estão apresentados nas Figuras 10 e 11 respectivamente.

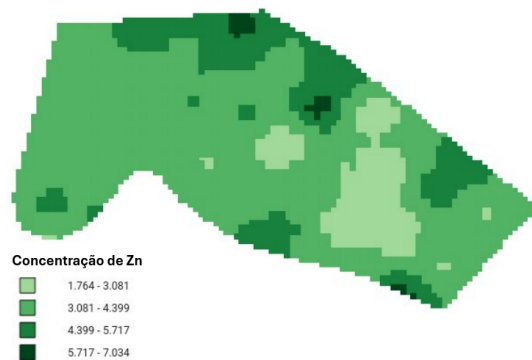
Figura 10 – Mapas gerados da área 1 com grade 10x10.



(a) Área 1 - Neo4j ID grade 10x10



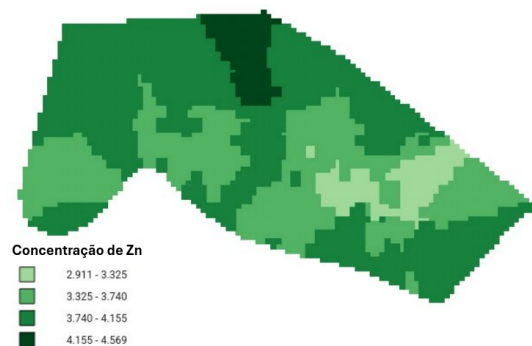
(b) Área 1 - PostgreSQL ID grade 10x10



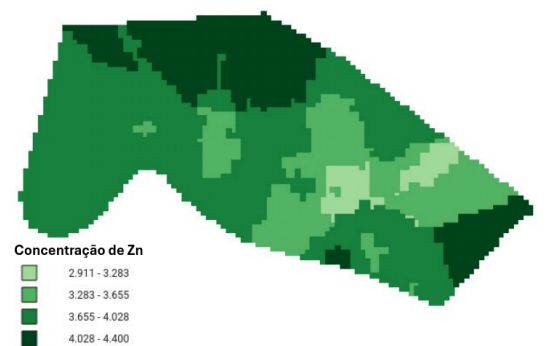
(c) Área 1 - Neo4j IDQ grade 10x10



(d) Área 1 - PostgreSQL IDQ grade 10x10



(e) Área 1 - Neo4j MM grade 10x10



(f) Área 1 - PostgreSQL MM grade 10x10

Fonte: Autoria própria (2025).

Para a área 1, foram executados os três métodos implementados — ID, IDQ e MM — tanto no Neo4j quanto no PostgreSQL.

No mapa interpolado com o método ID, os valores de concentração de zinco na implementação no Neo4j variaram de 2,32 a 5,79, enquanto no PostgreSQL variaram de 2,26 a 5,59, apresentando uma redução do valor mínimo e uma leve diminuição do valor máximo.

Em relação ao método IDQ, os valores no Neo4j variaram de 1,76 a 7,03, e no PostgreSQL, de 1,75 a 7,25.

Por fim, para o método MM, os valores no Neo4j variaram de 2,91 a 4,57, enquanto no PostgreSQL variaram de 2,91 a 4,40. Embora essa tenha sido uma das menores diferenças entre os valores mínimos e máximos, foi encontrado o menor índice de concordância, segundo o índice Kappa, conforme apresentado na Tabela 3.

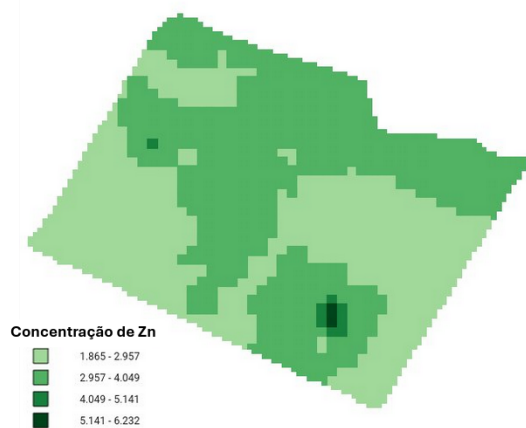
Comparando os mapas gerados pelas implementações nos dois ambientes — Neo4j e PostgreSQL — foram observados os seguintes níveis de concordância:

Tabela 3 – Concordância entre os mapas de amostras interpoladas na área 1 - grade 10x10

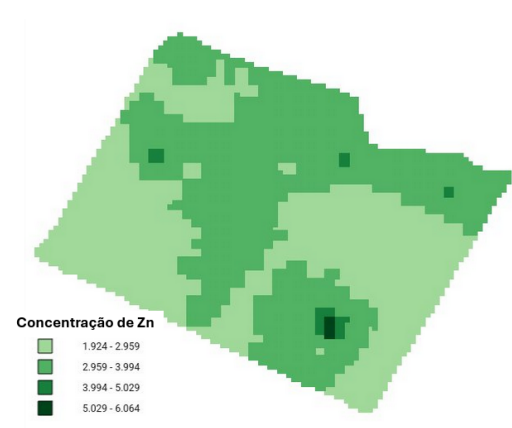
Mapa	Kappa	Interpretação
ID	0,79	Substancial
IDQ	0,84	Quase Perfeita
MM	0,39	Razoável

Fonte: Autoria própria (2025).

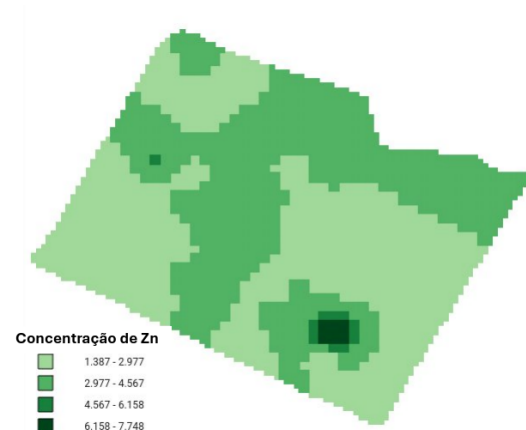
Figura 11 – Mapas gerados da área 2 com grade 10x10.



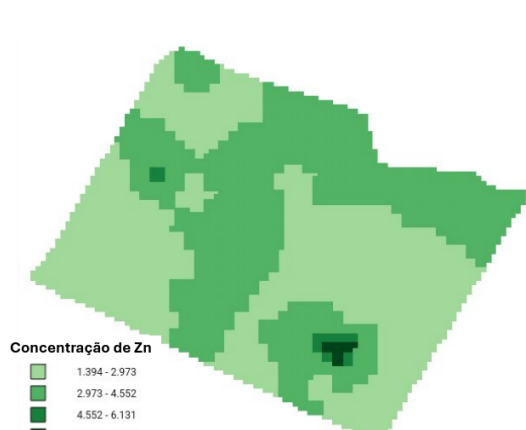
(a) Área 2 - Neo4j ID grade 10x10



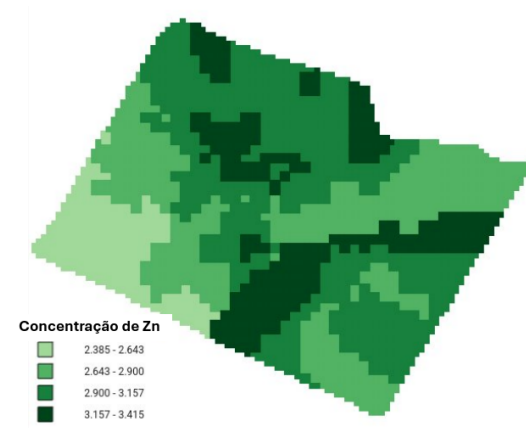
(b) Área 2 - PostgreSQL ID grade 10x10



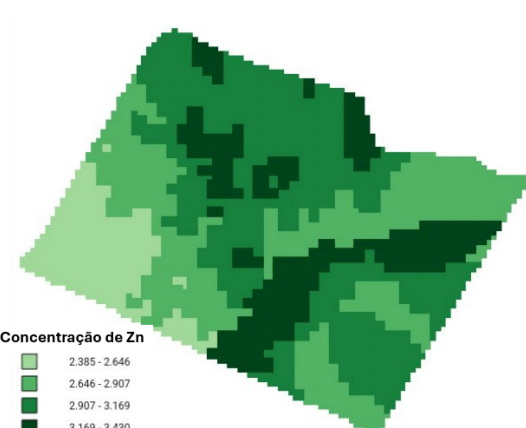
(c) Área 2 - Neo4j IDQ grade 10x10



(d) Área 2 - PostgreSQL IDQ grade 10x10



(e) Área 2 - Neo4j MM grade 10x10



(f) Área 2 - PostgreSQL MM grade 10x10

Fonte: Autoria própria (2025).

Assim como na área 1, foram executados na área 2 os três métodos de interpolação – ID, IDQ e MM – tanto no Neo4j quanto no PostgreSQL.

Nos mapas gerados pelas interpolações na área 2, também foram observadas variações nos valores de concentração de zinco. Para o método ID, os valores no Neo4j variaram de 1,86 a 6,23, enquanto no PostgreSQL variaram de 1,92 a 6,06.

Para o método IDQ, os valores no Neo4j variaram de 1,39 a 7,75, enquanto no PostgreSQL variaram de 1,39 a 7,71.

Por fim, para o método MM, os valores no Neo4j variaram de 2,38 a 3,41, enquanto no PostgreSQL variaram de 2,38 a 3,43.

Os níveis de concordância observados estão apresentados na Tabela 4.

Tabela 4 – Concordância entre os mapas de amostras interpoladas na área 2 — grade 10×10

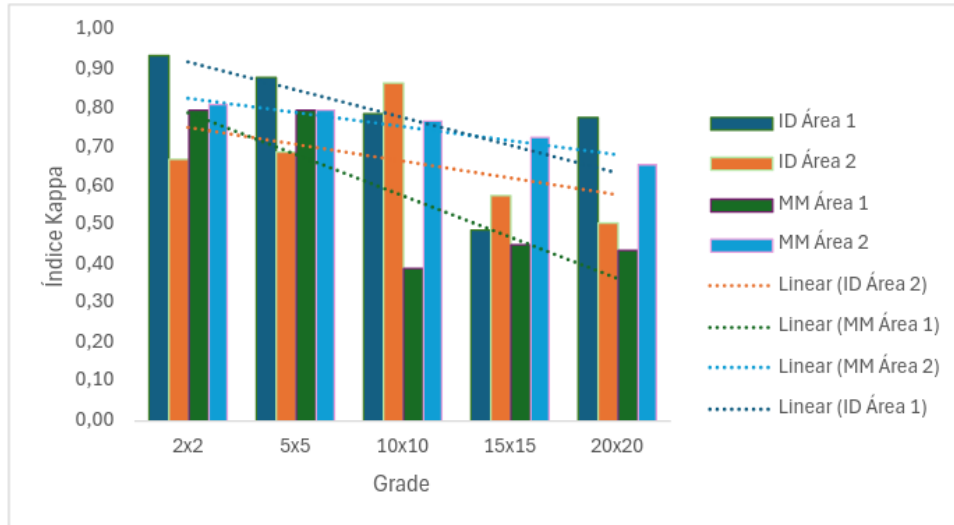
Mapa	Kappa	Interpretação
ID	0,87	Quase perfeita
IDQ	0,88	Quase perfeita
MM	0,77	Substancial

Fonte: Autoria própria (2025).

Comparando as Tabelas 3 e 4, observa-se que os níveis de concordância foram superiores na área 2. Um dos fatores que pode explicar essa diferença é a maior densidade amostral da área 2, que, conforme mencionado na Seção 3.4, apresenta 4,24 pontos por hectare, enquanto a área 1 possui 2,58 pontos por hectare.

Essa diferença sugere que áreas com maior densidade de pontos amostrais tendem a gerar mapas com maior concordância entre os métodos, tanto no Neo4j quanto no PostgreSQL. Tal tendência também pôde ser observada ao variar o tamanho das grades de interpolação, como ilustra a Figura 12. No entanto, esse comportamento foi mais evidente para o método de Médias Móveis, em que o índice Kappa aumentou com o uso de grades mais densas. Já no método ID, a concordância foi mais alta nas grades intermediárias, sem um padrão crescente claro nas grades mais densas.

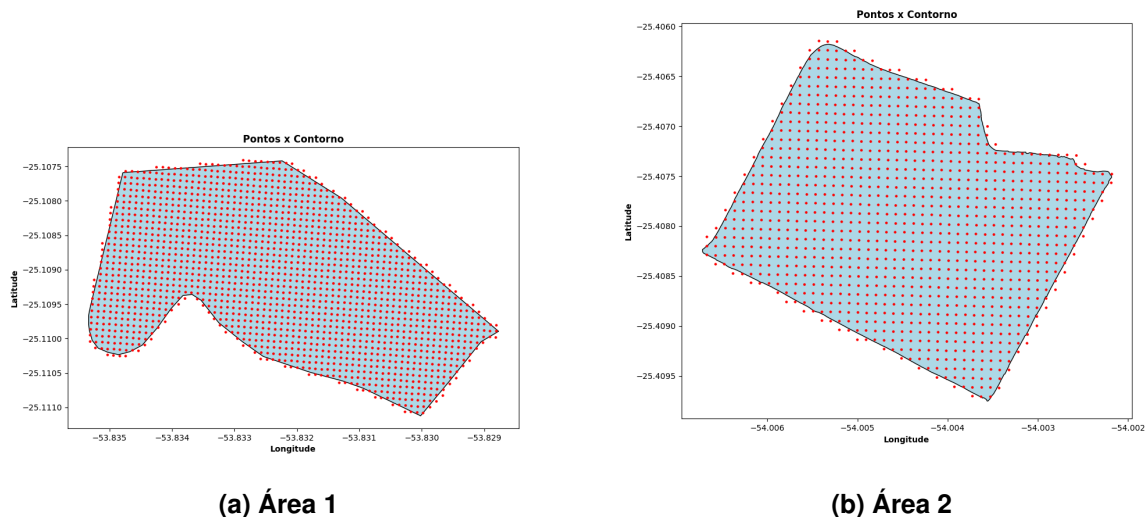
Figura 12 – Relação entre a densidade de pontos na grade de interpolação e o índice de concordância.



Fonte: Autoria própria (2025).

Outro fator relevante é o número de pontos interpolados – pelos métodos implementados em ambiente PostgreSQL – fora do polígono que delimita o contorno da área. Na área 1, o método ID com grade 10×10 plotou 112 pontos fora do contorno (Figura 13a), enquanto na área 2 foram 81 pontos (Figura 13b), como ilustrado na Figura 11.

Figura 13 – Comparação dos pontos interpolados fora do contorno nas áreas 1 e 2, utilizando o método ID no PostgreSQL.



Fonte: Autoria própria (2025).

4.6 Interpolação com R

Observadas as diferenças entre os mapas gerados pelos métodos implementados nos ambientes Neo4j e PostgreSQL, foi proposta a implementação dos métodos ID, IDQ e MM na linguagem R.

Embora o R não seja um SGBD, e portanto não seja válida a comparação direta com os outros dois SGBDs analisados, sua utilização serve ao propósito de comparar os mapas gerados, buscando esclarecer as diferenças observadas entre os resultados obtidos nas implementações anteriores.

Comparando as Tabelas 5 e 6, é possível observar que, na área 1, os dados interpolados em R apresentaram maior concordância com os resultados do Neo4j do que com os do PostgreSQL.

Esse comportamento foi especialmente evidente no método de médias móveis, no qual o índice de concordância Kappa entre R e Neo4j foi de 0,72, enquanto, entre R e PostgreSQL, o mesmo índice foi de apenas 0,26.

Tabela 5 – Concordância entre os mapas gerados R x Neo4j na área 1 — grade 10×10

Mapa	Kappa	Interpretação
ID	0,90	Quase perfeita
IDQ	0,88	Quase perfeita
MM	0,72	Substancial

Fonte: Autoria própria (2025).

Tabela 6 – Concordância entre os mapas gerados R x PostgreSQL na área 1 — grade 10×10

Mapa	Kappa	Interpretação
ID	0,78	Substancial
IDQ	0,87	Quase perfeita
MM	0,26	Razoável

Fonte: Autoria própria (2025).

Na área 2, comparando as Tabelas 7 e 8, embora a diferença tenha sido mais sutil em relação à observada na área 1, também se verificou maior nível de concordância entre os resultados obtidos no R e no Neo4j do que entre R e PostgreSQL.

Tabela 7 – Concordância entre os mapas gerados R x Neo4j na área 2 — grade 10×10

Mapa	Kappa	Interpretação
ID	0,83	Quase perfeita
IDQ	0,88	Quase perfeita
MM	0,85	Quase perfeita

Fonte: Autoria própria (2025).

Tabela 8 – Concordância entre os mapas gerados R x PostgreSQL na área 2 — grade 10×10

Mapa	Kappa	Interpretação
ID	0,83	Quase perfeita
IDQ	0,87	Quase perfeita
MM	0,77	Substancial

Fonte: Autoria própria (2025).

4.7 Análise de tempo de consulta

Embora o tempo de execução dos métodos em R não seja considerado parte da comparação entre os SGBDs analisados (Neo4j e PostgreSQL), esses tempos foram registrados com o objetivo de disponibilizar dados adicionais sobre os mesmos métodos interpoladores, ainda que implementados em uma linguagem de programação e não diretamente em um SGBD.

Para o registro dos tempos de execução dos métodos no Neo4j, visando uma comparação justa, foi necessário realizar as medições com o banco em *cold start*. Isso porque o Neo4j armazena em cache informações sobre nós utilizados em instruções anteriores, o que poderia interferir nos tempos de execução subsequentes.

Inicialmente, foram tentadas estratégias de mitigação desse efeito, como a redução do tamanho disponível para cache e o preenchimento do cache com dados irrelevantes, distintos dos pontos amostrais utilizados nos testes. No entanto, após alguns testes, constatou-se que essas abordagens foram ineficazes, pois o Neo4j continuava a manter em cache dados relevantes para as execuções seguintes, comprometendo as medições.

Assim, a solução adotada foi reiniciar o banco de dados a cada execução, garantindo a limpeza completa do cache e, consequentemente, maior confiabilidade na medição dos tempos de execução.

Os tempos de execução para as grades 2x2, 5x5, 10x10, 15x15 e 20x20 estão expressos nas tabelas 9, 10, 11, 12 e 13 respectivamente.

Tabela 9 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 2x2

Área	PostgreSQL			NEO4J			R		
	ID	IDQ	MM	ID	IDQ	MM	ID	IDQ	MM
Céu Azul	14551,9	14898,4	13131,0	1773,0	823,0	1286,0	2110,0	2035,0	2193,0
Serranópolis	9166,4	9429,4	8449,7	1351,0	1298,0	1260,0	2477,0	2469,0	2575,0

Fonte: Autoria própria (2025).

Tabela 10 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 5x5

Área	PostgreSQL			NEO4J			R		
	ID	IDQ	MM	ID	IDQ	MM	ID	IDQ	MM
Céu Azul	2246,74	2225,63	2022,18	236,00	151,00	147,00	644,00	624,00	655,00
Serranópolis	1366,90	1400,86	1240,82	454,00	396,00	420,00	668,00	647,00	662,00

Fonte: Autoria própria (2025).

Tabela 11 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 10x10

Área	PostgreSQL			NEO4J			R		
	ID	IDQ	MM	ID	IDQ	MM	ID	IDQ	MM
Céu Azul	715,42	537,53	504,06	379,00	311,00	332,00	399,00	384,00	393,00
Serranópolis	434,94	394,47	351,82	589,00	280,00	239,00	408,00	395,00	404,00

Fonte: Autoria própria (2025).

Tabela 12 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 15x15

Área	PostgreSQL			NEO4J			R		
	ID	IDQ	MM	ID	IDQ	MM	ID	IDQ	MM
Céu Azul	285,69	271,89	227,80	100,00	108,00	91,00	358,00	347,00	356,00
Serranópolis	213,30	182,44	172,92	118,00	200,00	197,00	362,00	350,00	358,00

Fonte: Autoria própria (2025).

Tabela 13 – Tempo de execução (ms) das operações utilizando PostgreSQL e NEO4J em duas áreas - Grade 20x20

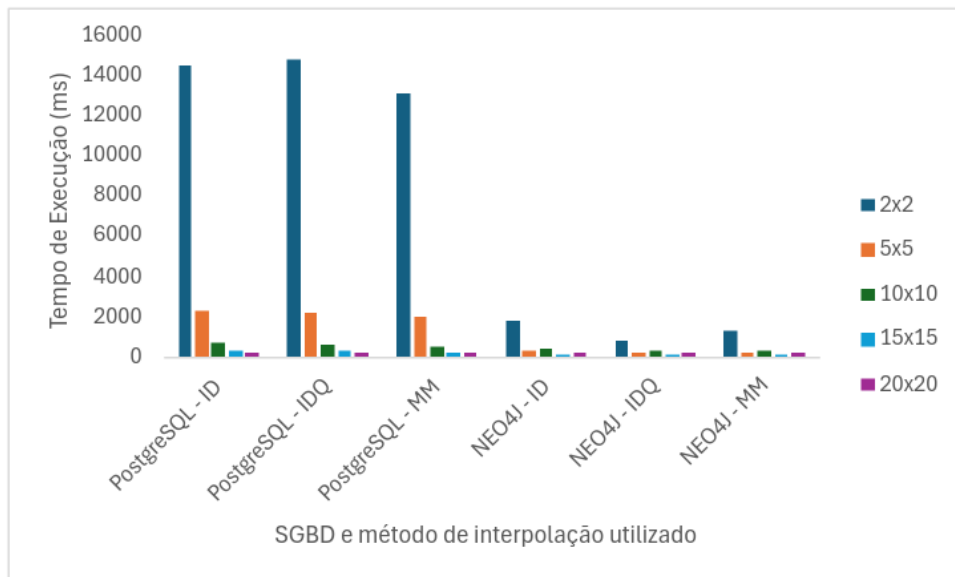
Área	PostgreSQL			NEO4J			R		
	ID	IDQ	MM	ID	IDQ	MM	ID	IDQ	MM
Céu Azul	173,94	155,81	143,23	146,00	142,00	147,00	383,00	371,00	379,00
Serranópolis	177,00	126,38	97,38	245,00	201,00	192,00	338,00	327,00	335,00

Fonte: Autoria própria (2025).

Ao analisar os tempos de execução do Neo4j e do PostgreSQL, considerando cinco diferentes tamanhos de grade, foi possível observar que, à medida que a distância entre os pontos da grade diminui – e, conseqüentemente, a densidade de pontos aumenta –, o tempo de execução tende a crescer em ambos os SGBDs.

Entretanto, o aumento do tempo de execução em função da redução do tamanho da grade foi consideravelmente mais acentuado no PostgreSQL do que no Neo4j, conforme ilustrado na Figura 14.

Figura 14 – Relação entre os métodos de interpolação no Neo4j e PostgreSQL e tempo de execução.



Fonte: Autoria própria (2025).

5 CONCLUSÃO

Embora o trabalho tenha levantado questões importantes, como a diferença no número de pontos interpolados entre os ambientes e o motivo pelo qual a implementação no PostgreSQL interpolou pontos fora do contorno, os objetivos inicialmente propostos foram sendo gradualmente atingidos ao longo do desenvolvimento.

O objetivo principal, que intitula este trabalho — verificar a viabilidade do uso do SGBD NoSQL Neo4j na interpolação de pontos amostrais —, demonstrou-se não apenas factível, mas também bastante promissor, especialmente considerando os tempos de execução dos métodos analisados.

Em comparação com os métodos implementados no PostgreSQL, o Neo4j apresentou desempenho significativamente superior, principalmente à medida que a densidade dos pontos interpolados aumentou, com ganho de performance — no que se refere à velocidade de execução — de até 17 vezes no caso mais extremo (grade 2×2 com o método IDQ).

Os resultados obtidos com a implementação em R corroboraram as diferenças observadas entre os dados interpolados com diferentes *stacks*. Embora os níveis de concordância entre os dados interpolados na implementação em R e no Neo4j tenham sido superiores em comparação aos resultados do PostgreSQL, ainda foram observadas diferenças nos dados resultantes. O motivo para tais discrepâncias não foi identificado, o que justifica futuras investigações sobre suas causas.

Diante dos resultados alcançados, este estudo não apenas evidenciou a viabilidade do uso do Neo4j para a interpolação de dados no contexto da agricultura de precisão, como também indicou caminhos promissores para futuras pesquisas na área. A identificação de discrepâncias entre diferentes implementações e o desempenho superior do Neo4j em cenários de alta densidade amostral reforçam a importância de aprofundar a investigação sobre a modelagem e o processamento de dados geoespaciais em bancos de dados orientados a grafos. Assim, este trabalho estabelece uma base sólida para o desenvolvimento de soluções mais eficientes e escaláveis, incentivando a continuidade dos estudos e o aprimoramento das técnicas aplicadas.

REFERÊNCIAS

- BASSOI, L. H. *et al.* Agricultura de precisão e agricultura digital. TECCOGS, n. 20, jul./dez., 2019., 2019. Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1121544/agricultura-de-precisao-e-agricultura-digital>. Acesso em: 16 nov. 2024.
- BAZZI, C. *et al.* A software for the delineation of crop management zones (sdum). **Australian Journal of Crop Science**, v. 13, p. 26–34, 01 2019. Disponível em: https://www.cropj.com/bazzi_13_1_2019_26_34.pdf. Acesso em: 06 dez. 2024.
- BORGES, L. F. *et al.* Software para web para criar mapas temáticos para agricultura de precisão. **Pesquisa Agropecuária Brasileira**, v. 55, n. X, mar. 2020. Disponível em: <https://apct.sede.embrapa.br/pab/article/view/26669>. Acesso em: 27 mai. 2025.
- BURROUGH, P. A.; MCDONNELL, R. A.; LLOYD, C. D. **Principles of geographical information systems**. [S.l.]: Oxford university press, 2015.
- CAMARGO, L. C. *et al.* Restrições de integridade e regras ativas em bancos de dados distribuídos. Florianópolis, SC, 2001. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/79996/182871.pdf?sequence=1&isAllowed=y>. Acesso em: 15 nov. 2024.
- CUNJIN, X. *et al.* An ocean current-oriented graph-based model for representing argo trajectories. **Computers Geosciences**, v. 166, p. 105143, 2022. ISSN 0098-3004. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0098300422000991>. Acesso em: 10 nov. 2024.
- DATE, C. J. **An Introduction to Database Systems**. 8. ed. Boston, MA, USA: Addison-Wesley, 2004. Comprehensive coverage of database design and relational systems. ISBN 9780321197849.
- EZENNE, G. *et al.* Current and potential capabilities of uas for crop water productivity in precision agriculture. **Agricultural Water Management**, v. 218, p. 158–164, 2019. ISSN 0378-3774. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378377418318298>. Acesso em: 24 nov. 2024.
- FERGUSON, R. B.; HERGERT, G. W. Soil sampling for precision agriculture. **Extension, EC**, v. 154, 2009. Disponível em: <https://extensionpubs.unl.edu/publication/ec154/2009/pdf/view/ec154-2009.pdf>. Acesso em: 20 fev. 2025.
- FRANCIS, N. *et al.* Cypher: An evolving query language for property graphs. *In*: **Proceedings of the 2018 international conference on management of data**. [s.n.], 2018. p. 1433–1445. Disponível em: <https://dl.acm.org/doi/10.1145/3183713.3190657>. Acesso em: 16 nov. 2024.
- GERMINIAN, J. F.; S.T., T. E. W. Utilizing postgis extension to process spatial data stored in neo4j database. *In*: **2023 IEEE International Conference on Data and Software Engineering (ICoDSE)**. [s.n.], 2023. p. 250–255. Disponível em: <https://ieeexplore.ieee.org/document/10291400>. Acesso em: 24 out. 2024.
- GONZÁLEZ-APARICIO, M. T. *et al.* Testing of transactional services in nosql key-value databases. **Future Generation Computer Systems**, v. 80, p. 384–399, 2018. ISSN 0167-739X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X1731467X>. Acesso em: 17 nov. 2024.

GREEN, A. *et al.* Updating graph databases with cypher. **Proc. VLDB Endow.**, VLDB Endowment, v. 12, n. 12, p. 2242–2254, ago. 2019. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/3352063.3352139>. Acesso em: 16 dez. 2024.

IDZIASZEK, P. *et al.* Identification of the condition of crops based on geospatial data embedded in graph databases. *In*: FALCO, C. M.; JIANG, X. (Ed.). **Ninth International Conference on Digital Image Processing (ICDIP 2017)**. SPIE, 2017. v. 10420, p. 1042006. Disponível em: <https://doi.org/10.1117/12.2281722>. Acesso em: 24 out. 2024.

KARIM, L.; BOULMAKOUL, A.; ZEITOUNI, K. From raw pedestrian trajectories to semantic graph structured model—towards an end-to-end spatiotemporal analytics framework. **Procedia Computer Science**, v. 184, p. 60–67, 2021. ISSN 1877-0509. The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187705092100630X>. Acesso em: 12 nov. 2024.

KOUTSOS, T. M.; MENEXES, G. C.; ELEFTHEROHORINOS, I. G. The use of spatial interpolation to improve the quality of corn silage data in case of presence of extreme or missing values. **ISPRS International Journal of Geo-Information**, v. 11, n. 3, 2022. ISSN 2220-9964. Disponível em: <https://www.mdpi.com/2220-9964/11/3/153>. Acesso em: 25 nov. 2024.

KUHN, M.; KAMINSKI, E. T.; FRANKE, J. Track and trace: Integrating static and dynamic data in a hybrid graph-based traceability model. **Procedia CIRP**, v. 112, p. 250–255, 2022. ISSN 2212-8271. 15th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 14-16 July 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2212827122012434>. Acesso em: 12 nov. 2024.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. **Biometrics**, 1977. Disponível em: <https://www.jstor.org/stable/2529310>.

MACÁRIO, C. G. d. N.; BALDO, S. M. O modelo relacional. **Instituto de Computação Unicamp. Campinas**, p. 1–15, 2005. Disponível em: <https://www.ic.unicamp.br/~geovane/mo410-091/Ch03-RM-Resumo.pdf>. Acesso em: 16 nov. 2024.

MALEIKA, W. Moving average optimization in digital terrain model generation based on test multibeam echosounder data. **Geo-Marine Letters**, v. 35, n. 1, p. 61–68, 2015. ISSN 1432-1157. Disponível em: <https://doi.org/10.1007/s00367-014-0389-8>. Acesso em: 15 dez. 2024.

MATHEW, A. B.; KUMAR, S. An efficient index based query handling model for neo4j. **IJCST, Citeseer**, v. 3, n. 2, p. 12–18, 2014. Disponível em: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e46ba09fa246a786905fc88665802014466cbee3>. Acesso em: 14 dez. 2024.

MEENA, A.; MUTHUSAMY, M.; BEHERA, H. Spatial econometrics in agriculture: Modelling spatial dependencies in data. **Archives of Current Research International**, v. 24, p. 488–506, 08 2024. Acesso em: 25 nov. 2024.

MEHMOOD, N.; CULMONE, R.; MOSTARDA, L. Modeling temporal aspects of sensor data for mongodb nosql database. **Journal of Big Data**, v. 4, 03 2017. Disponível em: https://www.researchgate.net/publication/315907342_Modeling_temporal_aspects_of_sensor_data_for_MongoDB_NoSQL_database. Acesso em: 21 nov. 2024.

MONTEIRO, J.; Sá, F.; BERNARDINO, J. Experimental evaluation of graph databases: Janusgraph, nebula graph, neo4j, and tigergraph. **Applied Sciences**, v. 13, n. 9, 2023. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/13/9/5770>. Acesso em: 9 nov. 2024.

Neo4j. **Neo4j Docs**. [S.l.], 2024. Disponível em: <https://neo4j.com/docs/getting-started/graph-database/>. Acesso em: 16 nov. 2024.

NEO4J. **Graph Database Company**. 2025. Disponível em: <https://neo4j.com/company/>.

ORACLE. **Schema Diagrams**. 2024. Disponível em: <https://docs.oracle.com/en/database/oracle/oracle-database/21/comsc/schema-diagrams.html#GUID-01BFEF14-C6BD-42CD-8558-DDD214DB3AE0>. Acesso em: 25 nov. 2024.

PERÇUKU, A.; MINKOVSKA, D.; STOYANOVA, L. Modeling and processing big data of power transmission grid substation using neo4j. **Procedia Computer Science**, v. 113, p. 9–16, 2017. ISSN 1877-0509. The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017) / Affiliated Workshops. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187705091731685X>. Acesso em: 10 nov. 2024.

RAJŠP, A.; FISTER, I. Neo4j graph dataset of cycling paths in slovenia. **Data in Brief**, v. 48, p. 109251, 2023. ISSN 2352-3409. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352340923003700>. Acesso em: 21 out. 2024.

REIMOV, M. *et al.* Spatial analysis in decision-making on agriculture activities. *In*: EDP SCIENCES. **E3S Web of Conferences**. [S.l.], 2023. v. 386, p. 01002. Acesso em: 25 nov. 2024.

RONDÓN, A. **¿Conoces Neo4j o sabes de qué va?** 2022. Disponível em: <https://www.enmilocalfunciona.io/conoces-neo4j-o-sabes-de-que-va/>. Acesso em: 27 mai. 2025.

ROSA, A. d. S. da *et al.* Agrograph: a graph-based system for precision agriculture. **Memorias de las JALIO**, v. 8, n. 4, p. 67–80, 2022. Disponível em: <https://publicaciones.sadio.org.ar/index.php/JALIO/article/view/418>. Acesso em: 9 nov. 2024.

SCHENATTO, K. **Influência do interpolador na definição de unidades de manejo utilizando métodos de agrupamento**. 2014. Dissertação (Mestrado) — Universidade Estadual do Oeste do Paraná, 2014.

SHIRATSUCHI, L. S. *et al.* **Geração de mapas multitemáticos em agricultura de precisão**. [S.l.], 2007. Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/572273/geracao-de-mapas-multitematicos-em-agricultura-de-precisao>. Acesso em: 27 mai. 2025.

SU, K. Analysis on the research of database programming technology in computer software engineering. *In*: **2022 International Conference on Cloud Computing, Big Data Applications and Software Engineering (CBASE)**. [s.n.], 2022. p. 111–114. Disponível em: <https://ieeexplore.ieee.org/document/10092543>. Acesso em: 14 dez. 2024.

SUN, Y.; SARWAT, M. A spatially-pruned vertex expansion operator in the neo4j graph database system. **Geoinformatica**, Springer, v. 23, p. 397–423, 2019. Disponível em: <https://link.springer.com/article/10.1007/s10707-019-00361-2>.

TECNOLOGIA, L. **Banco de dados NoSQL para iniciantes**. 2022. Disponível em: <https://logap.com.br/blog/banco-de-dados-nosql-para-iniciantes/>. Acesso em: 25 nov. 2024.

TRUICĂ, C.-O. *et al.* The forgotten document-oriented database management systems: An overview and benchmark of native xml dodbmses in comparison with json dodbmses. **Big Data Research**, v. 25, p. 100205, 2021. ISSN 2214-5796. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214579621000228>. Acesso em: 15 nov. 2024.

WANG, S. *et al.* Prediction of wide range two-dimensional refractivity using an idw interpolation method from high-altitude refractivity data of multiple meteorological observatories. **Applied Sciences**, v. 11, n. 4, 2021. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/11/4/1431>. Acesso em: 14 dez. 2024.

YAMAMOTO, J. K.; LANDIM, P. M. B. **Geoestatística: conceitos e aplicações**. [S.l.]: Oficina de Textos, 2013.

ZHENG, L. *et al.* The spatio-temporal data modeling and application based on graph database. *In: 2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. [s.n.], 2017. p. 741–746. Disponível em: <https://ieeexplore.ieee.org/document/8110386>. Acesso em: 24 out. 2024.

ČEREŠŇÁK, R.; KVET, M. Comparison of query performance in relational a non-relation databases. **Transportation Research Procedia**, v. 40, p. 170–177, 2019. ISSN 2352-1465. TRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352146519301887>. Acesso em: 21 nov. 2024.

ŘEZŇÍK, T. *et al.* Interpolation of data measured by field harvesters: Deployment, comparison and verification. *In: ____*. [s.n.], 2020. p. 258–270. ISBN 978-3-030-39814-9. Disponível em: https://www.researchgate.net/publication/338887892_Interpolation_of_Data_Measured_by_Field_Harvesters_Deployment_Comparison_and_Verification. Acesso em: 25 nov. 2024.