UNIVERSIDADE DA INTEGRAÇÃO INTERNACIONAL DA LUSOFONIA AFRO-BRASILEIRA

INTITUTO DE ENGENHARIAS E DESENVOLVIMENTO SUSTENTÁVEL - IEDS

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

ANTONIO PAULO UAMBA

SOLUÇÃO PARA QUERIES COMPLEXAS EM BIG DATA UTILIZANDO MAPREDUCE COM BANCO DE DADOS MONGODB: UM ESTUDO DE CASO COM DADOS DO CAGED

REDENÇÃO - CE

2024

ANTONIO PAULO UAMBA

SOLUÇÃO PARA QUERIES COMPLEXAS EM BIG DATA UTILIZANDO MAPREDUCE COM BANCO DE DADOS MONGODB: UM ESTUDO DE CASO COM DADOS DO CAGED

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Intituto de Engenharias e Desenvolvimento Sustentável - IEDS da Universidade da Integração Internacional da Lusofonia Afro-Brasileira, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Antonio Manoel Ribeiro

REDENÇÃO - CE

2024

Universidade da Integração Internacional da Lusofonia Afro-Brasileira
Sistema de Bibliotecas da UNILAB
Catalogação de Publicação na Fonte.

Uamba, Antonio Paulo.

U11s

Solução para queries complexas em big data utilizando mapreduce com banco de dados mongodb: um estudo de caso com dados do Caged / Antonio Paulo Uamba. - Redenção, 2024.
54f: il.

Monografia - Curso de Engenharia De Computação, Instituto De Engenharias E Desenvolvimento Sustentável, Universidade da Integração Internacional da Lusofonia Afro-Brasileira, Redenção, 2024.

Orientador: Prof. Dr. Antonio Manoel Ribeiro.

1. Big data. 2. Informações governamentais. 3. MapReduce. 4. MongoDB. 5. Processo distribuido. I. Título

CE/UF/BSCA CDD 005.7

ANTONIO PAULO UAMBA

SOLUÇÃO PARA QUERIES COMPLEXAS EM BIG DATA UTILIZANDO MAPREDUCE COM BANCO DE DADOS MONGODB: UM ESTUDO DE CASO COM DADOS DO CAGED

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Intituto de Engenharias e Desenvolvimento Sustentável - IEDS da Universidade da Integração Internacional da Lusofonia Afro-Brasileira, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em:

BANCA EXAMINADORA

Documento assinado digitalmente
ANTONIO MANOEL RIBEIRO DE ALMEIDA
Data: 03/07/2024 12:02:12-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Antonio Manoel Ribeiro (Orientador)
Universidade da Integração Internacional da Lusofonia Afro-Brasileira - UNILAB

Documento assinado digitalmente
ANTONIO CARLOS DA SILVA BARROS
Data: 01/07/2024 14:19:43-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Antonio Carlos Da Silva Barros
Universidade da Integração Internacional da Lusofonia Afro-Brasileira - UNILAB

Documento assinado digitalmente
gov.br
JOHN HEBERT DA SILVA FELIX
Data: 01/07/2024 17:54:47-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. John Hebert Da Silva Felix
Universidade da Integração Internacional da
Lusofonia Afro-Brasileira - UNILAB

AGRADECIMENTOS

Primeiramente, expresso minha gratidão a Deus pelo amor, pela vida, pela salvação e pela sabedoria que me concedeu ao longo deste curso. Agradeço também aos meus irmãos Armando Muchanga, Nelson Muchanga, Celestina,pelo apoio incondicional, pelo incentivo nos estudos, à minha mãe pelo amor, carinho, ensinamentos e educação, e à minha família em geral pelo suporte inabalável, pelo incentivo e pelo companheirismo, sempre acreditando em mim. Agradeço à a minha Namorada Tamires da Conceição Mendes Semedo pela amizade, companheirismo, ideias e apoio constante quando necessário. Aos meus amigos de longa data, Orlando Alberto Buque e Wagner José Devete e ao Mario benjamin expresso minha gratidão pelo companheirismo e suporte nos momentos bons e ruins da vida,tambem agradecer ao Edilson Cuambe, Elêncio, pela convivencia, Também não posso deixar de agradecer ao Prof. Dr. Antonio Manoel Ribeiro por sua orientação em meu trabalho de conclusão de curso e pelo apoio fornecido, Tambem ao João Gabriel que me deu um grande suporte durante a pesquisa. Aos meus colegas de curso, Manuel Finda Evaristo e Manuel Lucala Zengo, agradeço por tornarem essa jornada acadêmica inspiradora e motivadora, e a todos os colegas do curso que me ajudaram em tudo. Agradeço a todos os professores por compartilharem seu conhecimento e também por demonstrarem afeto e caráter na educação, contribuindo não apenas para meu crescimento intelectual, mas também para o desenvolvimento de minha pessoa. Um agradecimento especial ao Thales Robson Barbalho e à Camila Campos por me ajudarem em meu desenvolvimento na área de dados durante o estágio. Por fim, sou grato a todos os colegas que, direta ou indiretamente, contribuíram para minha jornada de graduação. Obrigado a todos pelo apoio e pela confiança depositada em mim.

RESUMO

O processamento de grandes volumes de dados representa um desafio crescente em diversas esferas, abrangendo desde a gestão pública até setores privados e acadêmicos. À medida que a quantidade e a variedade de dados disponíveis para as agências governamentais aumentam exponencialmente, surge a necessidade urgente de desenvolver estratégias eficazes para analisar e utilizar essas informações de maneira inteligente e proativa na tomada de decisões. No âmbito das políticas públicas, a análise de dados desempenha um papel crucial ao fornecer insights valiosos para medir resultados, avaliar o desempenho de programas e projetos governamentais e embasar a formulação de políticas mais eficientes e eficazes. No entanto, o processamento de grandes volumes de dados requer abordagens e tecnologias especiais para garantir resultados precisos e oportunos. Nesse contexto, este trabalho propõe a aplicação da tecnologia de MapReduce em conjunto com o banco de dados MongoDB para lidar com os desafios associados ao processamento de grandes volumes de dados governamentais. O MapReduce, um modelo de programação paralela e distribuída, permite a divisão de tarefas em várias etapas de mapeamento e redução, possibilitando o processamento eficiente de grandes conjuntos de dados em ambientes distribuídos. Por sua vez, o MongoDB, um banco de dados NoSQL altamente escalável e flexível, oferece um ambiente propício para armazenar e manipular grandes volumes de dados de forma eficiente. O principal objetivo deste trabalho é investigar a viabilidade e a eficácia da utilização da tecnologia de MapReduce com MongoDB no contexto do processamento de dados governamentais. Por meio de um estudo prático, serão avaliadas as capacidades dessas tecnologias em lidar com as demandas específicas de análise e processamento de dados governamentais, visando aprimorar a capacidade das agências governamentais de extrair insights valiosos e tomar decisões fundamentadas com base em evidências. Ao finalizar esta pesquisa, espera-se contribuir significativamente para o avanço do conhecimento no campo do processamento de grandes volumes de dados governamentais, além de fornecer insights práticos e recomendações para o aprimoramento dos processos de análise e utilização de dados nas esferas governamentais.

Palavras-chave: Big data.Informações governamentais.MapReduce.MongoDB.Processo distribuido

ABSTRACT

The processing of large volumes of data represents a growing challenge in several spheres, ranging from public management to private and academic sectors. As the amount and variety of data available to government agencies increases exponentially, there is an urgent need to develop effective strategies to analyze and utilize this information intelligently and proactively in decision-making. In the context of public policies, data analysis plays a crucial role in providing valuable insights to measure results, evaluate the performance of government programs and projects, and support the formulation of more efficient and effective policies. However, processing large volumes of data requires special approaches and technologies to ensure accurate and timely results. In this context, this work proposes the application of MapReduce technology in conjunction with the MongoDB database to deal with the challenges associated with processing large volumes of government data. MapReduce, a parallel and distributed programming model, allows the division of tasks into several mapping and reduction steps, enabling the efficient processing of large data sets in distributed environments. In turn, MongoDB, a highly scalable and flexible NoSQL database, offers a suitable environment for storing and manipulating large volumes of data efficiently. The main objective of this work is to investigate the feasibility and effectiveness of using MapReduce technology with MongoDB in the context of government data processing. Through a practical study, the capabilities of these technologies in dealing with the specific demands of analyzing and processing government data will be evaluated, aiming to improve the ability of government agencies to extract valuable insights and make informed decisions based on evidence. Upon completion of this research, it is expected to contribute significantly to the advancement of knowledge in the field of processing large volumes of government data, in addition to providing practical insights and recommendations for improving data analysis and use processes in government spheres.

Keywords: Big data. Government information. MapReduce. MongoDB. Distributed processing.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Atualmente, os bancos de dados desempenham um papel crucial na vida cotidiana, contribuindo para o aumento do volume de informações, tanto em organizações quanto na internet. Como afirmou (CODD, 1970), "Os bancos de dados relacionais representam uma abordagem para gerenciar dados usando uma estrutura de tabelas". No entanto, com o crescimento exponencial de dados, especialmente na web, surgiram alternativas como os bancos de dados NoSQL, que seguem o paradigma BASE, como descrito por (BREWER; GILBERT, 2000) e seus colegas (2000), focando em desempenho, disponibilidade, escalabilidade e consistência.

Existem diversos modelos de bancos de dados no mercado, cada um com suas especificidades e usos. Os modelos relacionais, como o MySQL, são amplamente adotados em aplicações devido à sua estrutura tabular e capacidade de realizar consultas complexas. Por outro lado, os modelos não relacionais, como o RavenDB, se destacam pelo desempenho e escalabilidade em certas aplicações de software. Como afirmou (STONEBRAKER *et al.*, 2007), "Os bancos de dados NoSQL são uma alternativa viável para lidar com grandes volumes de dados de forma eficiente".

Os bancos de dados NoSQL são divididos em quatro categorias principais: orientados a colunas, documentos, grafos e armazenamento chave-valor. Cada categoria possui suas próprias características e é adequada para diferentes tipos de aplicações. Este trabalho visa analisar o desempenho dos bancos de dados relacionais e não relacionais, fornecendo dados para auxiliar na escolha do modelo mais adequado para diferentes necessidades. Como afirmou (DATE, 2003), "A escolha do modelo de banco de dados é crucial para o sucesso de um sistema de software, pois afeta diretamente a eficiência e a escalabilidade da aplicação".

Portanto, a relevância deste estudo reside na importância de selecionar o modelo de banco de dados mais apropriado para uma aplicação de software, considerando suas características distintas e as necessidades específicas do projeto. Como afirmou (OREN, 2013), "A escolha entre um banco de dados relacional e não relacional deve ser baseada em uma análise cuidadosa dos requisitos da aplicação e das capacidades de cada modelo".

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Analisar a eficiência de queries complexas em big data utilizando MAPREDUCE com banco de dados MONGODB.

1.1.2 Objetivos Específicos

- Analisar a Eficiência do MapReduce em Queries Complexas;
- Integração do MongoDB com MapReduce para Análise de Big Data;
- Otimização de Performance e Escalabilidade;
- Comparação com Outras Técnicas de Processamento de Big Data;

1.2 Justificativa

O estudo de caso que propõe a utilização do MapReduce em conjunto com o banco de dados MongoDB para abordar queries complexas em Big Data, usando como exemplo os dados do Cadastro Geral de Empregados e Desempregados (CAGED) no Brasil, apresenta diversas justificativas importantes. Primeiramente, os dados do CAGED representam um vasto conjunto de informações sobre o mercado de trabalho brasileiro, com uma complexidade que envolve variáveis demográficas, setoriais e geográficas, tornando a abordagem de Big Data essencial. O uso do MapReduce é particularmente eficiente neste contexto, pois permite a distribuição e o processamento de grandes volumes de dados de forma mais ágil, o que é crucial para análises oportunas e aprofundadas (CODD, 1970).

A escolha do MongoDB, um banco de dados NoSQL, reflete a necessidade de flexibilidade no armazenamento e na consulta de dados, que frequentemente são não estruturados ou semi-estruturados em projetos de Big Data. Esta combinação tecnológica não apenas aborda desafios de performance e escalabilidade, comuns no processamento de grandes conjuntos de dados, mas também se mostra inovadora no contexto de análise de dados governamentais (STONEBRAKER *et al.*, 2007). A relevância deste estudo se estende para além das fronteiras tecnológicas; ele tem o potencial de fornecer insights valiosos para a formulação de políticas públicas de emprego no Brasil, contribuindo significativamente para a compreensão das dinâmicas do mercado de trabalho. Em termos acadêmicos, o estudo serve como um exemplo prático da

aplicação de tecnologias de Big Data, oferecendo uma contribuição substancial para a literatura na área e servindo como referência para futuras pesquisas (BREWER; GILBERT, 2000).

A análise e oferecer uma perspectiva comparativa, o estudo inclui uma avaliação do PostgreSQL, um sistema de gerenciamento de banco de dados relacional, como ferramenta de contraste em termos de desempenho com o MongoDB. O PostgreSQL é reconhecido por sua robustez, integridade de dados e suporte avançado a SQL, elementos que representam um paradigma diferente de gestão de dados em relação ao MongoDB (DATE, 2003). A comparação entre MongoDB e PostgreSQL visa ilustrar as diferenças fundamentais em escalabilidade, performance e eficácia no tratamento de consultas complexas em volumes de dados significativos, como os presentes no CAGED. Este segmento do estudo ressalta a importância de selecionar o sistema de gerenciamento de banco de dados mais adequado às necessidades específicas de cada projeto de Big Data, destacando as contribuições únicas que ambos, MongoDB e PostgreSQL, podem trazer para a análise de dados em larga escala (OREN, 2013).

A utilização de técnicas estatísticas avançadas, como análise de regressão e séries temporais, pode aprimorar a compreensão dos padrões e tendências nos dados do CAGED, permitindo previsões mais precisas e embasadas. Essas abordagens estatísticas podem fornecer insights valiosos para a tomada de decisões estratégicas no âmbito do mercado de trabalho, auxiliando na identificação de oportunidades de crescimento e no desenvolvimento de políticas mais eficazes (EFRON; HASTIE, 2016).

Uma estatística relevante que destaca a importância do trabalho estatístico no contexto do CAGED é o registro de empregos formais no Brasil ao longo dos anos. Em 2020, apesar dos desafios da pandemia de COVID-19, observou-se um saldo positivo de empregos formais em vários meses, evidenciando a resiliência e capacidade de recuperação do mercado de trabalho.

Por exemplo, em dezembro de 2020, o Brasil registrou um saldo de 67.906 empregos formais, representando um aumento significativo em comparação com meses anteriores. Esses dados ressaltam a importância da análise estatística dos dados do CAGED para compreender tendências, identificar oportunidades de melhoria e embasar decisões estratégicas.

A análise estatística dos dados do CAGED também pode fornecer insights valiosos sobre a distribuição geográfica dos empregos, setores mais impactados, variações sazonais no mercado de trabalho, subsidiando a formulação de políticas públicas e ações governamentais.

1.3 Contextualização

A crescente disponibilidade de dados administrativos em larga escala apresenta tanto um desafio quanto uma oportunidade para os governos. Segundo o relatório "Data for Good: A Decade of Digital Innovation in Public Services"da OCDE (Organização para a Cooperação e Desenvolvimento Econômico), a quantidade de dados gerados pelos governos está crescendo exponencialmente, com previsão de um aumento de 30% ao ano. Esses dados incluem informações variadas sobre a população, serviços públicos, orçamentos, entre outros, e a capacidade de processá-los eficientemente é fundamental para a tomada de decisões informadas(OCDE, 2023).

A utilização do modelo de programação MapReduce em conjunto com o sistema de banco de dados MongoDB tem se destacado como uma abordagem eficaz para lidar com consultas complexas em grandes volumes de dados governamentais. Estudos recentes, como o artigo "Big Data Analytics in Government: A Systematic Literature Review"(CHEN *et al.*, 2023)., têm demonstrado que a combinação dessas tecnologias permite o processamento paralelo e distribuído de grandes conjuntos de dados, facilitando a extração de tendências, a identificação de padrões e a condução de análises preditivas.

As pesquisas como o relatório "Data-Driven Decision Making: The Engine of Government Transformation"da IBM Institute for Business Value destacam a importância da análise de dados para impulsionar a inovação e a eficiência nos governos. A capacidade de transformar dados em insights acionáveis não só melhora a prestação de serviços públicos, mas também contribui para a formulação de políticas mais eficazes e a alocação eficiente de recursos (VALUE, 2023)

1.4 Organização do Trabalho

O presente trabalho está estruturado em seis capítulos, conforme descrito a seguir.

- **Capitulo 1**: apresenta a introdução do trabalho, a descrição dos objetivos, gerais, específicos bem como a justificativa.
- **Capitulo 2**: apresenta o referencial teórico, contextualizando a unidade específica para a qual o projeto é destinado. Este capítulo também inclui uma análise dos principais trabalhos e pesquisas relacionados ao conteúdo abordado."
- **Capitulo 3**: traz a descrição dos principais temas e tecnologias referentes ao desenvolvimento deste trabalho.

- **Capitulo 4**: apresenta, de forma detalhada, todo o processo metodológico que foi seguido para o desenvolvimento do trabalho.
- **Capitulo 5**: traz os resultados obtidos no desenvolvimento do trabalho.
- **Capitulo 6**: apresenta as conclusões e questões que podem ser exploradas futuramente.

2 REFERENCIAL TEÓRICO

2.1 Big Data

De acordo com (SALINAS; LEMUS, 2017), o termo Big Data foi criado em 1997 por Michael Cox e David Ellsworth, pesquisadores da NASA que tinham que trabalhar com conjuntos de dados geralmente muito grandes, o que sobrecarrega a memória principal, disco local e capacidade de disco remoto. Eles chamaram isso de problema do Big Data.

Embora seja amplamente referenciado, o conceito de Big Data não possui uma definição rigorosa e consensual. Geralmente, está associado ao processamento de grandes volumes de dados provenientes de diversas fontes e sem estruturas pré-definidas (SALINAS; LEMUS, 2017) De acordo com (GANDOMI; HAIDER, 2015), cercade 95% dos dados tratados por tecnologias de Big Data são dados não estruturados.

Para alguns autores, Big Data é simplesmente um conjunto de dados cujo tamanho ultrapassa as capacidades das ferramentas tradicionais de bancos de dados para capturar, armazenar, gerenciar e analisar (SALINAS; LEMUS, 2017).

Segundo a (SAS Institute Inc., 2022), Big Data refere-se a conjuntos de dados tão grandes, rápidos ou complexos que são difíceis ou até impossíveis de serem processados utilizando métodos convencionais. Embora o ato de acessar e armazenar grandes quantidades de informações para análise exista há muito tempo, o conceito de Big Data ganhou destaque no início dos anos 2000.

Para (ANAND, 2019), Big Data é uma tecnologia utilizada para armazenar dados, incluindo formatos não estruturados, semi estruturados e estruturados, fazendo uso de dispositivos de armazenamento mais econômicos. O processamento dos dados é descentralizado e distribuído em vários servidores para acelerar o processamento. Os dados são armazenados em seu formato nativo, sem um esquema ou modelagem definidos.

Segundo (OUSSOUS *et al.*, 2018), o termo Big Data refere-se a conjuntos de dados grandes e em constante crescimento, que englobam diferentes formatos de dados estruturados, não estruturados e semi-estruturado. O Big Data possui uma natureza complexa e requer tecnologias sofisticadas e algoritmos avançados. Nesse novo contexto, as ferramentas tradicionais de Business Intelligence mostram-se ineficientes para lidar com aplicações de Big Data.

2.1.1 Características

A expressão Big Data surge da proliferação dos avanços tecnológicos e da abundante geração de dados. Em resumo, representa grandes conjuntos de dados heterogêneos, desafiando abordagens computacionais convencionais devido à sua dinâmica e complexidade. Inicialmente, o conceito de Big Data incorporava três propriedades fundamentais dos dados, identificadas por (LANEY, 2001) como os 3Vs: Volume, Variedade e Velocidade.

- **Volume** Grandes volumes de dados são gerados mediante o uso de recursos computacionais abundantes. Com a evolução das mídias sociais e outros recursos e serviços da Internet, as pessoas produzem mais e mais conteúdo, vídeos, fotos, tweets, entre outros tipos de dados.
- **Velocidade** Os dados são gerados em grande velocidade, à medida que os recursos computacionais têm sua capacidade de produção, captura e processamento de dados aumentada.
- **Variedade** Os dados advêm de variadas fontes (sistemas legados, e-mails, posts em mídias sociais, arquivos de vídeo/áudio, gráficos, dispositivos ou sensores), as quais implementam tecnologias distintas para representação e armazenamento de recursos digitais.

Ao analisar o atual panorama da adoção de Tecnologias de Comunicação e Informação, novos atributos são incorporados aos 3Vs iniciais, variando de acordo com a perspectiva de especialistas ou o domínio de aplicação. Nessa linha, (AKHTAT, 2018) destaca apresença de 6Vs, ampliando as propriedades com:

- **Veracidade** Refere-se à integridade e à precisão dos dados, contrapondo o fenômeno GIGO (garbage-in, garbage-out – lixo entra, lixo sai) na recuperação da informação. Neste sentido, deve-se evitar ruídos e incertezas no armazenamento dos dados de modo a não interferir, consequentemente, na análise da informação e no Processo de Tomada de Decisão.
- **Variabilidade** Relaciona-se à compreensão e ao tratamento dos fenômenos subliminares e temporariamente presentes nos dados. Por exemplo, sazonalmente, alguns eventos específicos (virais nas mídias sociais, como a estreia de um filme muito aguardado ou o acontecimento de um fato midiático) podem refletir em padrões de comportamento que não se sustentam ao longo do tempo.
- **Valor** É a característica mais importante em termos dos dados, independente das demais dimensões (volume, velocidade, variedade, variabilidade e veracidade). O valor em Big Data é, principalmente, percebido mediante a análise com dados precisos e,por conseguinte,

a aquisição de informação e insights úteis para o processo de tomada de decisão.

2.2 MapReduce como Paradigma de Processamento em Big Data

2.2.1 MapReduce

O MapReduce (VELOSO, 2019) é um modelo de programação paralela para processamento de grandes volumes de dados, proposto inicialmente pela Google em 2004. É adequado para problemas que podem ser divididos em subproblemas menores. Este paradigma é eficaz na distribuição do processamento em muitas máquinas, permitindo que grandes volumes de dados sejam processados de maneira eficiente e paralela. A base do MapReduce é dividir e processar conjuntos de dados usando duas funções fundamentais: Map e Reduce. A função Map processa blocos de arquivos em paralelo em várias máquinas, produzindo tuplas chave-valor. A função Reduce é responsável pelo resultado final do processamento, agrupando e reduzindo essas tuplas.

2.2.2 Modelo de Programação

O modelo de programação MapReduce simplifica o processamento de grandes volumes de dados. Sua estrutura se baseia em duas funções principais: Map e Reduce.

A função Map, definida pelo usuário, recebe uma tupla (chave, valor) e gera um conjunto intermediário de tuplas (chave, valor). Por outro lado, a função Reduce, também definida pelo usuário, é executada para cada chave intermediária, combinando todos os valores associados a ela. Enquanto a tarefa de mapeamento geralmente busca algo, a função de redução faz a sumarização do resultado.

Algoritmo 1: Mapeamento (Map)

mapString key, String value **forall** *word w in value*
do- **endemit**(w,1)$Emiteatupla(palavra, 1)$

Algoritmo 2: Redução (Reduce)

reduceString key, List<Int> values $word\_count \leftarrow 0$ **for** *count in values*
do- **end**word_count $\leftarrow word\_count + count$ **emit**(*key*, *word_count*) Emite a tupla (palavra, total de ocorrências)

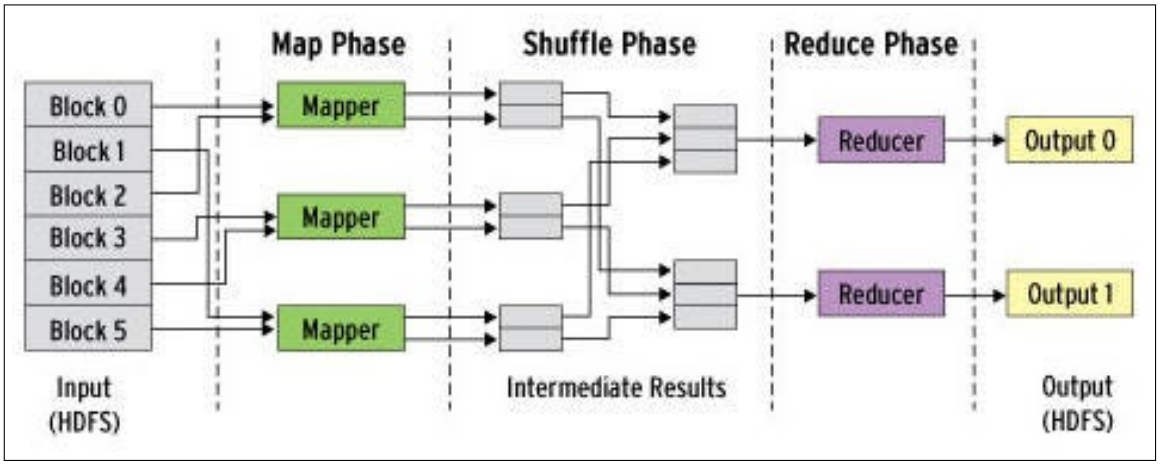
Nos pseudocódigos 1 e 2, a função Map recebe o nome de um e-book como chave e

seu conteúdo como valor. Para cada palavra no conteúdo, ela emite uma tupla (palavra, 1). A função Reduce, por sua vez, recebe uma palavra como chave e um iterador para todos os valores emitidos pela função Map para essa palavra. Todos os valores são somados, resultando em uma tupla (palavra, total de ocorrências).

2.2.3 Etapas do Processamento MapReduce

O processo MapReduce envolve três etapas principais: Map, onde os dados de entrada são divididos e processados; Shuffle, Combine e Partition, onde os dados são redistribuídos e preparados para redução; e Reduce, onde os dados são agregados. Este modelo, no entanto, apresenta desafios na expressão de lógicas complexas e na manipulação de dados, especialmente em iterações e no uso intensivo de leitura/gravação no HDFS.(Hadoop Distributed File System).

Figura 1 – Processo de Shuffle no MapReduce



Fonte: (MACHADO, 2023)

2.3 Banco de Dados Não Relacional (NoSQL)

Inegavelmente, o avanço exponencial no volume de dados e informações na internet é uma realidade palpável em nosso cotidiano. Contudo, essa proliferação de dados pode gerar desafios significativos em termos de infraestrutura para as aplicações que lidam com grandes quantidades de informação. Em 2000 o pesquisador Eric Brewer elaborou o Teorema CAP (Consistency, Availability e Partition Tolerance), tendo em mente a conjectura de que não existem garantias de que um sistema computacional distribuído possa ter simultaneamente consistência, disponibilidade e tolerância ao particionamento. Segundo Brewer, um sistema distribuído pode garantir apenas duas dessas três características ao mesmo tempo (GILBERT; LYNCH, 2012)

2.3.1 Consistência

É a garantia de que, após a finalização de uma transação, o sistema mantém a integridade dos dados, ou seja, a transação não pode infringir as regras do base de dado (WEI *et al.*, 2009)

2.3.2 Tolerância ao particionamento

É a garantia de que um sistema continue operando mesmo após ocorrerem particionamentos na rede. Assim, se um sistema dispuser dessa propriedade, ele deverá ser capaz de realizar operações como leitura e escrita após ocorrerem particionamentos na rede (GILBERT; LYNCH, 2012) Segundo (GILBERT; LYNCH, 2012), uma aplicação voltada para a Internet pode atender até duas dessas propriedades. Como qualquer estratégia de escalabilidade horizontal é baseada no particionamento de dados, os desenvolvedores devem decidir entre consistência e disponibilidade.

Nesse contexto, a arquitetura NoSQL possui uma melhor disponibilidade, desempenho e flexibilidade, porém tem uma baixa consistência. O modelo NoSQL utiliza a Consistência Eventual, na qual poucas atualizações são esperadas, e quando elas ocorrem não são propagadas instantaneamente. Esse modelo requer que as atualizações sejam disseminadas para todas as cópias de tempos em tempos. Com isto, quando uma escrita for realizada no base, não tem como garantir que, daquele momento em diante todos os processos terão acesso ao dado atualizado (VOGELS, 2008) Ao analisar o teorema e as particularidades do NoSQL em relação à consistência, percebe-se que, embora a abordagem ACID destaque a consistência como um de

seus elementos centrais, emergiu o conceito BASE (Basically Available, Soft state, Eventual consistency). O BASE sugere uma abordagem mais flexível à constante necessidade de consistência de dados, priorizando a disponibilidade e a resistência a problemas de particionamento. A disponibilidade é alcançada através do gerenciamento de falhas locais. Assim, se um componente do sistema falhar, isso significa apenas a redução de um nó disponível na rede (PRITCHETT, 2008) A estratégia mais comum atualmente para enfrentar esse desafio é a expansão horizontal, ou seja, a adição de mais servidores para suportar as demandas da aplicação. No entanto, essa solução não é isenta de complicações, pois a integração de um novo servidor em um sistema distribuído pode ser uma tarefa árdua, especialmente em aplicações que dependem de bancos de dados relacionais, devido à complexidade de configuração envolvida. Quando se lida com uma quantidade significativa de registros, o uso de bancos de dados relacionais pode afetar negativamente o desempenho, prejudicando a experiência do usuário. Surge então a tecnologia NoSQL como uma solução inovadora, oferecendo uma maneira alternativa de armazenamento de dados focada na disponibilidade, desempenho e escalabilidade. Atualmente, existe uma ampla variedade de modelos e sistemas de bancos de dados não-relacionais (NoSQL), cada um com seus próprios conceitos e características distintas, atendendo às diversas necessidades de armazenamento e gestão de dados.

Os modelos de dados NoSQL apresentam essas característica:

- Baseados em documento (document-store),
- Chave/valor(key/value),
- Column Store Database,
- Grafo (Graph)

2.3.2.1 Baseados em documento (document-store)

Em um banco de dados orientado a "documentos", a informação é armazenada e recuperada como pares chave-valor, onde os valores são conhecidos como ’Documentos’. Estes documentos representam estruturas de dados complexas, podendo incluir textos, arrays, strings, ou formatos como JSON e XML. É comum o uso de documentos aninhados, aumentando a eficiência do sistema, especialmente considerando que muitos dados gerados atualmente estão em JSON e tendem a não ser estruturados.

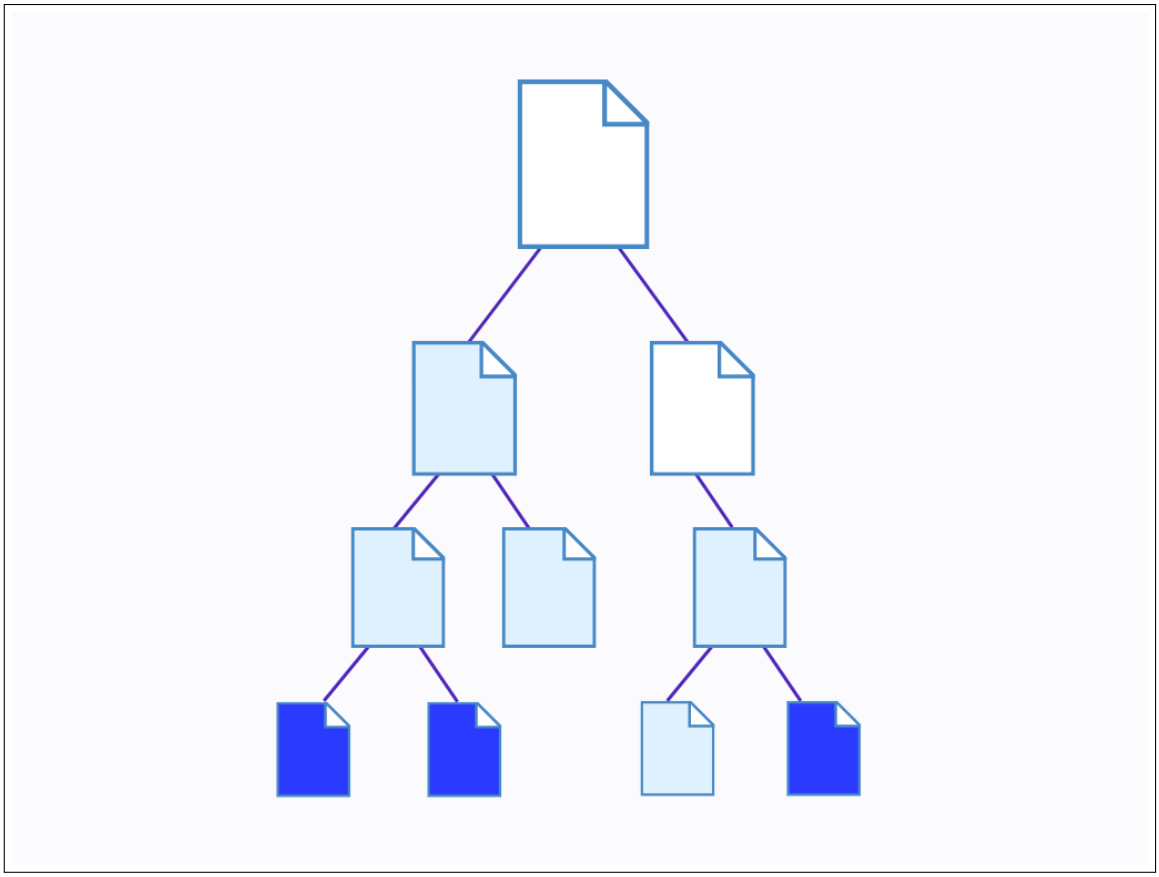
2.3.2.2 *Principais Vantagens:*

- Adequação para Dados Semi-estruturados:Este formato é especialmente útil para lidar com dados que não seguem uma estrutura rígida.
- Facilidade no Gerenciamento:O armazenamento e recuperação de documentos são processos simplificados, facilitando a gestão dos dados.

2.3.2.3 *Desafios e Limitações*

- Gestão de Múltiplos Documentos:A manipulação de um grande número de documentos pode se tornar complexa.
- Precisão em Operações de Agregação:Pode haver desafios na execução precisa de operações de agregação devido à natureza variável dos documentos.

Figura 2 – Modelo de Armazenamento de Documentos



Fonte: elaborado pelo autor (2024).

2.3.2.4 Chave/valor(key/value)

Banco de dados baseado em valor-chave representa uma abordagem simples e eficaz dentro do espectro dos bancos de dados NoSQL. Neste modelo, os dados são organizados como pares de valor-chave. As chaves geralmente são identificadas por strings, números inteiros ou caracteres, podendo também adotar formatos mais complexos. Cada chave está associada a um valor, que pode variar em tipo - desde JSON e BLOBs (Objetos Binários Grandes) até simples strings. Essa estrutura é semelhante a uma tabela hash, onde cada chave é única e direciona para um valor específico. Este modelo é frequentemente utilizado em plataformas de comércio eletrônico e sites de vendas online devido à sua eficiência.

2.3.2.5 Principais Vantagens:

- Capacidade de Escala - Ideal para gerenciar grandes volumes de dados e cargas de trabalho intensas.
- Recuperação Simples - Permite um acesso rápido e fácil aos dados utilizando as chaves.

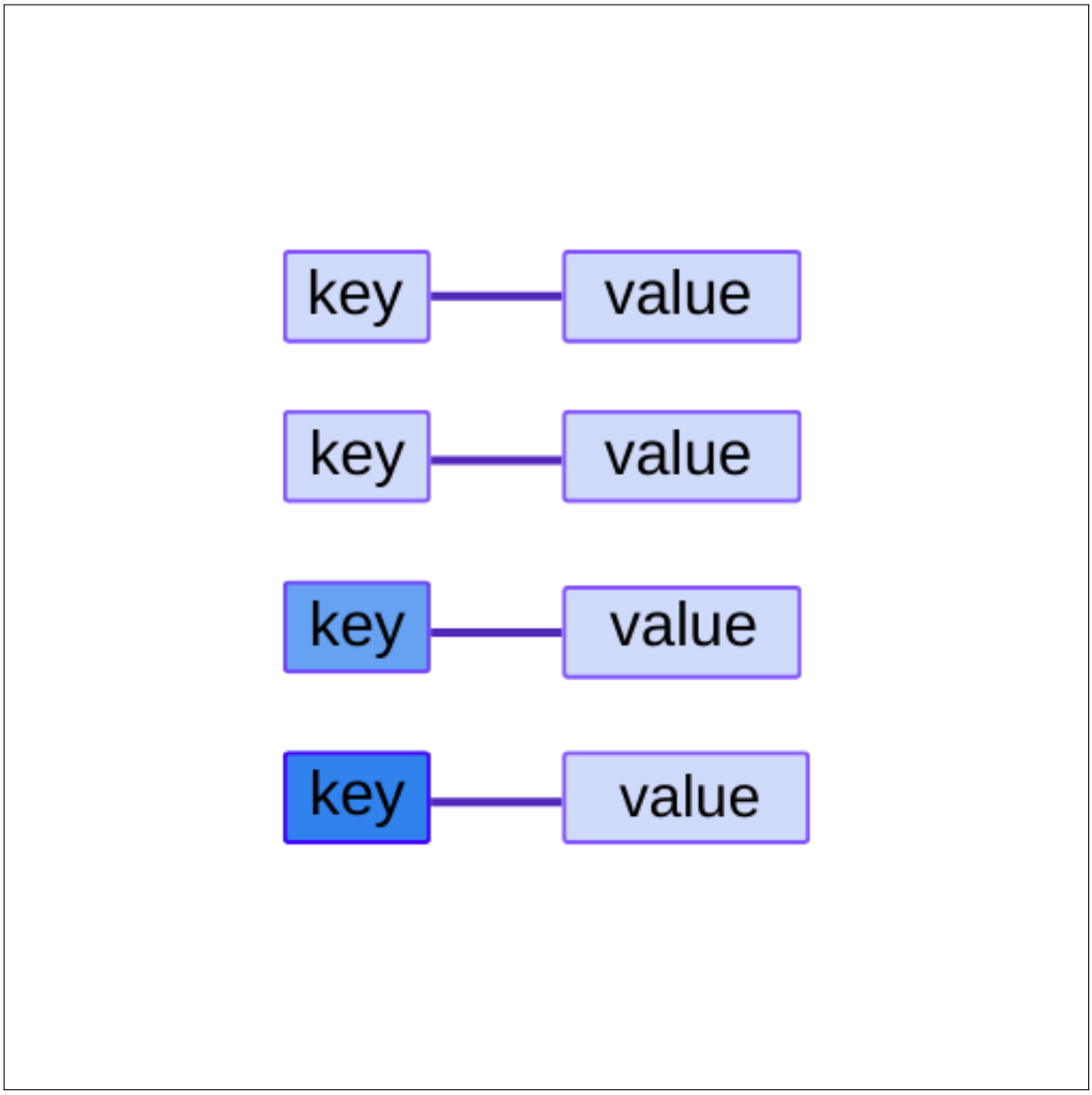
2.3.2.6 Desafios e Limitações

- Complexidade nas Consultas - Realizar consultas complexas que envolvam múltiplos pares valor-chave pode reduzir o desempenho do sistema.
- Precisão em Operações de Agregação:Pode haver desafios na execução precisa de operações de agregação devido à natureza variável dos documentos.

2.3.2.7 Column Store Database

Em bancos de dados orientados a colunas, a organização dos dados difere significativamente do modelo relacional tradicional. Aqui, ao invés de armazenar dados em filas ou tuplas, eles são mantidos em células que são agrupadas por colunas. Este método destaca a coluna como a unidade principal de armazenamento, permitindo que grandes volumes de dados sejam armazenados eficientemente em colunas adjacentes. Notavelmente, a estrutura das colunas – seu formato e títulos – pode variar de uma linha para outra, proporcionando flexibilidade. Cada coluna é gerenciada de forma independente, e dentro de cada uma, é possível encontrar subconjuntos de outras colunas, assemelhando-se aos bancos de dados tradicionais em certo grau.

Figura 3 – Modelo de Armazenamento de Chave/valor

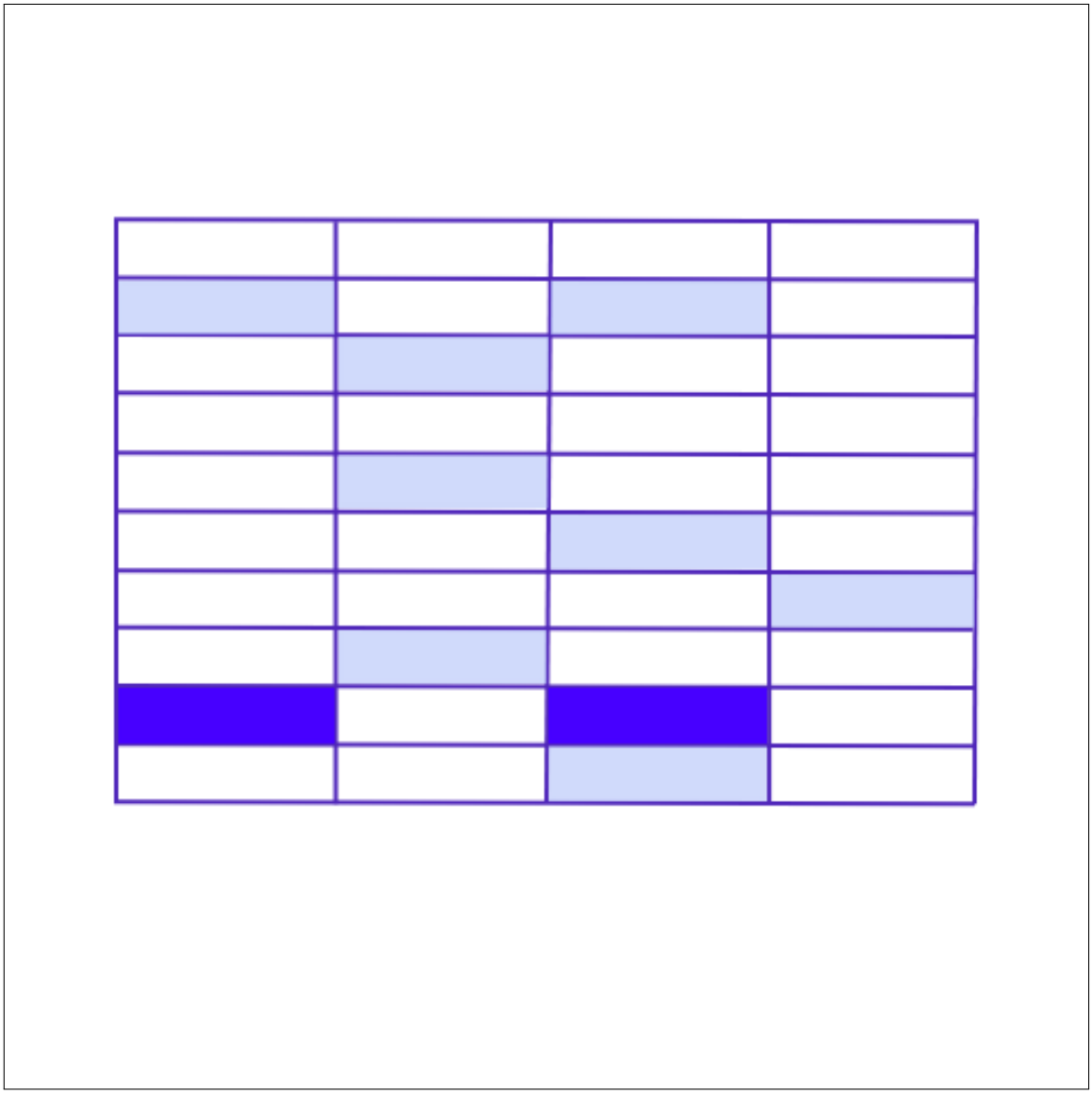


Fonte: elaborado pelo autor (2024).

2.3.2.8 Grafo (Graph)

A arquitetura especializada dos bancos de dados baseados em grafos, focando em como eles lidam com o armazenamento e a gestão de informações estruturadas em forma de grafos. Essencialmente, um grafo é uma representação de ligações entre dois ou mais elementos em um conjunto de dados. Estes elementos, conhecidos como nós, estão interconectados através de relações denominadas arestas, cada uma com um identificador único. Os nós atuam como pontos de interação chave no grafo. Este tipo de banco de dados é particularmente prevalente em redes sociais, caracterizadas por uma ampla gama de entidades interligadas por diversas

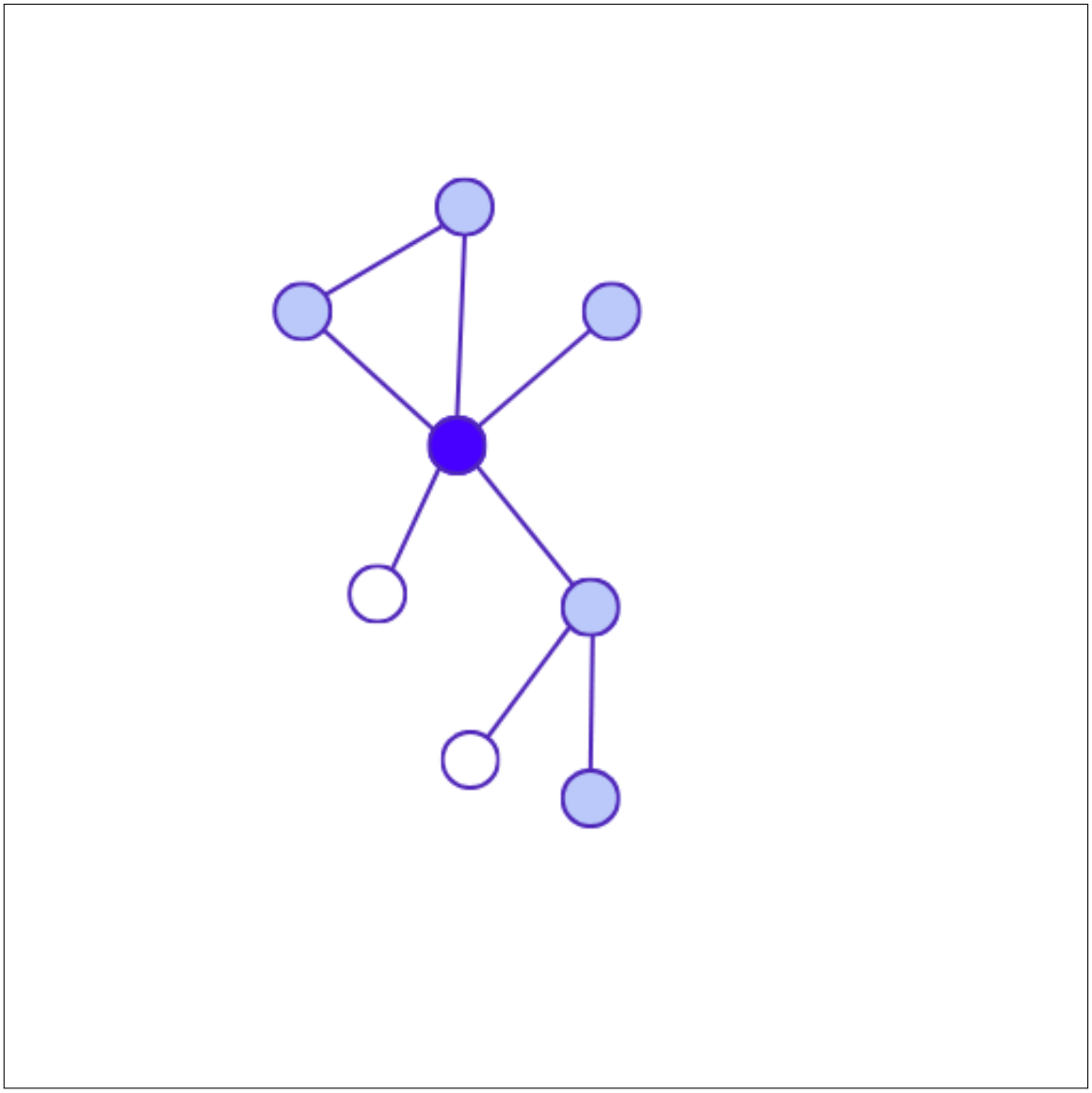
Figura 4 – Modelo de Armazenamento de
Column Store Databaser



Fonte: elaborado pelo autor (2024).

características através das arestas. Em contraste com os bancos de dados relacionais, onde as tabelas possuem conexões mais tênues, os grafos se destacam por suas ligações fortes e estruturas rígidas.

Figura 5 – Modelo de Armazenamento de Grafo (Graph)



Fonte: elaborado pelo autor (2024).

2.4 MongoDB: Características e Vantagens para Dados Complexos

2.4.1 MongoDB

O MongoDB,é um sistema de base de dados NoSQL, foi lançado em 2009 pela empresa 10 gen, que mais tarde foi renomeada para MongoDB Inc. Este sistema opera sob uma licença híbrida, combinando elementos da GNU Affero General Public License e da Apache License. Desenvolvido em linguagens de programação como C, C++ e JavaScript, o MongoDB rapidamente se estabeleceu como um líder entre os sistemas de bases de dados NoSQL. Ele está listado entre os cinco principais sistemas de base de dados, ao lado de gigantes do mercado de

bases de dados relacionais como Oracle, MS SQL Server, MySQL e PostgreSQL.

A popularidade do MongoDB é avaliada com base em vários critérios. Isso inclui menções em motores de busca populares como Google, Bing e Yandex; tendências de pesquisa no Google Trends; discussões técnicas em fóruns como Stack Overflow e DBA Stack Exchange; a frequência com que é mencionado em anúncios de emprego nos sites Indeed e Simply Hired; o número de perfis profissionais que o mencionam no LinkedIn; e sua relevância em redes sociais como o Twitter. Esses múltiplos indicadores refletem a ampla adoção e a importância crescente do MongoDB no cenário das tecnologias de informação. O MongoDB se destaca como um sistema de base de dados orientado a documentos. Diferente dos sistemas relacionais que organizam dados em linhas ou tuplas, o MongoDB utiliza documentos. Estes documentos são estruturas que podem incluir uma variedade de campos, contendo valores simples, vetores, ou até outros documentos incorporados. Essa abordagem resulta em uma flexibilidade muito maior do que as estruturas tabulares tradicionais. Além disso, um aspecto notável dos sistemas baseados em documentos como o MongoDB é a ausência de esquemas rígidos. Dentro de uma mesma "coleção"(análoga à "tabela"nos sistemas relacionais), os documentos não são obrigados a ter os mesmos campos, permitindo uma maior flexibilidade na gestão de dados. A adição ou remoção de campos em um documento é um processo bastante simples nesse ambiente

2.5 CAGED (Cadastro Geral de Empregados e Desempregados)

O CAGED é uma fonte crucial de informações sobre o mercado de trabalho em escala nacional, com atualização mensal, sendo de grande importância para o mercado financeiro. Foi criado como um instrumento de acompanhamento e fiscalização do processo de admissão e demissão de trabalhadores regidos pela CLT, com o propósito de apoiar os desempregados e promover medidas contra o desemprego. A partir de 1986, passou a ser utilizado como suporte ao pagamento do seguro-desemprego e, mais recentemente, tornou-se um recurso significativo para a reciclagem profissional e recolocação dos trabalhadores no mercado de trabalho. Dentro do CAGED, existem três tipos de arquivos: MOV (dados em movimentação), FOR (dados fora de execução ou movimentação) e EXC (dados excluídos). Vale ressaltar que os dados do CAGED são gerados mensalmente. É importante considerar a possibilidade de correções, podendo haver meses nos quais todos os dados fornecidos são retificados.

3 TRABALHOS RELACIONADOS

3.1 Revisão Bibliográfica sobre Soluções em Big Data

Big Data é um termo que se refere a grandes conjuntos de dados que excedem a capacidade de processamento dos softwares de banco de dados tradicionais. Este conceito foi detalhado por (BRUCE *et al.*, 2013) e (OHLHORST, 2013) . O Big Data é caracterizado principalmente por três atributos: volume, variedade e velocidade, conforme discutido por Simon em 2013 e Castro em 2014. O volume diz respeito ao crescente montante de dados, a variedade à diversidade de tipos de dados, e a velocidade à rapidez de processamento dos dados.

Segundo (NOVO; NEVES, 2013), as características distintas do Big Data se diferenciam dos sistemas tradicionais de Business Intelligence (BI). Big Data envolve uma abordagem nova no armazenamento e análise de dados, permitindo a integração e interação de todos os dados de uma organização. Davenport em 2014 destaca que as empresas que se beneficiam do Big Data focam no fluxo de dados ao invés de apenas em dados históricos, priorizando modelos preditivos.Os benefícios do Big Data na gestão incluem redução de custos, aumento de eficiência operacional, melhor tomada de decisão, aprimoramento de produtos e serviços, e inovação, conforme explorado por (LEEFLANG *et al.*, 2014) e (SILVA; CAMPOS, 2014).

Em relação à adoção do Big Data, existem desafios que ainda não foram amplamente explorados na literatura. (DAVENPORT, 2014) e (YEOH; KORONIUS, 2010) discutem que, assim como os sistemas de BI, as tecnologias de Big Data visam melhorar o processo de tomada de decisão. Além disso, muitas das tecnologias e conceitos de análise sob a égide do Big Data não são novos.

Para uma adoção eficaz de sistemas de Big Data, é crucial considerar a estratégia, os processos e a liderança, como apontado por (YEOH; KORONIUS, 2010) A alta gestão deve estar envolvida para fornecer apoio, acesso a recursos e superação de barreiras. No que se refere aos recursos humanos, há um desafio relacionado à falta de profissionais qualificados, uma lacuna que pode ser preenchida por meio de educação formal e desenvolvimento de talentos internos, segundo (LEEFLANG *et al.*, 2014).

A gestão da implementação também apresenta desafios. Metodologias ágeis de desenvolvimento são consideradas mais adequadas para grandes projetos analíticos de dados, como discutido por (DAVENPORT, 2014) Além disso, a abordagem faseada é vista como crítica para o sucesso na adoção de sistemas de informação, conforme (GUPTA *et al.*, 2014)

Por fim, a ética e a privacidade são aspectos fundamentais no contexto do Big Data. Questões como o uso apropriado de dados e a segurança da privacidade são essenciais, conforme indicado por (SIMON, 2013) A privacidade se relaciona com a proteção de informações pessoalmente identificáveis e o desafio de manter a utilidade dos dados enquanto se protege a anonimidade dos indivíduos, como discutido por (MINELLI *et al.*, 2013).

3.2 Revisão Bibliográfica sobre Soluções CAGED

Um dos principais pontos abordados é a insegurança gerada por mudanças nos parâmetros do CAGED, como apontado em um artigo da Agência O Globo (iG). As constantes revisões nos dados do CAGED, como a criação de empregos formais, geram dúvidas e reduzem a utilidade dessas informações para agentes econômicos. Por exemplo, uma revisão nos dados de emprego de um mês específico alterou significativamente o número de vagas criadas, impactando a percepção do mercado sobre o cenário econômico.

O Caged, coordenado pelo Ministério do Trabalho (MTb), foi instituído pela Lei no 4.923/1965, com o objetivo de registrar "admissões e dispensas de empregados nas empresas abrangidas pelo sistema da Consolidação das Leis do Trabalho" (Brasil, 1965, Artigo 1o). Trata-se de um registro administrativo alimentado mensalmente por estabelecimentos formais mediante sistema eletrônico próprio. Os procedimentos para declaração da movimentação de empregados admitidos ou desligados são sistematizados quando da publicação do Manual de Orientações do Caged, por meio de portaria do MTb, de periodicidade anual. A declaração é realizada por meio eletrônico, até o dia 7 do mês subsequente à admissão ou ao desligamento do empregado. As informações encaminhadas fora do prazo legal implicam multa automática ao estabelecimento, variando de acordo com o tempo de atraso (entre R$ 4,70 por empregado até trinta dias de atraso e R$ 13,40 por empregado acima de sessenta dias de atraso).(EMPREGADOS, 1965).

Em 2017, uma média mensal de 7,6 milhões de estabelecimentos informou 29,5 milhões de movimentações, sendo 14,7 milhões de admissões e 14,8 milhões de desligamentos, gerando saldo de -15,3 mil empregos celetistas. Entre janeiro-junho/2018, uma média mensal de 7,4 milhões de estabelecimentos declarou 15,4 milhões de movimentações, das quais 7,9 milhões de admissões e 7,5 milhões de desligamentos, com saldo de 392,5 mil empregos celetistas.(EMPREGADOS, 1965).

4 METODOLOGIA

A metodologia, segundo (BRUYNE *et al.*, 1982), tem como objetivo esclarecer a unidade subjacente a uma multiplicidade de procedimentos científicos particulares. Ela ajuda a desimpedir os caminhos da prática concreta da pesquisa dos obstáculos que esta encontra.

A metodologia não pretende refletir a progressão concreta de cada pesquisa particular, pois esta é eminentemente variável, mas quer se pensar em sua própria progressão e em suas relações com os procedimentos concretos da prática científica.

Esta pesquisa é de caráter qualitativo e exploratório.A abordagem qualitativa se deve a análise de informações descritivas de situações que o estudo teve contato (DUARTE, 2009) O método utilizado foi o Estudo de caso, definido por (GIL, 2002) como um "estudo profundo e exaustivo de um ou poucos objetos, de maneira que se permita seu amplo e detalhado conhecimento". Esta técnica é a mais apropriada para o estudo de fenômenos contemporâneos inseridos dentro de um contexto real, como é o caso dessa pesquisa (YIN, 2015)

No contexto da teoria de Estudo de Casos, esse é considerado um Estudo de Caso único, por considerar apenas um evento. Apesar de existirem críticas em relação a esse tipo de estudo ser convincente, ele é defendido para esse estudo por ser um caso raro, devido ao protagonismo e ao número reduzido de outros casos com as mesmas características (YIN, 2001)

Nesta Seção, serão apresentados todos os métodos e procedimentos adotados para o desenvolvimento da pesquisa.

4.1 Levantamento e Preparação dos Dados

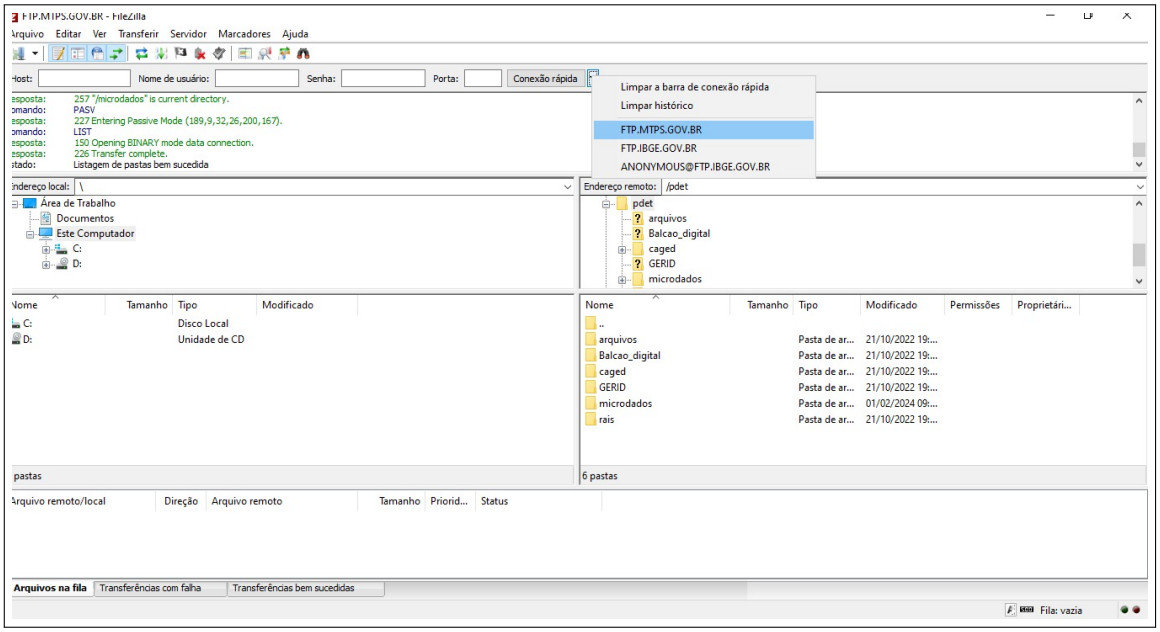
O primeiro passo consistiu em acessar os dados essenciais no site do PDET(Programa de Disseminação das Estatísticas do Trabalho), usando o Filezila atravez do comando FTP. Nesse portal há diversos conjuntos de dados disponivel para analise. Para o escopo desta pesquisa específica, optamos por utilizar os dados do Cadastro Geral de Empregados e Desempregados (CAGED), devido à sua relevância para o estudo em questão.

4.1.1 Coleta dos dados do Cadastro Geral de Empregados e Desempregados (CAGED)

NNa presente seção, descrevemos sucintamente o processo de coleta de dados do Cadastro Geral de Empregados e Desempregados (CAGED). Inicialmente, na Figura 6, demonstramos como acessar o site do Ministério do Trabalho, um passo crucial para obter

informações atualizadas e precisas sobre emprego e desemprego no Brasil, essenciais para análises e estudos do mercado de trabalho. Ao utilizar o protocolo FTP para acessar o site, somos direcionados para uma pasta chamada 'pdet', conforme ilustrado na Figura 6. Dentro desta pasta, estão armazenados os dados relevantes para nossa análise, o que simplifica significativamente o processo de recuperação e utilização das informações.

Figura 6 – Acesso ao dados de CAGED

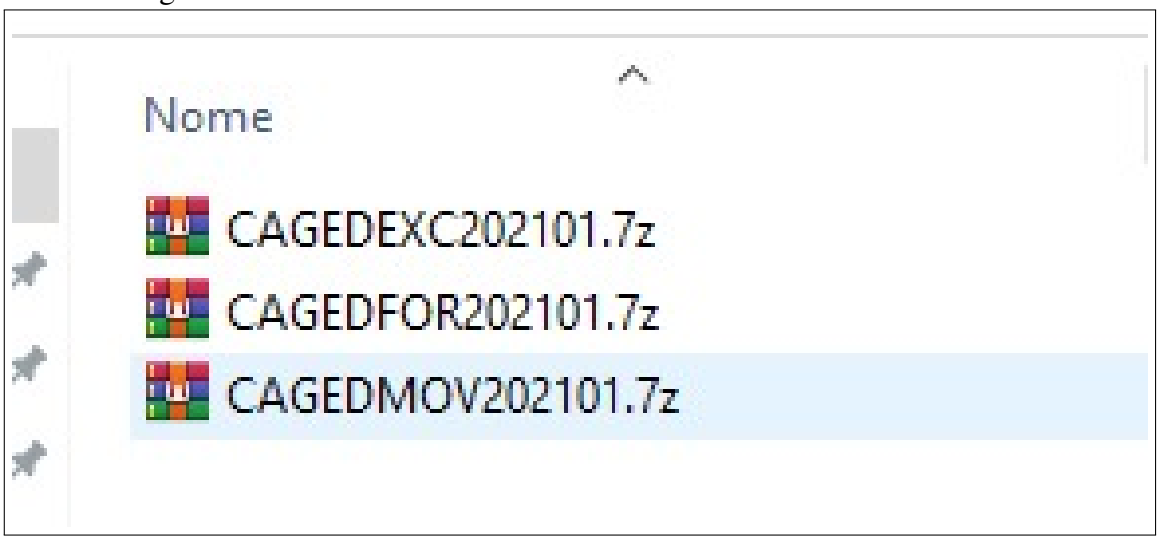


Fonte: elaborado pelo autor (2024).

4.1.2 Preparação dos dados, incluindo limpeza, transformação e estruturação para atender aos requisitos da análise

Nesta fase, foi necessário utilizar várias ferramentas para extrair os dados, que são disponibilizados em formato 7z, conforme ilustrado na figura 7. Esse processo é essencial para acessar as informações após o download.

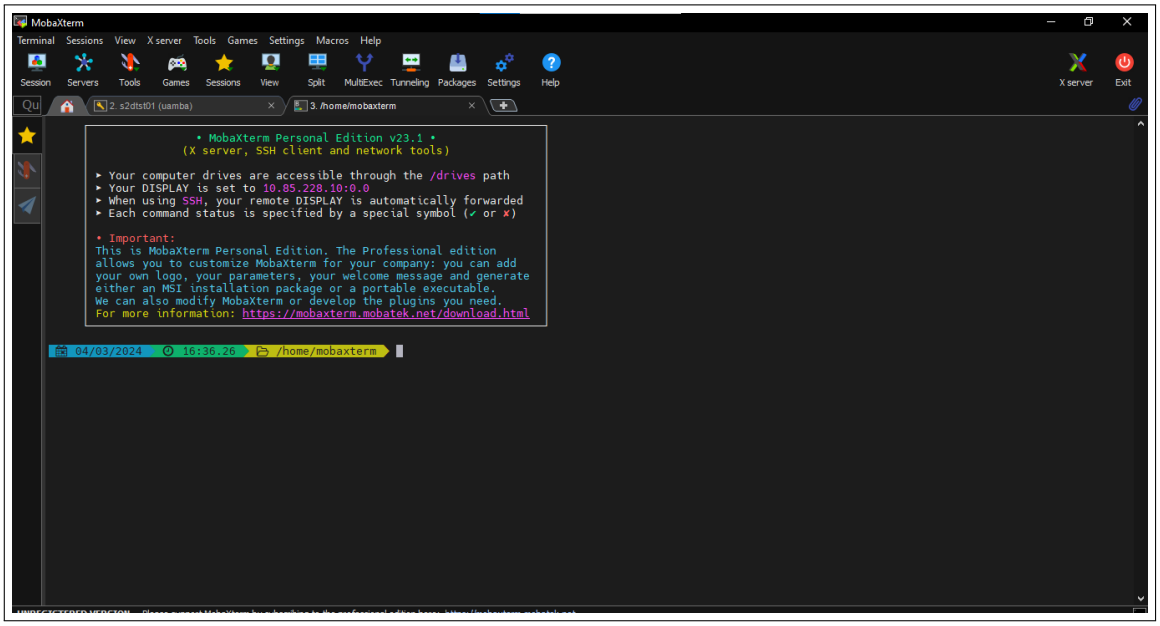
Figura 7 – Dados em 7z



Fonte: elaborado pelo autor (2024).

Após o download dos arquivos, pode-se iniciar o processo de extração visto que os arquivos estão comprimidos em formato 7z. Com o MobaXterm, é possível gerenciar diversos aspectos da infraestrutura de TI, incluindo redes (equipamentos Cisco), servidores (Linux e Windows), data centers (armazenamento e processamento) e também infraestruturas de laboratório na nuvem. Este software foi fundamental para o manuseio eficiente dos dados. Na Figura 12, mostrada abaixo, podemos ver a interface utilizada para converter os arquivos de 7z para txt, facilitando a extração e análise dos dados

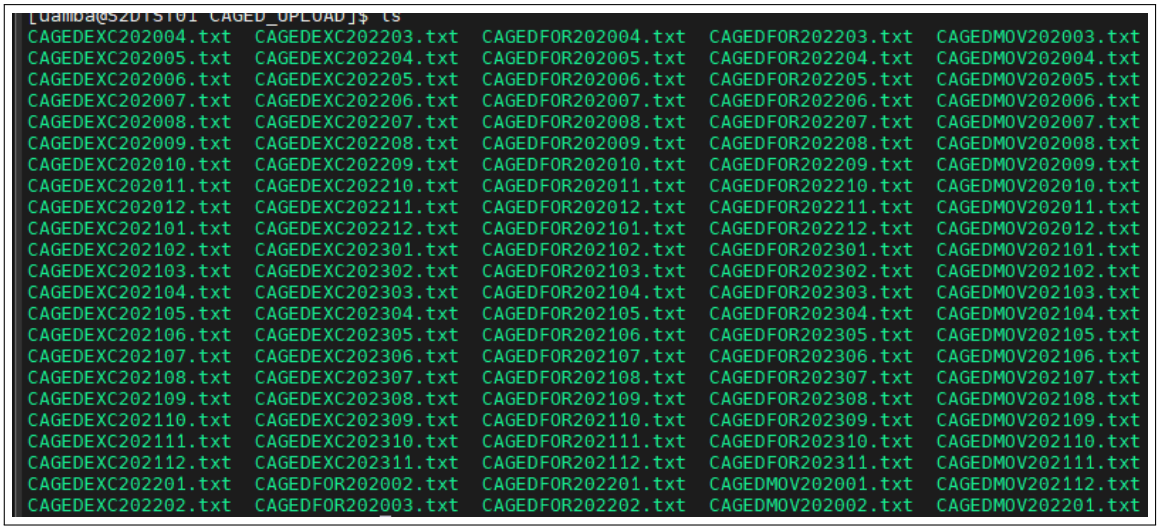
Figura 8 – Pagina inicial



Fonte: elaborado pelo autor (2024).

Na Figura 9, mostrada abaixo, são apresentados os dados já extraídos em formato txt, após serem processados no MobaXterm utilizando diversos comandos Linux específicos para o processamento de texto. Esta etapa é crucial para preparar e refinar os dados, permitindo uma análise mais eficaz e detalhada. A utilização desses comandos facilita a manipulação dos dados, otimizando o fluxo de trabalho e garantindo a precisão necessária para os próximos passos da pesquisa.

Figura 9 – Dados em txt



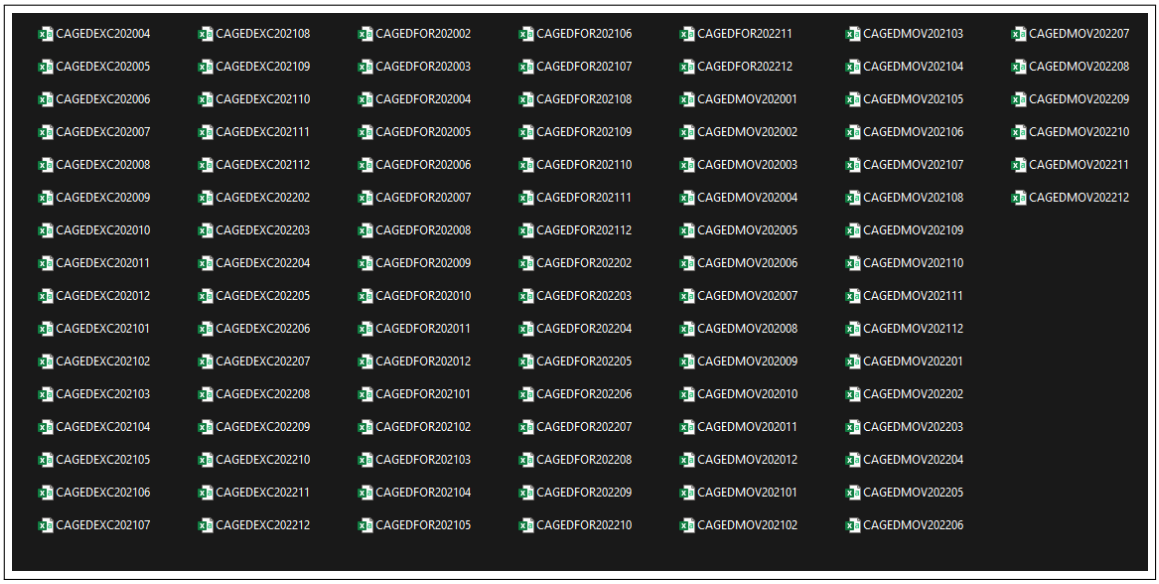
Fonte: elaborado pelo autor (2024).

Decidiu-se converter os dados para o formato CSV para maximizar a sua utilizabilidade. Para isso, utilizamos um script em Python que facilitou essa conversão, ajustando tanto o delimitador quanto a estrutura dos arquivos. Essa transformação tornou os dados mais acessíveis e fáceis de manusear, permitindo uma integração eficiente com outras ferramentas de análise e manipulação de dados. Com a conversão para CSV, os dados agora podem ser facilmente importados e processados em diversas plataformas de análise de dados.

Figura 10 – Script

Fonte: elaborado pelo autor (2024).

Figura 11 – Dados em CSV



Fonte: elaborado pelo autor (2024).

4.1.3 Cálculo da média

Para finalizar o processo podemos verificar a media dos dados obtidos no mongoDB e postgresSQL

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{4.1}$$

onde:

- N é o número total de observações no conjunto de dados,
- x_i representa cada valor observado,

- μ é a média dos valores.

4.1.4 Cálculo do desvio Padrão

O desvio padrão é uma medida que indica a dispersão ou variação dos valores de um conjunto de dados em relação à sua média. A fórmula para calcular o desvio padrão é:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{4.2}$$

onde σ é o desvio padrão, N é o número de observações, x_i são os valores individuais, e μ é a média dos valores.

Coeficiente de Variação

O coeficiente de variação em percentagem é calculado como:

$$CV = \left(\frac{\sigma}{\mu}\right) \times 100\% \tag{4.3}$$

4.2 Configuração o ambiente de desenvolvimento

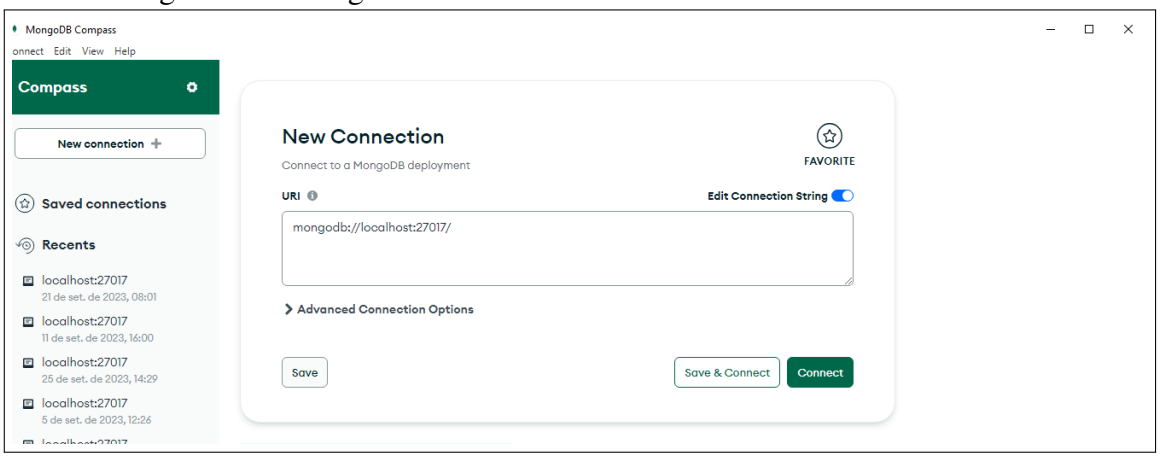
4.2.1 Configuração do Mongodb

O MongoDB Compass é uma ferramenta gráfica que facilita a interação com o MongoDB. A configuração foi realizada através do site oficial do MongoDB Compass, utilizando a versão 1.42.2 do software. O sistema operacional empregado foi o Windows 11, instalado em um notebook Lenovo equipado com um processador Core i5 de 8ª geração. O notebook também possui 1TB de memória interna e um SSD de 225GB, proporcionando excelente desempenho para o gerenciamento de dados

4.2.2 Configuração Python

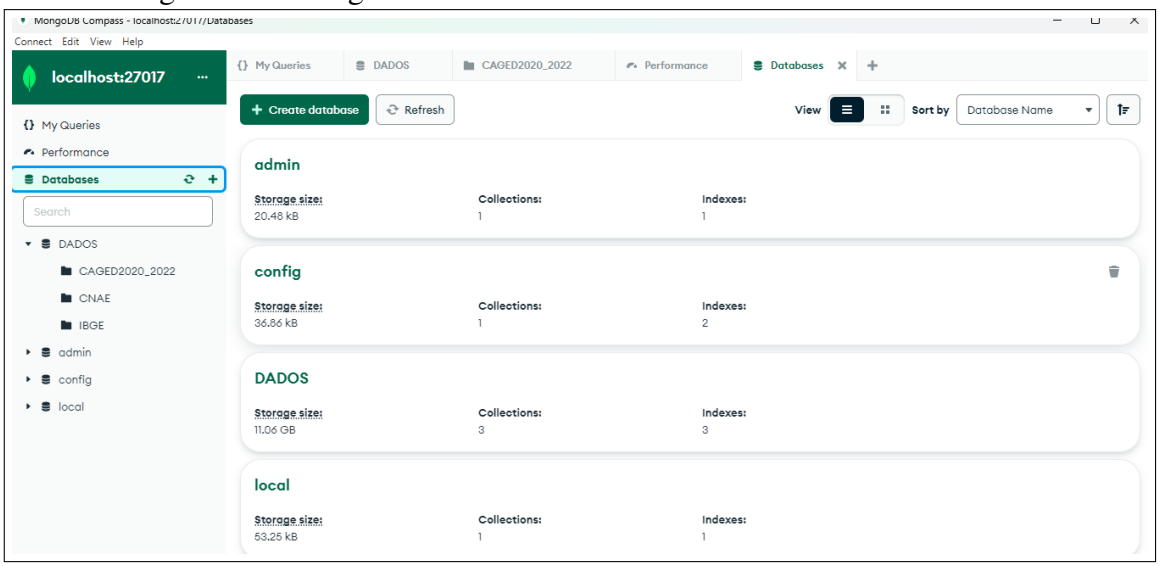
Durante a pesquisa sentiu-se a necessidade de buscar outras ferramentas para facilitar no proceso de otimização, nesse caso trabalhamos com a linguagem pyton para poder fazer as cargas dos dados como tambem fazer algumas conversões, para isso a IDE usada foi o Visual Studio, por preferencia, mas pode-se explorar varis IDE´s, como pycharm, jupter, Google Colab

Figura 12 – MongoDB



Fonte: elaborado pelo autor (2024).

Figura 13 – MongoDB

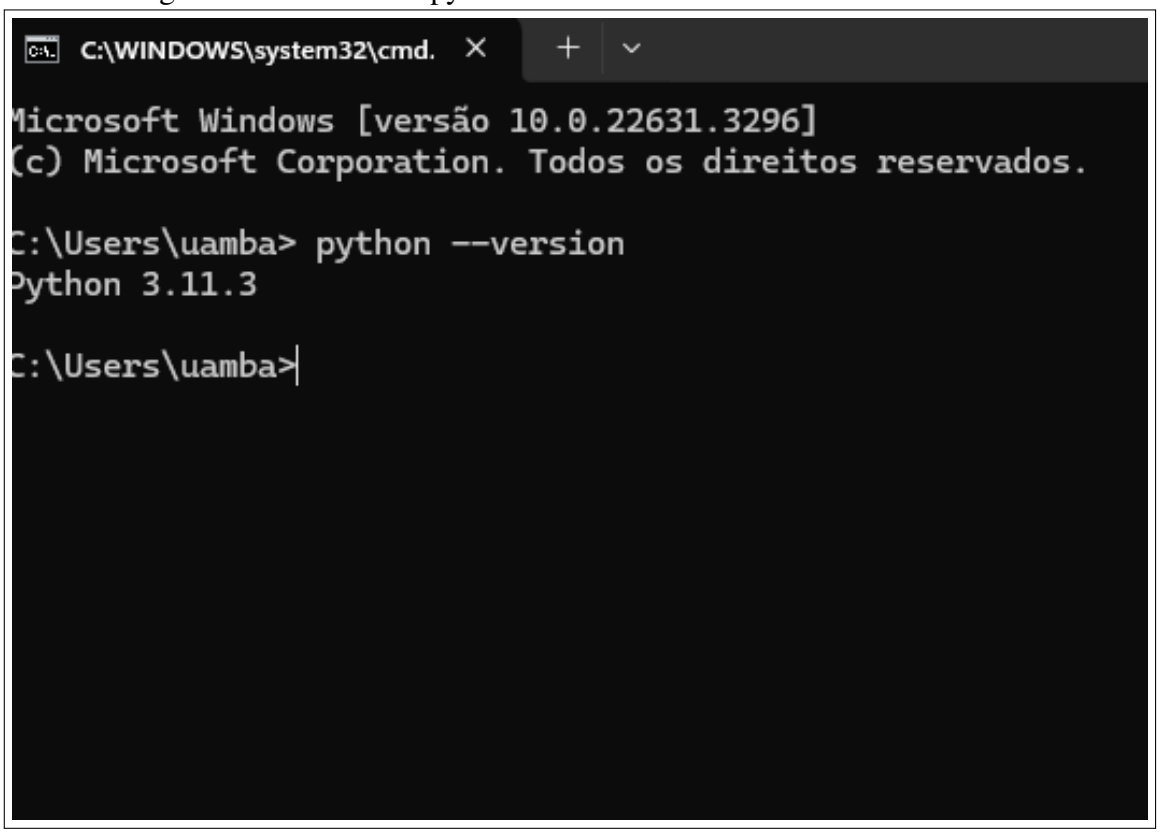


Fonte: elaborado pelo autor (2024).

entre outros. A linguagem de programação Python foi concebida por Guido Van Rossum em 1991, com o propósito de ser uma linguagem simples e acessível. Embora seja simples, Python é uma linguagem incrivelmente poderosa, capaz de suportar o desenvolvimento e administração de sistemas complexos. Uma das suas características distintivas é a sua legibilidade. Ao contrário de outras linguagens, que frequentemente exigem uma grande quantidade de marcações e palavras-chave, Python mantém uma sintaxe limpa e minimalista, facilitando a leitura e compreensão do código. Isso torna Python uma escolha popular para uma variedade de aplicações devido à sua clareza e facilidade de uso (PET ADS São Carlos, 2016).Para esta pesquisa foi usado Python 3.11.3.

Para instalar o Python, é possível acessar a documentação disponível no site oficial do Python e proceder com a instalação no sistema em uso. Na pesquisa realizada, foi utilizada

Figura 14 – Versão do python



Fonte: elaborado pelo autor (2024).

a IDE VS Code, sendo necessário acessar a extensão do Python na plataforma para realizar a instalação.

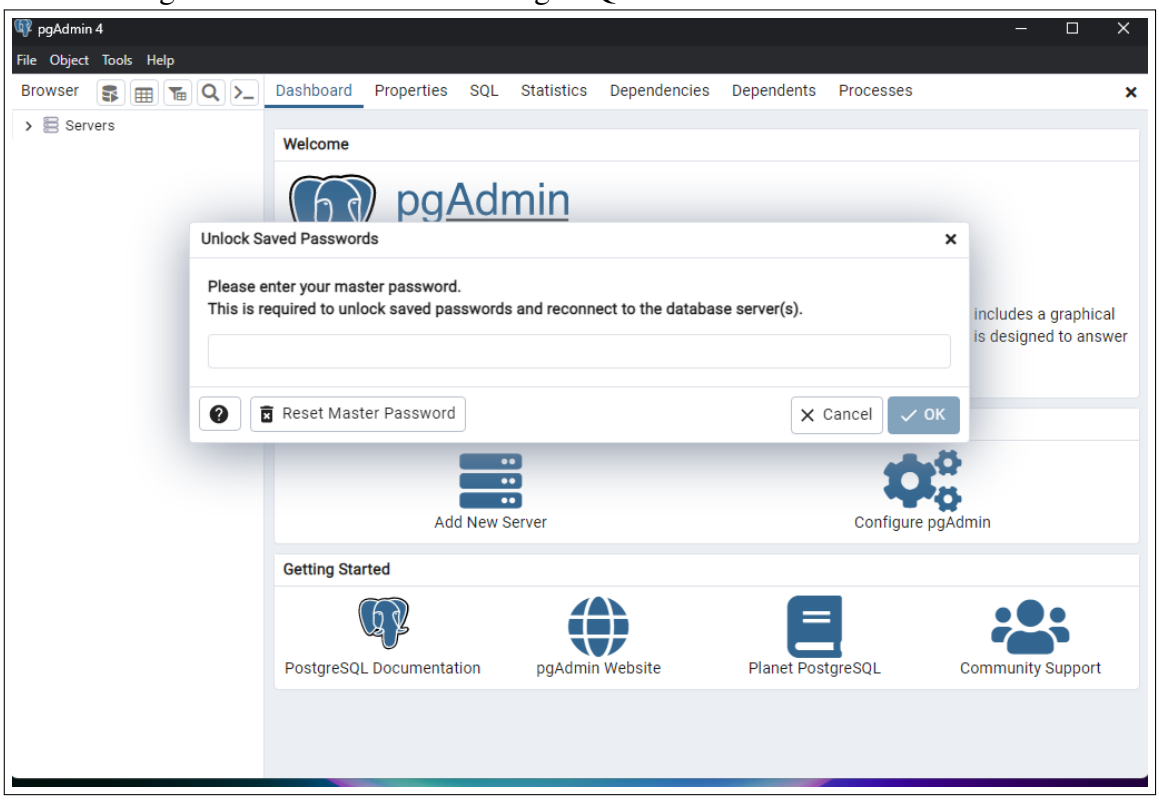
4.2.3 Configuração PostgreSQL

Para este experimento, foi realizada a instalação do PostgreSQL versão 15 no Windows 11, em um notebook Lenovo equipado com um processador Core i5 de 8ª geração. Durante o processo de instalação, foram definidos a senha do usuário "postgres"e a porta de conexão padrão. É essencial seguir as orientações do instalador e da documentação oficial para garantir uma instalação correta e funcional do PostgreSQL na versão desejada. O notebook utilizado possui 1TB de memória interna e um SSD de 225GB.

4.3 Carregar os dados do CAGED no ambiente do MongoDB e no PostgreSQL

Para carregar os dados do CAGED nos ambientes do MongoDB e PostgreSQL, optamos por automatizar o processo utilizando uma linguagem de programação. A escolha recaiu sobre o Python, devido à sua facilidade e agilidade no processamento de dados. Configuramos

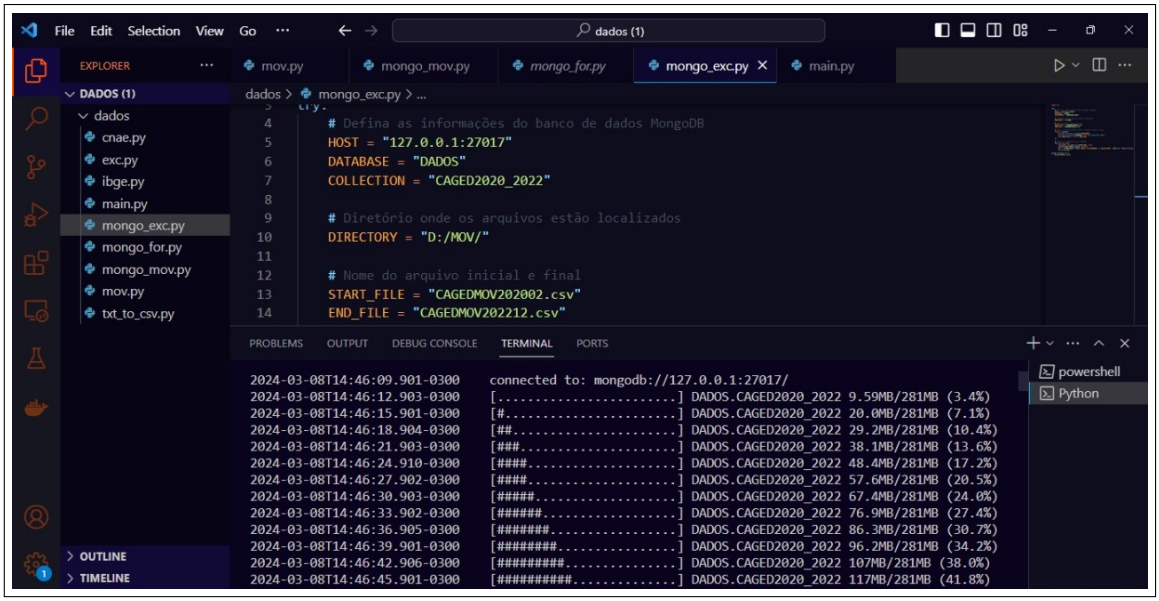
Figura 15 – Tela Inicial do PostgreSQL



Fonte: elaborado pelo autor (2024).

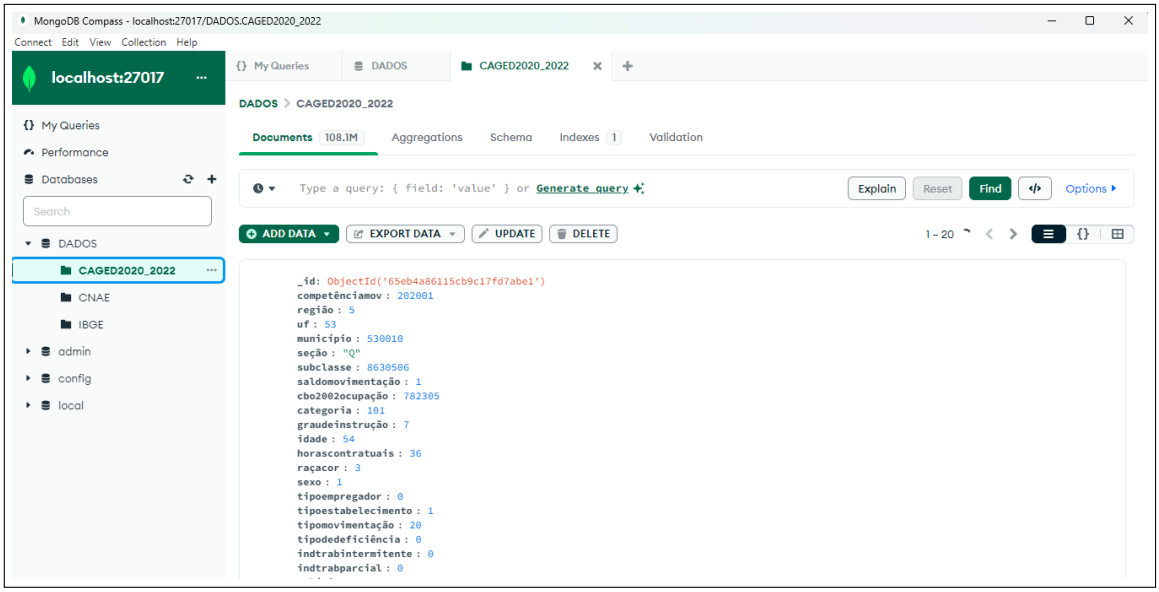
o ambiente de desenvolvimento no Visual Studio Code, embora outras IDEs, como Google Colab, Jupyter e PyCharm, também possam ser utilizadas. As Figuras 15, 16 , 17 e 18 ilustram o processo de carga dos dados utilizando scripts Python, demonstrando a eficiência e flexibilidade desta abordagem.

Figura 16 – Carregamento de dados no mongodb



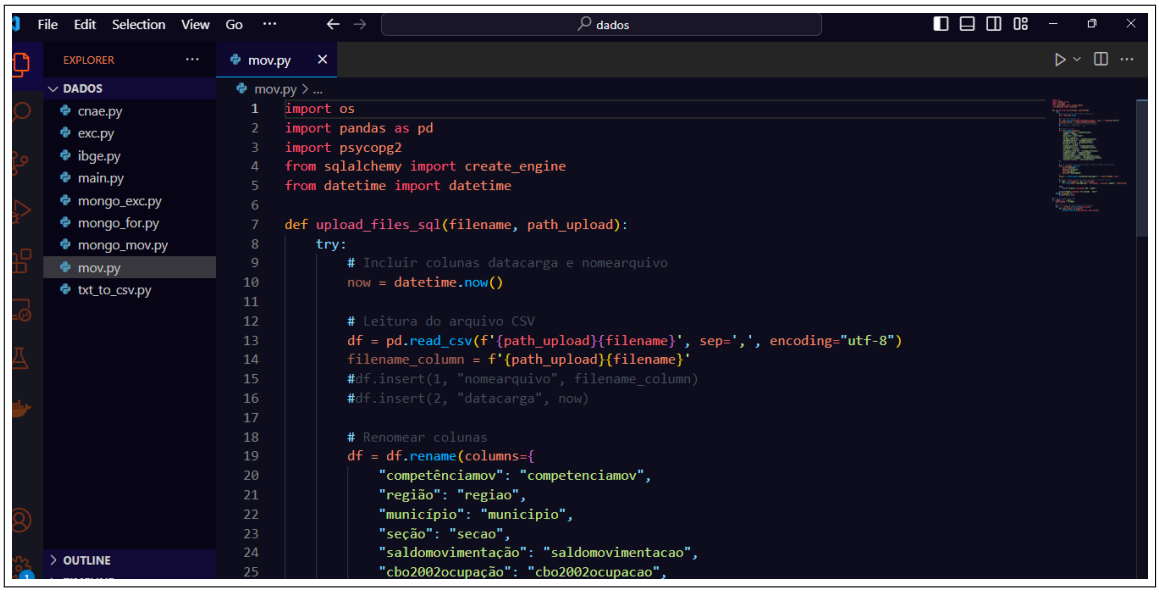
Fonte: elaborado pelo autor (2024).

Figura 17 – Dados Inseridos



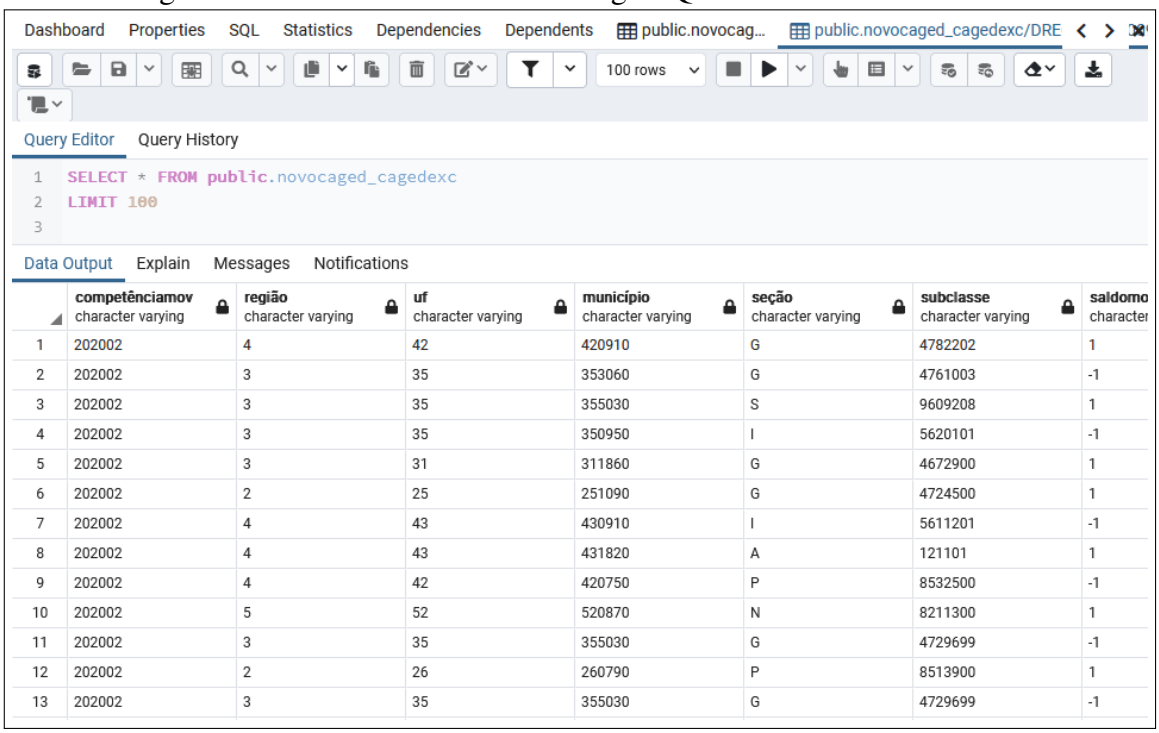
Fonte: elaborado pelo autor (2024).

Figura 18 – Carregamento de dados no PostgreSQL



Fonte: elaborado pelo autor (2024).

Figura 19 – Dados Inseridos NO PostgreSQL



	competênciamov character varying	região character varying	uf character varying	município character varying	seção character varying	subclasse character varying	saldomo character
1	202002	4	42	420910	G	4782202	1
2	202002	3	35	353060	G	4761003	-1
3	202002	3	35	355030	S	9609208	1
4	202002	3	35	350950	I	5620101	-1
5	202002	3	31	311860	G	4672900	1
6	202002	2	25	251090	G	4724500	1
7	202002	4	43	430910	I	5611201	-1
8	202002	4	43	431820	A	121101	1
9	202002	4	42	420750	P	8532500	-1
10	202002	5	52	520870	N	8211300	1
11	202002	3	35	355030	G	4729699	-1
12	202002	2	26	260790	P	8513900	1
13	202002	3	35	355030	G	4729699	-1

Fonte: elaborado pelo autor (2024).

5 RESULTADOS

A seção de experimento apresenta as análises e conclusões derivadas da implementação e testes realizados na solução para queries complexas em Big Data utilizando MapReduce com o banco de dados MongoDB, conforme estudado com os dados do CAGED.

Nesta seção, descrevemos e discutimos os resultados obtidos a partir da aplicação prática da solução, bem como as considerações e implicações desses resultados para o processamento eficiente de consultas complexas em grandes conjuntos de dados.

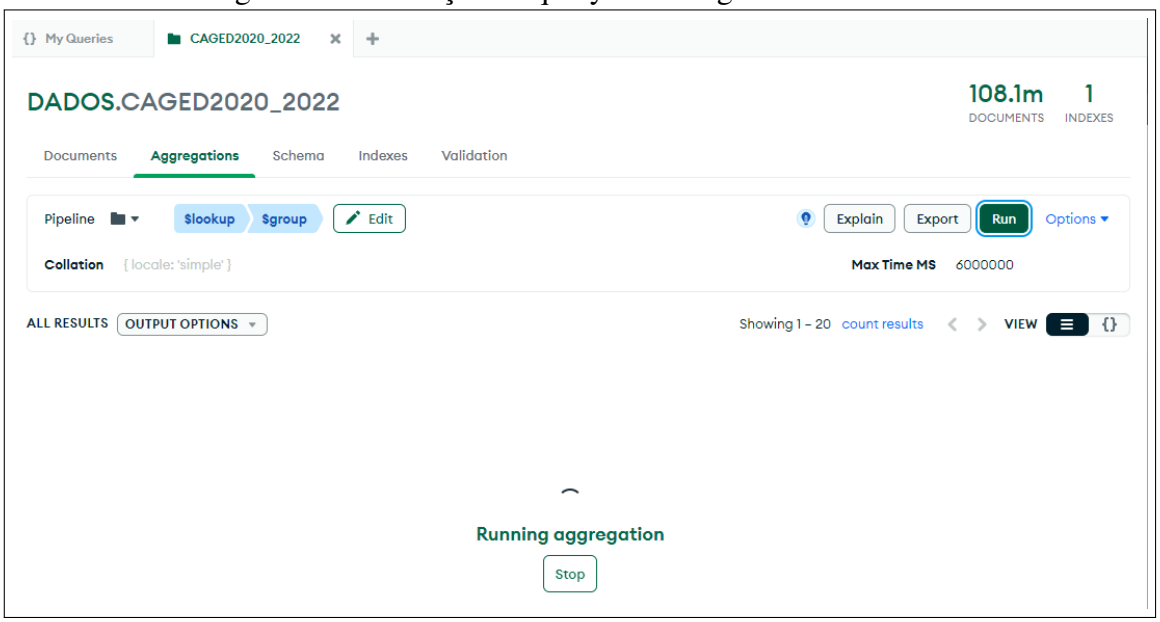
Neste caso, exploramos as potenciais melhorias, desafios encontrados e as perspectivas futuras decorrentes dos resultados alcançados, visando contribuir para aprimorar a eficiência e a escalabilidade do processamento de Big Data, para poder termos insighs valiosos tabem usamos dados de IBGE, para saber sobre a polulação e também o nome do municipio, e CNAE para saber a atividade que teve desempregados ou empregados

5.1 Implementação da Solução: Integração do MapReduce com MongoDB

A implementação da solução para queries complexas em Big Data utilizando MapReduce com o banco de dados MongoDB foi um processo fundamental neste trabalho. Essa integração desempenha um papel crucial na otimização do processamento de consultas complexas, proporcionando uma abordagem eficiente e escalável para lidar com grandes volumes de dados. Concebida com foco na performance e na flexibilidade, a implementação da solução oferece uma maneira robusta de processar consultas complexas de forma distribuída e paralela. Nesta seção, serão detalhados os componentes, a arquitetura e os resultados obtidos com a implementação da solução, destacando como ela se tornou uma ferramenta estratégica para lidar com os desafios de consultas complexas em Big Data.

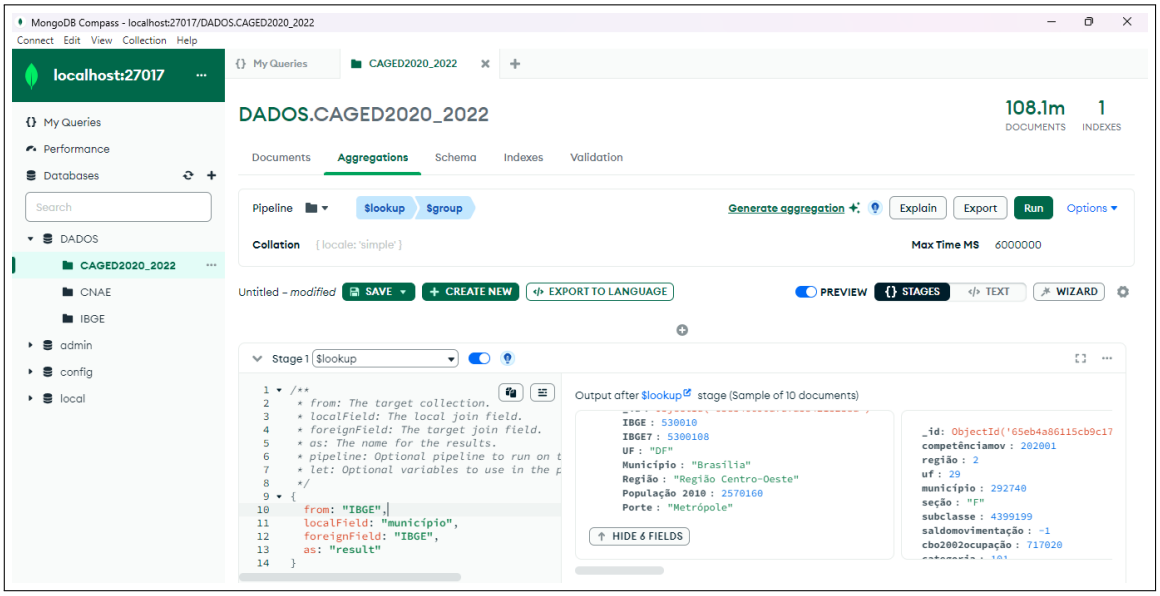
Neste experimento importamos o dataset do CAGED com 118,7 milhões de documentos e 12.3GB, executamos a query abaixo com recurso de Aggregation Pipeline do MongoDB que utiliza em seus estágios o MapReduce.

Figura 20 – Execução da query em mongoDB



Fonte: elaborado pelo autor (2024).

Figura 21 – Execução da query em mongoDB



Fonte: elaborado pelo autor (2024).

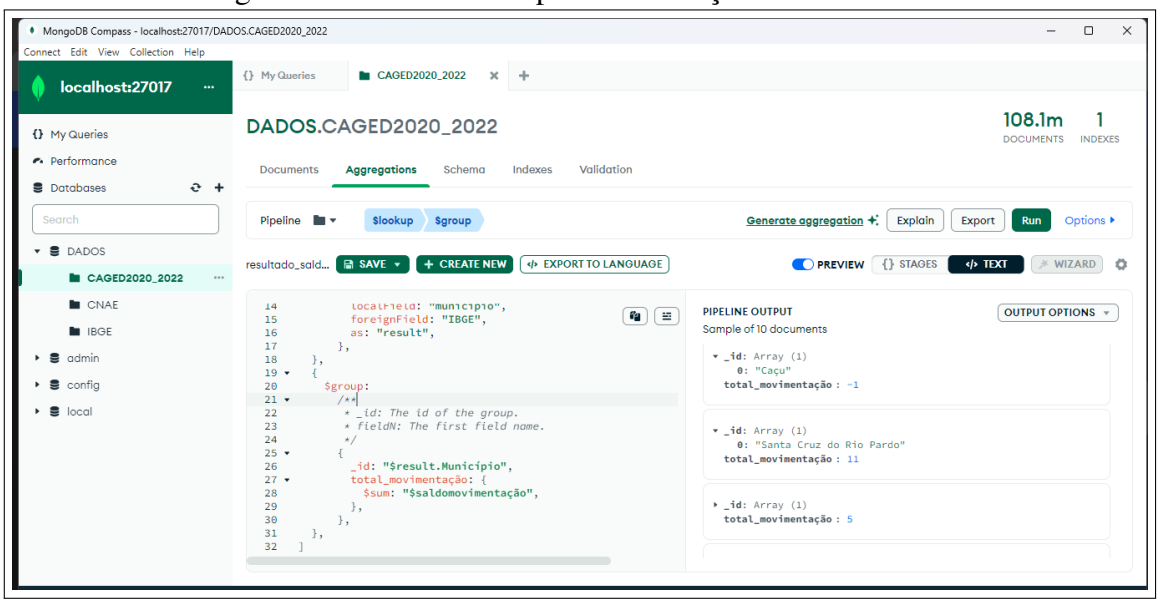
Depois da execução podemos verificar como os dados são entregues

5.2 Avaliação do desempenho do MongoDB

5.2.1 Eficiência

Os resultados foram examinados considerando o tempo de resposta das consultas complexas executadas no MongoDB, fornecendo uma visão abrangente do desempenho da solução. Discutiu-se a eficiência do MongoDB em processar consultas distribuídas e paralelas,

Figura 22 – Resuldado depois da execução



Fonte: elaborado pelo autor (2024).

aproveitando sua arquitetura de armazenamento orientada a documentos e indexação eficaz. Além disso, os resultados foram comparados com as expectativas iniciais e com o desempenho de outros sistemas de gerenciamento de banco de dados, levando em consideração o tempo de processamento e o consumo de recursos computacionais.

5.2.2 Escalabilidade

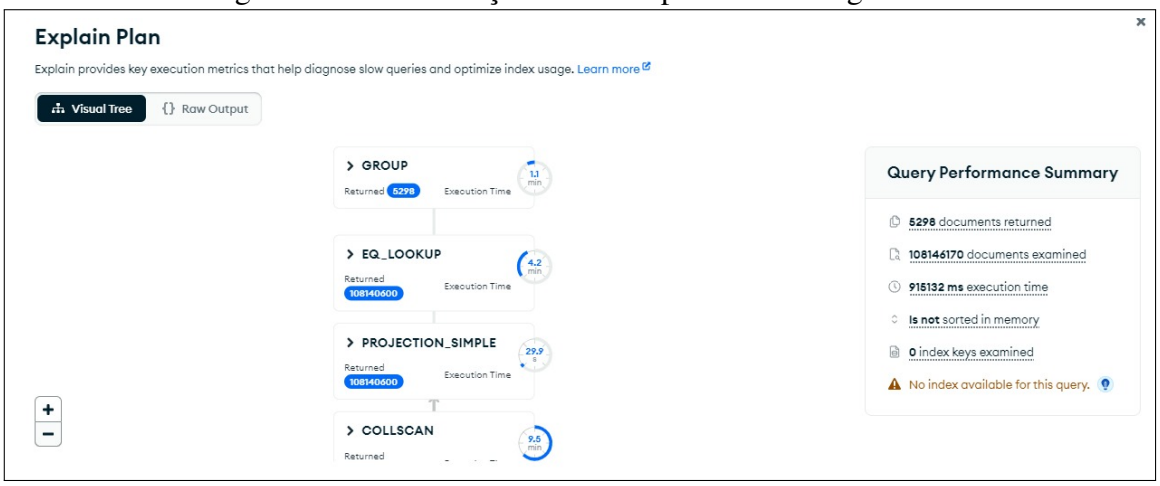
Foi avaliada a capacidade do MongoDB em escalar horizontalmente para lidar com conjuntos de dados em expansão e aumentar a carga de consultas sem prejudicar significativamente o desempenho. A discussão abordou como a arquitetura distribuída e o suporte a sharding do MongoDB contribuem para sua escalabilidade, permitindo a distribuição eficiente de dados e consultas entre vários nós. Os resultados foram analisados em termos de escalabilidade linear ou não linear, considerando o crescimento do tamanho do conjunto de dados e da carga de consultas.

5.2.3 Capacidade de Lidar com Consultas Complexas:

Capacidade de Lidar com Consultas Complexas: Avaliou-se a habilidade do MongoDB em lidar com consultas complexas, incluindo operações de junção, agregação, filtragem e projeção em grandes volumes de dados. A discussão explorou o suporte oferecido pelo MongoDB, como sua estrutura flexível de documentos, consultas ad hoc e recursos de análise integrados. Os resultados foram examinados em relação à eficácia e eficiência na execução de

consultas complexas, identificando possíveis limitações e propondo melhorias ou otimizações.

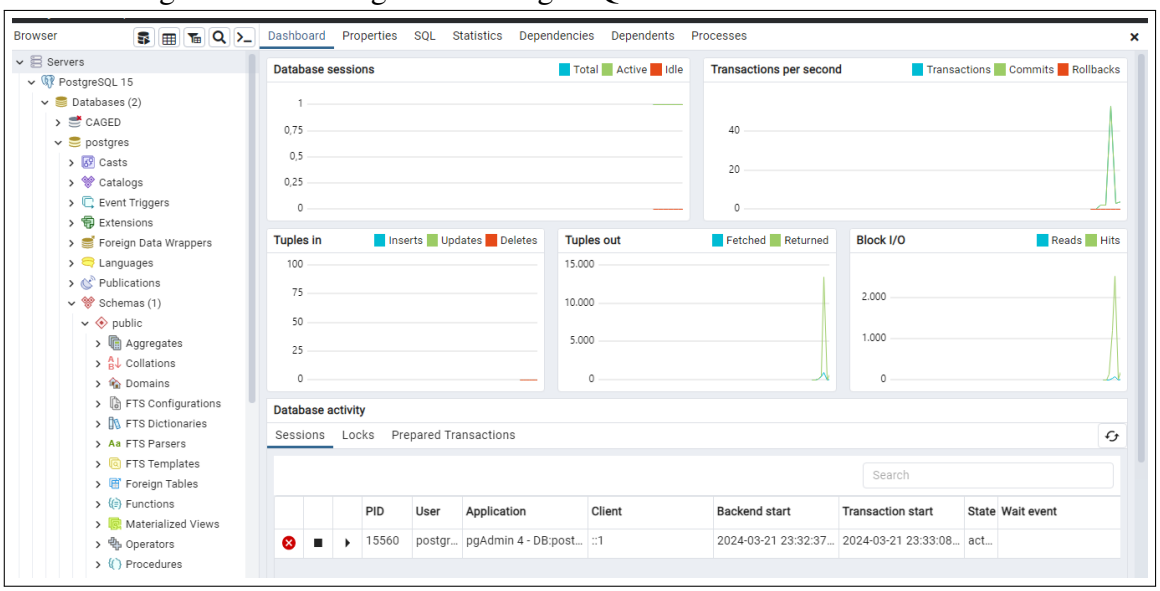
Figura 23 – Demostração do desempenho no MongoDB



Fonte: elaborado pelo autor (2024).

Para avaliar a flexibilidade do MongoDB, uma vez que estamos lidando com uma solução NoSQL, optamos por conduzir testes comparativos utilizando um banco de dados relacional. Escolhemos o PostgreSQL como ferramenta para essa análise. Ao realizar os mesmos testes de consulta em ambas as plataformas, pudemos observar uma diferença notável no tempo de processamento. O MongoDB demonstrou uma eficiência superior em comparação com o PostgreSQL, apresentando tempos de resposta mais rápidos. Essa diferença destaca a capacidade do MongoDB de lidar de forma eficaz com consultas complexas em grandes volumes de dados, ressaltando sua flexibilidade e desempenho em ambientes de Big Data.

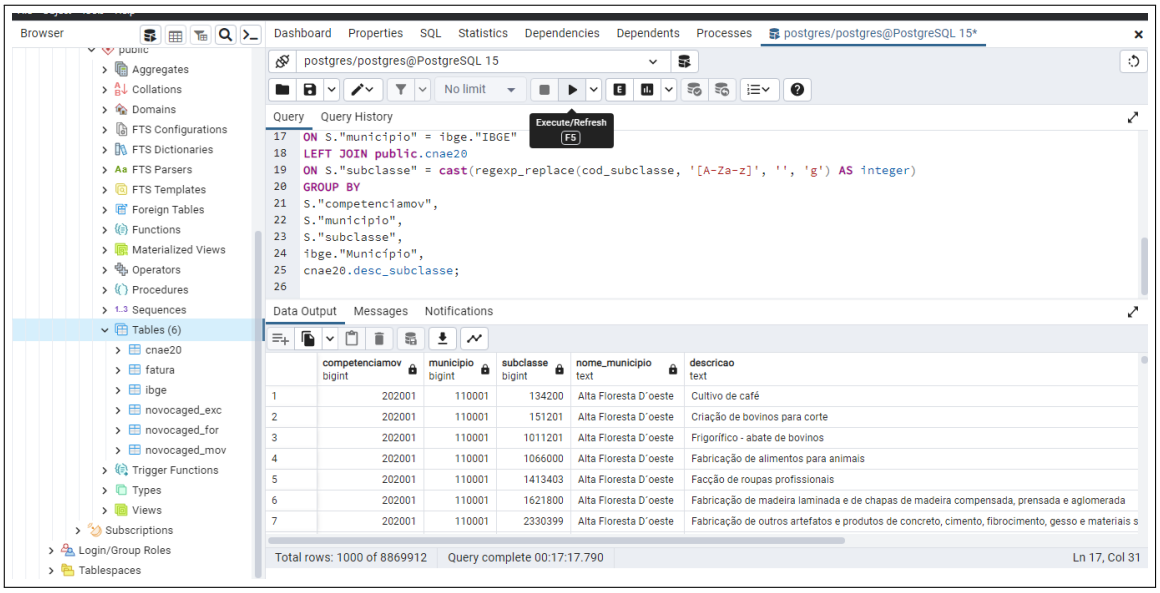
Figura 24 – visao geral do PostgreSQL



Fonte: elaborado pelo autor (2024).

Após compararmos a execução da mesma consulta tanto no MongoDB quanto no PostgreSQL, observamos uma diferença notável no tempo de processamento. No PostgreSQL, a consulta levou 22 minutos e 45 segundos para ser concluída, o que representa um período significativamente mais longo em comparação com o MongoDB. Essa demora na execução é problemática, especialmente considerando que a finalidade da consulta é simplesmente obter uma informação solicitada pelo usuário. Essa disparidade no tempo de resposta ressalta a eficiência superior do MongoDB ao lidar com consultas complexas e grandes volumes de dados em comparação com o PostgreSQL.

Figura 25 – Resusltado no PostgreSQL



Fonte: elaborado pelo autor (2024).

5.2.4 Análise comparativa do tempo de processamento das consultas entre o MongoDB e o PostgreSQL.

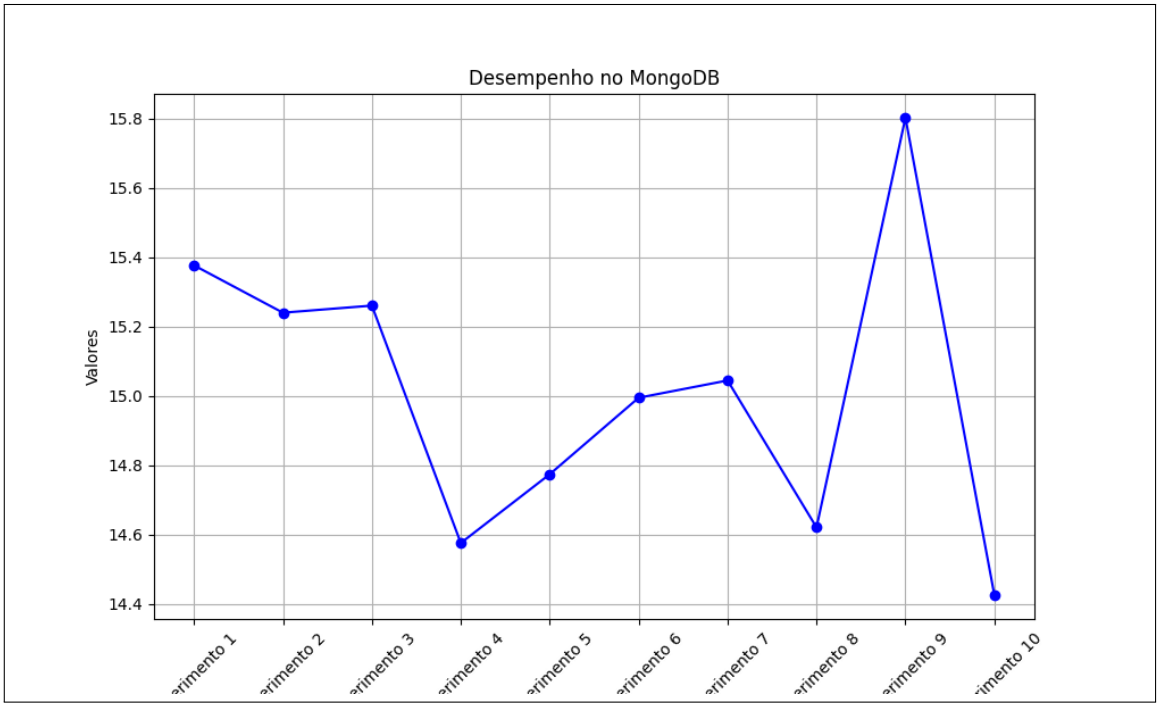
Para poder ter a certeza dos resultados obtidos outrora tivemos que trazer varios experimentos para cada ferramenta, segue a baixo na tabela os minutos

Tabela 1 – Comparação de Tempo

Esperimento(E)	MongoDB	PostgreSQL
E1	15,376	17,456
E2	15,240	19,765
E3	15,260	29,786
E4	14,575	22,129
E5	14,773	20,909
E6	14,994	18,765
E7	15,044	16,987
E8	14,620	18,765
E9	15,802	17,401
E10	14,251	20,005

Para melhor a visualização dos dados geramos um grafico que mosytra detalhamento do desempenho das feramentas

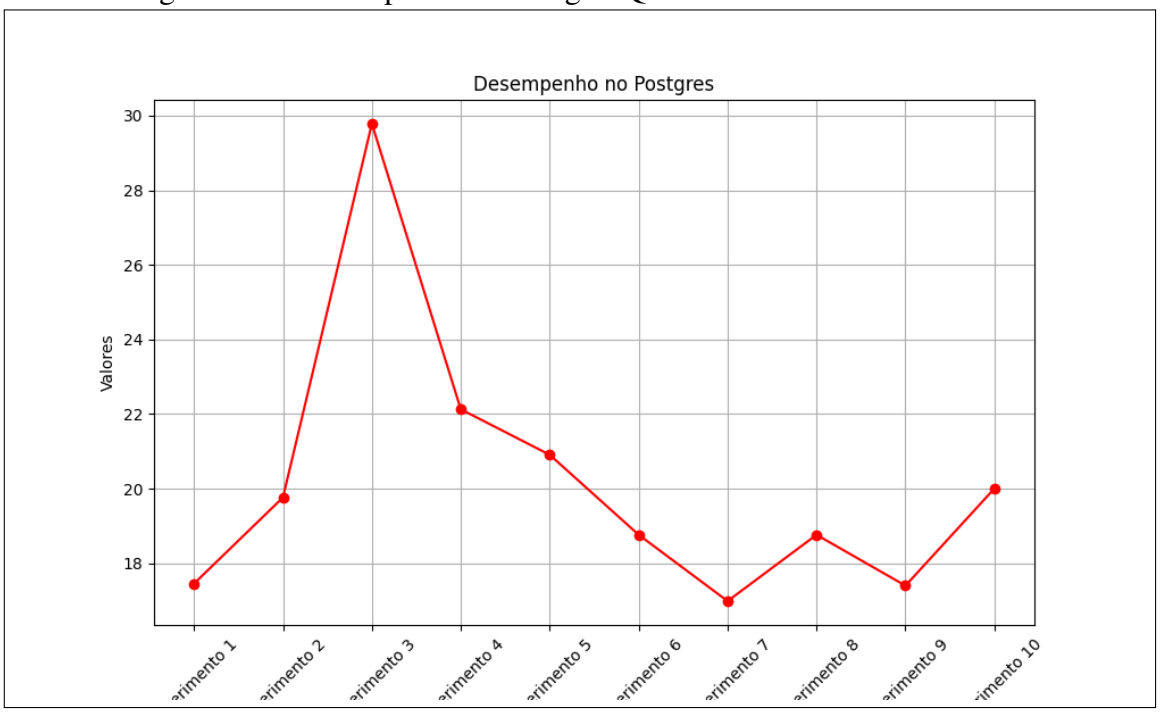
Figura 26 – Desempenho no MongoDB



Fonte: elaborado pelo autor (2024).

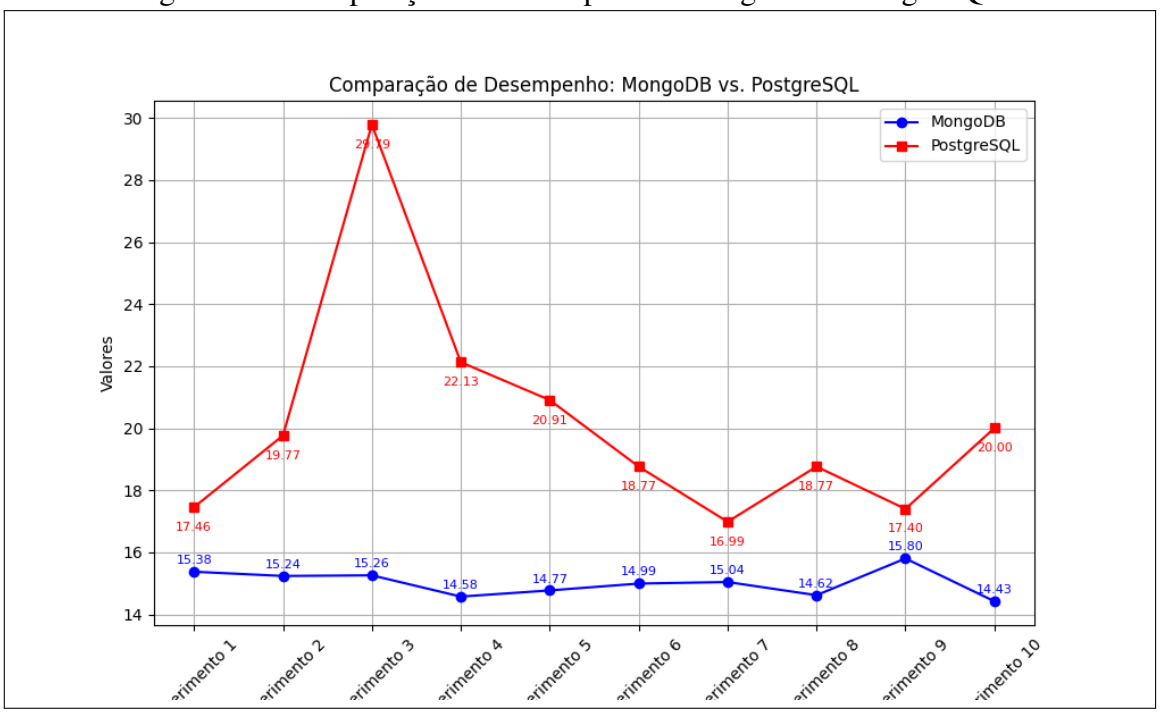
Agora veremos como eles se comportam juntando os dois graficos no mesmo sistema.

Figura 27 – Desempenho no PostgreSQL



Fonte: elaborado pelo autor (2024).

Figura 28 – Comparação de Desempenho: MongoDB e PostgreSQL



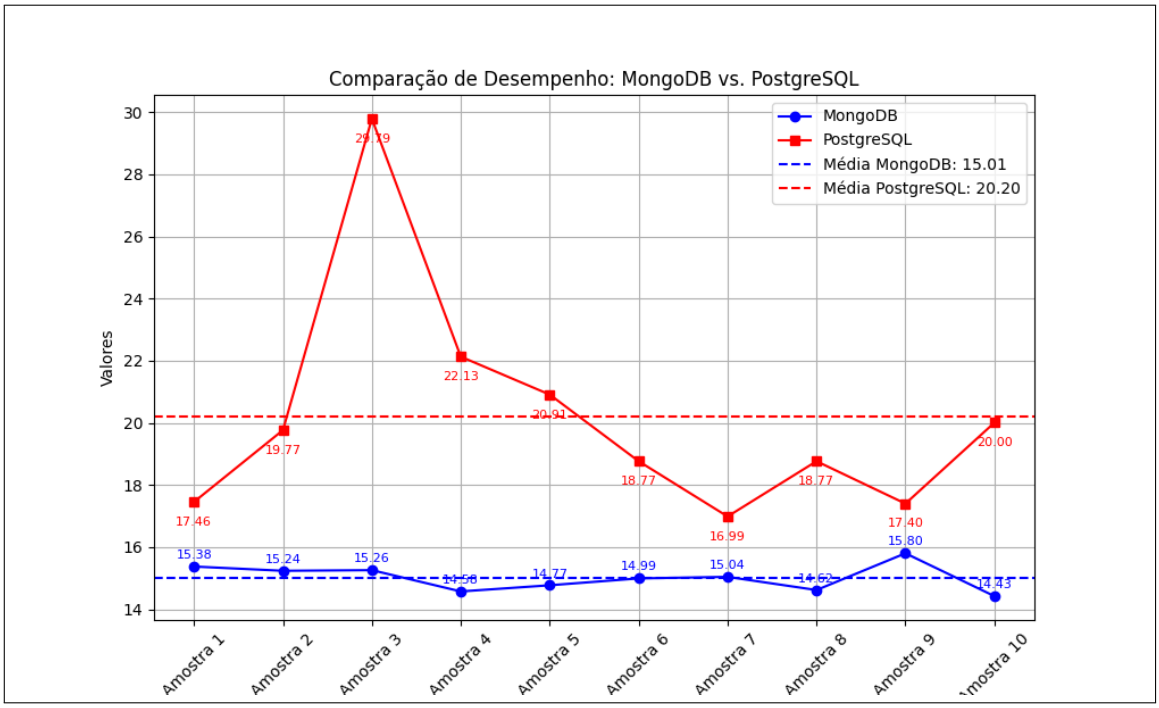
Fonte: elaborado pelo autor (2024).

5.2.5 Demostração da média

Para finalizar o processo podemos verificar a media dos dados obtidos no mongoDB e postgresSQL

MongoDB	PostgreSQL
15.011433	20.197

Figura 29 – Cálculo da media

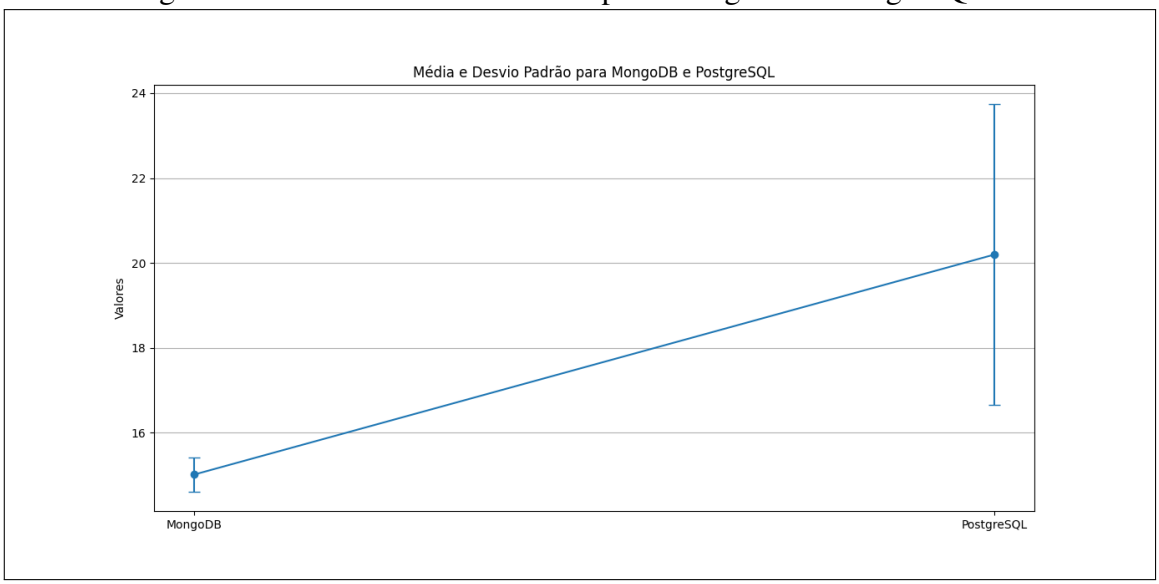


Fonte: elaborado pelo autor (2024).

5.2.6 Demostração do desvio Padrão

MongoDB	PostgreSQL
0.402312 (2.68%)	3.552 (17.59%)

Figura 30 – Media e Desvio Padrão para MongoDB e PostgreSQL



Fonte: elaborado pelo autor (2024).

6 CONCLUSÕES

A combinação de MapReduce e MongoDB oferece uma maneira promissora de lidar com a complexidade de consultas em grandes conjuntos de dados, como o CAGED. Os resultados preliminares deste estudo mostram avanços significativos no fornecimento de respostas rápidas e eficientes às questões mais complexas. A combinação de MapReduce e MongoDB oferece uma maneira promissora de lidar com a complexidade de consultas em grandes conjuntos de dados, como o CAGED. Os resultados preliminares deste estudo mostram avanços significativos no fornecimento de respostas rápidas e eficientes às questões mais complexas.

A capacidade de processar dados em paralelo com o MapReduce e a flexibilidade do MongoDB para armazenar e consultar dados não estruturados se complementam e fornecem uma solução sólida para grandes problemas.O MongoDB mostrou melhor consistência e menor desvio padrão do que o PostgreSQL, conforme indicado por quanto mais baixo coeficiente de variação no MongoDB, o que indica resultados mais homogêneos. Especificamente, o desvio padrão do MongoDB foi de 0,402312 (2,68%), enquanto o do PostgreSQL foi de 3,552 (17,59%), destacando a eficiência do MongoDB em alcançar resultados consistentes em comparação com o PostgreSQL.

Testes práticos em larga escala e simulações mostram que a solução proposta tem potencial para otimizar significativamente o processamento de grandes dados CAGED. No entanto, entendemos a importância de testar e validar totalmente esta solução em ambientes reais para garantir sua eficácia e escalabilidade sob diversas condições e cargas de trabalho.Além disso, é importante considerar o desenvolvimento e a expansão contínuos da solução sob diversas condições. Isso pode exigir a implementação de técnicas adicionais de otimização, melhorias na infraestrutura de processamento distribuído e a adoção de melhores práticas de engenharia de software para garantir viabilidade e escalabilidade a longo prazo.A implementação prática desta abordagem pode beneficiar significativamente pesquisadores e analistas desafiadores, e profissionais de dados.

REFERÊNCIAS

AKHTAT, S. M. F. **Big Data Architect's Handbook**. Birmingham: Pack Publishing, 2018.

ANAND, G. **Big Data Analytics: A Comprehensive Guide**. [S.l.]: Apress, 2019.

BREWER, E. A.; GILBERT, S. **CAP theorem: Revisiting the basics**. 2000. 51-57 p.

BRUCE, W.; LENITA, D.; PAUL, D. B. Perspectives on big data. **Journal of Marketing Analytics**, v. 1, n. 4, p. 187–201, 2013.

BRUYNE, P. de; HERMAN, J.; SCHOUTHEETE, M. de. **Dinâmica da pesquisa em ciências sociais**. Rio de Janeiro: Editora Zahar, 1982.

CHEN, J. *et al.* Big data analytics in government: A systematic literature review. **Nome do Periódico**, Volume, n. Número, p. Páginas, 2023. URL do artigo, se disponível.

CODD, E. F. **A relational model of data for large shared data banks**. 1970. 377-387 p.

DATE, C. J. **An introduction to database systems**. [S.l.]: Addison-Wesley, 2003.

DAVENPORT, T. H. **Big data at work: Dispelling the myths, uncovering the opportunities**. Boston: Harvard Business Press, 2014.

DUARTE, T. A possibilidade da investigação a 3: reflexões sobre triangulação (metodológica). **CIES e Working Papers**, v. 60, p. 1–24, 2009.

EFRON, B.; HASTIE, T. **Computer Age Statistical Inference: Algorithms, Evidence, and Data Science**. Cambridge, UK: Cambridge University Press, 2016. v. 5. (Institute of Mathematical Statistics Monographs, v. 5). ISBN 9781107149892.

EMPREGADOS, C. G. de. Caged. 1965.

GANDOMI, A.; HAIDER, M. **Getting Started with Data Science: Making Sense of Data with Analytics**. [S.l.]: Pearson, 2015. ISBN 978-0-13-399102-4.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002.

GILBERT, S.; LYNCH, N. Perspectivas do teorema cap. **Computer**, v. 45, n. 2, p. 30–36, 2012.

GUPTA, R.; GUPTA, S.; SINGHAL, A. Big data: An overview. **International Journal of Computer Trends and Technology**, v. 9, n. 5, p. 1–3, 2014.

LANEY, D. 3d data management: Controlling data volume, velocity, and variety. **Blog Gartner**, 2001. Disponível em: <https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>.

LEEFLANG, P. S. H.; VERHOEF, P. C.; DAHLSTRöM, P.; FREUNDT, T. Challenges and solutions for marketing in a digital era. **European Management Journal**, v. 32, n. 1, p. 1–12, 2014.

MACHADO, C. E. Mapreduce vs. spark: Um comparativo entre as principais ferramentas de big data. **InfoQ**, 2023. Disponível em: <https://www.infoq.com/br/articles/mapreduce-vs-spark/>.

MINELLI, M.; CHAMBERS, M.; DHIRAJ, A. **Big data, big analytics: Emerging business intelligence and analytic trends for today's businesses**. Hoboken: John Wiley Sons, 2013.

NOVO, R.; NEVES, J. M. S. D. Inovação na inteligência analítica por meio do big data: Características de diferenciação da abordagem tradicional. In: **VIII Workshop de Pós-Graduação e Pesquisa do Centro Paula Souza**. São Paulo, SP, Brasil: [s.n.], 2013. p. 32–44.

OCDE. **Data for Good: A Decade of Digital Innovation in Public Services**. 2023. Relatório da Organização para a Cooperação e Desenvolvimento Econômico. URL do relatório, se disponível.

OHLHORST, F. **Big Data Analytics: Turning Big Data into Big Money**. Hoboken, N.J.: John Wiley & Sons, 2013.

OREN, M. e. a. **NoSQL databases: A survey and decision guidance**. 2013. 1-34 p.

OUSSOUS, A.; BENJELLOUN, F. Z.; LAHCEN, A. A.; BELFKIH, S. Big data technologies: A survey. **Journal of King Saud University-Computer and Information Sciences**, v. 30, n. 4, p. 431–448, 2018.

PET ADS São Carlos. **Apostila Python**. 2016. Disponível em: <http://antigo.scl.ifsp.edu.br/portal/arquivos/2016.05.04_Apostila_Python_-_PET_ADS_S%C3%A3o_Carlos.pdf>. Disponível em: <http://antigo.scl.ifsp.edu.br/portal/arquivos/2016.05.04_Apostila_Python_-_PET_ADS_S%C3%A3o_Carlos.pdf>.

PRITCHETT, D. Base: An acid alternative. **ACM Queue**, v. 6, n. 5, 2008.

SALINAS, J. H.; LEMUS, R. **Tecnologias de Big Data**. [S.l.]: Editora Elsevier, 2017. ISBN 978-85-352-8250-2.

SAS Institute Inc. **What is Big Data?** 2022. Disponível em: <https://www.sas.com/en_us/insights/big-data/what-is-big-data.html>.

SILVA, I. M. d.; CAMPOS, F. C. d. New perspectives using big data: A study of bibliometric 2000-2012. In: **Proceedings of the International Conference on Information Systems and Technology Management - CONTECSI**. São Paulo, SP, Brasil: [s.n.], 2014. p. 11.

SIMON, P. **Too big to ignore**. Hoboken: John Wiley Sons, 2013.

STONEBRAKER, P. A.; OOI, E. E.; SHASHA, D. Nosql databases: A survey of current trends and future research directions. **ACM Computing Surveys**, ACM, v. 40, n. 3, p. 1–39, 2007.

VALUE, I. I. for B. **Data-Driven Decision Making: The Engine of Government Transformation**. 2023. Relatório do IBM Institute for Business Value. URL do relatório, se disponível.

VELOSO, L. **Desenvolvimento de um sistema de recomendação baseado em aprendizado de máquina para a biblioteca digital de teses e dissertações da UEPG**. 2019. Disponível em: <https://tede2.uepg.br/jspui/bitstream/prefix/127/1/Lays%20Veloso.pdf>.

VOGELS, W. Eventually consistent, scalable web services. **ACM Queue**, v. 6, n. 6, 2008.

WEI, Z.; PIERRE, G.; CHI, C. Transações escalonáveis para aplicativos da web na nuvem. In: **Proc. da Conferência Euro-Par**. [S.l.: s.n.], 2009.

YEOH, W.-K.; KORONIUS, A. Critical success factors for business intelligence systems. **Journal of Computer Information Systems**, v. 50, n. 3, p. 23–32, 2010.

YIN, R. K. **Estudo de caso: planejamento e métodos**. 2. ed. Porto Alegre: Bookman, 2001.

YIN, R. K. **Estudo de caso: planejamento e métodos**. 5. ed. Porto Alegre: Bookman, 2015.