UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ENIEDSON FABIANO PEREIRA DA SILVA JÚNIOR

BUSCA EM CATÁLOGO DE PRODUTOS: UMA COMPARAÇÃO ENTRE BANCO DE DADOS RELACIONAL E MOTOR DE BUSCA

CAMPINA GRANDE - PB

2023

ENIEDSON FABIANO PEREIRA DA SILVA JÚNIOR

BUSCA EM CATÁLOGO DE PRODUTOS: UMA COMPARAÇÃO ENTRE BANCO DE DADOS RELACIONAL E MOTOR DE BUSCA

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador : Professor Dr. Cláudio de Souza Baptista

CAMPINA GRANDE - PB

2023

ENIEDSON FABIANO PEREIRA DA SILVA JÚNIOR

BUSCA EM CATÁLOGO DE PRODUTOS: UMA COMPARAÇÃO ENTRE BANCO DE DADOS RELACIONAL E MOTOR DE BUSCA

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Cláudio de Souza Baptista

Orientador – UASC/CEEI/UFCG

Professor Dr. Maxwell Guimarães de Oliveira

Examinador – UASC/CEEI/UFCG

Professora Dra. Melina Mongiovi Cunha Lima Sabino

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 17 de novembro de 2023.

CAMPINA GRANDE - PB

RESUMO

O objetivo do TCE-AC é fiscalizar as despesas e receitas dos municípios e do estado do Acre. Para tanto, nos últimos anos tem modernizado a sua forma de trabalho. Em particular, o acesso rápido aos preços praticados é fundamental para a fiscalização e também para a população em geral. Para isso, o Banco de Preços é utilizado, sendo alimentado por uma base de dados em constante crescimento e que, atualmente, conta com dezenas de milhões de registros de notas fiscais. Diante desse cenário, por utilizar de banco de dados relacionais para a realização das consultas e devido a grande massa de dados existente, o sistema em questão acaba demorando para produzir resultados em diversas situações, além de retornar resultados pouco relevantes em algumas situações. Para solucionar o problema, propõe-se a implantação do Elasticsearch como o motor de busca do sistema. O Elasticsearch utiliza técnicas de indexação e possui ferramentas que otimizam a execução e resultados das queries realizadas. Além disso, serão implementadas estratégias para a carga contínua dos dados, além da documentação dos desafios enfrentados durante a implementação. Para avaliar a solução proposta, foram realizadas medições de estatísticas referentes ao tempo de resposta e qualidade das consultas antes e depois da implantação do Elasticsearch. A qualidade dos resultados foi verificada por meio de técnicas como NDCG (Normalized Discounted Cumulative Gain) e f1-score, a partir da definição dos documentos relevantes ou não para cada consulta. Como resultado, foi possível notar uma diminuição em 10 vezes do tempo de respostas das consultas realizadas no Elasticsearch quando comparado com os resultados envolvendo o Sql Server. Além disso, também foi possível observar uma melhora na relevância dos resultados retornados de cerca de 2%, chegando a um NDCG de 95,3% em média, para consultas com 10 resultados, utilizadas por padrão no sistema.

ON SEARCHING PRODUCT CATALOG: RELATIONAL DATABASE VERSUS SEARCH ENGINE APPROACHES

ABSTRACT

The objective of TCE-AC is to oversee the expenses and revenues of municipalities and the state of Acre. In recent years, it has modernized its working methods. In particular, fast access to the prices being practiced is crucial for both the oversight process and the general population. To achieve this, the Price Database is employed, being continuously updated and currently containing tens of millions of invoice records. Given the scenario, as the system relies on relational databases for conducting queries, it often experiences delays in producing results in various situations and occasionally yields less relevant outcomes. To address this issue, the proposal is to implement Elasticsearch as the search engine for the system. Elasticsearch employs indexing techniques and features tools that optimize query execution and results. Additionally, strategies for continuous data loading will be implemented, along with documenting the challenges encountered during the implementation. To evaluate the proposed solution, statistics related to response times and query quality were measured before and after the implementation of Elasticsearch. Result quality was assessed using techniques such as NDCG (Normalized Discounted Cumulative Gain) and F1-score, based on the determination of relevant and non-relevant documents for each query. As a result, it was observed that Elasticsearch reduced query response times by a factor of 10 when compared to results involving SQL Server. Furthermore, there was an improvement in result relevance of approximately 2%, leading to an average NDCG of 95.3% for queries with 10 results, which are the default in the system.

Busca em Catálogo de Produtos: uma Comparação entre Banco de Dados Relacional e Motor de Busca

Eniedson Fabiano P. S. Júnior
eniedson.junior@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraiba, Brasil

Cláudio de Souza Baptista
baptista@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraiba, Brasil

André Luiz F. Alves
andre.alves@ifpb.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraiba, Brasil

RESUMO

The objective of TCE-AC is to oversee the expenses and revenues of municipalities and the state of Acre. In recent years, it has modernized its working methods. In particular, fast access to the prices being practiced is crucial for both the oversight process and the general population. To achieve this, the Price Database is employed, being continuously updated and currently containing tens of millions of invoice records. Given the scenario, as the system relies on relational databases for conducting queries, it often experiences delays in producing results in various situations and occasionally yields less relevant outcomes. To address this issue, the proposal is to implement Elasticsearch as the search engine for the system. Elasticsearch employs indexing techniques and features tools that optimize query execution and results. Additionally, strategies for continuous data loading will be implemented, along with documenting the challenges encountered during the implementation. To evaluate the proposed solution, statistics related to response times and query quality were measured before and after the implementation of Elasticsearch. Result quality was assessed using techniques such as NDCG (Normalized Discounted Cumulative Gain) and F1-score, based on the determination of relevant and non-relevant documents for each query. As a result, it was observed that Elasticsearch reduced query response times by a factor of 10 when compared to results involving SQL Server. Furthermore, there was an improvement in result relevance of approximately 2%, leading to an average NDCG of 95.3% for queries with 10 results, which are the default in the system.

CCS CONCEPTS

• **Information systems** → **Relevance assessment**; **Retrieval efficiency**; *Document filtering*; **Retrieval effectiveness**.

KEYWORDS

Elasticsearch, SQL Server, Recuperação da informação

1 INTRODUÇÃO

A fiscalização das despesas e receitas dos municípios e estados em um país desempenha um papel fundamental na promoção da transparência e eficiência na administração pública. No Brasil, os Tribunais de Contas dos estados e municípios assumem a responsabilidade por essa importante função. Nos últimos anos, o Tribunal de Contas do Estado do Acre (TCE-AC) tem se dedicado a modernizar seus processos, buscando otimizar sua atuação e aprimorar sua capacidade de fiscalização dos jurisdicionados (prefeituras municipais, secretarias, etc). Para tal, uma das principais necessidades para um efetivo controle é o acesso ágil e preciso aos preços praticados pelos fornecedores de produtos e serviços contratados pelos órgãos públicos. Nesse sentido, o TCE-AC utiliza o Banco de Preços, uma plataforma que fornece estatísticas e medidas referentes aos preços praticados para órgãos públicos, pessoas físicas e jurídicas em todo o Estado.

A plataforma em questão também conta com um mecanismo de busca por descrições de produtos, sendo realizada a partir de consultas SQL em um banco de dados relacional. O banco de dados em questão possui um grande volume de dados, atualmente com dezenas de milhões de registros de notas fiscais e, em média, 120 mil novas inserções diárias. Neste cenário, as consultas realizadas demandam um alto número de comparações que podem comprometer o desempenho da aplicação.

Em sistemas de RI (Recuperação da Informação), existe um grande foco na medição da eficácia dos resultados obtidos, visando aprimorar a entrega de resultados mais relevantes para as consultas realizadas. A eficiência dos sistemas de RI também é considerada e pode ser avaliada, por exemplo, medindo o tempo que o sistema leva para retornar os resultados das consultas [1]. O Elasticsearch, como ferramenta de busca e análise de dados, baseia-se na indexação de documentos e oferece diversas funcionalidades e recursos destinados a aprimorar o desempenho das consultas. Adicionalmente, lida com a ordenação dos resultados, contribuindo para aumentar a relevância dos produtos retornados.

O objetivo deste trabalho é avaliar o desempenho dos tempos de execução de consultas e a relevância dos resultados obtidos no Elasticsearch[1] em comparação com o banco de dados relacional previamente utilizado, o SQL Server[2], a partir de experimentos envolvendo diversas consultas. Através desta análise comparativa, buscamos obter uma melhoria mensurável no desempenho das consultas realizadas na plataforma, proporcionando resultados mais rápidos e precisos aos seus usuários, aprimorando assim a experiência deles.

Com o intuito de avaliar os mecanismos de busca, foram realizados experimentos visando medir o tempo de resposta e a qualidade das consultas antes e após a implementação do Elasticsearch. Para aferir a qualidade dos resultados, recorreu-se a métricas como o NDCG (Normalized Discounted Cumulative Gain), Precision, Recall e F1-score [3, 5]. Quanto à avaliação da eficiencia das consultas, procedeu-se à comparação do tempo de resposta considerando vários de tipos de consultas. Além disso, foram conduzidos testes de hipótese com o propósito de validar estatisticamente as análises realizadas.

[1] https://www.elastic.co/pt/elasticsearch
[2] https://www.microsoft.com/pt-br/sql-server

O restante deste artigo está organizado como segue. Na seção 2 é abordada a metodologia. Na seção 3, é apresentada a metodologia. As seções 4 e 5 discutem os experimentos e resultados obtidos, respectivamente. A seção 6 foca em ameaças à validade. Por fim, a seção 7 apresenta as conclusões desta pesquisa.

2 TRABALHOS RELACIONADOS

Os motores de busca, como o Elasticsearch, desempenham funções de grande importância em diversas aplicações. Isso naturalmente conduz a uma demanda substancial por estudos que visam realizar comparativos referentes aos ganhos de performance e relevância obtidos através da sua implantação. Um exemplo disso pode ser encontrado no artigo de Fdhila [2], que contempla uma comparação de desempenho, segurança e autenticidade dos dados retornados por consultas realizadas em um banco de dados relacional (Oracle) e no Elasticsearch. O artigo não trata da relevância dos resultados, mas realiza uma avaliação nos tempos de resposta dos mesmos e apresenta resultados que colaboram com o objetivo deste trabalho, onde foi observada uma melhora significativa nos resultados com a utilização do Elasticsearch.

Outro exemplo pode ser encontrado no trabalho de Marwah [9] que apresenta um design de experimento para medição da qualidade da solução proposta no quesito "Relevância dos resultados". Além disso, também é tratado do BM25[3], algoritmo utilizado na implementação do Elasticsearch na aplicação avaliada nesse estudo. O trabalho também utiliza-se de métricas para avaliação dos resultados, como NDCG e f1-score, sendo uma base interessante para prosseguir na avaliação da qualidade dos resultados.

No trabalho conduzido por Greca et al. [4], foi abordada a integração do Elasticsearch para aprimorar a funcionalidade de buscas textuais em um banco de dados NoSQL, o MongoDb. Apesar de mencionar que a solução proposta resolveu o problema de busca, não foram apresentadas evidências detalhadas ou resultados mensuráveis que pudessem comprovar a melhoria na relevância dos resultados.

O estudo de Kannan et al. (2011) [7] propõe uma abordagem para realizar a correspondência de produtos utilizando a busca de texto completo no Elasticsearch. O artigo concentra-se na abordagem de desafios relacionados à busca e comparação de produtos, que decorrem das diversas descrições associadas a esses produtos, bem como da ampla utilização de abreviações. O artigo apenas propõe uma arquitetura, não realizando qualquer análise quanto a eficácia da mesma.

O trabalho de Vavliakis et al. [12] apresenta uma ideia interessante para trabalhos futuros, utilizando uma ideia de popularidade para os dados, sendo calculada a partir da quantidade de cliques, compras e visualizações. Para o Banco de Preços, essas medidas podem ser substituídas pela quantidade de produtos semelhantes na base de dados e a quantidade de cliques para cada descrição. A partir dessa melhoria, espera-se que resultados mais relevantes sejam retornados pelo Elasticsearch, melhorando ainda mais a experiência do usuário. Além disso, o artigo também utiliza o NDCG para avaliação dos resultados, o que corrobora com a estratégia definida neste trabalho.

[3]https://www.elastic.co/pt/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables

3 METODOLOGIA

Este estudo adota uma abordagem de pesquisa quantitativa para avaliar o desempenho em tempo de execução e relevância dos resultados no Elasticsearch em comparação com o SQL Server como motores de busca. A pesquisa é conduzida em duas fases principais: a primeira fase concentra-se na medição dos tempos de processamento das consultas, enquanto a segunda fase aborda a avaliação da relevância dos resultados obtidos. A pesquisa foi realizada em quatro etapas: design, preparação, execução e avaliação do experimento. A seguir, estas etapas são detalhadas.

3.1 Definição de design do experimento

O experimento tomou como base uma combinação dos conceitos propostos por David J. Lilja [8] e Raj Jain [6], que fazem uso das seguintes definições: variável de resposta, fatores, níveis, replicação e interação. Para ambos os experimentos realizados, foram considerados o Sql Server e o Elasticsearch para a realização de cada uma das consultas.

A interação entre os fatores de cada um dos experimentos se deu a partir de um modelo fatorial completo. Dessa forma, o número total de consultas realizadas durante a execução do experimento é regido de acordo com a equação 1, onde n representa a quantidade de experimentos, k representa os fatores, n_i representa a quantidade de níveis para o i-ésimo fator w x representa a quantidade de repetições.

$$n = x \prod_{i=1}^{k} n_i \tag{1}$$

3.1.1 Relevância.

O primeiro experimento realizado teve como foco a avaliação da relevância dos resultados das consultas. Nesse caso, as variáveis de resposta se deram a partir do cálculo das medidas referentes à relevância dos resultados e fatores definidos, já com os seus respectivos níveis, foram os seguintes:

- `Consultas`: Foram selecionadas 500 descrições de produtos que foram utilizadas como consultas, cada produto possuindo, ao menos, 40 variações de sua descrição. Tais variações foram utilizadas como gabarito no momento da avaliação dos resultados.
- `Quantidade de resultados retornados`: As métricas utilizadas para avaliação da relevância dos resultados são bastante afetadas pela quantidade de registros retornados. Portanto, a quantidade de registros retornados variou entre 1 e 100 registros.

3.1.2 Performance.

O segundo experimento buscou coletar dados referentes a performance das consultas, utilizando dos tempos de processamento das mesmas como variável de resposta. Para esse experimento, os seguintes fatores foram definidos:

- `Tamanho da consulta`: Define a quantidade de termos existentes na consulta, variando de 1 a 4 termos.
- `Filtros de intervalo aplicados`: Refere-se ao filtro temporal aplicado à consulta, podendo variar entre intervalos de 30, 60, 90, 180 e 365 dias. Esse fator é muito relevante dado

que as consultas realizadas no experimento são baseadas nas datas de emissão das notas fiscais, garantindo a existência de pelo menos 10 notas fiscais para cada produto no intervalo de datas definido. Isso é essencial para proporcionar visualizações significativas nos gráficos e visualizações presentes no Banco de Preços.
- `Frequência do produto na base de dados`: Define a popularidade dos produtos da consulta, podendo variar entre pouco popular, muito popular e consultas escolhidas empiricamente, que se refere às consultas que são realizadas frequentemente por usuários da aplicação.

No experimento realizado, utilizou-se da replicação das consultas, executando cada uma delas dez vezes. Essa abordagem foi adotada com o objetivo de aumentar a confiabilidade e a robustez dos resultados obtidos. A replicação não apenas ajudou a reduzir a incerteza, mas também desempenhou um papel fundamental na confirmação das conclusões e na contribuição para uma compreensão mais profunda durante a análise. A repetição das consultas permitiu uma avaliação mais precisa dos resultados, mitigando a influência de variações aleatórias.

3.2 Preparação do experimento

Nesta fase, foram elaboradas etapas para a implementação do arcabouço do experimento, que abrangem a configuração do ambiente experimental e a codificação dos testes necessários para a condução do referido experimento. O propósito fundamental dessa etapa é garantir que o experimento seja conduzido de maneira sistemática e replicável, assegurando, desse modo, que os resultados obtidos sejam confiáveis e comparáveis para ambas as abordagens de busca que estão sendo estudadas.

3.2.1 Preparação do ambiente.

Esta etapa demandou a configuração de elementos cruciais, como o Elasticsearch, o SQL Server e a aplicação Spring Boot. Adicionalmente, procedeu-se à instalação e configuração de todas as ferramentas e recursos necessários para garantir que o sistema experimental operasse de maneira eficaz e integrada.

3.2.2 Implementação dos scripts de teste.

O processo de seleção das consultas para a coleta de resultados é descrito na Figura 1, representando as expansões realizadas em cada consulta. Para cada nível de consulta (ou seja, "Populares", aquelas escolhidas empiricamente e as não populares), planejou-se a realização de um total de 400 consultas, resultado da multiplicação dos seguintes fatores: 5 consultas (representando diferentes intervalos de tempo), 4 (correspondendo às quantidades de termos nas consultas), 2 (indicando a execução tanto no SQL Server quanto no Elasticsearch) e 10 (refletindo a quantidade de repetições realizadas).

Com base nas consultas definidas, os casos de teste foram implementados de forma a garantir uma comparação justa entre às duas abordagens de busca, ao mesmo tempo que favorecem a consistência e reprodutibilidade do experimento.

3.3 Execução do experimento

Nesta fase, a execução do experimento foi conduzida de acordo com o arcabouço e os procedimentos previamente estabelecidos. Durante esse processo, os testes planejados foram conduzidos, os dados relevantes foram coletados, e as informações necessárias foram registradas para uma análise abrangente. Isso envolveu a medição do tempo de resposta do sistema e a avaliação da relevância dos resultados obtidos. Essa execução é essencial para assegurar que os dados coletados sejam confiáveis e proporcionem uma base sólida para a análise comparativa das abordagens de busca estudadas.

3.4 Análise dos resultados

A quarta e última etapa engloba a análise dos resultados obtidos durante a execução do experimento, utilizando as técnicas previamente definidas para obter uma comparação entre as duas versões do sistema. Neste contexto, para a avaliação da relevância dos resultados, serão utilizadas as métricas a seguir: Precisão, Recall, F1-score e NDCG. No caso da avaliação de performance das consultas, os tempos de respostas serão avaliados a partir dos seus valores médios. A escolha dessas métricas busca evidenciar os impactos da implementação do Elasticsearch como substituto das consultas em bancos de dados relacionais. Para uma avaliação rigorosa dos resultados alcançados, foram conduzidos testes de hipótese com o objetivo de validar as suposições realizadas e assegurar a obtenção de resultados confiáveis e conclusivos.

4 EXPERIMENTO

Esta seção oferece uma descrição dos experimentos, começando pelos seus objetivos e definições, e adentrando na definição do ambiente no qual foram realizados.

4.1 Objetivos do Experimento

O objetivo do experimento é analisar o desempenho e a relevância dos resultados obtidos a partir de consultas realizadas no Elasticsearch e no Sql Server, com foco na consulta de descrições de produtos em notas fiscais. Para atingir esse objetivo, dois experimentos distintos foram realizados.

4.1.1 Experimento 1: Relevância dos resultados.

No primeiro experimento foi realizada uma avaliação da relevância dos resultados retornados por ambas as abordagens, para esse caso, as seguintes hipóteses foram definidas:

- `H1-0 - (Hipótese nula)`: O Elasticsearch não retorna resultados mais relevantes em comparação com os resultados das consultas no Sql Server.
- `H1-1 - (Hipótese alternativa)`: O Elasticsearch retorna resultados mais relevantes em comparação com os resultados das consultas no Sql Server.

4.1.2 Experimento 2: Performance das consultas.

No segundo experimento, procedeu-se à avaliação de desempenho das consultas a partir dos tempos de resposta das mesmas, onde as hipóteses foram as seguintes:

- `H2-0 - (Hipótese nula)`: O Elasticsearch não apresenta um tempo de resposta inferior em comparação com o Sql Server.
- `H2-1 - (Hipótese alternativa)`: O Elasticsearch apresenta um tempo de resposta inferior em comparação com o Sql Server.

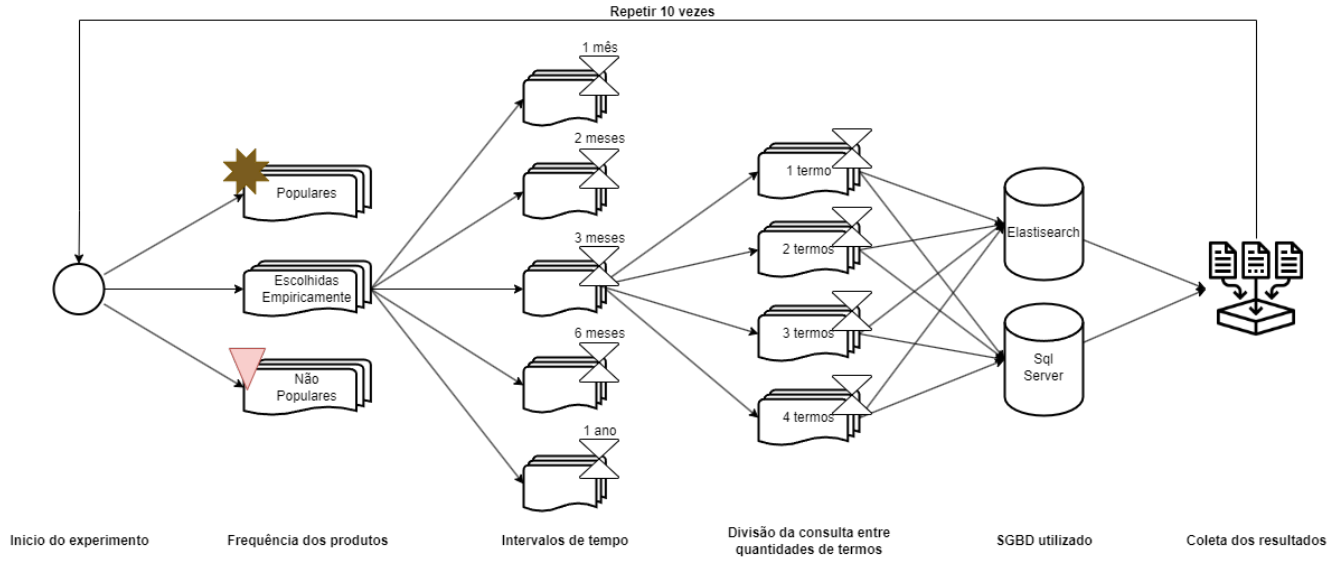

Figura 1: Fluxograma do experimento para coleta dos tempos de resposta das consultas.

Essas hipóteses orientam a análise comparativa das duas abordagens e ajudam a determinar qual delas proporciona os resultados mais relevantes e o melhor desempenho no contexto das consultas de produtos de notas fiscais.

4.2 Ambiente do Experimento

A máquina utilizada para a execução do experimento utiliza o sistema operacional Linux, com a distribuição Pop!_OS 20.04 LTS, o processador é um Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz, 32GB de memória ram DDR4 e um SSD (Solid State Drive) de 1TB. As especificações e definições referentes a cada ferramenta utilizada estão dispostas a seguir:

4.2.1 Aplicação para o backend.

Para intermediar a conexão com o Sql Server e o Elasticsearch, escolhemos utilizar uma aplicação Spring Boot[4]. Essa escolha se deve à necessidade de desenvolver uma interface Restful para padronizar e facilitar as requisições em ambas as ferramentas durante a execução do experimento.

Para a realização das consultas no SQL Server, adotamos o uso do Java Persistence API (JPA)[5] em conjunto com o Hibernate[6]. Essa decisão foi motivada pela eficácia e facilidade de uso, oferecidas por essas tecnologias quando se trata de interagir com bancos de dados relacionais, como o SQL Server. O Hibernate, por exemplo, simplifica a persistência de objetos Java no banco de dados, enquanto o JPA fornece um conjunto de padrões para mapeamento objeto-relacional.

Já para a execução das consultas no Elasticsearch, escolhemos o Elasticsearch Java API Client. A razão para essa escolha é a capacidade desse cliente de interagir de maneira eficiente com o Elasticsearch. A biblioteca em questão permite que nosso sistema se comunique de maneira eficaz com o Elasticsearch, aproveitando sua escalabilidade e recursos de pesquisa avançados.

Essas escolhas de tecnologia foram feitas com base na adequação das ferramentas aos requisitos do projeto, visando facilitar a conexão e otimizar a execução das consultas, de acordo com as características e necessidades específicas de cada implementação.

4.2.2 Microsoft Sql Server.

A versão utilizada do SQL Server foi a 2019 - 15.0.4322.2, executada em um container Docker. Para compreender a estrutura do banco de dados, é apresentado na Figura 2 um Modelo Entidade-Relacionamento (MER) representando o esquema utilizado durante a execução do experimento:

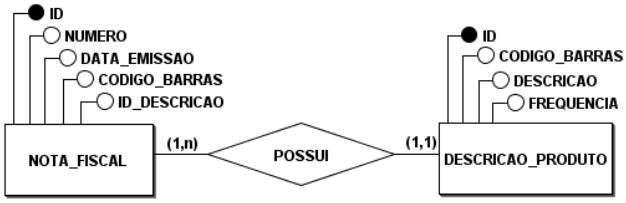

Figura 2: Modelo Entidade Relacionamento (MER) para o Sql Server.

Para otimizar essas consultas, que utilizam de filtros temporais, foram criados dois índices. O primeiro deles foi projetado para facilitar a filtragem com base nos códigos de barras e datas de emissão, campos críticos nas consultas do sistema. Abaixo é possível observar o script para a criação deste índice:

```
CREATE NONCLUSTERED INDEX IDX_NF_CODIGO_BARRAS_TIPO
ON dbo.NOTA_FISCAL (CODIGO_BARRA ASC, DATA_EMISSAO ASC)
INCLUDE ( CNPJ_CPF_DESTINATARIO,
CODIGO_MUNICIPIO_DESTINO, DESCRICAO_PRODUTO,
```

[4]https://spring.io/projects/spring-boot
[5]https://docs.spring.io/spring-data/jpa/docs/current/reference/html/
[6]https://hibernate.org/

```
QUANTIDADE_PRODUTO, VALOR_PRODUTO);
```

O segundo índice criado é um índice de texto completo (Fulltext index), sendo projetado para aprimorar a pesquisa de texto em colunas de texto longo, como descrições de produtos em notas fiscais. Em resumo, o Fulltext index simplifica e acelera a pesquisa de texto em grandes conjuntos de dados, tornando-o valioso para aplicativos que exigem análise e recuperação de informações de texto de maneira eficaz. Dessa forma, segue o código utilizado para a criação do índice:

```
CREATE FULLTEXT CATALOG DEFAULT_CATALOG;
CREATE FULLTEXT INDEX ON DESCRICAO_PRODUTO (DESCRICAO)
KEY INDEX PK_DESCRICAO_UNIQUE_INDEX;
ALTER FULLTEXT INDEX ON DESCRICAO_PRODUTO
START FULL POPULATION;
```

As consultas no Sql Server utilizam a função "FREETEXTTABLE", que faz uso do algoritmo OKAPI BM25. Esse algoritmo é o mesmo utilizado pelo Elasticsearch para o ranqueamento de resultados das consultas. Essa mesma função também foi utilizada no estudo realizado por Edson Marchetti da Silva e Lucas Meneses Mardegan [11], que comparou o desempenho de diferentes SGBDs (Sistema de gerenciamento de banco de dados) em consultas de documentos textuais.

Para uma compreensão mais aprofundada, apresenta-se um exemplo prático de uma consulta realizada no SQL Server. Neste exemplo, a consulta é direcionada à busca por resultados relacionados à consulta "Água mineral"que atendam ao requisito de ter pelo menos 10 notas fiscais no período de 1 mês.

```
SELECT DISTINCT DP.*, fta.RANK FROM DESCRICAO_PRODUTO DP
JOIN DESCRICAO_PRODUTO DP2 ON
   DP.CODIGO_BARRAS = DP2.CODIGO_BARRAS AND DP.RANK = 1
JOIN FREETEXTTABLE(DESCRICAO_PRODUTO , DESCRICAO,
    'AGUA MINERAL') FTA ON FTA."key" = DP2.ID_DESCRICAO
JOIN ( SELECT CODIGO_BARRAS FROM NOTA_FISCAL
   WHERE DATA_EMISSAO BETWEEN '2023-09-06 00:00:00.000'
        AND '2023-10-06 00:00:00.000'
    GROUP BY CODIGO_BARRAS HAVING COUNT(ID) >= 10
) NF ON DP.CODIGO_BARRAS = NF.CODIGO_BARRAS
ORDER BY fta.RANK DESC OFFSET 0 ROWS
    FETCH NEXT 10 ROWS ONLY;
```

4.2.3 Elasticsearch.

No caso do Elasticsearch, foi empregada a versão 8.6.2. Sua execução, assim como no SQL Server, ocorreu em um container Docker. O índice onde as consultas foram conduzidas foi criado com base na seguinte definição: o campo principal para busca é a descrição, juntamente com o "ean", e a data é utilizada para a filtragem por intervalo, de forma semelhante ao que é feito no SQL Server.

```
{
    "settings": {
        "number_of_shards": 5,
        "number_of_replicas": 1
    },
    "mappings": {
        "properties": {
            'ID': {'index': False, 'type': 'long'},
            'DATA': {'index': True, 'type': 'date'},
            'ean': {'type': 'keyword'},
          'DESCRICAO': {'index': True, 'type': 'text',
               'analyzer': 'brazilian'},
            'ID_DESCRICAO': {'type': 'keyword'}
        }
    }
}
```

Conforme mencionado anteriormente, as consultas realizadas no Elasticsearch também incluem filtros baseados em intervalo de tempo. Nesse caso, as filtragens e agregações dos dados podem ser observadas no exemplo a seguir, que ilustra uma consulta efetuada pelo sistema no Elasticsearch.

```
{
    "query": {
        "bool": {
            "must": [
                {
                    "multi_match": {
                        "query": "leite integral",
                        "operator": "AND",
                        "fuzziness": 2,
                       "fields": ["DESCRICAO", "ean"]
                    }
                }
            ],
            "filter": {
                "range": {
                    "DATA": {
                           "gte": "2022-07-11",
                           "lte": "2022-09-11"
                    }
                }
            }
     }
    }
    "aggregations": {
     "qtd": {
         "terms": {
             "field": "ean",
             "min_doc_count": 10,
         },
         "aggs": {
             "descricao": {
                 "top_hits": {
                     "_source": {
                    "includes": [ "ean", "DESCRICAO" ]
                        },
                     "size": 1
                 }
             }
         }
     },
    }
}
```

4.3 Definição do Experimento

Os experimentos foram realizados utilizando a linguagem Python [7] e cada experimento realizado está listado a seguir:

4.3.1 Relevância dos resultados.

Para avaliar a relevância dos resultados das consultas, iniciamos selecionando 500 descrições de produtos. Cada um desses produtos estava associado a, no mínimo, 40 descrições alternativas que compartilhavam o mesmo código de barras. Durante a execução do experimento, essas descrições alternativas foram empregadas como consultas. Durante a etapa de análise dos resultados, consideramos como relevantes aqueles que correspondiam a qualquer uma das descrições alternativas do produto em foco. A definição precisa dos resultados relevantes é fundamental para os cálculos das métricas estabelecidas na avaliação.

Por exemplo, se o produto fosse "REFRIGERANTE COCA COLA ORIGINAL 1LT", consideraríamos relevante qualquer resultado que correspondesse a uma descrição alternativa desse produto, como "REFRIG COCA COLA ORIGINAL PET 1LT". Essas descrições alternativas se referem ao mesmo produto portanto, são resultados relevantes.

Cada consulta foi submetida a uma série de experimentos nos quais a quantidade de resultados retornados variou de 1 a 100. Essa variação sistemática permitiu uma investigação detalhada do comportamento das métricas de relevância calculadas à medida que a demanda por resultados aumentava. Esse enfoque abrangente possibilitou a identificação de tendências e padrões na avaliação da relevância das abordagens de busca, fornecendo insights valiosos sobre como elas se comportam sob diferentes cargas de trabalho.

Em especial, o cálculo do NDCG, que exige uma pontuação para cada resultado, foi realizado seguindo os critérios estabelecidos da seguinte forma:

- Um resultado recebeu uma pontuação de 3 se correspondia exatamente ao produto buscado, ou seja, uma correspondência exata na descrição (por exemplo, "REFRIGERANTE COCA COLA ORIGINAL 1LT").
- Uma pontuação de 2 foi atribuída a resultados que se referiam ao mesmo produto, mesmo que não tivessem uma correspondência exata na descrição, desde que possuam o mesmo código de barras (por exemplo, "REFRIG COCA COLA ORIGINAL PET 1LT").
- Uma pontuação de 1 foi dada a produtos que não compartilhavam o mesmo código de barras, mas tinham descrições semelhantes. A semelhança entre as descrições foi definida calculando a distância do cosseno entre elas, e a pontuação foi definida como 1 quando o resultado desse cálculo era maior que 0,7.
- Todos os demais resultados que não se encaixavam nos critérios acima receberam uma pontuação de 0.

Essa abordagem permitiu atribuir pontuações aos resultados com base em sua relevância em relação ao produto buscado, o que, por sua vez, facilitou a avaliação da qualidade das abordagens de busca em relação aos resultados retornados.

Consulta	Categoria
Renopril com 20MG C/30	Produtos pouco frequentes
Laço Facil Presente Branco	
Gel Cola Pote Incolor	
Cilindro de Palheta 48MM	
Jogo Chave Hexagonal 8PCS	
Água Mineral Cristalina 20L	Produtos selecionados empiricamente, representando consultas realizadas com frequência pelos usuários
Leite Itambe 1L Integral	
Pao Forma Acrepan 500G	
Sabao Po Omo 800G	
Papel Toalha Stylus 50F	
Açúcar Cristal Itamarati 1KG	Produtos muito frequentes na base de dados
Sacola Especial Presente P	
Coca Cola Lata 350ML	
Óleo Soja Concordia 900ML	
Cerveja Pilsen Lata 350ML	

Tabela 1: Consultas realizadas no experimento para análise da performance das consultas e suas respectivas categorias.

4.3.2 Performance das consultas.

Para a coleta dos tempos de resposta das consultas, conforme definido na metodologia, as consultas utilizadas podem ser observadas na Tabela 1, juntamente com suas respectivas categorias.

Cada consulta passou por um processo de expansão, que envolveu a aplicação de cinco filtros com intervalos distintos. Além disso, de cada consulta original, seguindo a Formula 1, que representa o design fatorial utilizado, foram geradas quatro novas consultas, variando de 1 a 4 termos. Para garantir resultados robustos, cada consulta expandida foi repetida 10 vezes. Em resumo, iniciamos com 15 consultas originais, cada uma sujeita a 5 filtros e gerando 4 variações de quantidade de termos, sendo que cada variação foi repetida 10 vezes. Isso resultou em um total de 3.000 consultas. Todas essas consultas foram executadas tanto no Elasticsearch quanto no Sql Server, totalizando 6.000 consultas ao término do experimento.

Como exemplo, para a descrição "ÁGUA MINERAL CRISTALINA 20L", geramos quatro novas consultas, a saber:

- ÁGUA
- ÁGUA MINERAL
- ÁGUA MINERAL CRISTALINA
- ÁGUA MINERAL CRISTALINA 20L

[7] https://www.python.org/

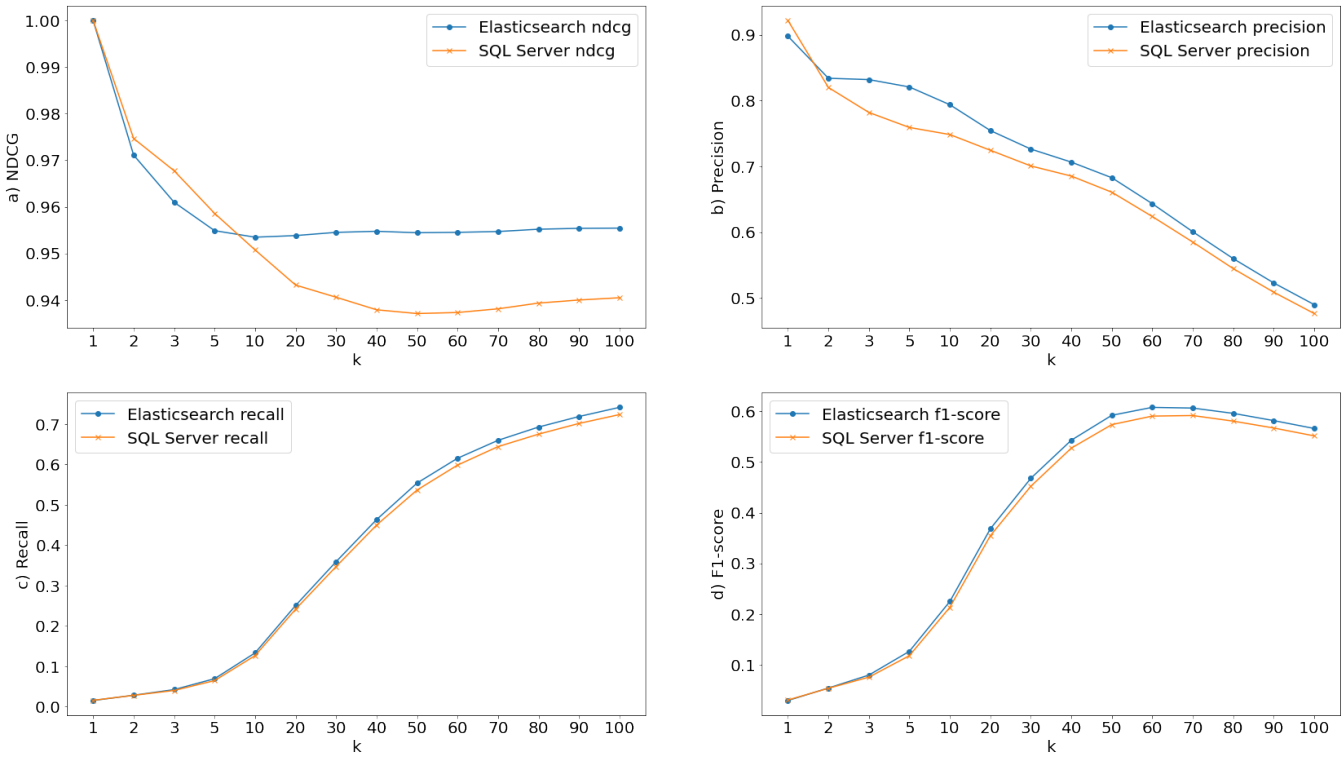

Figura 3: Métricas para avaliação da relevância dos resultados do Elasticsearch e SQL Server: a) NDCG; b) Precision; c) Recall; e d) F1-Score.

Cada uma dessas consultas foi associada aos 5 filtros de intervalo e repetida 10 vezes, resultando assim em 200 consultas. Esse procedimento permitiu analisar os impactos resultantes da implantação do Elasticsearch no sistema.

5 RESULTADOS

Nesta seção serão apresentados e analisados os resultados obtidos durante a execução dos experimentos.

5.1 Relevância dos resultados

Conforme mencionado anteriormente, as métricas utilizadas para avaliar a relevância dos resultados englobam Precision, Recall, F1-score e NDCG. A partir dos cálculos dessas métricas, os gráficos na Figura 3 exibem uma análise comparativa entre o SQL Server e o Elasticsearch, onde 'k' denota a quantidade de resultados retornados.

Na Figura 3 a), observa-se que o SQL Server manteve resultados superiores em termos de NDCG para valores de k entre 1 e 5. No entanto, a disparidade entre as curvas é baixa até esse ponto, possuindo uma diferença maior a partir de $k = 20$, quando o Elasticsearch demonstra um desempenho superior.

No que tange à precisão, representada na Figura 3 b), o SQL Server apresentou um desempenho superior para $k = 1$, mas, a partir desse ponto, o Elasticsearch passa a exibir resultados superiores, revelando sua vantagem a partir desse limiar.

Em relação à Recall e ao F1-score, destacados nas Figuras 3 c) e 3 d), respectivamente, as diferenças não são tão evidentes. Embora a curva do Elasticsearch pareça se posicionar acima, a discrepância entre ambas é bastante reduzida.

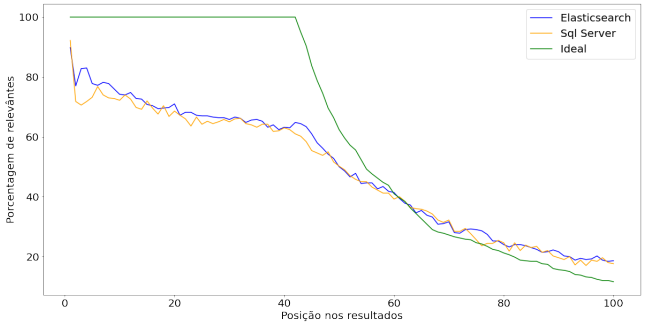

Figura 4: Porcentagem de resultados relevantes retornados em cada posição.

A Figura 4, por sua vez, apresenta a porcentagem de documentos relevantes retornados em cada posição nas buscas realizadas para as duas versões do sistema em comparação com a quantidade ideal de resultados relevantes na posição. O cálculo da porcentagem para cada posição é bastante semelhante ao da precisão, mas observa apenas a posição em questão, desconsiderando as anteriores. Na

posição 1, por exemplo, os valores estão próximos de 90%, isso quer dizer que em 90% das consultas realizadas, a primeira posição foi ocupada por um resultado relevante. Um fato interessante na análise do gráfico é que, na primeira posição, o SQL Server se mostrou mais preciso que o Elasticsearch, o que também pode ser observado no gráfico referente a precisão na Figura 3. A partir do segundo resultado, as medições favorecem ao Elasticseach. Vale ressaltar tambem que a grande semelhança nas curvas do SQL Server e Elasticsearch se dá devido à utilização de funções semelhantes para o ranqueamento dos resultados no SQL Server e no Elasticsearch (OKAPI BM25), conforme mencionado na seção 4.2.2.

Ainda na Figura 4, a linha representando o resultado ideal descreve a situação ótima de busca, onde todos os resultados relevantes são exibidos nas primeiras posições. Por exemplo, uma porcentagem de 100% na posição 1 indica que em todas as consultas realizadas essa posição foi ocupada por um resultado relevante. No entanto, a partir da posição 40, ocorrem consultas que existem menos resultados relevantes que o valor da posição,ou seja, a partir dessa posição algumas consultas não conseguem exibir resultados relevantes nessa posição em um resultado ideal. A diferença substancial entre a linha ideal e os resultados obtidos pelas ferramentas de busca indica que a análise lexical por si só não é suficiente para a eficácia de uma ferramenta de busca em alguns casos. Um exemplo notório disso é observado nas descrições "BISCOITO TIPO AGUA E SAL 400g"e "BOLACHA SALGADA", que, embora se refiram ao mesmo produto na base de dados, utilizam termos totalmente distintos. A solução para esse problema pode estar na utilização de abordagens que vão além da análise lexica, considerando o significado semântico das palavras.

5.1.1 Teste de hipótese.

Na Figura 5, apresentada abaixo, é possível observar a análise de histogramas das variáveis analisadas. Esses histogramas revelaram distribuições assimétricas e não gaussianas, indicando que os dados não seguem uma distribuição normal. Além disso, o Teste de Shapiro-Wilk também foi utilizado como forma de validar a não normalidade dos dados.

A não normalidade das variáveis analisadas é uma informação crucial, uma vez que impacta a escolha do método estatístico apropriado. A não aderência à distribuição normal compromete a aplicação de testes paramétricos, tornando necessária a utilização de métodos não paramétricos, como o teste de Mann-Whitney U [10].

Com aplicação do teste de Mann-Whitney U, a hipótese nula (H1-0) pode ser descartada na avaliação de todas as métricas, onde os p-values estão definidos a seguir: para a precisão o valor foi de 0,0000, para o recall o valor foi de 0,0236, para o f1-score o valor foi de 0,0079 e, para o NDCG, o valor foi de 0,000. Dessa forma é possível entender que, estatisticamente, as métricas coletadas para análise de relevância possuem resultados melhores no Elasticsearch, quando comparado com o Sql Server.

5.2 Performance

Inicialmente, realizou-se uma comparação abrangente dos tempos de resposta das consultas entre as duas estratégias de busca, Elasticsearch e Sql Server. A Figura 6 exibe um boxplot que apresenta uma análise geral, sem tratamento nos dados, dos resultados obtidos no

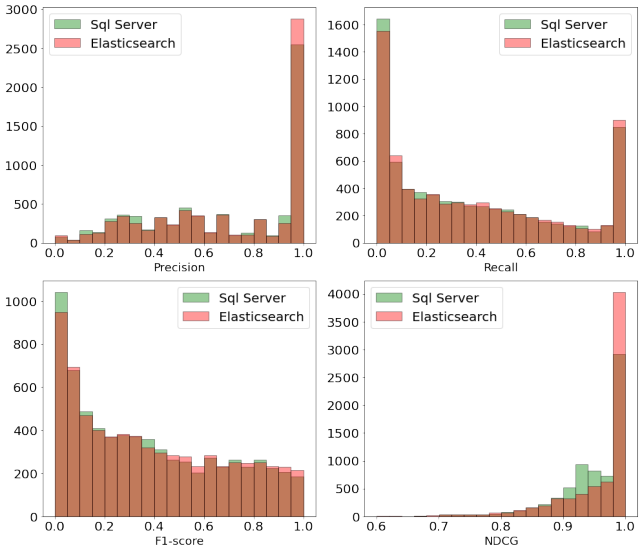

Figura 5: Distribuição dos valores de cada uma das métricas utilizadas.

experimento. A partir dessa figura, observa-se uma melhoria significativa no desempenho das consultas ao utilizar o Elasticsearch como motor de busca. Além disso, em relação à Figura 6, é possível observar a presença de instabilidade na estratégia que emprega o SQL Server para processar as consultas. Desconsiderando os valores atípicos (outliers), os tempos de resposta para consultas no Elasticsearch variaram entre 0,025 e 0,325 segundos, ao passo que no SQL Server, essa variação foi mais ampla, situando-se entre 0,75 e 2,95 segundos. Essa disparidade evidencia a considerável variabilidade no desempenho das consultas no SQL Server em comparação com a estabilidade proporcionada pelo Elasticsearch.

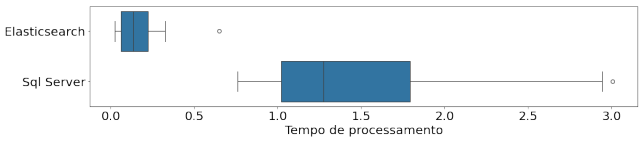

Figura 6: Boxplot do tempo de resposta das consultas.

5.2.1 Fator 1: Tamanho das consultas.

A Figura 7 apresenta um gráfico que ilustra a variação das médias dos tempos de resposta, juntamente com seus respectivos desvios padrão. A partir da análise do gráfico, é possível observar uma tendência de queda nos tempos de resposta à medida que a quantidade de termos aumenta no Elasticsearch. Por outro lado, as consultas realizadas no Sql Server não variaram o tempo de resposta com a alteração desse fator.

5.2.2 Fator 2. Frequência do produto na base de dados.

A Figura 8 mostra a influência da frequência das descrições nas consultas realizadas. No caso do Elasticsearch, observa-se um aumento

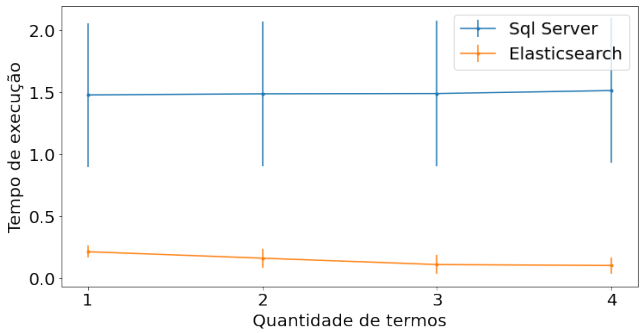

Figura 7: Tempo médio e desvio padrão para consultas a para cada quantidade de termos utilizados.

no tempo de resposta, o que provavelmente se deve à maior quantidade de documentos a serem processados. No entanto, mesmo com esse aumento, as consultas executadas no Sql Server apresentam tempos de resposta médios mais altos. É importante ressaltar que, mesmo que os tempos médios no Sql Server não estejam variando, a diferença em relação ao Elasticsearch permanece substancial.

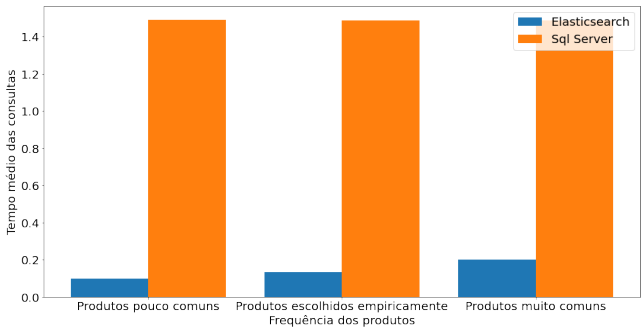

Figura 8: Tempos de processamento das consultas para produtos mais ou menos frequentes.

5.2.3 *Fator 3. Filtros por intervalo de tempo.*

A Figura 9 fornece uma representação visual dos tempos de resposta médios das consultas à medida que os intervalos usados como filtros nas consultas aumentam. A partir dos resultados apresentados, torna-se evidente um aumento significativo no tempo de processamento das consultas à medida que o intervalo aumenta no caso do Sql Server. Essa variação parece estar diretamente relacionada à quantidade de registros que precisam ser processados à medida que o intervalo cresce. Por outro lado, também é notável um aumento na linha azul, que representa o Elasticsearch, mas esse aumento ocorre de maneira mais gradual, mantendo resultados aceitáveis e proporcionando uma experiência satisfatória ao realizar as consultas.

5.2.4 *Teste de hipótese.*

Assim como no caso da análise das relevâncias dos resultados, os dados coletados referentes aos tempos de resposta das consultas também não seguem uma distribuição normal, como pode ser visto

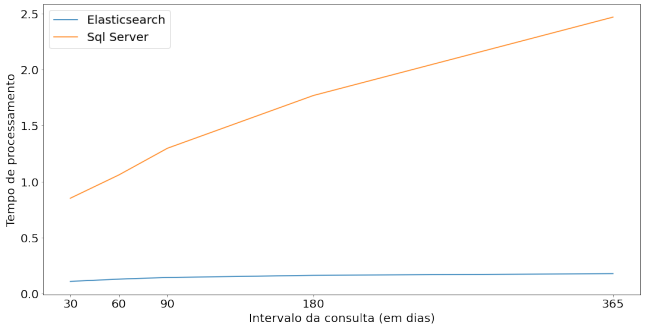

Figura 9: Gráfico mostrando o crescimento do tempo de processamento com o aumento do intervalo pesquisado.

no histograma da Figura 10. Assim como no caso anterior, foi utilizado o Teste de Shapiro-Wilk para validar a suposição da não normalidade dos dados.

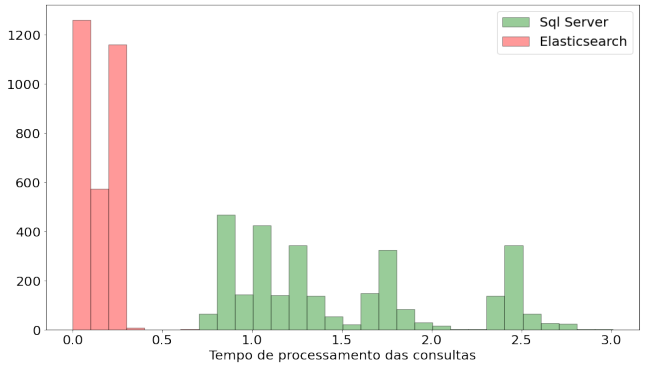

Figura 10: Distribuição dos tempos de resposta obtidos com o experimento.

A partir desta constatação, o teste de hipótese aplicado foi o de Mann-Whitney U [10], tendo como resultado um p-value de 0,0000, possibilitando assim a rejeição da hipótese nula (H2-0) e, com isso, aceitando a hipótese alternativa. Portanto, é possível dizer que o Elasticsearch possui um tempo de resposta menor que o Sql Server para o caso analisado neste trabalho.

6 AMEAÇAS À VALIDADE

Alguns fatores podem representar ameaças à validade deste estudo. No que se refere à comparação dos tempos de resposta, a presença de resultados armazenados em cache pode ocasionar em tempos de resposta menores, que não correspondem a cenários reais de aplicação. Para contornar esse problema, adotamos a estratégia de executar as consultas em ordem aleatória, reduzindo assim a possibilidade de o banco de dados ter resultados pré-computados e evitar o processamento da consulta.

Outro possível fator de risco envolve processos em segundo plano iniciados pelo sistema operacional, que podem resultar em um aumento no uso de recursos na máquina durante o experimento com uma das abordagens de busca. Para lidar com essa preocupação,

intercalamos a execução das consultas no Elasticsearch e Sql Server, minimizando o impacto de quaisquer processos em segundo plano e garantindo uma avaliação mais realista de seu desempenho.

7 CONCLUSÃO

Neste estudo comparativo, foram avaliados os desempenhos do Elasticsearch e do SQL Server na busca de produtos em notas fiscais, com o propósito de verificar a eficácia da implementação do Elasticsearch nesse contexto. O presente estudo envolveu a coleta e análise de métricas relacionadas aos tempos de resposta e à qualidade dos resultados obtidos em ambos os sistemas.

Os resultados obtidos, após análise e aplicação de testes estatísticos, demonstraram que o Elasticsearch supera o SQL Server no que diz respeito ao desempenho das consultas. Especificamente, o Elasticsearch demonstrou a capacidade de proporcionar respostas mais rápidas, aprimorando a experiência do usuário durante a interação com a ferramenta de busca. Além disso, a avaliação da relevância dos resultados obtidos também favoreceu o Elasticsearch, uma vez que métricas como NDCG, Precisão e Recall apresentaram valores superiores em comparação com o SQL Server na maioria dos cenários avaliados.

Portanto, com base nos resultados obtidos neste estudo, podemos concluir que o Elasticsearch demonstra ser mais promissor, considerando tempo de processamento das consultas e relevância dos resultados obtidos, na tarefa de busca de informações em notas fiscais, demonstrando superioridade em relação ao SQL Server para o contexto investigado. Este achado não apenas realça a importância do Elasticsearch como uma solução eficaz para consultas desse tipo, mas também contribui significativamente para o avanço da pesquisa no campo de sistemas de gerenciamento de informações e recuperação de dados.

Como trabalho futuro, considera-se a investigação de soluções que envolvam busca semântica ou abordagens mistas para abordar os desafios relacionados a descrições que apresentam termos consideravelmente distintos, mas que denotam o mesmo produto, conforme evidenciado na Seção 5.1. Tais abordagens detêm o potencial de aprimorar as operações de busca, fazendo uso do significado semântico das palavras. Ao incorporar a busca semântica, a capacidade de identificar produtos relacionados, independentemente de suas variações léxicas, é aprimorada, contribuindo assim para uma recuperação mais precisa e abrangente de resultados relevantes em cenários complexos.

Tais abordagens detêm o potencial de aprimorar as operações de busca, fazendo uso do significado semântico das palavras. Com a incorporação da busca semântica, a capacidade de identificação de produtos correlatos, independentemente de suas variações léxicas, é aprimorada. Esse aperfeiçoamento contribui, por conseguinte, para uma recuperação mais precisa e abrangente de resultados relevantes em cenários de complexidade.

REFERÊNCIAS

[1] Paul Clough and Mark Sanderson. 2013. Evaluating the performance of information retrieval systems using test collections. *Information Research* 18 (06 2013).

[2] Mehdi Bel Fdhila. 2023. *A COMPARISON OF SEARCHING DATA WITH, AND WITHOUT ELASTICSEARCH IN A SQL DATABASE.* Bachelor's Thesis. MÄLARDALEN UNIVERSITY SCHOOL OF INNOVATION, DESIGN AND ENGINEERING VÄSTERÅS, SWEDEN.

[3] Swapna Gottipati, David Lo, and Jing Jiang. 2011. Finding Relevant Answers in Software Forums. In *Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering (ASE '11)*. IEEE Computer Society, USA, 323–332. https://doi.org/10.1109/ASE.2011.6100069

[4] Silvana Greca, Anxhela Kosta, and Suela Maxhelaku. 2018. Optimizing Data Retrieval by Using Mongodb with Elasticsearch. In *International Conference on Recent Trends and Applications in Computer Science and Information Technology*. https://api.semanticscholar.org/CorpusID:57661138

[5] Donna Harman. 2011. *Information Retrieval Evaluation* (1st ed.). Morgan & Claypool Publishers.

[6] Raj Jain. 1991. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.* Wiley. I–XXVII, 1–685 pages.

[7] Anitha Kannan, Inmar Givoni, Rakesh Agrawal, and Ariel Fuxman. 2011. Matching Unstructured Product Offers to Structured Product Descriptions. (01 2011).

[8] D.J. Lilja. 2005. *Measuring Computer Performance: A Practitioner's Guide.* Cambridge University Press. https://books.google.com.br/books?id=jb68T-OuIC4C

[9] Divyanshu Marwah and Joeran Beel. 2020. Term-Recency for TF-IDF, BM25 and USE Term Weighting. In *Proceedings of the 8th International Workshop on Mining Scientific Publications*, Petr Knoth, Christopher Stahl, Bikash Gyawali, David Pride, Suchetha N. Kunnath, and Drahomira Herrmannova (Eds.). Association for Computational Linguistics, Wuhan, China, 36–41. https://aclanthology.org/2020.wosp-1.5

[10] Patrick E. McKnight and Julius Najab. 2010. *Mann-Whitney U Test.* John Wiley Sons, Ltd, 1–1. https://doi.org/10.1002/9780470479216.corpsy0524 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470479216.corpsy0524

[11] Edson Marchetti da Silva. 2019. Estudo comparativo de indexação complete de texto para recuperação de informações em sistemas gerenciadores de banco de dados. *InCID: Revista de Ciência da Informação e Documentação* 10, 1 (maio 2019), 281–301. https://doi.org/10.11606/issn.2178-2075.v10i1p281-301

[12] Konstantinos Vavliakis, George Katsikopoulos, and Andreas Symeonidis. 2019. E-commerce Personalization with Elasticsearch. *Procedia Computer Science* 151 (01 2019), 1128–1133. https://doi.org/10.1016/j.procs.2019.04.160