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

**LUCIAN JULIO FELIX DA COSTA**

# DETECÇÃO DE SOBREPREÇO E SUBPREÇO EM AUDITORIA DE LICITAÇÕES

**CAMPINA GRANDE - PB**

**2024**

**LUCIAN JULIO FELIX DA COSTA**

# DETECÇÃO DE SOBREPREÇO E SUBPREÇO EM AUDITORIA DE LICITAÇÕES

**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 : Cláudio de Souza Baptista**

**CAMPINA GRANDE - PB**

**2024**

**LUCIAN JULIO FELIX DA COSTA**

# DETECÇÃO DE SOBREPREÇO E SUBPREÇO EM AUDITORIA DE LICITAÇÕES

**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:

**Cláudio de Souza Baptista**

**Orientador – UASC/CEEI/UFCG**

**José Antão Beltrão Moura**

**Examinador – UASC/CEEI/UFCG**

**Francisco Vilar Brasileiro**

**Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 15 de maio de 2024**

**CAMPINA GRANDE - PB**

# RESUMO

Auditar processos de licitação é crucial para garantir transparência e eficiência na gestão dos fundos públicos. No entanto, a crescente complexidade das licitações representa um desafio para os auditores, especialmente na detecção oportuna de irregularidades. Este trabalho apresenta um software de comparação de preços desenvolvido para facilitar a análise de sobrepreço e subpreço em licitações de obras públicas, fazendo uso de mineração de dados e busca semântica. A ferramenta tem como objetivo fornecer visualizações que auxiliem na detecção de possíveis fraudes, utilizando dados atualizados de mercado. Espera-se que essa ferramenta contribua para a eficiência da fiscalização em licitações de obras públicas, melhorando a análise financeira e dificultando práticas fraudulentas por parte dos licitantes.

# DETECTION OF OVERPRICING AND UNDERPRICING IN TENDER AUDITING

## ABSTRACT

Auditing tender processes is crucial to ensure transparency and efficiency in the management of public funds. However, the increasing complexity of tenders poses a challenge for auditors, especially in the timely detection of irregularities. This work presents a price comparison software developed to facilitate the analysis of overpricing and underpricing in public works tenders, utilizing data mining and semantic search. The tool aims to provide visualizations that assist in detecting potential frauds, using updated market data. It is expected that this tool will contribute to the efficiency of supervision in public works tenders, improving financial analysis and hindering fraudulent practices by bidders.

# Detecção de Sobrepreço e Subpreço em Auditoria de Licitações

Lucian Julio Felix da Costa
lucian.costa@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

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

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

## RESUMO

Auditar processos de licitação é crucial para garantir transparência e eficiência na gestão dos fundos públicos. No entanto, a crescente complexidade das licitações representa um desafio para os auditores, especialmente na detecção oportuna de irregularidades. Este trabalho apresenta um software de comparação de preços desenvolvido para facilitar a análise de sobrepreço e subpreço em licitações de obras públicas, fazendo uso de mineração de dados e busca semântica. A ferramenta tem como objetivo fornecer visualizações que auxiliem na detecção de possíveis fraudes, utilizando dados atualizados de mercado. Espera-se que essa ferramenta contribua para a eficiência da fiscalização em licitações de obras públicas, melhorando a análise financeira e dificultando práticas fraudulentas por parte dos licitantes.

## PALAVRAS-CHAVE

Licitações, Fraudes, Busca Semântica, Sobrepreço, Subpreço, REST

## REPOSITÓRIO

https://github.com/Lucianjfc/auditoria-obras.

## 1 INTRODUÇÃO

A auditoria de processos licitatórios por parte das Cortes de Contas do Brasil é uma atividade essencial para garantir a integridade e transparência dos processos de contratação pública, bem como para promover a eficiência na gestão dos fundos públicos em diversos órgãos de governo, nas esferas municipal, estadual e nacional. Atualmente, essas auditorias são realizadas mediante análises conduzidas por auditores, visando detectar e corrigir possíveis irregularidades durante as etapas de um processo licitatório [3, 6–8].

O aumento de volume e complexidade das licitações representa um desafio considerável para a eficácia da análise por parte dos auditores, incluindo-se a tempestividade. Esse desafio é agravado pelo fato de que a identificação de sobrepreços ou subpreços pode ser uma tarefa complexa e demorada. O sobrepreço refere-se à prática de estabelecer preços acima do valor justo ou de mercado para um bem, produto ou serviço específico, enquanto o subpreço ocorre quando os valores são fixados abaixo do valor justo de mercado.

Além disso, as disparidades nos valores de compras podem resultar em má gestão dos recursos públicos, atos fraudulentos e, em última instância, na injustiça da contratação entre os licitantes, o que fere a legislação vigente.

Neste trabalho foi desenvolvido um software de comparação de preços, elaborado com o intuito de facilitar a identificação de sobrepreços e subpreços em processos licitatórios. A aplicação busca prover visualizações que contribuem na detecção de possíveis fraudes em licitações de obras considerando dados atuais do mercado. Para fins de simplificação, neste trabalho, foram utilizados dados provenientes apenas do estado da Paraíba.

Com o desenvolvimento dessa ferramenta, espera-se contribuir para a eficiência da fiscalização no que tange à auditoria de licitações públicas. Dessa forma, auxiliando na análise das finanças públicas e nas práticas de preços adotadas pelos licitantes em relação aos preços adotados pelo mercado, dificultando assim a ocorrência de licitações fraudulentas.

No decorrer deste artigo, são apresentados os trabalhos relacionados na seção 2, seguidos pela discussão das principais funcionalidades da aplicação na seção 3, que inclui a estratégia usada para a análise de sobrepreço e subpreço. Posteriormente na seção 4, são detalhadas as avaliações da solução desenvolvida. As considerações finais são abordadas na seção 5.

## 2 TRABALHOS RELACIONADOS

A detecção de sobrepreço e subpreço é um problema comum no que tange às compras realizadas pelos sistemas governamentais [13, 15]. Isso naturalmente conduz à realização de pesquisas de preços que buscam formas de detectar compras que apresentam indícios de fraudes visando auxiliar a transparência e integridade do governo brasileiro.

Em Silva et al. (2023) [14] é estudado a detecção de sobrepreço por meio de uma análise estatística, empregando técnicas de Processamento de Linguagem Natural (NLP) para agrupar itens. Embora não se concentre no desenvolvimento de uma aplicação para consulta de preços, o artigo explora três abordagens distintas de agrupamento. Essas abordagens podem ser utilizadas, posteriormente, visando melhorar a qualidade das buscas realizadas na ferramenta desenvolvida neste trabalho.

Correa e Leal (2018) [5] propõe uma metodologia para detectar sobrepreços em compras governamentais, especialmente focando nos medicamentos adquiridos pelo Ministério da Saúde, utilizando dados não estruturados disponíveis no Portal da Transparência[1]. O trabalho descreve técnicas de mineração de texto e agrupamento baseadas em ontologias para identificar e classificar automaticamente os produtos. Como resultado, é gerada uma base de preços consolidada para medicamentos, possibilitando a detecção de distorções nos preços e a revelação de possíveis fraudes.

No trabalho elaborado por Oliveira et al. (2023) [12], foi proposto uma abordagem para ranquear licitações públicas suspeitas de possuírem algum indício de fraude através da criação de dezenove trilhas de Auditagem que avaliam características do processo Licitatório e gerando um grau de risco para auxiliar e guiar os auditores para processos com um maior potencial de fraude. O artigo não trata de trilhas que analisam compras realizadas em licitações, uma

[1] https://portaldatransparencia.gov.br/

contribuição significativa seria a criação de trilhas que analisam o sobrepreço de compras públicas, possibilitando uma análise mais aprofundada para Licitações de Obras Públicas.

O uso da busca semântica para a correspondência de itens vem sendo pesquisada de maneira mais intensiva com o objetivo de superar as deficiências da correspondência léxica, como a sensibilidade às variantes ortográficas. Estudos como [1, 4, 11] têm explorado o emprego de modelos semânticos para aprimorar a correspondência de produtos.

A revisão da literatura ressalta a relevância da identificação de possíveis indícios de fraudes em aquisições públicas. Por fim, a variedade de abordagens destinadas a fortalecer a transparência nas compras governamentais é de suma importância para assegurar a eficácia na administração pública. Além dos estudos existentes, este trabalho contribui oferecendo uma aplicação para consulta histórica de preços de materiais e serviços de obras, juntamente com atualizações periódicas, o que garante uma precisão aprimorada nas comparações realizadas.

## 3 SOLUÇÃO PROPOSTA

Nesta seção, são destacadas as principais decisões tomadas para o desenvolvimento da ferramenta, abordando sua arquitetura e as ferramentas utilizadas, além de detalhar seus aspectos e especificidades. Também serão discutidas as bases de dados selecionadas como fonte para as comparações de preços.

### 3.1 Base de Dados: SICRO e SINAPI

A base de dados da SICRO é composta por mais de seis mil itens, refletindo a complexidade e a diversidade de materiais e serviços. Além disso, considera fatores temporais e espaciais, sendo possível consumir informações de diferentes regiões e datas. Possibilitando uma maior precisão para as consultas e comparações de preço.

O SINAPI é um banco de dados mantido pela Caixa Econômica Federal, que contém uma variedade de informações sobre os preços de insumos e serviços relacionados à construção civil em todo o Brasil. Embora tenha sido desenvolvido como um parâmetro para obras públicas, o SINAPI também serve como referência para projetos e serviços de construção civil no setor privado, especialmente para aqueles que ainda não foram executados. Os dados do SINAPI estão organizados por estado, permitindo o acompanhamento da evolução dos preços de insumos e serviços em cada região. Assim como na SICRO, os fatores temporais e espaciais também estão presentes.

Ambas as bases de dados, SICRO E SINAPI, possuem referências de preços desonerados e sem desoneração, possibilitando que a análise de sobrepreço e subpreço considere essa característica que implica totalmente no custo das obras. Itens desonerados referem-se aos custos de mão de obra que não incluem encargos sociais relacionados à contribuição de 20% do INSS sobre a folha de pagamento. Por outro lado, itens não desonerados indicam os custos de mão de obra que incluem tais encargos sociais.

Atualmente, obtém-se dados do site oficial da SICRO[2], do Governo Federal, e do SINAPI[3], mantido pela Caixa Econômica Federal. Essas plataformas fornecem informações detalhadas sobre custos e índices da construção civil, essenciais para análises precisas e tomadas de decisão no setor. Um exemplo dos dados pode ser visualizado na Tabela 1.

A escolha dessas bases de dados apresenta grandes vantagens para a análise dos preços nas licitações, proporcionando uma referência confiável e abrangente dos custos de materiais e serviços no setor da construção civil. No entanto, a complexidade das descrições dos dados coletados traz desafios significativos para a realização de buscas que garantam resultados satisfatórios. Essa complexidade pode incluir a variação nos formatos de apresentação dos dados, a falta de padronização das terminologias utilizadas e a necessidade de lidar com uma grande quantidade de informações dispersas.

Todos esses aspectos dificultam a eficiência da busca e a análise precisa dos preços, exigindo métodos e técnicas avançadas para a extração e interpretação dos dados. O conjunto de dados da SICRO e SINAPI possui diferentes características, a SICRO em comparação possui uma maior quantidade de materiais/serviços únicos provendo uma maior variedade de materiais e serviços para serem consultados. Além disso, os dados coletados se deram no período de janeiro de 2023 a fevereiro de 2024. As quantidades podem ser consultadas na Figura 1.

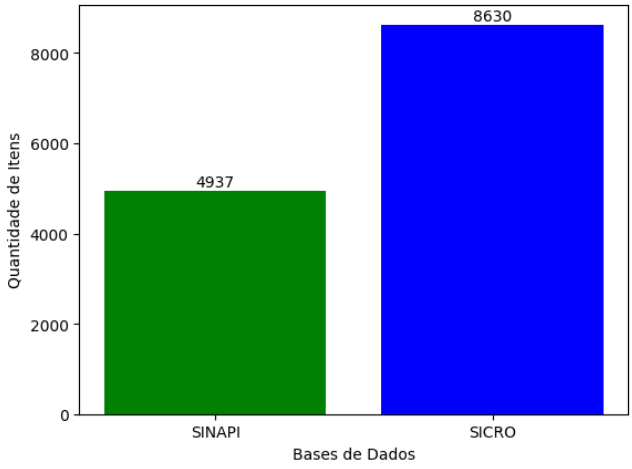


**Figura 1: Quantidade de itens nas bases de dados.**

Com acesso a informações atualizadas e detalhadas sobre preços, é possível identificar discrepâncias significativas entre os valores orçados e os valores praticados no mercado. Isso possibilita uma avaliação mais precisa dos custos envolvidos em um projeto, permitindo que sejam adotadas medidas corretivas para garantir a viabilidade financeira e a transparência nas licitações públicas, estimulando a competição justa e promovendo uma maior confiabilidade nos preços descritos pelos licitantes.

### 3.2 Visão Geral da Ferramenta

A ferramenta desenvolvida oferece mecanismos para consultar os preços de materiais e serviços de obras, utilizando um motor de busca integrado à busca semântica para aprimorar a qualidade dos resultados obtidos. Além disso, a aplicação disponibiliza uma

[2]https://www.gov.br/dnit/pt-br/assuntos/planejamento-e-pesquisa/custos-e-pagamentos/custos-e-pagamentos-dnit/sistemas-de-custos/sicro$_{antiga}$/nordeste/nordeste
[3]https://www.caixa.gov.br/site/Paginas/downloads.aspxcategoria$_{6}$64

**Tabela 1: Base de Dados.**

| Base de Dados | Descrição | Unidade | Preço |
|---|---|---|---|
| SINAPI | Eletricista - mensalista | Mês | R$ 1.955,05 |
| SINAPI | BLOCO DE CONCRETO ESTRUTURAL 14 X 19 X 29 CM, FBK 10 MPA | UN | R$ 4,71 |
| SINAPI | CABO DE COBRE NU 10 MM2 MEIO-DURO | m | R$ 10,68 |
| SICRO | CHAPA DE MDF CRU, E = 12 MM, DE *2,75 X 1,85* M | $m^2$ | R$ 38,31 |
| SICRO | Parafuso de cabeça chata em aço - D = 4,5 mm e bucha plástica - D = 8 mm (S8) | UN | R$ 0,30 |
| SICRO | Chapa de alumínio - E = 1,5 mm | $m^2$ | R$ 197,57 |

interface clara que apresenta as informações essenciais para análise de sobrepreço e subpreço, acompanhada do gráfico de curva ABC, que auxilia na identificação de elementos que carecem de maior atenção na relação de custos da obra.

Destacam-se nas principais funcionalidades desta aplicação: a coleta automatizada dos dados das tabelas do Sistema de Custos Referenciais de Obras (SICRO) e Sistema Nacional de Pesquisa de Custos e Índices da Construção Civil (SINAPI), a realização de consultas rápidas através do ElasticSearch[4], utilizando-se de busca semântica para o retorno de resultados mais relevantes, correspondência de materiais/serviços de forma automática para análise de sobrepreço e subpreço dos itens descritos em licitações em relação ao mercado.

O projeto arquitetural da aplicação baseia-se numa arquitetura REST, segmentada em *frontend*, *backend* e associada a um banco de dados relacional Sql Server e ao ElasticSearch, conforme detalhado na Figura 2.

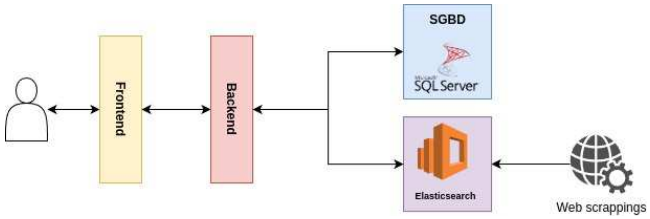


**Figura 2: Arquitetura da Aplicação.**

## 3.3 Frontend

O frontend é composto por um aplicativo React [5] que adota a implementação do padrão Flux [6] para o fluxo de dados no frontend, conforme ilustrado na Figura 3. O padrão Flux é constituído por quatro elementos essenciais: a View, representada pelos componentes React responsáveis por apresentar os dados da aplicação ao usuário; o Store, que consiste em estruturas encarregadas de adquirir, manipular e armazenar os dados na aplicação; o Dispatcher, cuja função é gerenciar o fluxo de dados, distribuindo as ações originadas da View para seus respectivos Stores; e, por último, a Action, que se refere a funções que transportam dados destinados aos Stores e originados de ações realizadas pelo usuário. Esses elementos, em conjunto, formam uma arquitetura coesa que viabiliza uma interação fluida e eficiente entre os diferentes componentes da aplicação frontend.

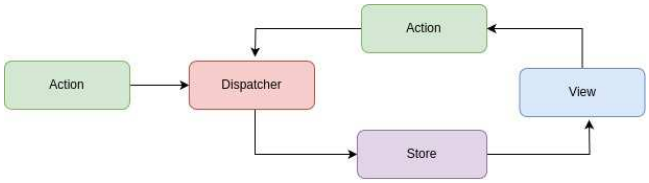


**Figura 3: Estrutura de fluxo de dados de frontend.**

Existem diferentes implementações do padrão Flux, porém a utilizada foi em associação com o MobX [7], uma biblioteca de gerenciamento de estado que desempenha simultaneamente as funções de Action e Dispatcher. A camada do padrão Flux é abstraída, tornando-se transparente para o desenvolvedor com o uso do MobX.

No núcleo do frontend, destaca-se a visualização da análise de sobrepreço, desenvolvida com o apoio das bibliotecas PrimeReact e Echarts. Todos os dados utilizados são armazenados em um Store e obtidos por meio de requisições HTTP ao backend, utilizando a biblioteca Axios [8], e posteriormente armazenados em um banco de dados no SQL Server.

A busca no Elasticsearch é realizada por meio de um proxy configurado no backend da aplicação. Para estruturar a query enviada ao Elasticsearch e fazer uso da busca semântica, é empregada a biblioteca Elasticsearch Client [9]. Os dados retornados pelo Elasticsearch são convertidos em objetos Java antes de serem enviados ao frontend da aplicação.

Esses dados são então utilizados para renderizar os materiais/serviços pesquisados pelo usuário, utilizando o componente DataTable do PrimeReact. Isso permite a visualização das informações de preços de materiais ou serviços específicos, além de possibilitar a comparação de preços.

## 3.4 Backend

O backend da aplicação é composto por um servidor web que processa requisições HTTP seguindo o padrão de API REST. Utilizando o framework Spring Boot em conjunto com Java, foi desenvolvida uma API que adota uma arquitetura dividida em três camadas principais: os controladores, que direcionam as requisições HTTP para os serviços correspondentes; os serviços, responsáveis por manipular

[4]https://www.elastic.co
[5]https://react.dev/
[6]https://facebookarchive.github.io/flux/
[7]https://mobx.js.org/README.html
[8]https://www.npmjs.com/package/axios/
[9]https://elasticsearch-py.readthedocs.io/en/v8.13.0/

os dados e aplicar regras de negócio com base nas informações processadas pelos repositórios; por fim, os repositórios, encarregados da construção de consultas e administração dos dados, incluindo consultas específicas ao Elasticsearch.

A maioria dos repositories no *backend* simplesmente redireciona as chamadas da API para o banco de dados por meio de consultas básicas, exceto pelo *sinapiElasticRepository* e pelo *sicroElasticRepository*, responsáveis por elaborar consultas específicas para o Elasticsearch. Além disso, o *backend* é configurado com a Java Persistence API (JPA) em conjunto com o Hibernate. Essa escolha foi motivada pela eficácia e facilidade de uso oferecidas por essas tecnologias ao interagir com bancos de dados relacionais, como o SQL Server. O Hibernate simplifica a persistência de objetos Java no banco de dados, enquanto o JPA estabelece padrões para o mapeamento objeto-relacional.

Essa ferramenta auxilia para uma interação eficiente e robusta com o banco de dados relacional da aplicação. Abaixo, será exemplificado o esquema relacional utilizado para armazenar as informações das análises realizadas pelo usuário, assim como as referências aos itens correspondentes da análise de sobrepreço e subpreço.

Também foi adotado o liquibase, uma ferramenta que proporciona um método estruturado e controlado para gerenciar e aplicar alterações no esquema do banco de dados. Ferramenta crucial para manter a consistência e a integridade dos dados, especialmente em ambientes de desenvolvimento colaborativo ou em projetos de grande escala. Além disso, o Liquibase simplifica o controle de versionamento do banco de dados, permitindo que todas as alterações sejam registradas e rastreadas ao longo do tempo. Isso facilita a identificação de mudanças.

Na Figura 4 é possível visualizar o esquema conceitual utilizando-se do modelo entidades e relacionamentos (MER) das tabelas utilizadas no mapeamento ORM.

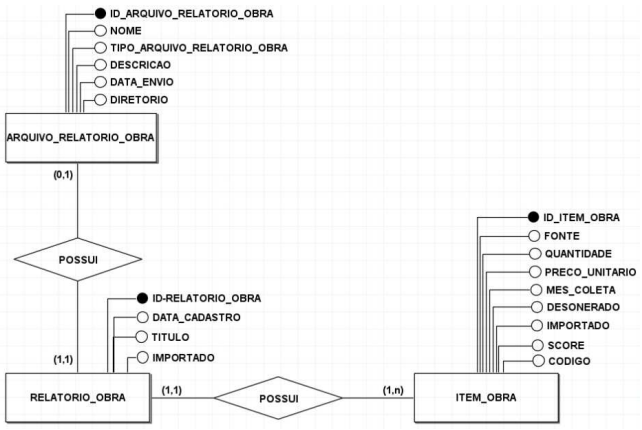


**Figura 4: Esquema conceitual para o Sql Server.**

### 3.4.1 *Análise Automática.*

Uma funcionalidade essencial do sistema é a capacidade do usuário de submeter um arquivo XLSX seguindo um template prédefinido. Esse arquivo contém informações como código, descrição, quantidade e preço unitário, representando as compras realizadas em uma licitação. Após a submissão do arquivo para o backend, é realizada uma validação de sua estrutura para verificar se o mesmo está conforme o esquema esperado. Em seguida, inicia-se o processo de correspondência automática dos itens contidos no arquivo.

Através da busca semântica, é realizada a correspondência de cada item do arquivo com os itens armazenados nos índices do Elasticsearch. Cada item é processado individualmente, onde se o código tiver sido fornecido é realizada uma busca apenas usando o código para retornar o item exato. No caso em que o código não é fornecido, é realizada uma busca semântica utilizando sua descrição, visando encontrar os itens mais semelhantes. Para avaliar os itens retornados pelo Elasticsearch, utiliza-se o Score, uma métrica fornecida pelo próprio Elasticsearch que indica a relevância dos resultados em relação à consulta realizada.

Finalmente, os itens retornados com um Score superior a 0.85 são considerados potencialmente similares. O item que apresenta o maior Score é então selecionado como o correspondente ao item descrito no documento. No entanto, caso nenhum item retornado alcance este limiar de relevância, a correspondência não é realizada e o usuário é notificado sobre a necessidade de associação manual deste item pendente.

### 3.4.2 *ElasticSearch.*

Atualmente, os motores de busca vêm desempenhando um papel cada vez mais importante em diversos tipos de aplicações, dada a necessidade de melhorar o tempo de respostas e a precisão na busca de informações. Entre esses motores, o Elasticsearch tem se destacado como uma solução robusta e versátil. Desenvolvido pela Elastic, o Elasticsearch é um mecanismo de busca distribuído e de código aberto, baseado no Apache Lucene[10], que permite armazenar, pesquisar e analisar grandes volumes de dados de forma eficiente e em tempo real.

Uma das principais características do Elasticsearch é sua capacidade de escalabilidade horizontal, o que significa que ele pode lidar com quantidades abundantes de dados distribuídos em vários nós de forma eficaz. Isso o torna ideal para lidar com o crescente volume de informações geradas em aplicações modernas, como registros de logs, métricas de aplicativos, documentos e dados de análise de negócios [2].

Além dos recursos mencionados anteriormente, o Elasticsearch também oferece suporte à busca semântica, permitindo que os usuários executem consultas mais avançadas e intuitivas. Neste contexto, o ElasticSearch contribuiu no desenvolvimento deste projeto, integrando-se à busca semântica para aprimorar a correspondência entre os itens listados nas licitações e aqueles catalogados nos bancos de dados SICRO e SINAPI.

A versão utilizada do Elasticsearch foi a 8.6.2. Sua execução deu-se via um container Docker[11]. Para otimizar as consultas e as correspondências, foram criados dois índices para os dados da SICRO e da SINAPI, dessa forma, segue o código utilizado para a criação do índice da SINAPI e SICRO respectivamente.

```
{
    "settings":{
        "number_of_shards": 2,
```

[10] https://lucene.apache.org/
[11] https://www.docker.com/

```
        "number_of_replicas": 1,
        "analysis": {
            "filter": {
                    "brazilian_stop": {
                    "type":       "stop",
                    "stopwords":  "_brazilian_"
                },
                "brazilian_stemmer": {
                    "type":       "stemmer",
                    "language":   "brazilian"
                }
            },
            "analyzer": {
                "rebuilt_brazilian": {
                    "tokenizer":  "standard",
                    "filter": [
                        "lowercase",
                        "brazilian_stop",
                        "brazilian_stemmer"
                    ]
                }
            }
        }
    },
    "mappings": {
        "properties": {
            "CODIGO": {
                "type": "keyword"
            },
            "DESCRICAO": {
                "type": "text",
                "analyzer": "rebuilt_brazilian"
            },
            "ALL_MINI_VECT": {
                "type": "dense_vector",
                "dims": 384,
                "index": True,
                "similarity": "cosine"
            },
            . . .
        }
    }
}
```

Para a indexação dos dados, foi utilizada a biblioteca Elasticsearch Client que provê uma API para se comunicar com o Elasticsearch. O código desenvolvido foi codificado na linguagem Python de forma similar aos Scrapers citados na seção 3.5.

## 3.5 Coleta dos Dados

Para a coleta de dados da SICRO e SINAPI, foram desenvolvidos dois Web Scrapers na linguagem Python em conjunto com as bibliotecas Requests[12] e BeautifulSoup[13]. Os scrappers têm como responsabilidade extrair os documentos contendo informações sobre materiais/serviços de obras e seus respectivos preços nos sites da SICRO e SINAPI.

Na primeira etapa, os scrapers extraem o HTML das páginas que contêm os links de redirecionamento para os documentos de download. Cada link é analisado com o objetivo de identificar características relevantes, como estado, ano e mês referenciados pelo mesmo. Apenas os links novos, ou aqueles que resultaram em erro em tentativas anteriores, são considerados na nova extração. Os documentos coletados são armazenados e os links processados são registrados no histórico de coletas bem-sucedidas.

Para gerenciar o histórico de coletas, foi utilizado um banco de dados SQL Server, onde são armazenadas as informações sobre as coletas bem-sucedidas. Esse banco de dados permite acompanhar o progresso das coletas ao longo do tempo e evita a coleta duplicada de dados. Ademais, é realizado um pré-processamento das informações extraídas, incluindo a extração das informações dos documentos no formato XLSX.

Na etapa final, os dados são estruturados em um formato de documento onde cada material ou serviço contém informações sobre seus preços ao longo do tempo. Se um item já tiver sido coletado anteriormente, apenas o novo preço é incluído. Por fim, os dados são indexados no ElasticSearch.

Na Figura 5 é possível visualizar a arquitetura dos componentes utilizados nesse processo.

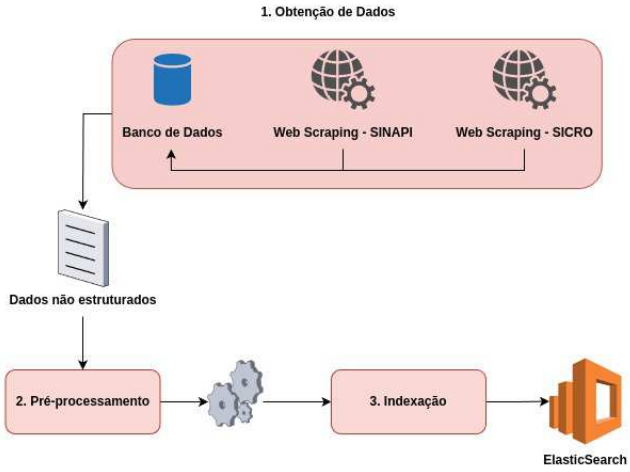


**Figura 5: Arquitetura dos componentes na coleta de dados.**

## 3.6 Busca Semântica

A busca semântica vai além da correspondência exata de palavras-chave e considera o significado e o contexto das consultas. Isso é alcançado por meio de recursos como análise de linguagem natural, que permite que o Elasticsearch compreenda a intenção por trás das consultas e retorne resultados mais relevantes, garantindo uma melhor correspondência para as buscas realizadas pelos usuários.

Para alcançar uma melhor correspondência de itens, a busca semântica foi integrada em conjunto com o Elasticsearch. Isso resultou em um desempenho satisfatório para as consultas, permitindo

[12] https://pypi.org/project/requests/
[13] https://beautiful-soup-4.readthedocs.io/en/latest/

uma compreensão mais profunda da intenção semântica por trás das buscas realizadas pelos usuários. Um exemplo do uso da busca semântica pode ser visualizado na Tabela 2, onde são exibidos os 5 resultados mais relevantes retornados.

**Tabela 2: Exemplo de Busca Semântica.**

| Consulta | Resultado da Consulta |
|---|---|
| AUXILIAR DE PINTOR | AJUDANTE DE PINTOR (HORISTA)<br>AJUDANTE DE PINTOR (MENSALISTA)<br>PINTOR DE LETREIROS (HORISTA)<br>PINTOR (HORISTA)<br>AUXILIAR DE MECÂNICO |

## 3.7 Métodos de Busca

Para implementação da busca semântica, foi avaliado o uso de quatros modelos semânticos diferentes: all-mpnet-base-v2[14], quora-distilbert-multilingual[15], all-MiniLM-L6-v2[16] e LaBSE[17]. Visando avaliar o melhor modelo para aplicação, foram realizados dois testes independentes para cada base de dados, com o intuito de avaliar qual modelo semântico desempenha melhor na tarefa de retornar resultados relevantes.

A detecção de sobrepreço ocorre quando o valor de um item excede o seu preço de mercado, enquanto o subpreço é identificado quando o valor fica abaixo. Recuperar resultados relevantes por meio da busca semântica é essencial na identificação de discrepâncias de preço, permitindo uma análise mais precisa e eficaz do mercado.

Nesse contexto, para garantir a qualidade dos resultados, recorreu-se à métrica NDCG (Normalized Discounted Cumulative Gain) [9, 10], uma métrica comumente utilizada para avaliar motores de busca, comparando as classificações com uma ordem ideal, onde todos os itens ideais estão no topo da lista.

O NDCG combina a relevância dos itens com suas posições na lista para calcular uma pontuação que varia de 0 a 1, representando a qualidade da classificação. O NDCG é calculado com base no DCG (Discounted Cumulative Gain), que leva em consideração a relevância dos itens ponderados pela posição na lista. A fórmula para calcular o NDCG é apresentada na Equação 1, onde a relevância dos itens é somada e descontada de acordo com sua posição na lista.

$$\text{nDCG} = \frac{\text{DCG}}{\text{IDCG}} \tag{1}$$

O experimento foi realizado de forma individual nas bases da SICRO e SINAPI, onde, para cada base de dados, foi separado um conjunto de 100 descrições com variações de tamanho. A métrica utilizada para avaliação da relevância dos resultados é bastante afetada pela qualidade de registros retornados. Portanto, a quantidade de registros retornados variou entre 1 e 100 registros. Além disso, o mesmo conjunto de descrições de cada base de dados foi executado para cada modelo semântico.

[14]https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[15]https://huggingface.co/sentence-transformers/quora-distilbert-multilingual
[16]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[17]https://huggingface.co/sentence-transformers/LaBSE

## 3.8 Conteinerização

Para simplificar as operações relacionadas ao deployment da aplicação, tanto o frontend quanto o backend possuem arquivos Dockerfile, sendo os arquivos de configuração para a construção de containers Docker. Esses Dockerfiles especificam as imagens Docker base e intermediárias a serem usadas na construção da imagem final, além de definir parâmetros específicos para essa construção.

No frontend, foi utilizada a imagem base do NodeJS com Alpine Linux para o processo de compilação da aplicação React. Além disso, foi usado o Webpack, um empacotador de módulos JavaScript, e o Babel, um transpilador JavaScript, empregados para a compilação. Isso resulta na geração de conjuntos de arquivos JavaScript, CSS e HTML, que são então copiados para o diretório interno de uma imagem NGINX, também executada em um ambiente Alpine Linux. O servidor NGINX é responsável por servir os arquivos resultantes do build da aplicação React em resposta às requisições HTTP. No backend, foi usada a imagem base da JDK8 com Alpine Linux, seguindo uma abordagem similar.

Além dos contêineres de frontend e backend, também foram criados contêineres para execução dos scrapings responsáveis por extrair os dados da SICRO e SINAPI, e a indexação dos dados no Elasticsearch. Por fim, foi definido um arquivo de configuração do Docker Compose que executa um banco de dados SQL Server com o Elasticsearch, facilitando o deployment da aplicação em algum servidor ou em uso de ambiente de desenvolvimento. Na Figura 6, a arquitetura da solução utilizando a conteinerização Docker é apresentada.

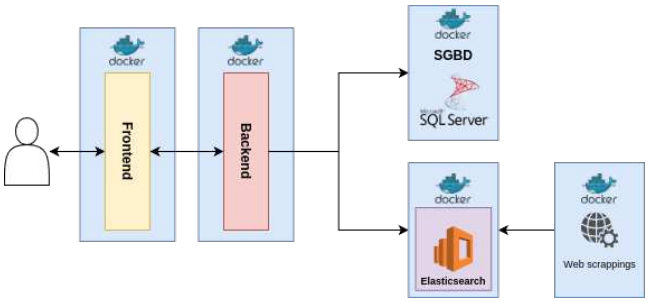


**Figura 6: Arquitetura base utilizando conteinerização Docker.**

# 4 RESULTADOS

Nesta seção será apresentado o uso da ferramenta para a criação de um relatório para identificação de sobrepreço e subpreço. Além disso, serão apresentados e analisados os resultados obtidos durante a execução dos experimentos da busca semântica.

## 4.1 Ferramenta

O desenvolvimento da ferramenta foi idealizado visando a facilidade de usuários Auditores construírem relatórios para a identificação de irregularidades nos preços das obras. Na Figura 7, é possível visualizar a tela de cadastro de um relatório exigindo as informações do título, autor e valor da licitação.

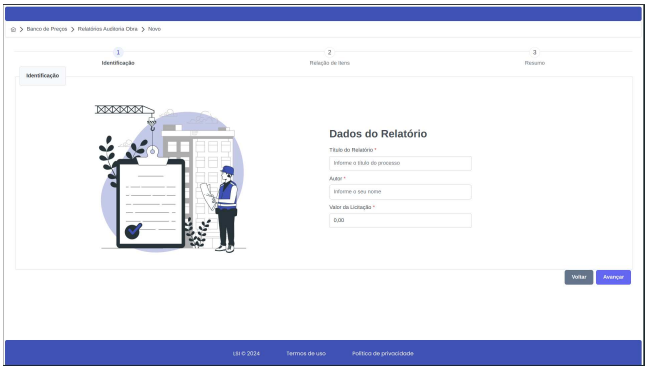


**Figura 7: Tela de cadastro das informações do relatório.**

Em seguida, para ser realizada a análise dos materiais e serviços adquiridos na licitação, é disponibilizada uma tela para a pesquisa e consulta dos preços nas bases de dados da SICRO e SINAPI, conforme apresentado na Figura 8.

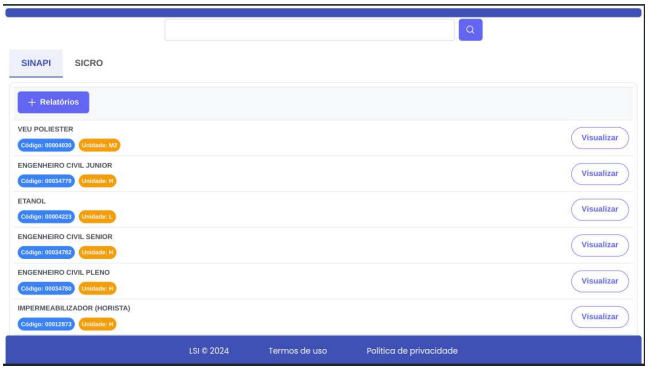


**Figura 8: Tela de consultas de materiais/serviços.**

Conforme ilustrado na Figura 9, é possível visualizar as informações de preço do item selecionado como também consultar o seu preço de forma histórica através do gráfico de linha. Além disso, é possível considerar o valor do item selecionado, incluindo a desoneração ou não, fator que colabora para o seu preço. Outras informações, como a quantidade e valor do item adquirido na licitação, também devem ser preenchidas para a realização da análise de sobrepreço e subpreço.

Com a construção do relatório finalizada, é apresentada a tela de resumo das informações, onde é possível visualizar os itens que possuem sobrepreço ou subpreço, além de consultar a curva ABC para identificação dos itens que mais encarecem a obra. Um exemplo de um relatório construído pode ser visto na Figura 11 onde existem 3 itens com sobrepreço detectado.

Por fim, com o objetivo de facilitar a construção dos relatórios, a aplicação também disponibiliza a importação desses itens adquiridos por meio de um arquivo XLSX, seguindo o template pré-definido exemplificado na seção 3.3.1. Essa funcionalidade é essencial para possibilitar a integração dessa aplicação com os sistemas de tribunais de contas, tornando a montagem desses relatórios de forma automatizada. Um exemplo da tela de importação pode ser visualizado na Figura 10.

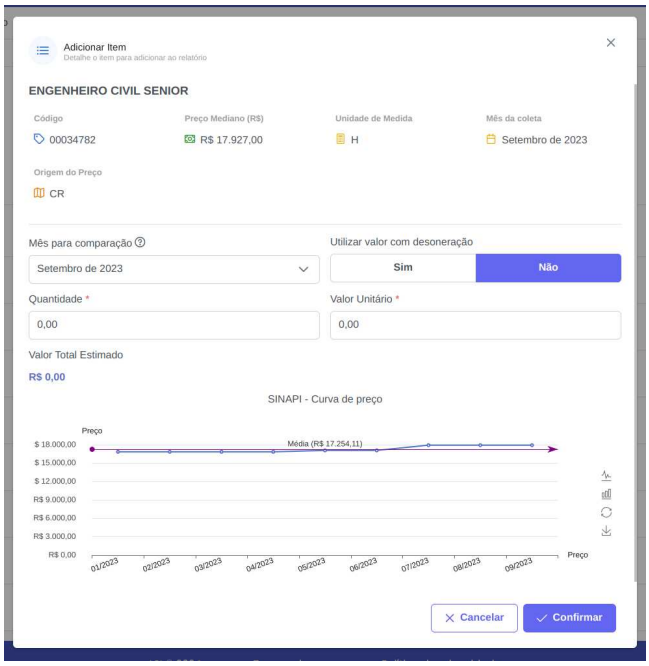


**Figura 9: Tela de adição de um item para o relatório.**

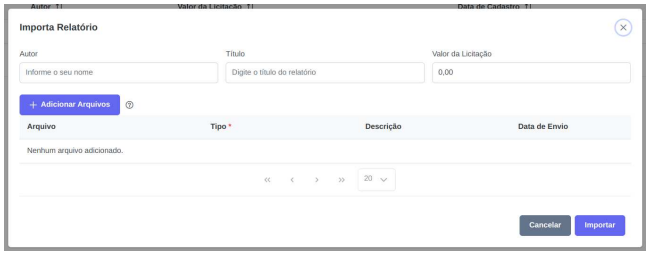


**Figura 10: Tela de importação do relatório por XLSX.**

## 4.2 Busca Semântica

Na Figura 12 é apresentada a execução na base da SINAPI, observa-se que o All Mini manteve resultados superiores em termos de NDCG para quaisquer valores de k. No entanto, a disparidade entre a curva do Mpnet se manteve pequena ao longo de todo o experimento. Por fim, o Labase e Quora se mantiveram com resultados bastante inferiores se comparados ao All Mini e o Quora, não se demonstrando eficientes no retorno de resultados relevantes.

Similarmente, na Figura 13 pode-se observar a execução na base da SICRO, de forma semelhante, o All Mini manteve resultados superiores em comparação com os outros modelos. No entanto, todos os modelos mantiveram resultados próximos.

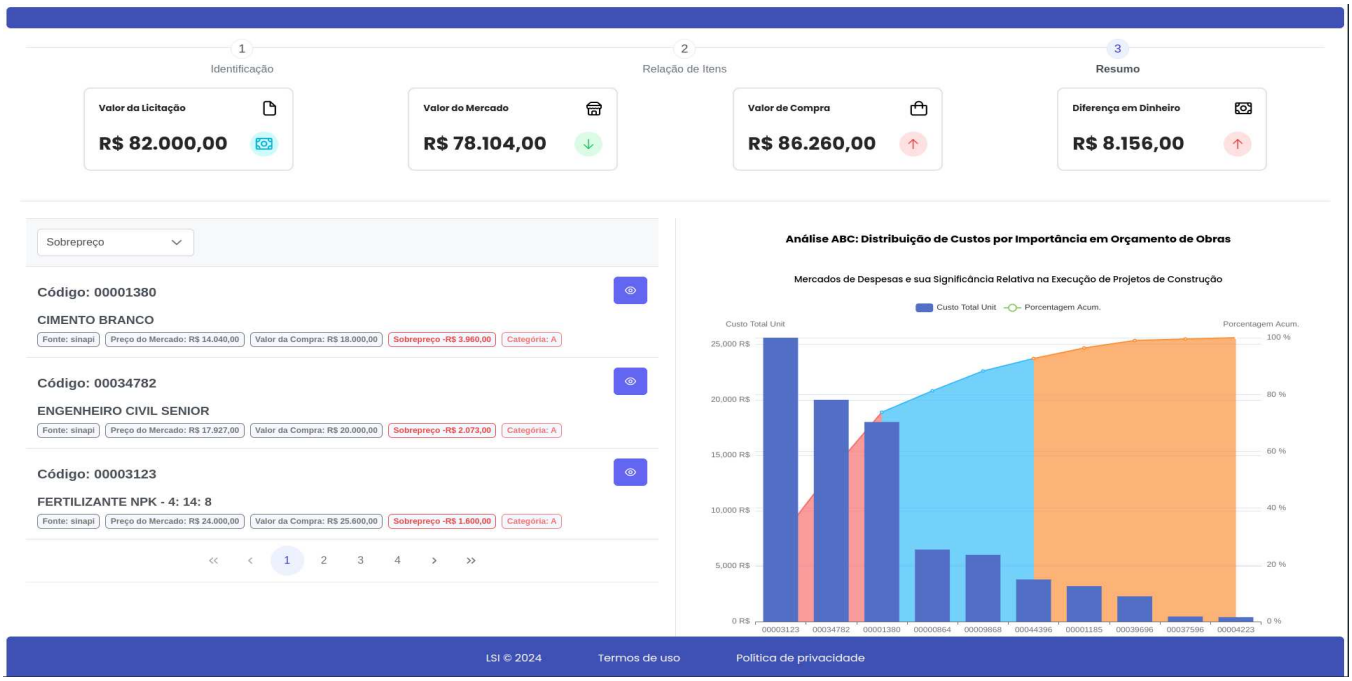


**Figura 11: Tela de resumo do relatório.**

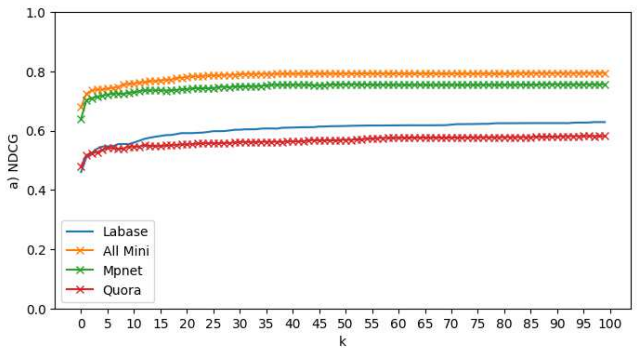


**Figura 12: Avaliação da relevância dos resultados na SINAPI.**

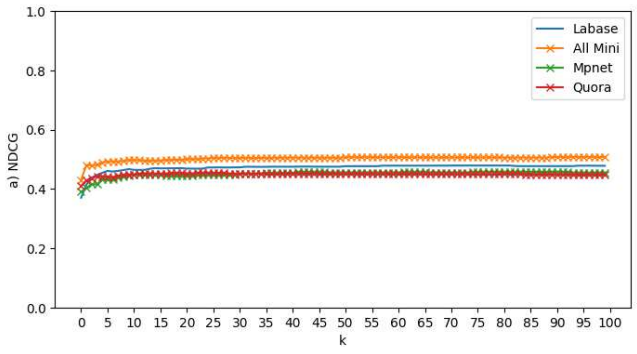


**Figura 13: Avaliação da relevância dos resultados na SICRO.**

## 5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, foi desenvolvida uma aplicação destinada a auxiliar na identificação de possíveis sobrepreços e subpreços em licitações de obras, por meio da correspondência de itens. Para efetuar essa correspondência de forma eficiente, foi realizado um comparativo entre quatro modelos semânticos empregados na busca semântica do Elasticsearch, cujos resultados foram avaliados pela métrica NDCG.

A ferramenta também conta com uma base de dados montada por meio de dois scrapers especializados, responsáveis pela extração de informações sobre materiais e serviços de obras dos sites da SICRO e SINAPI. Além disso, destaca-se a importância da estruturação dos dados para agrupar itens semelhantes, mantendo um histórico de preços simplificado para consultas.

A busca semântica desempenhou um papel significativo ao permitir que as consultas considerassem a similaridade semântica entre os itens, possibilitando uma análise abrangente e precisa das informações. Adicionalmente, a métrica NDCG proporcionou uma avaliação objetiva e comparativa da eficácia dos modelos semânticos utilizados, fornecendo *insights* valiosos para avaliar a busca e a correspondência de itens nas licitações, sendo identificado o modelo All Mini como o mais eficaz.

Como trabalhos futuros, pretende-se melhorar o processo de correspondência, utilizando estratégias de pós-filtragem baseadas nas características dos materiais e serviços, e otimizando a busca semântica apenas para palavras que possam inferir diferentes contextos. Além disso, planeja-se integrar essa ferramenta com sistemas de tribunais de contas, automatizando a identificação de sobrepreços e subpreços no momento do cadastro das licitações de obras.

## AGRADECIMENTOS

Gostaria de expressar minha gratidão a diversas pessoas que desempenharam papéis fundamentais em minha jornada acadêmica e profissional. Ao Professor Cláudio de Souza Baptista PhD, meu orientador, dedico sinceros agradecimentos por sua orientação, ensinamentos e confiança, os quais foram essenciais para o meu crescimento pessoal e profissional. Também gostaria de agradecer ao Professor Doutor Hugo Feitosa de Figueirêdo que me incentivou e apoiou durante meus estudos no IFPB e se tornou um espelho do profissionalismo que desejo alcançar um dia. Ao Professor André Luiz F. Alves, que contribuiu de forma significativa no desenvolvimento deste trabalho. Agradeço também ao corpo docente do curso de Ciência da Computação da Universidade Federal de Campina Grande (UFCG), assim como a todos os funcionários da instituição, por proporcionarem uma estrutura acadêmica de excelência. Não posso deixar de mencionar meus amigos, especialmente aqueles do Laboratório de Sistemas de Informação, cuja convivência e compartilhamento de conhecimentos foram essenciais para meu crescimento ao longo dos anos. Agradeço profundamente aos meus pais e ao meu irmão pelo amor incondicional, apoio constante e ensinamentos valiosos que moldaram quem sou hoje e garantiram minha boa educação. Por fim, expresso minha gratidão à minha namorada, cujo incentivo, compreensão e apoio foram fundamentais para enfrentar os desafios da jornada acadêmica.

## REFERÊNCIAS

[1] Gustavo Almeida, Kate Revoredo, Claudia Cappelli, and Cristiano Maciel. 2018. Improvement of transparency through mining techniques for reclassification of texts: the case of brazilian transparency portal. In *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*. 1–9.

[2] Arindam Bhattacharya, Ankit Gandhi, Vijay Huddar, Ankith MS, Aayush Moroney, Atul Saroop, and Rahul Bhagat. 2023. Beyond hard negatives in product search: Semantic matching using one-class classification (smocc). In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1012–1020.

[3] Camila S Braz, Bárbara MA Mendes, Gabriel P Oliveira, Lucas L Costa, Mariana O Silva, Michele A Brandão, Anisio Lacerda, and Gisele L Pappa. 2023. Análise de irregularidades em licitaçoes públicas com foco em empresas de pequeno porte. In *Anais do XI Workshop de Computação Aplicada em Governo Eletrônico*. SBC, 94–105.

[4] Haoming Chen, Yetian Chen, Jingjing Meng, Yang Jiao, Yikai Ni, Yan Gao, Michinari Momma, and Yi Sun. 2023. Improving product search with season-aware query-product semantic similarity. In *Companion Proceedings of the ACM Web Conference 2023*. 864–868.

[5] Marco Aurelio OS Correa and Adriano Galindo Leal. 2018. Identification of overpricing in the purchase of medication by the federal government of brazil, using text mining and clustering based on ontology. In *Proceedings of the 2018 2nd International Conference on Cloud and Big Data Computing*. 66–70.

[6] Lucas L Costa, Arthur PG Reis, Clara A Bacha, Gabriel P Oliveira, Mariana O Silva, Matheus C Teixeira, Michele A Brandao, Anisio Lacerda, and Gisele L Pappa. 2022. Alertas de fraude em licitações: Uma abordagem baseada em redes sociais. In *Anais do XI Brazilian Workshop on Social Network Analysis and Mining*. SBC, 37–48.

[7] Praveen M Dhulavvagol, Vijayakumar H Bhajantri, and SG Totad. 2020. Performance analysis of distributed processing system using shard selection techniques on elasticsearch. *Procedia Computer Science* 167 (2020), 1626–1635.

[8] Larissa D Gomide, Guilherme Bezerra dos Santos, Lucas L Costa, Michele A Brandão, Anisio Lacerda, and Gisele L Pappa. 2023. Mineração de Dados sobre Despesas Públicas de Municípios Mineiros para Gerar Alertas de Fraudes. In *Anais do XXXVIII Simpósio Brasileiro de Bancos de Dados*. SBC, 378–383.

[9] Swapna Gottipati, David Lo, and Jing Jiang. 2011. Finding relevant answers in software forums. In *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*. IEEE, 323–332.

[10] Donna Harman. 2011. *Information retrieval evaluation*. Morgan & Claypool Publishers.

[11] Aashiq Muhamed, Sriram Srinivasan, Choon-Hui Teo, Qingjun Cui, Belinda Zeng, Trishul Chilimbi, and SVN Vishwanathan. 2023. Web-scale semantic product search with large language models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 73–85.

[12] Gabriel P Oliveira, Bárbara MA Mendes, Camila S Braz, Lucas L Costa, Mariana O Silva, Michele A Brandão, Anisio Lacerda, and Gisele L Pappa. 2023. Ranqueamento de Licitações Públicas a partir de Alertas de Fraude. In *Anais do XII Brazilian Workshop on Social Network Analysis and Mining*. SBC, 1–12.

[13] LCLPC Ribeiro. 2013. Obras Públicas-alguns aspectos: da licitação à auditoria. *Especialize–revista online. Instituto de Pós-Graduação–IPOG. Disponível em: http://www. uniaodaserrageral. mg. gov. br/wp-content/pdf/Edital_23042014. pdf. Acesso em* 30, 05 (2013), 2021.

[14] Mariana O Silva, Lucas L Costa, Guilherme Bezerra, Larissa D Gomide, Henrique R Hott, Gabriel P Oliveira, Michele A Brandao, Anisio Lacerda, and Gisele Pappa. 2023. Análise de sobrepreço em itens de licitações públicas. In *Anais do XI Workshop de Computação Aplicada em Governo Eletrônico*. SBC, 118–129.

[15] Ramon Vaqueiro, Ana Vargas, Tatiana Escovedo, and Marcos Kalinowski. 2023. Machine Learning Applied to Open Government Data for the Detection of Improprieties in the Application of Public Resources. In *Proceedings of the XIX Brazilian Symposium on Information Systems*. 213–220.