UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

KEVYN EDUARDO CARVALHO DE SOUZA
MARCOS VINICIUS FRANCISCO DA SILVA

Héracles: Um Sistema de Monitoramento de Termos no Twitter

Prof. Filipe Braida do Carmo, D.Sc.
Orientador

Nova Iguaçu, Julho de 2024

Héracles: Um Sistema de Monitoramento de Termos no Twitter

Kevyn Eduardo Carvalho de Souza

Marcos Vinicius Francisco da Silva

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Apresentado por:

Kevyn Eduardo Carvalho de Souza

Marcos Vinicius Francisco da Silva

Aprovado por:

Prof. Filipe Braida do Carmo, D.Sc.

Prof. Leandro Guimarães Marques Alvim, D.Sc.

Prof. Bruno José Dembogurski, D.Sc.

NOVA IGUAÇU, RJ - BRASIL

Julho de 2024

DOCUMENTOS COMPROBATÓRIOS Nº 11391/2024 - CoordCGCC (12.28.01.00.00.98)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 22/07/2024 13:36)
BRUNO JOSE DEMBOGURSKI
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###249#4

(Assinado digitalmente em 22/07/2024 17:14)
FILIPE BRAIDA DO CARMO
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###295#4

(Assinado digitalmente em 22/07/2024 16:26)
LEANDRO GUIMARAES MARQUES ALVIM
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###008#2

(Assinado digitalmente em 22/07/2024 13:42)
KEVYN EDUARDO CARVALHO DE SOUZA
DISCENTE
Matrícula: 2020######7

(Assinado digitalmente em 22/07/2024 18:55)
MARCOS VINICIUS FRANCISCO DA SILVA
DISCENTE
Matrícula: 2020######6

*Visualize o documento original em https://sipac.ufrrj.br/documentos/ informando seu número: **11391**, ano: **2024**, tipo: **DOCUMENTOS COMPROBATÓRIOS**, data de emissão: **22/07/2024** e o código de verificação: **5caa5d64c0***

Agradecimentos

Kevyn Eduardo Carvalho de Souza

Primeiramente, agradeço a Deus, por estar sempre comigo, me dar forças, bençãos, direção e sabedoria ao longo de toda a minha jornada acadêmica e pessoal. Agradeço também aos meus professores, que compartilharam comigo o conhecimento necessário para a realização deste projeto. Cada lição e orientação foi fundamental para o meu desenvolvimento e crescimento intelectual.

À minha família, que me proporcionou todo o suporte e condições para que eu pudesse focar plenamente neste trabalho. Em especial, minha gratidão à minha tia Valdete de Carvalho Cavalcante e ao meu tio Jorge Alves de Barros, cujo apoio e incentivo foram inestimáveis. À minha namorada, que me acompanhou e esteve comigo durante todo o processo, me ajudando e incentivando constantemente. Seu apoio foi essencial para me motivar a superar os desafios e seguir em frente.

Agradeço profundamente ao meu orientador, Filipe Braida do Carmo, por todas as orientações, ensinamentos, paciência e confiança depositada. Sua liderança e suporte foram essenciais para a concretização deste projeto. Um agradecimento especial ao meu colega de TCC e de vida, Marcos Vinicius Francisco da Silva, por ter me ajudado e dividido este desafio comigo. Sua parceria e amizade tornaram este percurso muito mais leve e enriquecedor.

E, por fim, agradeço a mim mesmo, pelo esforço, estudo, dedicação e perseverança neste trabalho de projeto final. Este foi um caminho de muito aprendizado e superação, e estou orgulhoso de cada passo dado para a conclusão desta etapa.

Marcos Vinicius Francisco da Silva

Agradeço a Deus por me proporcionar uma trajetória que, apesar das dificuldades e desafios, me ajudou a ser uma pessoa melhor. Sou grato por todo o direcionamento e foco que Ele me deu e reconheço que, sem Ele, eu não seria nada.

Dedico este trabalho à minha família, que me apoiou ao longo de toda minha trajetória e nunca me deixou desamparado. Especialmente à minha avó, Severina dos Santos Francisco, que, apesar de não ser familiarizada com a tecnologia, sempre esteve ao meu lado, demonstrando apoio com palavras de incentivo ou com um café para sustentar as longas noites de estudo. Dedico também a minha mãe Joseane dos Santos Francisco, que com todo seu esforço, suor e trabalho fez com que seu primeiro filho fosse a primeira pessoa da família a ingressar em uma universidade. Agradeço minha namorada que esteve comigo nos dias em que precisei de apoio, me motivando com palavras para me lembrar de onde eu vim e aonde eu quero chegar.

Meus agradecimentos ao meu orientador, Filipe Braida do Carmo, que, com suas palavras de incentivo e motivação, além das valiosas orientações e ensinamentos, foi responsável pelo meu desenvolvimento como aluno e como ser humano. Sua dedicação e apoio foram primordiais para a conclusão deste projeto. Agradeço ao meu confidente e grande amigo pessoal Kevyn Eduardo Carvalho de Souza, por ter se dedicado como pesquisador e como amigo. Agradeço também, ao meu amigo Gabriel Domiciano Lacerda, me estar apoiando sempre que preciso.

Enfim, agradeço a mim mesmo por ter alcançado um grande objetivo de vida ao concluir este projeto final. Essa jornada de desenvolvimento, tanto como aluno quanto como pessoa, trouxe inúmeros ensinamentos e aprendizados. Tenho orgulho do que me tornei e aguardo ansiosamente pelo que ainda está por vir. Me sinto realizado e pronto para enfrentar qualquer desafio.

RESUMO

Héracles: Um Sistema de Monitoramento de Termos no Twitter

Kevyn Eduardo Carvalho de Souza e Marcos Vinicius Francisco da Silva

Julho/2024

Orientador: Filipe Braida do Carmo, D.Sc.

O aumento do uso de substâncias para melhorar o desempenho esportivo evidencia problemas no esporte, exacerbados pelas redes sociais que promovem práticas ilícitas. Nesse contexto, o *Twitter* representa uma valiosa fonte de dados para a análise de percepções e comportamentos relacionados ao *doping*. Este trabalho apresenta o desenvolvimento de um sistema web para monitoramento de termos relacionados ao *doping* no *Twitter*, utilizando métodos de índices invertidos para aprimorar a eficiência de busca. O objetivo principal é criar uma ferramenta que facilite a identificação e análise de discursos e menções a substâncias proibidas nos esportes, auxiliando pesquisadores, profissionais da área esportiva e responsáveis por políticas públicas a monitorar e compreender melhor o uso e a percepção do *doping* no esporte e na sociedade. Os resultados mostram que a utilização de técnicas como cache e busca de texto completo melhora significativamente o desempenho do sistema em comparação com outras soluções da literatura. Dessa maneira, os experimentos realizados confirmam a eficácia do sistema na coleta e análise de dados, contribuindo para uma melhor compreensão e controle do *doping* no esporte.

ABSTRACT

Héracles: Um Sistema de Monitoramento de Termos no Twitter

Kevyn Eduardo Carvalho de Souza and Marcos Vinicius Francisco da Silva

Julho/2024

Advisor: Filipe Braida do Carmo, D.Sc.

The increased use of substances to enhance athletic performance highlights issues in sports, exacerbated by social media that promote illicit practices. In this context, Twitter represents a valuable data source for analyzing perceptions and behaviors related to doping. This work presents the development of a web system for monitoring terms related to doping on Twitter, utilizing inverted index methods to improve search efficiency. The main objective is to create a tool that facilitates the identification and analysis of discussions and mentions of prohibited substances in sports, assisting researchers, sports professionals, and policymakers in better monitoring and understanding the use and perception of doping in sports and society. The results show that using techniques such as caching and full-text search significantly improves the system's performance compared to other solutions in the literature. Thus, the experiments conducted confirm the system's effectiveness in data collection and analysis, contributing to a better understanding and control of doping in sports.

Lista de Figuras

Lista de Tabelas

Lista de Códigos

Lista de Abreviaturas e Siglas

COI	Comitê Olímpico Internacional
WADA	World Anti-Doping Agency
API	Interface de Programação de Aplicação
APIs	Interfaces de Programação de Aplicação
ER	Entidade Relacionamento
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
NoSQL	Bancos de dados Não-Relacionais
PNL	Processamento de Linguagem Natural
NPM	Node Package Manager
MVC	Model-View-Controller
ORM	Object-Relational Mapping
CRUD	Create, Read, Update, Delete
IDs	Identificadores Únicos
TF	Frequências de Termos
TF-IDF	Frequência Inversa de Termos em Documentos Frequentes
GiN	Generalized Inverted Index
GiST	Generalized Search Tree
IA	Inteligência Artificial

Sumário

Capítulo 1

Introdução

A crescente prevalência no uso de substâncias para o aumento de performance no esporte reflete não apenas a busca incessante por desempenho e vantagem competitiva, mas também revela aspectos mais profundos e problemáticos do mundo esportivo atual (MASALA et al., 2019). Segundo Borges et al. (2021), a busca por melhores condições físicas, melhor desempenho em atividades esportivas de alto rendimento, sejam elas profissionais ou amadoras, além da estética corporal vem crescendo ao longo dos anos.

De acordo com Barbosa, Matos e Costa (2011), desde os tempos da Grécia Antiga existe a concepção de "corpo ideal", caracterizado pela beleza estética e excelente condição física. Assim, é indiscutível que a procura por um corpo a qual atenda aos padrões de beleza impostos pela sociedade não é um problema recente. Contudo, tal fenômeno é intensificado e influenciado pelas redes sociais, que se tornaram um fator crucial na formação da opinião pública e na difusão de padrões de beleza e sucesso num todo (STEFANO, 2017).

Dessa forma, as plataformas de mídia social possuem capacidade de alcançar um público amplo e diversificado, causando um impacto significativo na maneira como atletas profissionais e amadores percebem a necessidade de melhorar suas condições físicas e desempenho esportivo. As narrativas de sucesso esportivo e a constante exposição a imagens de corpos esteticamente idealizados podem criar uma pressão

implícita para atingir esses padrões, muitas vezes inatingíveis.

Assim sendo, as redes sociais têm se tornado plataformas cruciais onde a glória e os lucros associados ao sucesso esportivo são promovidos e exaltados extensivamente. Esta exposição tem ampliado a competição para além das arenas esportivas. A busca pela visibilidade e reconhecimento nas mídias sociais muitas vezes se torna algo lucrativo e importante na carreira, incentiva atletas a buscar métodos rápidos para alcançarem sucesso e prestígio rapidamente.

Paralelamente, observa-se nas últimas décadas um aumento significativo nos lucros gerados por competições esportivas, acarretando em uma rivalidade não apenas atlética, mas também de natureza política e econômica (CARTIGNY et al., 2020). Esta realidade, aliada ao desejo de alcançar e sustentar o sucesso esportivo e padrões estéticos, tem levado muitos atletas a recorrerem a métodos ilícitos, muitas vezes com consequências severas para sua saúde e carreira.

Embora as redes sociais apresentem desvantagens, elas também desempenham um papel crucial na coleta de dados e informações relevantes. O uso intenso dessas plataformas na vida cotidiana da população as estabelece como um repositório abrangente de pensamentos, rotinas e ações dos indivíduos. Essa grande gama de dados fornece um campo fértil para pesquisa, análise e tomada de decisões.

Concomitante a isso, a rede social *Twitter*[1] se tornou um dos principais meios de interação online existentes. Dados de início de 2022 mostram que há cerca de 19,05 milhões de usuários ativos no Brasil. Considerando que tal meio de comunicação impõe uma restrição de idade mínima de 13 anos para o uso da sua plataforma, é pertinente observar que em relação em público "elegível", 10,8% da população brasileira está ativa na rede. (KEMP, 2022)

Assim sendo, o *Twitter*, com sua vasta base de usuários, principalmente o público jovem e conteúdo dinâmico, apresenta-se como uma fonte rica e atualizada de informações e dados sobre este tópico. Neste contexto, a análise de postagens, *hashtags* e discussões na plataforma permitiu a coleta de dados qualitativos e quantitati-

[1] <http://www.twitter.com>

vos, facilitando um entendimento mais profundo das percepções e comportamentos relacionados ao *doping*.

Nota: Este estudo foi conduzido durante o período em que a plataforma de mídia social referida era conhecida como "Twitter". Subsequentemente, a plataforma passou por um processo de *rebranding* e atualmente é denominada "X". Esta mudança de nome reflete uma atualização na identidade da marca, mas as funcionalidades e a relevância da plataforma no contexto do presente trabalho permanecem inalteradas.

1.1 Objetivo

O presente estudo tem como objetivo desenvolver um sistema web para monitoramento de termos. Este sistema será projetado para demonstrar sua capacidade e potencial, especialmente no contexto de termos relacionados ao doping. A principal função será focada em identificar e analisar menções ao doping no *Twitter*, uma das plataformas de mídia social mais amplamente utilizadas.

A proposta é criar uma ferramenta que facilite a identificação e análise de discursos e menções ao doping em plataformas de mídia social. O objetivo principal é fornecer uma solução tecnológica que auxilie pesquisadores, profissionais da área esportiva e responsáveis de políticas públicas a monitorar e compreender melhor o uso e a percepção do doping no esporte e na sociedade.

Além disso, será feito um estudo para viabilizar o uso dos dados com mecanismos de otimização de consulta como índices invertidos, visando aprimorar o desempenho do sistema e tornar as buscas mais rápidas e precisas, mesmo diante do grande volume de dados gerado diariamente nas redes sociais.

1.2 Organização do Trabalho

Esta monografia é composta por cinco capítulos, consistidos em uma introdução ao tema, seguida de fundamentação teórica, o sistema proposto em si, experimentos ou resultados experimentais e finalmente, a conclusão. A seguir, é apresentado a

descrição de forma mais abrangente cada capítulo:

- Capítulo 1: Este capítulo aborda a problemática do uso de substâncias para o aumento de performance no esporte, destacando a relevância do estudo do doping no contexto esportivo atual. Explora também o impacto das redes sociais na opinião pública e nos padrões de beleza e sucesso. Além disso, apresenta o objetivo do estudo: desenvolver um sistema web de monitoramento de termos relacionados ao doping utilizando métodos de índices invertidos.

- Capítulo 2: Fornece uma base teórica, a qual explora os princípios e fundamentos das tecnologias cruciais para o projeto, como busca e seus componentes, incluindo *Web Crawler*, índice invertido em busca de texto completa e mineração de texto. Este capítulo estabelece a importância da busca eficiente e precisa em grandes volumes de dados não estruturados e como essas tecnologias facilitam a tomada de decisões informadas.

- Capítulo 3: Apresenta a proposta do sistema capaz de monitorar informações sobre doping na rede social *Twitter*. Inclui a motivação para o desenvolvimento da ferramenta, trabalhos relacionados, modelagem de dados, e a descrição detalhada das funcionalidades do sistema, como a busca por termos, cadastramento de termos e categorias, e o uso de sinônimos para melhorar a precisão da captura de dados.

- Capítulo 4: Detalha a solução desenvolvida, as ferramentas selecionadas para a execução do projeto, e os critérios técnicos que orientaram essas escolhas. Apresenta os experimentos realizados para validar a eficácia do sistema, as tecnologias utilizadas, e discute as contribuições do projeto, as melhorias implementadas e as limitações encontradas.

- Capítulo 5: Resume os principais pontos discutidos na monografia, apresentando as considerações finais sobre o trabalho realizado, as limitações do estudo e as possíveis extensões para trabalhos futuros. Este capítulo busca contextualizar a relevância do trabalho dentro do campo de pesquisa e sugere direções para aprimoramento e expansão do sistema proposto.

Capítulo 2

Fundamentação

Nos últimos anos, a explosão de dados digitais gerados por indivíduos e organizações evidenciou a urgente necessidade de desenvolver tecnologias avançadas de busca e recuperação de informações. O crescimento exponencial do volume de informações ao longo dos anos tornou essencial a criação de métodos eficazes para a localização rápida e precisa de informações em amplos repositórios de texto. Assim, a busca é apresentada como um processo fundamental para navegar e extrair valor desses vastos e complexos conjuntos de dados.

Este capítulo fornece uma base teórica essencial para a proposta e dedica-se a explorar os princípios e fundamentos de algumas tecnologias cruciais neste contexto: Mineração de Dados e seus componentes, incluindo *Web Crawler*, Índice Invertido em busca de texto completa e Mineração de Texto. Desse modo, a compreensão dessas tecnologias é crucial para o entendimento deste projeto e para entender a obtenção eficaz de informações a partir de grandes volumes de dados não estruturados, o que facilita tomadas de decisões.

2.1 Mineração de Dados

A compreensão do conceito de Mineração de Dados é essencial para o entendimento deste estudo. Dessa forma, em Han, Kamber e Pei (2011), este conceito é definido

como o processo de analisar e extrair informações significativas dos dados armazenados em grandes bases de dados. Este processo envolve o uso de métodos provenientes de várias áreas, como aprendizado de máquina, estatística e sistemas de bancos de dados.

A importância da Mineração de Dados reside na sua capacidade de transformar grandes quantidades de dados em informações úteis, permitindo que pessoas e organizações melhorem a tomada de decisões, identifiquem oportunidades de negócios, prevejam comportamentos sociais, detectem problemas a serem solucionados e otimizem monitoramentos. Desta forma, a Mineração de Dados se estabelece como uma ferramenta essencial na era da informação, proporcionando *insights* valiosos a partir de dados massivos.

A partir da exploração inicial do conceito de Mineração de Dados, uma aplicação específica da mesma é a Busca de Texto Completo, que se destaca pela sua capacidade de localizar sequências de texto em grandes volumes de registros, ao utilizar técnicas de análise para transformar dados brutos em informações úteis. Desse modo, Busca refere-se ao processo de localização de informações específicas e relevantes em um repositório de informações, geralmente em resposta a uma consulta do usuário (MANNING; RAGHAVAN; SCHÜTZE, 2008).

Este processo envolve a varredura de grandes volumes de dados textuais para encontrar termos ou frases que correspondam à consulta realizada, utilizando algoritmos eficientes para garantir rapidez e precisão na recuperação das informações. Esta técnica é fundamental em vários domínios, incluindo motores de busca na internet, sistemas de gestão documental e bases de dados acadêmicas onde a rapidez e precisão na recuperação de informações são críticas.

Dessa forma, a Mineração de Dados se concentra em extrair informações significativas de grandes volumes de dados e a Busca de Texto Completo é uma ferramenta que auxilia nesse processo ao permitir que usuários localizem rapidamente as informações textuais necessárias. Por exemplo, um sistema de Mineração de Dados pode usar Busca de Texto Completo para identificar padrões em documentos textuais que indicam tendências de comportamentos de consumidores de drogas ilícitas.

Contudo, para que a Mineração de Dados seja eficaz deve-se entender a ordem dos fatores e as dependências entre técnicas como índice invertido, *Web Crawler* e mineração de texto. Essas tecnologias são fundamentais para coletar, organizar e analisar dados e desempenham papéis vitais em aplicações como mecanismos de busca, comportamento em mídias sociais e análise de dados em larga escala. A seguir, há o detalhamento de cada uma dessas etapas e suas inter-relações.

2.2 Web Crawler

O *Web Crawler* é geralmente o ponto de partida no processo de captura de dados na web. Segundo Zia et al. (2022), *Web Crawler* é definido como um processo de coleta de informações de um website, extração de hiperlinks incluídos e o acompanhamento desses links. É descrito como um método usado principalmente por grandes motores de busca como Google[1] e Bing[2]. Assim sendo, o propósito é coletar informações gerais, salvando-as em uma base de dados ou indexando-as para permitir buscas.

Embora o foco da pesquisa seja no *Web Crawler*, no contexto da coleta de dados na web, é fundamental distinguir entre *web crawling* e *Web Scraping*. Como descrito anteriormente, enquanto o *Web Crawler* é geralmente mais abrangente e menos específico em termos de coleta de dados, pois visa indexar a web de forma extensa, o *Web Scraping* opera em uma escala mais específica, focando em dados detalhados de páginas específicas.

Esta abordagem mais específica é empregada para coletar dados detalhados e específicos de interesse, como preços de produtos, horários de voos e preços de passagens aéreas. O *Scraping* é frequentemente usado para extrair conjuntos de dados detalhados para análises subsequentes ou para alimentar aplicações específicas, sendo uma ferramenta valiosa para empresas e pesquisadores que necessitam de informações precisas para análises detalhadas ou marketing direcionado.

Desta maneira, as principais diferenças entre *Web Crawling* e *Web Scraping*

[1] <http://www.google.com>
[2] <http://www.bing.com>

residem no escopo e na profundidade da coleta de dados. O primeiro possui um processo mais abrangente focado na coleta e organização de grandes quantidades de dados enquanto o segundo é mais direcionado e detalhado, empregando identificadores específicos como estruturas HTML para localizar e extrair dados precisos de interesse.

A diferença entre o funcionamento dessas duas tecnologias são apresentadas visualmente na Figura 2.1. Ao lado esquerdo temos o Web Crawler. O *crawler* visita sites para coletar dados e à medida que visita os mesmos, constrói uma lista de páginas visitadas para referência futura. As informações coletadas são então indexadas, ou seja, organizadas de forma a facilitar a recuperação rápida, e finalmente, os dados são armazenados em um banco de dados para uso futuro.

Desse mesmo modo, ao lado direito da Figura 2.1, há o funcionamento do *Web Scraping*. Similar ao *Crawling*, o *Scraping* começa com a visita a um *website* específico. O processo principal é extrair dados específicos da página, usando técnicas que podem identificar precisamente os dados necessários. Assim, os dados extraídos podem ser salvos em vários formatos, como XML, Excel, JSON, e PDF, dependendo das necessidades da tarefa de *Scraping*.

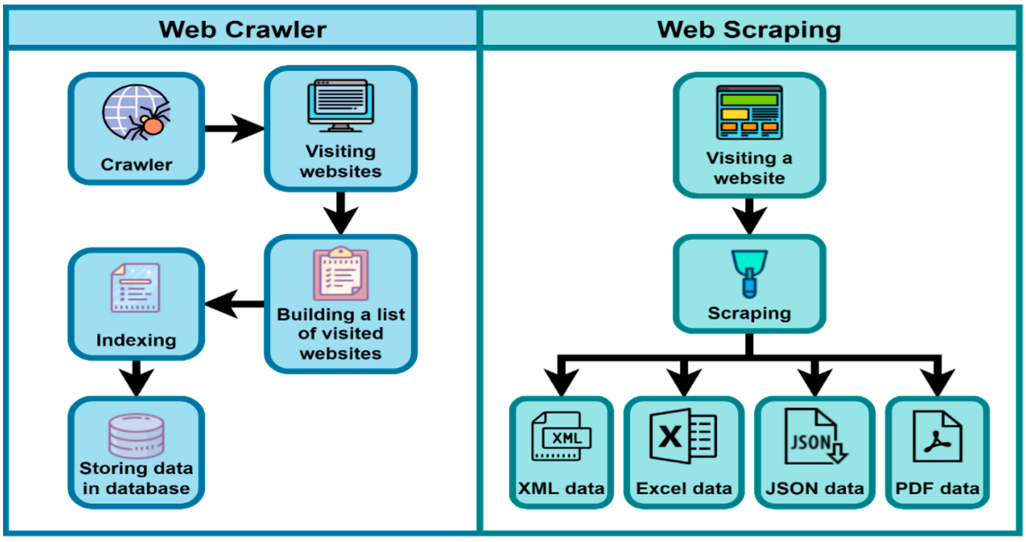

Figura 2.1: Comparação entre *Web Crawling* and e *Scraping*. Fonte: (ZIA et al., 2022)

Essa diferenciação é crucial para compreender como essas tecnologias são adap-

tadas para satisfazer necessidades distintas dentro da coleta de dados na internet. Ao compreender as diferenças fundamentais entre essas duas ferramentas, é possível apresentar um recurso essencial para melhorar a eficiência na recuperação de dados: o índice invertido.

Este organiza os dados coletados e indexados de uma forma que permite consultas rápidas e eficazes. Assim, a integração entre as técnicas de *Crawling* com o método de índice invertido pode otimizar a gestão de dados e facilitar a análise e recuperação de informações em grandes volumes de dados na internet.

2.3 Índice Invertido

Após a coleta de dados pelo *Web Crawler*, entra em cena o índice invertido. Este é uma estrutura de dados essencial usada principalmente por mecanismos de busca para permitir buscas rápidas. O índice invertido organiza as informações coletadas pelo *Crawler* indexando o conteúdo de cada página de modo que cada palavra ou termo importante. Isso facilita a recuperação rápida de informações, que é crucial para a eficiência dos mecanismos de busca.

Os conceitos e práticas presentes neste trabalho sobre a Busca de Texto Completo e a utilização de índices invertidos como estruturas otimizadoras, foram baseadas no livro "Expert Performance Indexing in SQL Server" por Jason Strate (STRATE, 2019). Essa obra oferece uma base teórica e prática sobre como as técnicas de indexação e recuperação de texto são essenciais para o desenvolvimento de sistemas de busca eficientes e precisos.

Dessa forma, segundo Strate (2019), índice invertido é uma estrutura de dados essencial para otimizar a busca de texto completo. Esta estrutura mapeia o conteúdo de texto para sua localização dentro de um banco de dados, documento ou conjunto de documentos. A ideia é semelhante ao índice no final de um livro que direciona você para as páginas onde um termo específico aparece.

Na prática, tal funcionalidade associa cada termo a uma lista de postagem, ou *posting list*, que contém os Identificadores Únicos (IDs) dos documentos associados a

esse termo específico. Durante uma consulta, o sistema não realiza uma varredura completa, mas acessa diretamente no índice e reduz significativamente o tempo de busca ou de coleta de informações.

Para otimizar o índice invertido, os termos são normalizados por meio de *Stemming* ou lematização, reduzindo-os às suas formas raiz, demonstrado melhor no Capítulo 3. Isso permite que variações do mesmo termo sejam tratadas como equivalentes, melhorando a busca e a indexação. O índice também ignora conectores e pontuações, ampliando sua habilidade de processar termos em ordem invertida e aumentando a flexibilidade do sistema de busca.

Outra característica importante do índice invertido é sua capacidade de ser atualizado incrementalmente. Quando novos documentos são introduzidos ao sistema, apenas as *posting list* relevantes são atualizadas, evitando a necessidade de uma reindexação completa. Essa eficiência torna o índice invertido particularmente adequado para sistemas dinâmicos, onde novos dados são continuamente incorporados.

Em suma, os índices invertidos desempenham um papel crucial na otimização dos processos de busca em sistemas de recuperação de informações e mecanismos de busca. Por conseguinte, facilitam a recuperação rápida, tratam eficientemente as variações linguísticas, proporcionam flexibilidade de consulta e permitem uma ordenação precisa dos resultados por relevância.

2.4 Mineração de Texto

Por fim, a mineração de texto é aplicada aos dados organizados, usando o índice invertido como ponto de partida para extrair conhecimento ou padrões implícitos de dados textuais. Esse processo se distingue da simples recuperação de informações, pois visa descobrir novos conhecimentos que não estão explicitamente armazenados no texto (JO, 2019). A mineração de texto utiliza técnicas de processamento de linguagem natural e aprendizado de máquina para transformar texto em dados estruturados que podem ser analisados.

Embora índices invertidos sejam essenciais para a recuperação rápida de documen-

tos com base em termos de consulta, a mineração de texto é crucial para entender o conteúdo desses documentos a um nível mais profundo. Este método pode ser usado para enriquecer os dados que são indexados, o qual permite buscas mais sofisticadas que vão além da simples correspondência de palavras-chave.

Desse modo, após a indexação, o texto é codificado em vetores numéricos que representam as características textuais de forma estruturada. Assim, ocorre a atribuição de valores às características e atribui valores numéricos às características selecionadas, como Frequências de Termos (TF) ou Frequência Inversa de Termos em Documentos Frequentes (TF-IDF), que pondera a importância de uma palavra no contexto do corpus inteiro (JO, 2019).

O corpus completo refere-se ao conjunto total de documentos ou textos que estão sendo analisados em um processo de mineração de texto. Sendo assim, a TF é uma medida básica utilizada na mineração de texto para quantificar a ocorrência de uma palavra em um documento enquanto a TF-IDF combina essa frequência com a frequência inversa dos documentos, uma medida utilizada na mineração de texto e recuperação de informações para ponderar a importância de uma palavra em um corpus de documentos.

Posteriormente, são aplicados algoritmos de aprendizado de máquina aos dados textuais codificados, como agrupamento de texto e associação de texto. Com o texto transformado em dados estruturados e valores atribuídos, técnicas de aprendizado de máquina analisam esses dados. Esse processo inclui a criação de vetores de características, a aplicação de métricas de similaridade e o uso de algoritmos para agrupar textos com base em suas similaridades, conhecido como *clustering*.

Além do agrupamento, a classificação categoriza documentos em classes predefinidas, enquanto a extração de entidades identifica nomes de pessoas, locais e organizações. A descoberta de tópicos revela os principais temas no corpus de documentos. Em contrapartida, a associação de texto utiliza algoritmos para identificar regras entre palavras, avaliando a frequência de coocorrência e outras métricas para determinar a força dessas associações.

A partir de todo esse processo, a mineração de texto consegue realizar uma análise profunda do texto e extrair informações e padrões valiosos. Utilizando índices invertidos, técnicas de processamento de linguagem natural e algoritmos de aprendizado de máquina, o texto é transformado em dados estruturados, permitindo a compreensão detalhada do conteúdo dos documentos.

Portanto, enquanto os índices invertidos são ótimos para localizar rapidamente documentos em grandes conjuntos de dados, a mineração de texto permite uma análise mais profunda desses documentos, contribuindo a extrair significado e *insights* valiosos. A combinação de ambos auxiliam a criar sistemas poderosos que não apenas recuperam rapidamente a informação mas também a entender e a organizar de maneira que seja mais acessível e útil para os usuários.

Em resumo, o índice invertido depende dos dados coletados pelo *Web Crawler* para construir uma estrutura de dados que permita consultas eficientes e a mineração de texto depende tanto dos dados coletados pelo *Web Crawler* quanto do índice invertido para realizar análises eficazes. Esses componentes, embora possam funcionar de forma independente em diferentes contextos, são frequentemente usados juntos para potencializar sistemas de busca e análise de dados na internet.

Capítulo 3

Héracles - Sistema de Monitoramento de Termos no Twitter

Esse capítulo propõe um sistema capaz de monitorar informações sobre *doping* e seus assuntos relacionados, *e.g.*, esportes, na rede social *Twitter*. Através desse sistema, gestores da área de prevenção e educação ao *doping* poderão monitorar termos sobre o assunto e poderão utilizar essa informação para tomadas de decisão. Além disso, serão apresentados a motivação para o desenvolvimento dessa ferramenta e trabalhos relacionados a ele.

3.1 Motivação

Para entender a motivação do sistema é importante entender primeiramente o conceito de *doping*. É definido como *doping* a ocorrência de qualquer violação das onze regras anti*doping* estabelecidas no Código Mundial Antidopagem (AGENCY, 2021). Uma dessas regras é o uso de substâncias ou métodos presentes na lista proibida, capazes de aumentar o desempenho esportivo de um atleta de forma artificial.

A partir do fato de atletas utilizarem esse método e do entendimento desse conceito, o Comitê Olímpico Internacional (COI) definiu três valores como fundamentais para

os esportes olímpicos: excelência, amizade e respeito (COMMITTEE, 2021). Esses princípios servem como exemplo para o esporte em escala global. No caso da excelência, ela promove a ideia de sempre aprender, se capacitar e se desenvolver para alcançar o máximo potencial possível.

Em continuidade, a amizade envolve a colaboração, o afeto, a lealdade e o engajamento de todos os membros da comissão. Por sua vez, o respeito representa o princípio ético que implica em respeitar o adversário, reconhecer suas virtudes, seguir as regras estabelecidas e praticar o "jogo limpo", no qual o atleta não deve recorrer a métodos ilegais para superar seu desempenho em relação aos demais.

Dessa forma, as competições esportivas em geral têm como base o esforço digno e merecido dos atletas para alcançarem os melhores resultados. No entanto, essa não é a realidade frequentemente observada. Conforme Yesalis e Bahrke (2002) argumentam, quando os seres humanos competem uns contra os outros, seja na guerra, nos negócios ou no esporte, os competidores, por definição, buscam obter uma vantagem sobre o adversário.

Dessa maneira, com frequência, recorrem ao uso de drogas e outras substâncias para alcançar essa vantagem. Como medida preventiva, em 1967, o COI condenou a prática do *doping* e apresentou uma lista de substâncias consideradas proibidas, iniciando assim o controle anti*doping* pela World Anti-Doping Agency (WADA) (YESALIS; BAHRKE, 2002).

É amplamente reconhecido que o esporte desempenha um papel fundamental no desenvolvimento da cultura humana, remontando aos tempos antigos, com o surgimento dos Jogos Olímpicos na Grécia. Diante disso, segundo Cartigny et al. (2020), nas últimas décadas observa-se um aumento significativo da influência do esporte sobre a sociedade, impulsionado pela mídia, questões políticas e econômicas, em decorrência do grande investimento e lucro pelas competições esportivas.

Torna-se assim, o indivíduo não apenas em um atleta, como também em uma figura de relevância para a sociedade, expandindo o seu raio de influência, para o cotidiano de pessoas que o acompanham. Essa influência abrangente do esporte é

evidente em áreas como entretenimento, política e estilo de vida, moldando o cenário social de forma significativa e desempenhando um papel crucial na maneira como as pessoas se relacionam com a cultura esportiva.

Assim, diversos jovens ao redor do mundo se inspiram e idolatram personalidades esportivas, tendo como exemplo o estilo de vida dos mesmos. Um exemplo disso, é o fato de que entre os cinco perfis mais seguidos na rede social Instagram[1], três são ou já foram atletas (BELING, 2023). Levando isso em consideração, é notório que os atletas possuem grande influência sobre as redes sociais, cujo na sociedade contemporânea, se encontram como foco das atenções de grande parte da população, principalmente os mais jovens.

Seguindo esse princípio, a importância de projetos educacionais conforme em Masala et al. (2019), é evidenciada pela necessidade de potencializar o conhecimento em saúde entre os jovens sobre os perigos do *doping*. Masala et al. (2019) destaca que a escola é um ambiente apropriado para a prevenção do *doping*, oferecendo um contexto onde a informação pode ser transmitida de maneira eficaz e onde os estudantes podem desenvolver uma compreensão crítica dos efeitos negativos das substâncias dopantes. Através da educação, os jovens são capacitados a tomar decisões informadas e a resistir às pressões de usar substâncias ilegais para melhorar o desempenho esportivo.

Com base nas informações dissertadas sobre a definição e o impacto do *doping*, o sistema desenvolvido foi pensado em atender a demanda de informações sobre *doping* obtidas através da rede social *Twitter*, a fim de monitorar as publicações sobre o tema, e analisar os dados. Tendo isso em mente, sabe-se que as redes sociais possuem grande número de publicações diárias onde os usuários publicam sobre seu cotidiano incluindo suas rotinas envolvidas no uso de, por exemplo, anabolizantes, conhecido como o principal meio de adquirir grandes ganhos em um período curto de tempo.

A ideia deste trabalho é elaborar um sistema que tenha a capacidade de monitorar redes sociais, capturar informações sobre publicações através de termos, e permitir usuários do sistema a usar essas informações para tomada de decisão e

[1] <http://www.instagram.com>

desenvolvimento de políticas preventivas do uso de *doping*. Desta maneira, diversos trabalhos existentes se tornam semelhantes e necessários para tal finalidade, estes alguns que serão apresentados na seção a seguir e posteriormente os Requisitos do Sistema.

3.2 Trabalhos Relacionados

Nesta seção, serão apresentados trabalhos, ferramentas, artigos e aplicativos relacionados a aplicação de monitoramento de redes sociais, demonstrando a utilização na qual pode servir de solução em áreas diversas como na reputação de uma empresa ou marca e até na saúde pública. Serão abordadas as limitações e desafios enfrentados pelas pesquisas, além de destacar os principais debates e controvérsias em torno do monitoramento de redes sociais e nas implicações éticas e sociais.

Em diversas competições esportivas amadoras e universitárias, não existem recursos suficientes para o controle de substâncias ilícitas por parte de seus atletas, bem como para a realização de testes anti*doping* regulares. No contexto dos esportes universitários, que representam uma etapa fundamental na formação de atletas profissionais, muitos jovens podem ser influenciados a utilizar substâncias proibidas com o intuito de aprimorar seu desempenho.

Segundo Stavrakantonakis et al. (2012), a monitorização das redes sociais é mais precisa, rápida e econômica em comparação com a análise convencional realizada por um painel de especialistas. Em consonância com esta afirmação, a vigilância de termos nas redes sociais pode representar uma solução para detectar possíveis casos de *doping*, identificar influenciadores que promovem o uso dessas substâncias e, ademais, contribuir para preservar a integridade ética e moral das competições. Com base nesse contexto, serão abordados artigos, ferramentas e aplicações competentes em auxiliar na resolução desse problema.

O Brandwatch[2] é uma ferramenta de monitorização de redes sociais que interpreta e analisa interações online. Após o monitoramento, realiza-se uma análise dos dados

[2] <http://www.brandwatch.com>

capturados e os categoriza com base em palavras-chave. Utilizando inteligência artificial, o algoritmo consegue analisar e detectar os sentimentos dos usuários nas mídias utilizadas diariamente, ou seja, identifica desejos, necessidades e demandas relacionadas a produtos, experiência digital do cliente e outros temas, conforme visto na Figura 3.1.

Dessa forma, além de ser o parceiro oficial do *Twitter*, é amplamente utilizado por grandes empresas. Seus planos são negociados diretamente com vendedores e especialistas, não sendo oferecida a opção de teste gratuito ou demonstração. Assim, é uma ferramenta paga com custos elevados, mais adequada para empresas com grande orçamento.

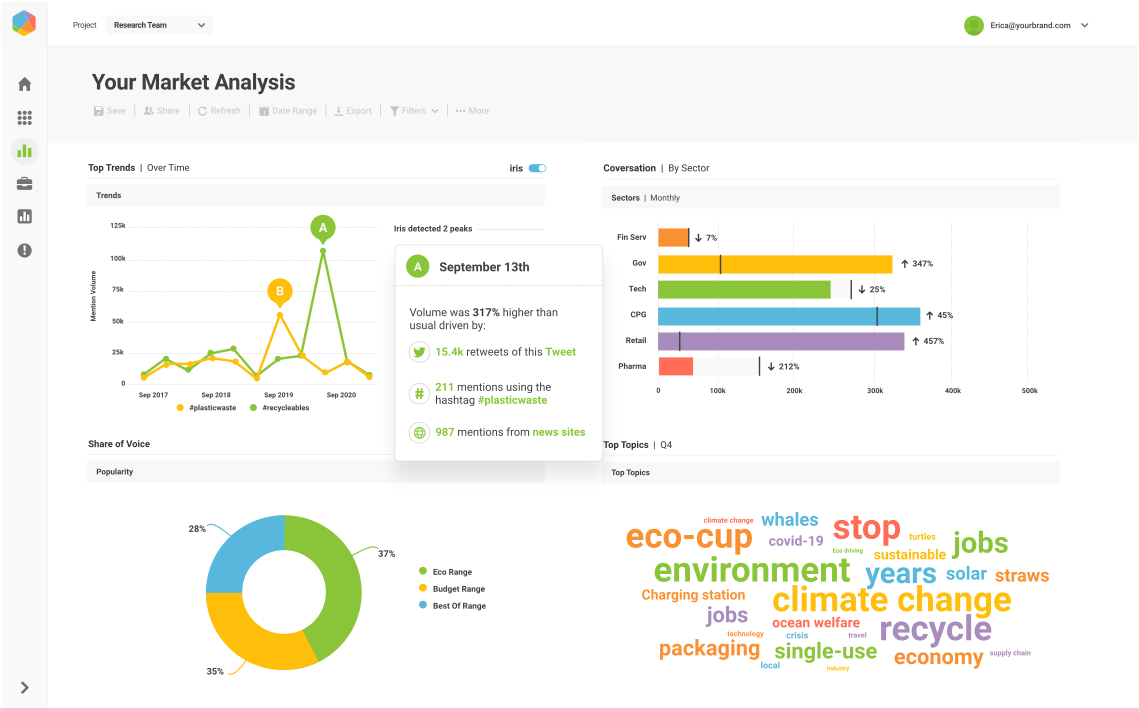

Figura 3.1: Captura de tela disponibilizada pela própria plataforma Brandwatch, a qual contém uma análise de mercado como gráficos mostrando as principais tendências ao longo do tempo e tópicos principais.

No estudo de Srikanth et al. (2021), diferente de alguns métodos de rastreamento, propõe-se a construção de monitores dinâmicos de mídia social capazes de se adaptar às mudanças nas conversas online e prever novas *hashtags* virais relacionadas ao tema em monitoramento. Logo, a ideia é usar uma palavra-chave para capturar o conjunto dos trinta vizinhos mais próximos, definidos pela métrica de similaridade, e,

a partir desse conjunto, selecionar as *hashtags* ou menções mais relevantes para o monitoramento.

Como exemplo, em 2021, Srikanth et al. (2021) conduziram um estudo de caso para acompanhar as conversas dinâmicas relacionadas à posse do presidente Joe Biden, a fim de testar o sistema dinâmico de coleta de dados usando a palavra-chave "*inauguration*". A abordagem dinâmica manteve as antigas palavras-chave frequentemente discutidas pelo monitor estático (#inaugurationday), enquanto identificou as novas virais (#trump, #biden), fornecendo uma cobertura de tópicos mais eficaz do que o monitor estático.

Contudo, requer uma infraestrutura robusta para processamento e armazenamento de dados em tempo real, por utilizar técnicas como *word embeddings* e modelos preditivos a qual exige uma capacidade computacional significativa, o que pode ser um desafio para pequenos projetos ou instituições. Além de que a implementação do sistema é bastante complexa, em que os custos associados ao desenvolvimento, manutenção e operação de um sistema tão complexo são altos.

O Mention[3] é uma plataforma online hábil em monitorar termos, frases, marcas e usuários, permitindo a realização de análises comparativas entre esses dados. O usuário tem a possibilidade de especificar as fontes desejadas em monitorar e receber alertas sempre que o objeto de monitoramento for mencionado. Além disso, pode escolher os idiomas a qual deseja incluir e as menções são atualizadas em tempo real, sendo listadas em um *feed* direto em um painel disponibilizado.

Um recurso interessante é a "Análise Competitiva" oferecida por essa ferramenta. Ela fornece um painel com possibilidade de comparar as análises das menções feitas pelo usuário com as dos seus concorrentes, lado a lado, um exemplo na Figura 3.2. Dessa forma, é possível obter dados de qualquer concorrente adicionado à lista de rastreamento.

A plataforma oferece um teste gratuito de 15 dias com algumas limitações, e após esse período, são disponibilizados planos com preços variados de $41 a $149. Para

[3] <http://www.mention.com>

empresas, é possível negociar valores personalizados.

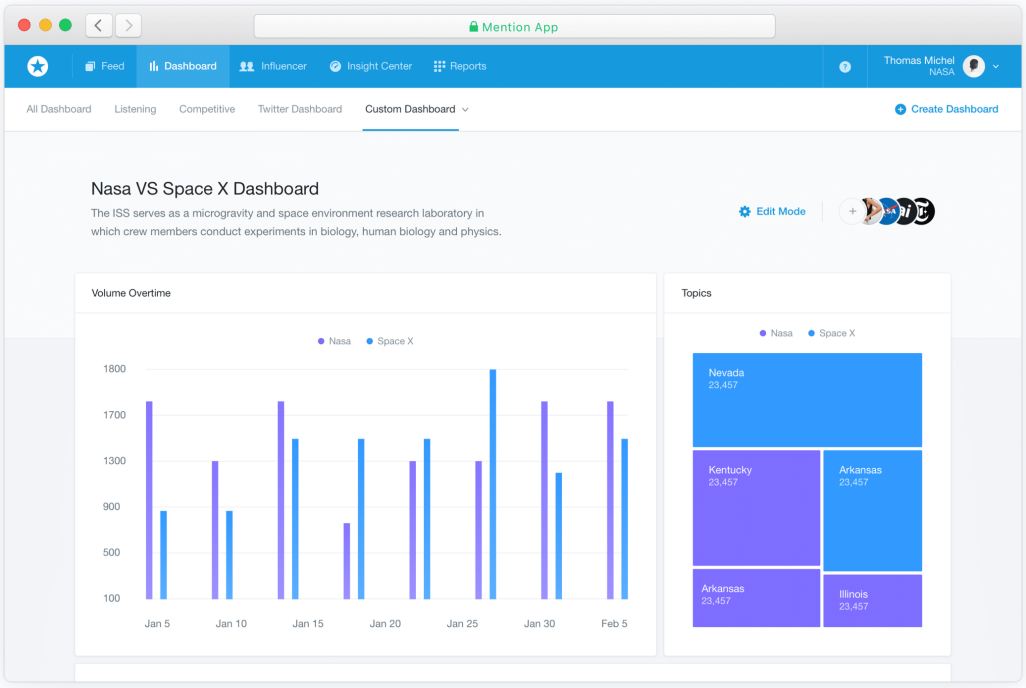

Figura 3.2: Gráfico de Comparação entre os termos "NASA" e "SpaceX" na plataforma do Mention.

Thelwall (2018) conduziu um estudo na Universidade de Wolverhampton, onde é proposto e disponibilizado um software de código aberto chamado "*Mozdeh*". Esse software tem como objetivo a coleta e análise de *tweets* e comentários do YouTube[4], com o potencial de aprimorar significativamente a análise e compreensão das interações nas redes sociais, proporcionando valiosos *insights* para a pesquisa e o monitoramento online.

Os principais recursos dessa ferramenta incluem a capacidade de coletar dados relacionados a palavras-chave ou a um conjunto de usuários, representar graficamente o volume de textos coletados ao longo do tempo ou a porcentagem de textos correspondentes a consultas específicas, utilizar a mineração de associação de palavras para identificar problemas relacionados a determinados tópicos e extrair conexões entre os textos para ilustrar os padrões de comunicação entre os usuários mais interativos.

[4] <https://www.youtube.com>

No estudo de Jordan et al. (2018), é abordado o papel das plataformas digitais como uma fonte relevante de informações relacionadas à saúde, devido à quantidade de dados compartilhados por cidadãos e fontes oficiais. O artigo apresenta uma revisão da literatura não apenas sobre a utilização da mineração de dados no *Twitter*, mas também resume vinte e oito pesquisas que discutem o uso desses dados no contexto da vigilância em saúde pública.

A pesquisa evidencia como a análise de dados digitais possibilita uma resposta ágil por meio de um aprimoramento na vigilância, a qual se mostra fundamental no combate a doenças infecciosas emergentes, como o Ebola e o Zika, mencionados no artigo. Além disso, o estudo ressalta a utilidade dos dados das redes sociais em diversas aplicações na área de saúde pública, tais como o monitoramento de doenças e a análise das reações do público, sejam elas positivas ou negativas.

3.3 Requisitos de Sistema

Os requisitos do sistema consistem em descrever de forma clara e sucinta as funções e comportamentos cujo o sistema deve ter, fornecendo uma compreensão completa do que é necessário e esperado. O processo de levantamento de requisitos se baseou em uma descrição do que era necessário para o sistema.

Tendo em vista as funcionalidades requeridas, foi-se identificados os requisitos, os quais serão descritos na tabela 3.1.

Número	Descrição do Requisito
RF 01	O sistema permitirá que o usuário visualize um *Dashboard*, a qual apresenta *links* para diversas partes internas, além de obter a opção de visualizar o perfil do visitante.
RF 02	O sistema deve apresentar um gráfico para mostrar a recorrência de termos buscados.
RF 03	O sistema deverá conter diferentes categorias para os termos, para facilitar a filtragem do usuário ao buscar o termo desejado.
RF 04	O sistema deve apresentar todos os termos cadastrados, em conjunto com seus respectivos sinônimos e categorias a qual estão inseridos.
RF 05	O sistema permitirá aos administradores adicionar ou excluir termos, assim como as categorias.
RF 06	O sistema deve permitir aos administradores a atribuição de sinônimos aos termos adicionados.
RF 07	O sistema deverá conter diferentes níveis de acesso, sendo o maior dele o nível administrador, no qual gozará de diferentes possibilidades, tais como, cadastrar termos, cadastrar usuários, elevar nível de usuário, remover ou bloquear acesso dos usuários, configurar o agendador.

Tabela 3.1: Requisitos do Sistema

3.4 Proposta

A partir da conjuntura apresentada, a coleta de dados e materiais assume um papel de extrema relevância com o propósito de embasar as decisões relacionadas ao controle do uso de substâncias que podem resultar em casos de *doping*. Essa abordagem contribuirá para a formulação de estratégias eficazes voltadas para a prevenção e o combate ao *doping*, tornando-se, assim, um pilar fundamental em nosso projeto acadêmico.

É importante ressaltar que este trabalho não possui o propósito de invadir a privacidade da população, nem a indução de algum crime cibernético, em decorrência de, ao concluir a criação de uma conta na rede social aqui utilizada, o usuário concorda com os termos de serviço e consequentemente com a política de privacidade vigente.

Mediante o exposto, propõe-se neste tratado acadêmico, criar um sistema capaz de detectar termos e palavras-chave acerca do *doping*, tendo como fonte de dados a rede social *Twitter*. Para facilitar a compreensão do sistema pelo leitor, as entidades e seus relacionamentos foram projetados e estruturados através da modelagem de dados. Esta metodologia esclarece a organização e a interação dos componentes do sistema, oferecendo uma base robusta para análise e interpretação dos dados.

3.4.1 Modelagem de Dados

A modelagem de dados é o processo de criação de uma representação visual de um sistema de informação inteiro ou de partes dele para comunicar conexões entre pontos de dados e estruturas (IBM, 2020).

Conforme apresentado por (VINHAS, 2016 apud PIONTKOSKI, 2017), um banco de dados consiste em uma coleção organizada de dados, que são armazenadas de modo a facilitar o acesso e a manipulação de forma eficaz. Ressaltado pelo autor, os bancos de dados exercem um papel crucial ao converter os computadores em repositórios robustos de informações, capacitando-os a armazenar e recuperar dados de maneira mais aprimorada.

Continuando com essa abordagem, após uma análise aprofundada sobre o modelo de banco de dados mais adequado para o sistema em questão, chegamos à conclusão de que a escolha mais apropriada seria a base de dados relacional. No entanto, surge a pergunta: por que não optar pelo modelo não relacional?

Os bancos de dados relacionais e não-relacionais possuem características distintas na estruturação e gerenciamento dos dados. Os relacionais armazenam dados em tabelas com linhas e colunas. Possuem um esquema de dados rígido, definido previa-

mente (*schema-on-write*), ou seja, a estrutura e os tipos de dados das tabelas, colunas, restrições e relacionamentos devem ser especificados e configurados previamente, antes que qualquer dado seja inserido.

Acrescente-se ainda que o relacionamento entre os dados é estabelecido por chaves primárias e estrangeiras, criando relações entre as tabelas. Utilizam SQL como linguagem de consulta para gerenciamento e manipulação dos dados e oferecem suporte a transações relacionadas a Atomicidade, Consistência, Isolamento e Durabilidade (ACID) para garantir confiabilidade. Exemplos comuns incluem ferramentas como MySQL, *PostgreSQL*, Oracle e SQL Server.

Por outro lado, os Bancos de dados Não-Relacionais (NoSQL) apresentam modelos de dados variados, como chave-valor, documentos, colunas amplas ou grafos. Possuem um esquema de dados (*schema-on-read*) mais flexível que o *schema-on-write*, permitindo alterações a qualquer momento, onde a estrutura dos dados é definida dinamicamente durante a leitura dos dados.

Ademais, menos ênfase é dada às relações tabelares, com foco na escalabilidade horizontal. A linguagem de consulta varia dependendo do tipo de NoSQL, por exemplo, o *MongoDB* utiliza uma Interface de Programação de Aplicação (API) de consulta baseada em documentos. Alguns bancos de dados deste tipo oferecem transações ACID, porém muitos priorizam a performance e escalabilidade, ainda que os dados possam não ser atualizados imediatamente em todos os lugares. Exemplos incluem *MongoDB*, *Cassandra*, *Redis* e *Neo4j*.

A partir desses conceitos, devido à propensão do sistema em receber grandes volumes de informações ao longo do tempo, e pela necessidade de alta confiabilidade em cima destes dados de usuários reais, foi então decidido utilizar o banco de dados relacional. Além do mais, sentiu-se a necessidade de relacionar as tabelas para facilitar a comunicação entre elas, e foi dado prioridade ao *schema-on-write* por evitar inconsistência e por garantir consultas aos dados mais eficientes.

Assim, se torna fundamental no desenvolvimento de sistemas de banco de dados relacionais, aprimorar a compreensão e identificação das entidades e seus respectivos

relacionamentos. Ademais, tal metodologia oferece uma visão unificada dos dados da organização.

A partir do contexto mencionado, por ser um sistema robusto foi utilizada a modelagem Entidade Relacionamento (ER). Esta técnica foi escolhida pela sua ampla simplicidade e legibilidade, produzindo um modelo que seja inteligível para o desenvolvedor do banco, assim como pelo usuário final, além de descrever o modelo de dados de um sistema com alto nível de abstração (FRANCK et al., 2021).

Ao longo desta pesquisa, os diagramas serão apresentados de acordo com as entidades abordadas. As classes principais são *categories*, *terms*, *synonyms*, *users*, *twitter_profiles*, *twitter_users* e *tweets*. Acrescente-se que os demais elementos estão diretamente relacionados a pelo menos uma dessas classes principais.

A priori, o sistema deve disponibilizar uma página inicial na qual apresenta de forma concisa a plataforma e suas funcionalidades, permitindo ao usuário fazer seu cadastro e solicitar a aprovação, ou realize o *login* caso já esteja devidamente registrado e com a autorização concedida.

Por conseguinte, para o usuário visualizar as informações desejadas do sistema, é importante que a aplicação tenha capacidade de buscar e extrair tais dados. Assim sendo, é disposto uma abstração nomeada "termo", que serão as palavras-chaves na qual o usuário pode cadastrá-las na aplicação e nesse sentido, investigadas nas redes sociais e utilizadas para a coleta de informações e armazenamento no banco de dados.

Dentro da classe *terms* é possível observar a existência de campos informativos específicos, como *fetched_at*, *updated_at* e *total_tweets*. O campo *fetched_at* representa o instante no qual o termo foi capturado nas rede social em análise. Por outro lado, *updated_at* indica o momento preciso em que ocorreu a última atualização do termo e *total_tweets* refere-se ao número de *tweets* nos quais esse termo específico está associado.

É permitido, também, ao usuário a criação de categorias em português e inglês, com a opção de removê-las posteriormente. É acrescido a definição de que um termo deve pertencer a uma categoria com o intuito de promover a organização dentro

do sistema. A categoria proposta é responsável por reunir palavras-chave similares, simplificando a organização desses elementos no sistema.

Além disso, para ampliar o escopo de informações sobre um termo e facilitar a análise, é possível vincular sinônimos a um termo cadastrado. A inclusão de sinônimos no sistema solucionou o problema das variações de terminologia, que anteriormente afetavam a precisão dos dados coletados. Consequentemente, a implementação dos sinônimos resultou em um aumento substancial no volume de informações disponíveis para um determinado tópico, aumentando a complexidade do monitoramento.

Na Figura 3.3, é possível observar a forma como foi modelado o esquema dos termos dentro de suas categorias e seus sinônimos.

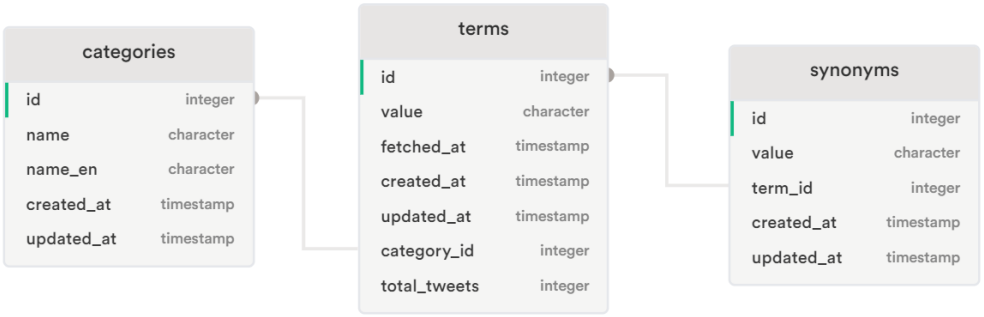

Figura 3.3: Diagrama ER acerca da entidade *Terms* no banco de dados

Desta maneira, o sistema permite ao usuário ter a aptidão de observar informações de *tweets*. Isso consiste em demonstrar a data de postagem, o quantitativo de respostas, *retweets* e *likes*, além de exibir informações adicionais da publicação, tais como a linguagem, número de citações, o sistema operacional do dispositivo e a última atualização do perfil.

Através da busca pelos termos, o sistema deve permitir, também, que o usuário consiga ver as informações de quem fez o *post*. Ou seja, será detalhado a biografia do perfil, a localização, quantidade de seguidores, se é verificado ou não, a quantidade de publicações já feitas e os links relacionados em que o dono da conta pode adicionar

ao perfil. É importante ressaltar que o usuário pode ser capturado várias vezes pelas publicações adquiridas.

Contudo, as informações capturadas ficam estáticas pela inviabilidade de atualizar continuamente cada *tweet* coletado devido ao alto custo operacional. Isso implica estabelecer um prazo para a atualização das informações do usuário capturado. Se o usuário for capturado dentro desse período, a atualização não é necessária. Caso contrário, um novo registro é gerado com informações atualizadas. Assim, as informações são armazenadas no momento da captura, como uma foto do *tweet* no instante da coleta.

Cabe salientar, que um *tweet* fará parte de um termo apenas se o termo ou seus sinônimos estiverem presentes no texto do *tweet*, e assim, um *tweet* pode pertencer a diversos termos. Então, foi tomada a decisão de não haver relação direta entre um termo e um *tweet*, a fim de reduzir a complexidade associada à verificação repetida da associação entre *tweets* e termos. Dessa forma, somente quando for necessário, a associação apenas será realizada através de uma busca específica.

Na plataforma Twitter, o usuário tem a capacidade de realizar alterações em seu perfil de maneira flexível, como, por exemplo, modificar sua imagem de perfil, ajustar sua descrição, atualizar sua localização, entre outras possibilidades. Diante dessa característica, o sistema foi concebido considerando a viabilidade de realizar capturas periódicas do referido perfil. Dessa forma, possibilita-se a atualização do perfil armazenado de acordo com o estado mais recente do mesmo.

Dessa forma, tornou-se necessário a criação de duas tabelas: *twitter_users* e *twitter_profile*, visto na Figura 3.4. A tabela *twitter_users* armazena o perfil mais recentemente atualizado, enquanto a tabela *twitter_profile* contém as informações capturadas no momento da atualização do perfil. Essa estrutura possibilita a realização de múltiplas atualizações no mesmo perfil, ao permitir a atualização contínua das informações na tabela *twitter_profile*.

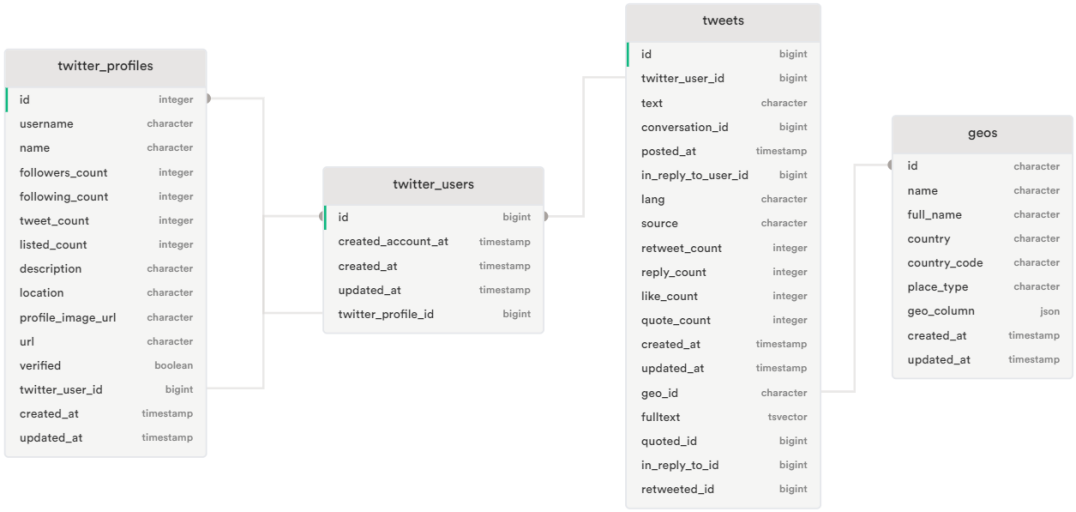

Figura 3.4: Modelagem acerca de *Tweets* contidos na base de dados

Dentro do sistema, observa-se na Figura 3.5 a entidade *users* que representa os indivíduos registrados na plataforma. Esta classe *users* está relacionada à modelagem da classe *api_tokens*. Dentro de *users* contém informações de email, senha, um indicador booleano de status administrativo, nome e data de criação do registro do usuário.

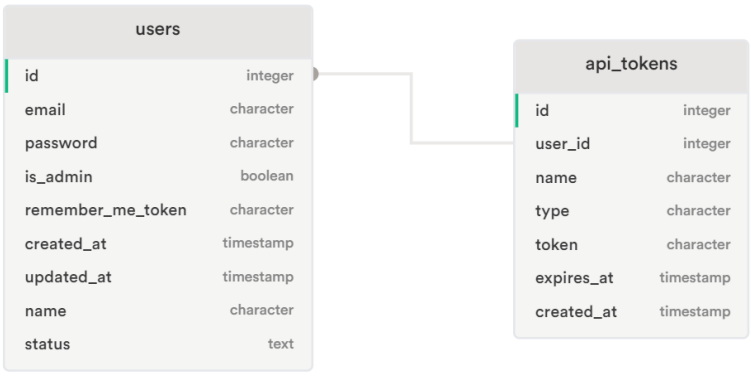

Figura 3.5: Modelagem acerca dos usuários do sistema contidos na base de dados

De forma geral, na Figura 3.6 é possível observar a modelagem na atual conjuntura, de acordo com todas as entidades e relacionamentos citados.

Figura 3.6: Diagrama da modelagem ER utilizada no sistema

A partir do que foi mencionado, disponibiliza-se ao leitor desta monografia uma arquitetura do sistema baseada no modelo C4 na Figura 3.7. O intuito é padronizar de maneira coerente e eficiente a representação da arquitetura de um software, facilitando assim a comunicação entre todos os envolvidos no projeto de software.

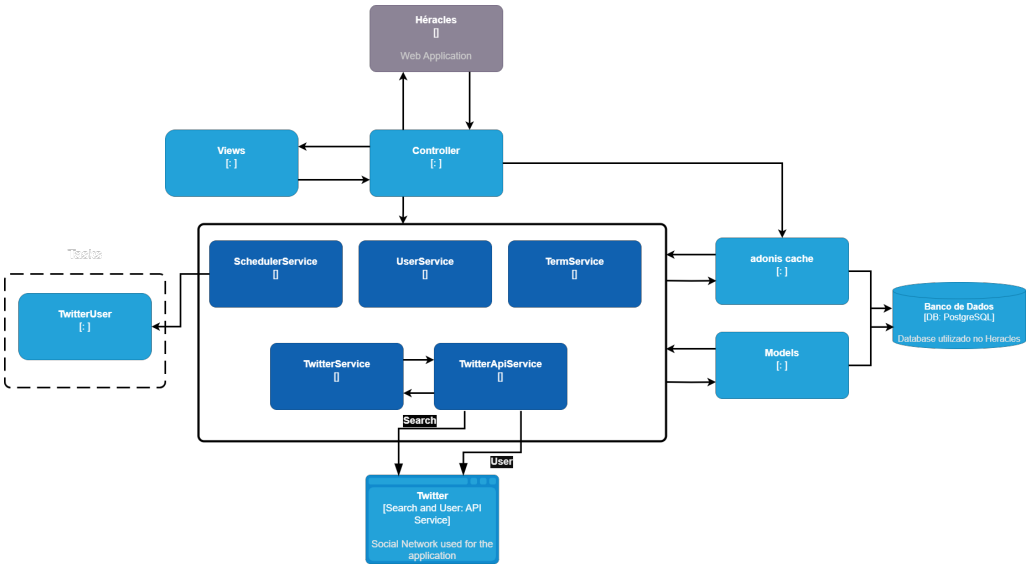

Figura 3.7: Arquitetura do sistema Héracles

3.4.2 Desempenho de Sistema

Um dos pontos discutidos na implementação do sistema, foi seu desempenho. Sabendo que ao tratar de extração e armazenamento de dados, este seria um fator crucial para o bom uso da plataforma pelos seus usuários, foi pensado em técnicas para aprimorar a inserção e as consultas destas informações no sistema. Nesta subseção será discutido estes procedimentos.

Inicialmente, na parte do front-end, a forma que os dados são exibidos poderia afetar no tempo de espera para carregar a página. Pensando nisso, foi utilizada a técnica de Paginação de dados. Esta, é uma técnica usada para dividir grandes conjuntos de dados em partes menores e mais gerenciáveis, chamadas de "páginas".

Por ter milhares de registros em uma tabela de banco de dados, carregar e exibir todos esses registros de uma só vez pode ser ineficiente e desnecessário. A paginação permite mostrar apenas uma parte desses registros por vez, e a partir desta estratégia, não apenas reduz a carga no banco de dados, mas também melhora a experiência do usuário, por diminuir o tempo de carregamento e tornar a interface mais fácil de navegar.

Ao adotar essa abordagem, o tempo de resposta das consultas de paginação é significativamente melhor, especialmente quando se trata de um grande número de páginas. Dessa forma, foi utilizada para melhorar a performance e a usabilidade ao lidar com grandes quantidades de dados. Além disso, foi implementado algoritmos e estratégias que priorizavam o tratamento de dados no banco, como Cache e Pesquisa em Texto Completo.

3.4.2.1 Cache

Sabendo do custo de desempenho envolvido para realizar consultas e acessar informações que estão recorrentemente sendo acessadas, utilizou-se a estratégia de armazenar dados em cache. Esta técnica serve como um local de armazenamento temporário de dados, a qual permite que futuras requisições destes mesmos dados, sejam acessados de forma mais rápida em comparação a uma busca no servidor, por

exemplo.

Dessa forma, o desempenho do sistema é elevado em comparação com a necessidade de consultar uma informação toda vez que for preciso. Ao salvar essas informações em cache, o custo é reduzido de forma satisfatória. Concomitante a isso, existem várias estratégias e cálculos direcionados a orientar o que, quando e como os dados devem ser armazenados em cache, tornando-se essenciais para maximizar a eficiência do sistema.

A aplicação proposta tem por previsão armazenar um grande número de dados capturados e realizar diversas consultas ao longo do tempo. Portanto, é de extrema necessidade utilizar uma estratégica para lidar com estas informações de forma bastante otimizada. Dessa maneira, em concordância com o modelo de banco de dados utilizado, foi priorizado ao desempenho de inserção de elementos no sistema ao capturar dados.

Através desta técnica, é possível colher o máximo de dados possível, por dar prioridade ao desempenho de elementos no sistema. O objetivo não é exibir dados, e sim capturá-los, de acordo com o *schema-on-write* do banco de dados. Tal abordagem não apenas facilita a acumulação eficaz de dados mas também reforça a capacidade de resposta e a eficiência da aplicação frente às demandas operacionais.

Nos gráficos, por não necessitar de captura a todo momento e ser apenas Estatística, foi utilizada a técnica de cache por expiração. A configuração foi a partir do conhecido “Absolute Expiration”, a qual determina um tempo de vida para os dados em cache, independente do seu uso ou não e assim, posteriormente eles são atualizados.

Outra funcionalidade que poderia impactar na performance do sistema, é a de contar o total de *tweets* que um termo capturou, pois a quantidade de termos cadastrado poderia afetar na velocidade do cálculo. Por exemplo, ao capturar um termo “maconha” possa ser que dentro dos *tweets* capturados contenha algum outro termo, como “deposteron”, sendo assim melhor recalcular após o *fetch* em todos os termos.

Diante dessa consideração, optou-se por calcular o *CountTweets* sempre que um termo é capturado. Assim garante uma abordagem mais eficiente em termos de desempenho, pois não há a sobrecarga associada ao processamento em segundo plano. Isso não apenas simplifica a lógica do sistema, mas também proporciona uma resposta mais ágil e em tempo real quando um termo é identificado.

Por fim, a implementação da estratégia de cache no sistema revelou-se altamente vantajosa em termos de desempenho. Ao manter essas informações em cache, o sistema demonstra uma notável agilidade de resposta ao recuperar dados já armazenados, superando significativamente o desempenho que teria caso essa abordagem de armazenamento em cache não fosse adotada.

Diante dos fatos apresentados, a plataforma tende e dá prioridade a capturar dados. Contudo, e quanto a leitura destas informações? A abordagem utilizada foi a pesquisa em texto completo, ou *Full Text Search*, que permite a busca por palavras ou frases dentro de grandes volumes de texto de forma eficiente e rápida. Na subseção seguinte, exploraremos em detalhes a metodologia aplicada, destacando sua importância no processamento e na recuperação de informações em bases de dados volumosas.

3.4.2.2 Full Text Search

A busca de texto completo (*Full Text Search*) é uma técnica avançada de pesquisa em bancos de dados que permite realizar buscas eficientes em grandes quantidades de texto. Diferentemente das buscas tradicionais, que procuram correspondências exatas nos dados, a busca de texto completo analisa e indexa o conteúdo textual, permitindo buscas por palavras-chave, frases, ou até variações linguísticas dentro de grandes volumes de texto (GROUP, 2024).

No contexto de desempenho do sistema, o uso deste método aprimorou significativamente a eficiência e a velocidade das buscas. Um dos aspectos mais notáveis dessa melhoria é a incorporação do Processamento de Linguagem Natural (PNL). Esta técnica permite ao nosso sistema entender o contexto e a semântica das palavras utilizadas nas consultas, além das relações entre elas.

Além da PNL, é importante entender sobre Generalized Inverted Index (GiN) e Generalized Search Tree (GiST). Estes, são dois métodos de indexação oferecidos pelo *PostgreSQL* a qual aprimora significativamente a eficiência e a velocidade das buscas. Dessa maneira, melhoram a experiência do usuário ao permitir a descoberta de informações relevantes, contudo, ambos possuem características distintas que os tornam mais adequados para diferentes cenários de aplicação.

O índice GiN é especialmente projetado para lidar com tipos de dados que podem conter múltiplos valores dentro de uma única coluna, como *arrays* e, principalmente, texto completo. A estrutura desta funcionalidade é otimizada para operações de contenção, ou seja, consultas que verificam se um valor específico está presente em uma coleção ou conjunto de valores, e assim permite buscas rápidas e eficientes em grandes volumes de texto (STRATE, 2019).

Por outro lado, o índice GiST é mais flexível e pode ser usado para uma variedade mais ampla de tipos de dados e operações de consulta. É uma estrutura de árvore balanceada que permite implementar vários algoritmos de busca personalizados, como para dados geométricos e intervalos. No contexto de *Full Text Search*, o GiST é útil para consultas complexas, como proximidade semântica ou busca por similaridade, onde a estrutura de árvore contribui a podar o espaço de busca de forma mais eficiente (STRATE, 2019).

Outra metodologia utilizada foi através da expansão de sinônimos já citada acima, enriquecendo o escopo das buscas. Ao inserir uma consulta, ele automaticamente analisa e inclui termos similares cadastrados na tabela *synonyms*, permitindo aos usuários acessar um leque mais amplo de informações.

Essa funcionalidade assegura que conteúdos relevantes sejam encontrados, mesmo que não correspondam exatamente às palavras-chave especificadas na busca inicial. Por exemplo, a busca por “Olimpíadas” automaticamente se estende para incluir sinônimos como “Jogos Olímpicos”, capturando uma gama diversificada de dados pertinentes.

Essas metodologias não apenas melhora significativamente a experiência do

usuário, como também aprimora a descoberta de informações. Os usuários se beneficiam ao acessar um espectro mais amplo de dados relevantes, muitas vezes encontrando informações que ultrapassam suas expectativas iniciais e não apenas atenda às necessidades imediatas de informação dos usuários.

Considerando a grande quantidade de dados textuais envolvidos, torna-se essencial a utilização de uma abordagem para gerenciar esse volume de informação. A partir disso, foi-se utilizado o método de indexação de texto completo. Essa abordagem utiliza uma estrutura de dados projetada para facilitar buscas rápidas e eficazes em extensos conjuntos de dados textuais.

Inicialmente, faz-se uma análise detalhada de cada parte de texto presente em um determinado conjunto de dados, como textos encontrados em arquivos ou documentos. O processo inicia com a eliminação dos diacríticos, que são sinais gráficos adicionados a letras para alterar a sua realização fonética ou fonológica, *e.g.*, acento agudo, cedilha e acento circunflexo.

Posteriormente, empregam-se algoritmos específicos para o idioma do texto para identificar e remover as palavras consideradas de preenchimento, conhecidas como *"filler words"*, que são termos frequentemente usados mas de pouco valor para a busca, como preposições e conjunções. Concomitantemente, o processo de *Stemming* é aplicado, o qual consiste em reduzir as palavras ao seu radical básico, facilitando a classificação de diferentes formas da mesma palavra como equivalentes.

Em relação ao *Stemming*, um exemplo a ser dado para melhor entendimento é o verbo "correr". Em diversas formas, como "correu", "correndo", e "corrida", todas essas variantes são reduzidas ao seu radical "corr", com o objetivo de simplificar a busca e análise de texto. Logo após esta etapa, os termos são convertidos todos para minúsculo ou maiúsculo.

A Figura 3.8 serve como exemplificação deste processo, a qual decidimos no último passo converter as palavras para letras maiúsculas, processo conhecido como *"toUpperCase"*.

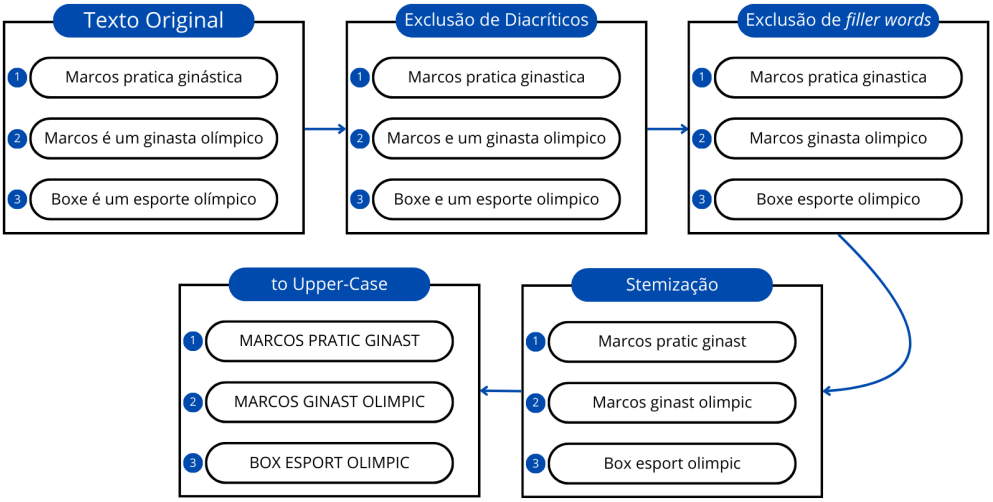

Figura 3.8: Processos inicial do índice de busca de texto completo

Por fim, a construção do índice se dá pela organização do dicionário gerado, no qual se armazenam as referências indicativas dos locais específicos, por exemplo, em qual parte do documento, em que cada termo identificado, seja uma palavra ou uma expressão, está localizado, demonstrado na Figura 3.9. Este processo facilita a rápida localização dos termos dentro dos documentos em que eles aparecem, aprimorando o desempenho na busca dos termos.

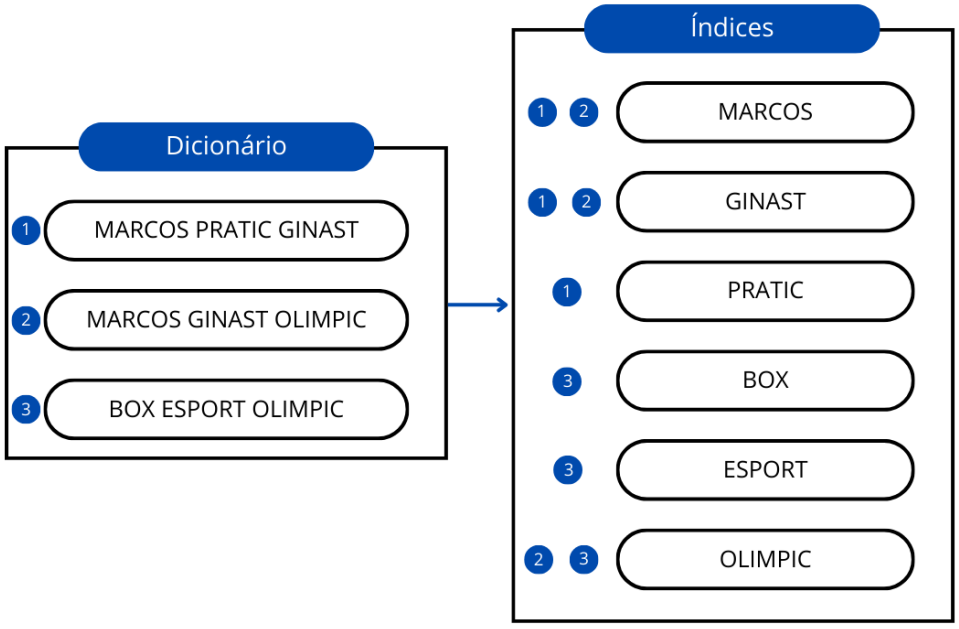

Figura 3.9: indexação final do *Full Text Search*

3.4.3 Casos de Uso

Os casos de uso não se concentram nas funcionalidades do sistema, mas sim nas ações e interações dos diferentes tipos de usuários, no qual classificamos como atores. Enquanto a Tabela 3.2 apresenta os tipos de usuários envolvidos, a Tabela 3.3 explica os casos de uso abordados do visitante, já na Tabela 3.4 referencia os usuários do sistema e na Tabela 3.5 os administradores são abordados.

Ator	**Descrição do Ator**
Visitante	Usuário antes de solicitar o cadastro e ter o mesmo aprovado pelos administradores do sistema.
Usuário	Usuário público após ter o cadastro aprovado pelos administradores.
Administrador	Usuário com o nível hierárquico mais alto da aplicação, com permissões acrescidas em relação aos demais usuários.

Tabela 3.2: Atores do Sistema

Nome: **Ator:** **Descrição:**	Realizar *login* Visitante O visitante pode efetuar *login* caso tenha cadastro aceito no sistema.
Nome: **Ator:** **Descrição:**	Cadastrar *login* Visitante O visitante pode se cadastrar, mas precisa ser aprovado para efetuar *login*.

Tabela 3.3: Descrição Casos de uso do Visitante

Nome:	Cadastrar termo
Ator:	Usuário
Descrição:	O usuário pode cadastrar um termo a ser capturado no sistema.
Nome:	Editar termo
Ator:	Usuário
Descrição:	O usuário pode editar atributos de um termo cadastrado.
Nome:	Deletar termo
Ator:	Usuário
Descrição:	O usuário pode deletar um termo previamente cadastrado.
Nome:	Buscar termo
Ator:	Usuário
Descrição:	O usuário pode capturar *tweets* relacionados a um termo cadastrado.
Nome:	Cadastrar categoria
Ator:	Usuário
Descrição:	O usuário pode cadastrar uma categoria para classificar os termos cadastrados.
Nome:	Deletar categoria
Ator:	Usuário
Descrição:	O usuário pode deletar uma categoria.
Nome:	Atualizar perfil do usuário
Ator:	Usuário
Descrição:	O usuário pode atualizar um perfil capturado.
Nome:	Autorizar usuário
Ator:	Usuário
Descrição:	O usuário pode aprovar um pedido cadastro.
Nome:	Visualizar *dashboard*
Ator:	Usuário
Descrição:	O usuário pode visualizar o *dashboard* de dados.

Tabela 3.4: Descrição Casos de uso do Usuário

Nome:	Configurar agendador
Ator:	Administrador
Descrição:	O administrador pode configurar o agendador.
Nome:	Elevar nível de usuário
Ator:	Administrador
Descrição:	O administrador pode elevar o nível de um usuário.
Nome:	Bloquear usuário
Ator:	Administrador
Descrição:	O administrador pode bloquear um usuário.

Tabela 3.5: Descrição Casos de uso do Administrador

3.4.4 Caso de Uso Termos

Na abordagem do caso de uso Termos, foi pensado na maior abrangência de captura pelo usuário, a partir do termo selecionado. Sabendo que palavras possuem sinônimos, e que um termo em seu estado mais puro é uma palavra, foi decidido adicionar sinônimos aos termos cadastrados.

Todavia, dessa forma haveria a possibilidade de, pelos sinônimos de um termo, o raio de captura ser tão grande que acabaria reduzindo o efeito da objetividade ao escolher o termo a ser utilizado.

Por exemplo o termo futebol pertence a categoria Esportes e o termo “trembolona” à categoria Anabolizantes. A partir da criação da classificação, cada termo deve ser enquadrado em uma categoria previamente cadastrada, afim de ser classificado, dando objetividade à sua busca.

Capítulo 4

Tecnologias Utilizadas e Análises

Este capítulo detalha as ferramentas selecionadas para a execução do projeto, bem como os critérios que orientaram essas escolhas. Serão abordadas as decisões relacionadas às tecnologias utilizadas na construção e eficácia do sistema proposto, destacando os critérios técnicos e os benefícios específicos de cada uma para o alcance dos objetivos estabelecidos.

Adicionalmente, serão apresentadas análises das consultas realizadas, comparando o desempenho das soluções adotadas em termos de eficiência do sistema e identificando as limitações encontradas durante a implementação do projeto.

4.1 Tecnologias Utilizadas

Inicialmente, é importante entender que a quantidade de tecnologias em de desenvolvimento web é ampla e diversificada, com muitas opções para cada aspecto do sistema. A escolha de tecnologias foi baseada não apenas em popularidade ou capacidade de atender a requisitos técnicos, mas também na integração com outras ferramentas e na disponibilidade de documentação.

O desenvolvimento do sistema web proposto exigiu a escolha de tecnologias que atendem às necessidades de funcionalidade, desempenho, segurança e escalabilidade. Desse modo, a seguir serão detalhadas as tecnologias escolhidas, explicando como

elas contribuem para a arquitetura do sistema, as razões para sua seleção e o papel que desempenham na solução proposta, desde o *backend* até o *frontend*.

4.1.1 *Node.js*

Node.js é um ambiente de execução *JavaScript* de código aberto focado na eficiência de aplicações autônomas sem a necessidade de um navegador para a execução. Esse software distingue-se por sua capacidade de interpretar e converter código *JavaScript* em linguagem de máquina. Essa funcionalidade é viabilizada pelo motor V8 da Google[1], um interpretador *JavaScript* avançado desenvolvido na linguagem C++, cujo propósito é acelerar a execução de aplicações *JavaScript*.

Algo importante do *Node.js* é sua arquitetura assíncrona e orientada a eventos, a qual facilita a realização de operações de Entrada/Saída (I/O) sem bloqueio. Isso torna o *Node.js* especialmente adequado para tarefas que exigem uso intensivo de dados ou análises em tempo real, oferecendo excelente desempenho. Sua natureza assíncrona permite que processos demorados sejam executados em segundo plano, sem interromper o fluxo de execução da aplicação.

Dessa maneira, diferente de outras plataformas que utilizam múltiplas threads para processar tarefas em paralelo, o *Node.js* adota um modelo *single thread*. Essa característica elimina a necessidade de gerenciar múltiplas *threads*, e otimiza o uso de memória e recursos da CPU, além de evitar o desperdício de recursos enquanto aguarda respostas. Tal abordagem facilita a gestão de numerosas solicitações simultâneas, sem comprometer a eficiência.

A capacidade do *Node.js* de operar sem bloqueios é atribuída ao sistema de *callbacks* do *JavaScript* e ao seu *loop* de eventos. Quando uma solicitação é feita ao servidor web, a recursividade de eventos processa o pedido e o encaminha para processamento sem reter a *thread* principal. Essa metodologia assegura que operações de E/S possam ser realizadas de maneira fluida e eficiente.

Vale abordar também, um componente fundamental do *Node.js*: O Node Package

[1] <http://www.google.com>

Manager (NPM). Além de ser uma ferramenta de linha de comando para a instalação e gestão de dependências, o npm atua como um vasto repositório público de pacotes, aumentando a eficiência e a produtividade no desenvolvimento de projetos *Node.js*. Dessa maneira, o npm desempenhou um papel fundamental na otimização da gestão de dependências.

Assim, foi possível integrar de forma simples e eficaz uma série de bibliotecas e módulos externos, enriquecendo a aplicação com funcionalidades adicionais sem a necessidade de criar esses elementos do zero. Além de facilitar o gerenciamento de pacotes, também suporta a execução de *scripts*, o que automatizou diversas tarefas de desenvolvimento, como iniciar o servidor, verificar possíveis erros de código e compilação.

Embora o *Node.js* privilegie operações assíncronas, é importante destacar que métodos síncronos são suportados e necessários em determinados cenários. A preferência por processos assíncronos trouxe ao projeto maior agilidade do sistema, e diminuiu a latência em operações que poderiam, de outra forma, resultar em bloqueios. Portanto, o *Node.js* foi uma solução robusta e eficiente para o desenvolvimento do projeto aqui trabalhado, por ser capaz de lidar com desafios de performance e escalabilidade de forma tão eficaz.

4.1.2 *AdonisJS*

AdonisJS consiste num *framework* robusto baseado em TypeScript, com o objetivo de otimizar o desenvolvimento de aplicações *Node.js*. Este *framework* destaca-se por introduzir padrões preestabelecidos de organização de projetos. Ou seja, uma característica marcante do *AdonisJS* é sua natureza conhecida como "opinativa", a qual promove um modelo de desenvolvimento específico e induz sos desenvolvedores a aderirem este modelo.

Ao adotar o TypeScript como sua linguagem de implementação, tal *framework* garante que as aplicações construídas utilizando sua arquitetura sejam igualmente fundamentadas nesta linguagem atual e avançada. Esta escolha contribuiu significativamente para a coesão técnica do projeto, além de otimizar o processo de manutenção

do código. A seleção desta linguagem, reconhecida por adotar características atuais e modernas, revelou-se uma feliz decisão por ampliar a robustez e a escalabilidade das aplicações.

Como dito anteriormente, o *AdonisJS* promove um modelo de desenvolvimento específico. Esta estrutura conhecida como arquitetura Model-View-Controller (MVC) é um padrão de design de software amplamente adotado para o desenvolvimento de aplicações web, a qual oferece uma estrutura organizada e intuitiva para desenvolvedores. A seguir, é detalhado de forma mais aprofundada sobre cada componente:

- Models:

  Este representa a camada de dados da aplicação e simplifica a interação com o banco de dados. Essa simplificação é alcançada por meio do Object-Relational Mapping (ORM) Lucid, uma ferramenta que elimina a necessidade de redigir consultas SQL complexas, e facilita a execução de operações no banco de dados de forma orientada a objetos como um Create, Read, Update, Delete (CRUD).

  Esta abordagem contribui significativamente para uma interação segura e eficaz com o banco de dados. Assim, os Models no contexto do *AdonisJS* não apenas representam as tabelas do banco de forma abstrata, mas também encapsulam a complexidade da gestão de dados.

- Views:

  Esta camada utiliza o Edge como sua *engine* de template padrão, a qual é projetada para ser rápida e com recursos ricos, permitindo aos desenvolvedores escreverem HTML de forma expressiva e dinâmica. Desse modo, é focada exclusivamente na camada de apresentação e suporta a reutilização através de componentes e layouts o que reduz a duplicação de código.

  Além disso, podem receber e apresentar dados dinâmicos que são passados pelos *Controllers*, estas detalhadas no próximo tópico. Esses dados são usados para preencher o conteúdo das páginas, a permitir a personalização da experiência do usuário com base em informações específicas, como dados formados por consultas ao banco de dados, mensagens de erro e muito mais.

Algumas Views do atual projeto serão demonstradas. Na figura 4.1 é demonstrado a tela de "*login*", a qual o usuário insere o *e-mail* cadastrado e sua senha. Caso ainda não possua registro no sistema, o usuário clica em "Esqueceu a senha?" e é direcionado para a página de cadastro representada pela figura 4.2 em que o visitante insere seu nome, *e-mail*, senha e novamente a senha como confirmação.

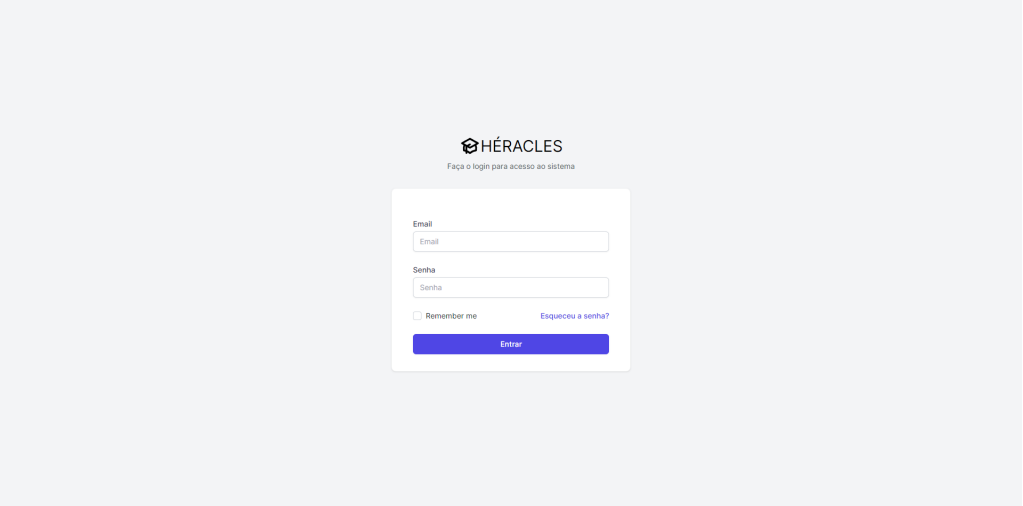

Figura 4.1: Tela de Login

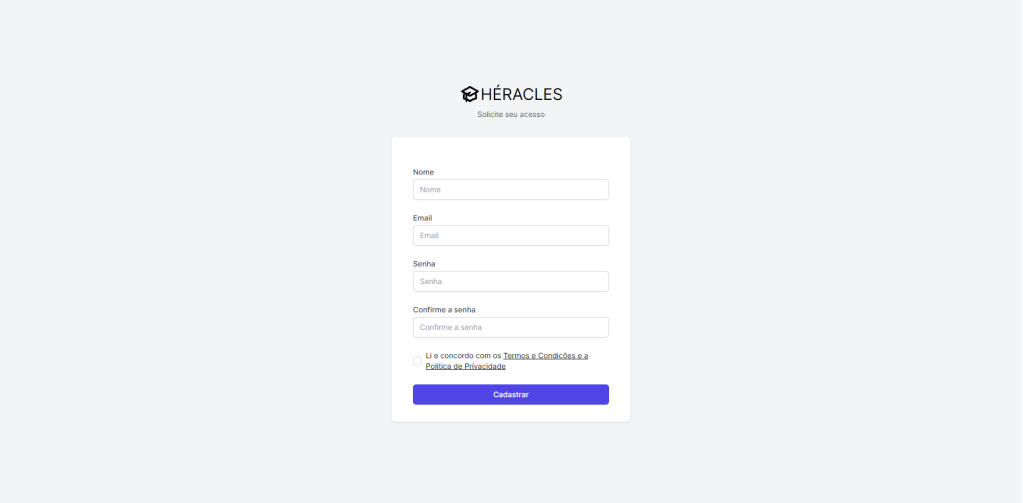

Figura 4.2: Tela de cadastro no sistema

Desse modo, temos a nossa "*Home*" demonstrada pela imagem 4.3 com *links* que encaminham o usuário para diversas partes do sistema. Como exemplificação,

algumas destas partes são as categorias cadastradas com a opção de adicioná-las, denotada pela gravura 4.4.

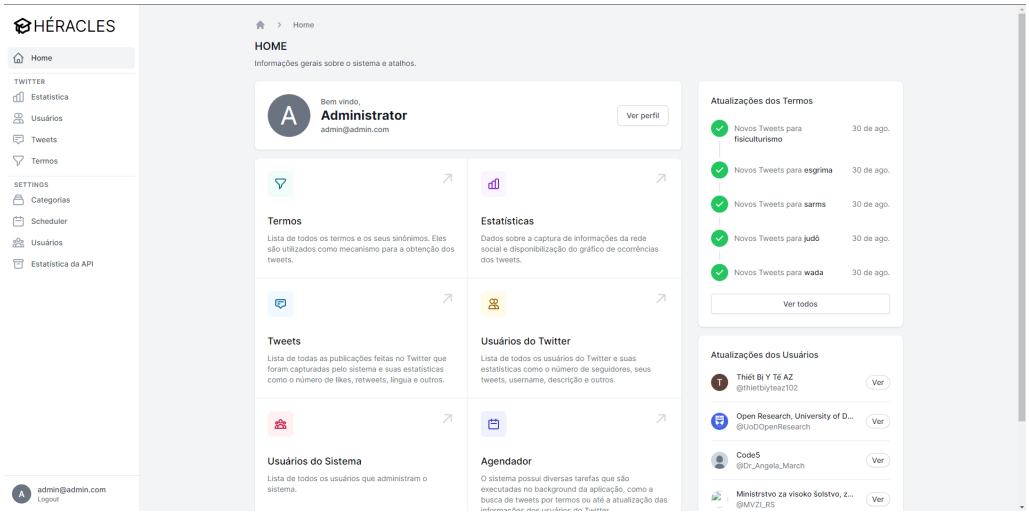

Figura 4.3: Tela Home do sistema

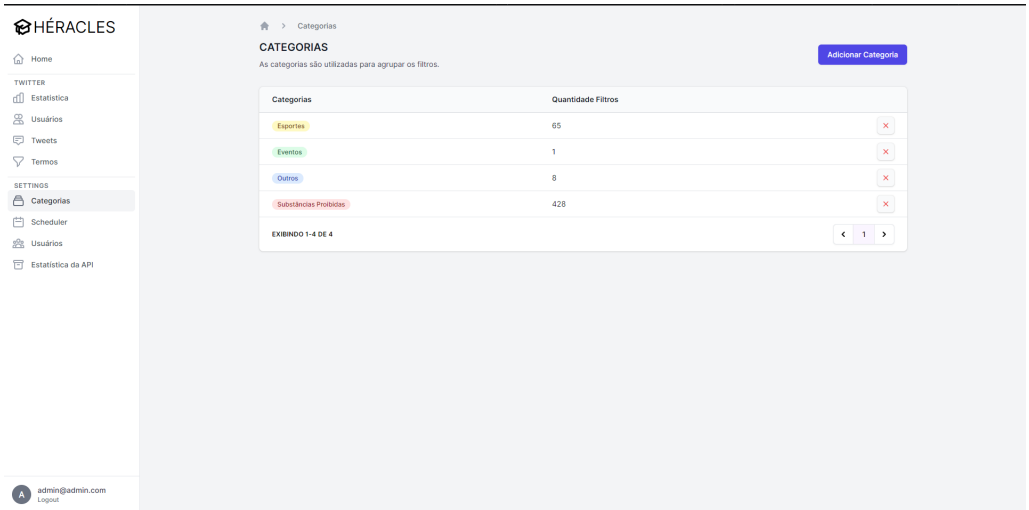

Figura 4.4: Tela de Categorias no sistema

Além das categorias, são exemplificadas as Views do "Scheduler" e dos termos, representados pelas Figuras 4.5 e 4.6, respectivamente. Na tela do agendador, é possível algumas configurações, como alterar o regex do *CronJobs* e declarar quantos *tweets* máximos serão capturados por chamada. Nos termos, há a possibilidade de cadastrar novos termos, capturar *posts* com estes já cadastrados e são demonstrados os mesmos com suas informações, além de filtrá-los.

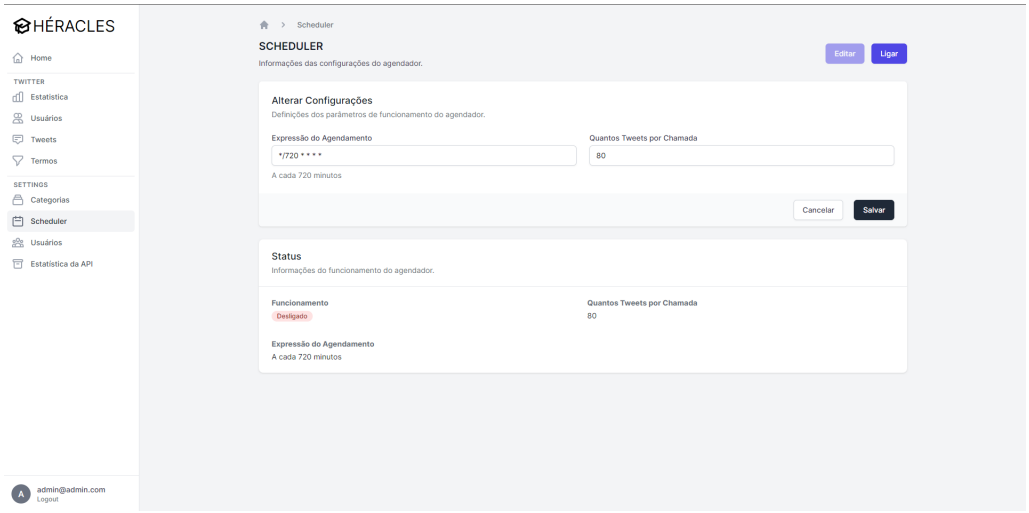

Figura 4.5: Tela do Agendador no sistema

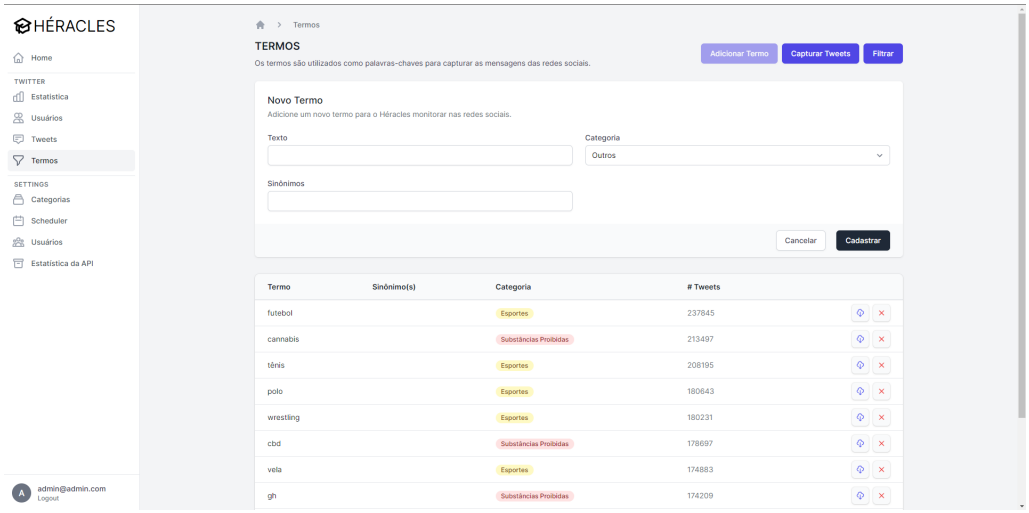

Figura 4.6: Tela de Termos no sistema

- Controller:

  São componentes responsáveis pela comunicação entre a camada das *Views* e *Models*, gerenciando a lógica de interação entre o usuário e os dados da aplicação. Após receberem as requisições encaminhadas através do processamento de solicitações HTTP, os controladores processam esses dados, interagem com os modelos (quando necessário) e retornam uma resposta ao cliente.

  Dessa maneira, os *Controllers* no *AdonisJS* vão além de meros intermediários entre *Views* e *Models*; são fundamentais na definição da lógica e comportamento

da aplicação, e faciilitam a interação entre usuários e o sistema de forma eficiente e organizada. Assim, se torna importante na manutenção da estrutura do projeto, como uma ferramenta essencial para o desenvolvimento da solução web aqui presente se tornar eficiente e bem estruturada.

Portanto, o *AdonisJS* é um *framework Node.js* que oferece diversas funcionalidades destinadas a otimizar o desenvolvimento de aplicações web. Suas vantagens são projetadas para oferecer um ambiente de desenvolvimento coeso e eficiente. Em resumo, as vantagens do *AdonisJS* refletem seu design como um *framework* abrangente e moderno para o desenvolvimento de aplicações web em *Node.js*, ao oferecer uma combinação poderosa de eficiência, segurança e facilidade de uso.

4.1.3 adonis-cache

Pensando de forma estratégica em aprimorar a performance geral do atual sistema proposto por meio do caching, foi utilizado a biblioteca "adonis-cache". Esta ferramenta possibilita a rápida implementação de soluções de cache, essenciais para acelerar o tempo de resposta das aplicações e otimizar os recursos do servidor e a experiência do usuário final.

A escolha pelo adonis-cache se alinha não apenas com o objetivo de adotar práticas de otimização de desempenho desde as fases iniciais do desenvolvimento de software, mas também assegurar sua escalabilidade a longo prazo. Concomitante a isso, facilitou a reduzir o custo e a eficiência do sistema, no que diz respeito ao agendador, ao organizar os termos mais pesquisados e o volume de *tweets* capturados pelo sistema no gráfico.

4.1.4 *Tailwind CSS*

Pensando na praticidade para o desenvolvimento do *FrontEnd*, foi escolhido a biblioteca *Tailwind CSS*, que contém inúmeros componentes pré desenvolvidos, ao qual adaptamos para a nossa visão de como seria a parte visual de nosso sistema. O sistema foi pensado para ser utilizado em qualquer dispositivo, tanto computadores

quanto smartphones, por ser *mobile-first*, o *Tailwind CSS* foi a nossa escolha. Isso trouxe facilidade para o desenvolvimento de certa forma, permitindo que fosse dado mais ênfase para outras partes do sistema.

4.1.5 Scheduler ou Cron Jobs

Também conhecido como "Agendador", de forma simplista, é uma operação programada dentro do ambiente do sistema operacional Linux, com o objetivo de automatizar comandos, programas ou algo executável em um certo intervalo de tempo. Em vista disso, se torna recorrente sua aplicação em *backups* de dados, envio de *e-mails*, geração de relatórios, verificação de atualizações de software e diversas outras tarefas de automação.

A utilização desta função tem vantagens em cenários em que a execução precisa e pontual é essencial. Ela assegura uma coleta de dados consistente e regular, o que, por sua vez, facilita e aprimora a organização e a administração. Estes benefícios se originam pela automação da tarefa, e como consequência eliminam a variabilidade e os erros associados à intervenção manual.

Por outro lado, há a desvantagem de ter a possibilidade de sobrecarga do servidor, visto que, tarefas agendadas em grande quantidades e intervalos curtos, pode ocorrer uma sobrecarga significativa de processos, alocando muita memória e sobrecarregando serviços do servidor. Em decorrência disso, há a consequência imediata que é a lentidão do servidor, não apenas por excesso de processos, mas por tarefas que exigem alto processamento, como a execução de rotinas com loop infinito, por exemplo.

Na desenvolvimento do sistema abordado no projeto final presente, optou-se pela implementação do In-Process Cron Jobs. Esta escolha foi baseada no *middleware* "adonis5-scheduler", uma solução que basicamente atua como um wrapper para o "node-scheduler".

A decisão de empregar o "adonis5-scheduler" foi motivada por várias razões. Dentre elas, a realização de tarefas em segundo plano sem atividade manual ou externa, visto que as funções são executadas dentro do próprio processo da aplicação. Assim,

a arquitetura do sistema fica mais simples e reduz a complexidade operacional. Além disso, permite maior flexibilidade na manipulação de dependências e no tratamento de erros.

Outra vantagem adicional, se torna pela recorrente preocupação em não exceder a API do Twitter. Como já mencionado, a API do Twitter impõe limites rigorosos quanto ao número de solicitações que podem ser feitas em um determinado intervalo de tempo. Dessa forma, foi possível estabelecer um controle mais refinado sobre o número e a frequência das requisições feitas à API, ao agendar as solicitações de forma a distribuir as requisições ao longo do tempo, sem exceder os limites impostos.

Abaixo, há um exemplo de Regex no Código 4.1, que é uma sequência de caracteres para processar strings de acordo com regras definidas. Neste caso, é demonstrado uma tarefa agendada, em um script localizado em /heracles/captureTerms e de nome capture.sh, para todas as terças-feiras às 19:00. Logo após, na figura 4.7, uma simples tradução do Regex mencionado.

Código 4.1: Linha de comando para agendar uma tarefa

```
0 19 * * 2 /heracles/captureTerms/capture.sh
```

```
# ┌──────────────── minutos (0-59)
# │ ┌────────────── horas (0-23)
# │ │ ┌──────────── dia do mês (1-31)
# │ │ │ ┌────────── mês (1-12)
# │ │ │ │ ┌──────── dia da semana (0-6, sendo 0=domingo; 6=sábado)
# │ │ │ │ │
# │ │ │ │ │
# *  *  *  *  *  comando a ser executado
```

Figura 4.7: Interpretação de uma linha correspondente a um cron. Fonte: HostMídia (2021)

4.1.6 *PostgreSQL*

A decisão de usar o banco de relacional *PostgreSQL*, foi devido aos seus recursos no que diz respeito à escalabilidade do projeto. Escolher um banco de dados popular

e que oferece muitas funcionalidades, agregou positivamente na pesquisa, ao enfrentar problemas com a ferramenta, ter um suporte como documentação e comunidade ativa são fatores cruciais para essa escolha.

Além disso, o *PostgreSQL* tem grande eficiência em *Full Text Search*, que seria encontrar a busca não apenas comparando a *string* exata, mas sim incluindo resultados relevantes para a busca realizada. A ferramenta possui diversas funcionalidades que auxiliam a implementar o *Full Text Search*, facilitando a utilização e implementação na pesquisa.

Sabe-se que grandes volumes de dados podem afetar o custo de tempo de buscas complexas, além de dificultar a gestão desses dados armazenados. No *PostgreSQL* existem funções para lidar com diversos volumes de dados, além de ser uma ferramenta que possui grande eficiência em escalabilidade e particionamento de tabelas, alcançando assim todos os escopos desejáveis para a pesquisa.

4.2 Análise das Consultas

Através do conhecimento exposto acerca do índice invertido, foram realizadas comparações de desempenho entre consultas realizadas com índices invertidos e sem índices invertidos. Essas comparações foram feitas utilizando o sistema do *PgAdmin* para realizar as *queries* em *SQL*. A implementação de índices invertidos visa aprimorar a performance das buscas por termos relacionados ao *doping* nas redes sociais, especialmente no *Twitter*, onde a quantidade de dados gerada diariamente é imensa. Ao realizar as comparações podemos visualizar o desempenho com a utilização dos índices e sem a utilização dos mesmos.

Além disso, a precisão das buscas também foi aprimorada com o uso de índices invertidos. A técnica permitiu um melhor gerenciamento de sinônimos e termos relacionados, aumentando a relevância dos resultados retornados. Este aspecto é crucial em cenários de busca de texto completo, onde a variedade e complexidade dos termos podem dificultar a identificação precisa de conteúdos relevantes.

Do mesmo modo, a escolha entre GiN e GiST depende do tipo de consulta e das

características dos dados. O primeiro oferece melhor performance para consultas que envolvem a presença de termos específicos, devido à sua estrutura otimizada para contenção de múltiplos valores. Em contrapartida, o segundo pode ser mais eficiente em cenários onde as consultas requerem operações mais complexas ou onde os dados possuem uma estrutura hierárquica ou espacial.

Para bases de dados grandes, onde o *Full Text Search* é uma operação crítica, a implementação de índices adequados pode melhorar significativamente o desempenho. O GiN, com sua capacidade de lidar eficientemente com grandes volumes de texto e múltiplos valores, é geralmente a escolha preferida para buscas de texto completo onde a presença de termos é a principal preocupação. No entanto, se as consultas forem mais complexas e envolverem aspectos como similaridade ou proximidade, o GiST pode oferecer vantagens adicionais.

Dessa maneira, experimento foi estruturado da seguinte forma: primeiramente, uma base de dados significativa foi criada utilizando postagens reais do *Twitter* que continham termos relacionados ao *doping*. Em seguida, foram realizadas consultas utilizando índices invertidos e comparadas com consultas tradicionais que não utilizavam esses índices. As métricas de desempenho incluíram o tempo de resposta das consultas como é apresentado nos experimentos abaixo.

Realizando a *query* "flamengo":

Código 4.2: *Query* do termo "flamengo" sem índice invertido

```
SELECT COUNT(*)
    FROM tweets
    WHERE text LIKE '%flamengo%'
LIMIT 100;
```

Código 4.3: *Query* do termo "flamengo" utilizando GiN

```
SELECT COUNT(*)
        FROM tweets
        WHERE fulltext @@ phraseto_tsquery('flamengo')
LIMIT 100;
```

Código 4.4: *Query* do termo “flamengo” utilizando GiST

```
SELECT COUNT(*)
        FROM tweets
        WHERE to_tsvector('portuguese', text) @@ to_tsquery('
            ↪ flamengo')
LIMIT 100;
```

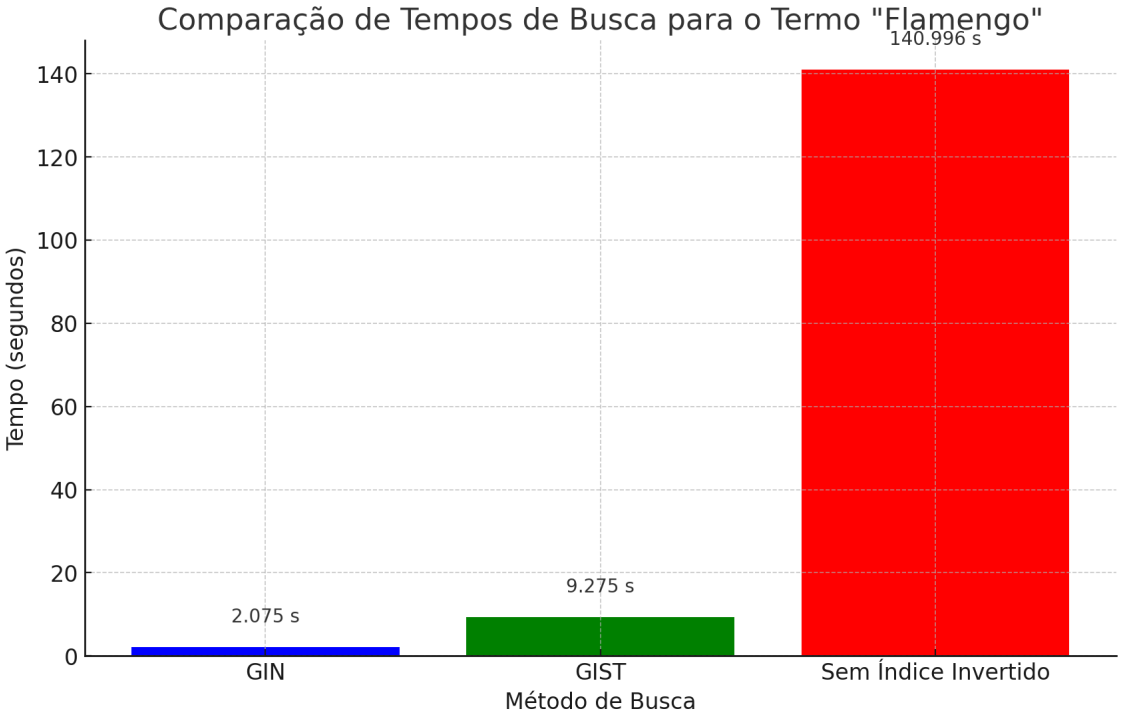

Figura 4.8: Gráfico comparativo de tempo relacionado ao termo “flamengo”.

A consulta utilizando o termo “flamengo” foi realizada sem a aplicação de índices invertidos e com os índices GiN e GiST. Conforme mostrado na Figura 4.8, a *query* sem índices levou cerca de 2 minutos, 20 segundos e 996 milissegundos. Utilizando GiN a busca completa levou 2,075 segundos, enquanto a mesma busca demorou 9,275 segundos com GiST.

Desse modo, ao comparar a consulta sem índice com a consulta utilizando GiN, a query com o índice GiN foi aproximadamente 68 vezes mais rápida, apresentando uma superioridade de cerca de 98,53%. Da mesma maneira, os resultados indicam

que a consulta com GiN apresentou um tempo de resposta 4,47 vezes mais rápido em comparação ao GiST, cerca de 77,63% de superioridade.

Realizando a *query* "trembolona":

Código 4.5: *Query* do termo "trembolona" sem índice invertido

```
SELECT COUNT(*)
    FROM tweets
    WHERE text LIKE '%trembolona%'
LIMIT 100;
```

Código 4.6: *Query* do termo "trembolona" utilizando GiN

```
SELECT COUNT(*)
        FROM tweets
        WHERE fulltext @@ phraseto_tsquery('trembolona')
LIMIT 100;
```

Código 4.7: *Query* do termo "trembolona" utilizando GiST

```
SELECT COUNT(*)
        FROM tweets
        WHERE to_tsvector('portuguese', text) @@ to_tsquery('
            ↪ trembolona')
LIMIT 100;
```

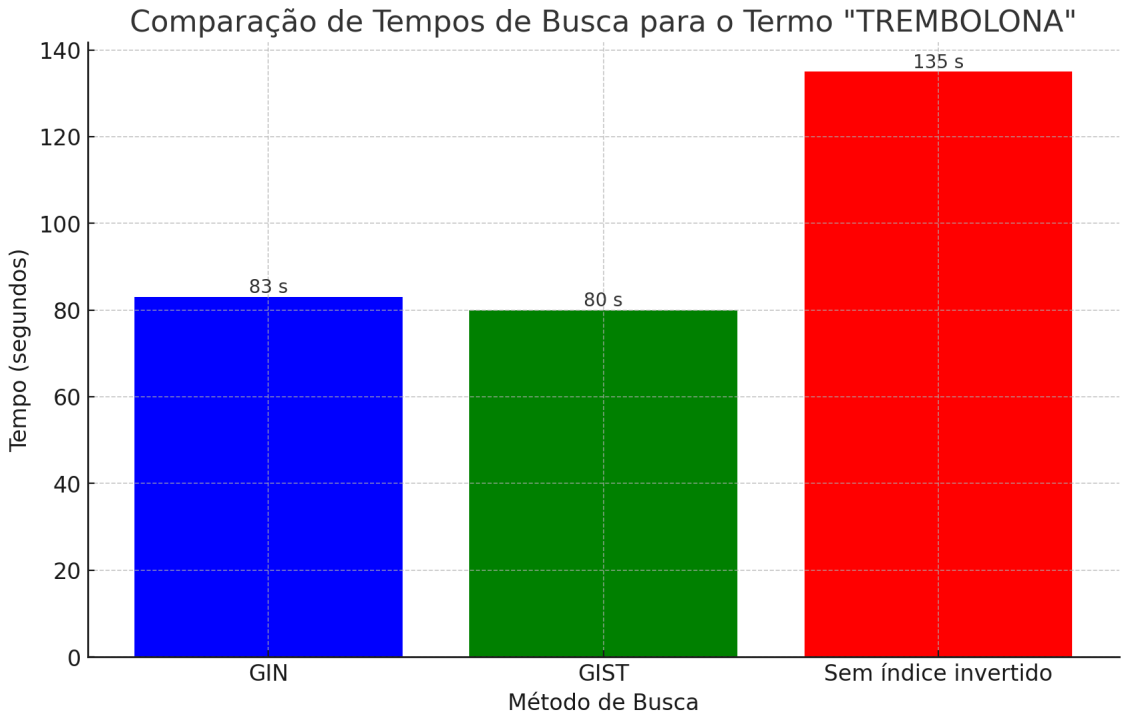

Figura 4.9: Gráfico comparativo de tempo relacionado ao termo “trembolona”.

Seguindo a comparação do termo “flamengo” foi feito com o termo “trembolona”. Assim, na Figura 4.9, sem utilizar índice, a *query* levou aproximadamente 2 minutos, 15 segundos e 957 milissegundos. Da mesma maneira, usando GiN, a busca completa levou um minuto, 20 segundos e 252 milissegundos, e com o GiST, mostra que demorou um minuto, 23 segundos e 121 milissegundos.

Diante dos resultados, ao comparar a consulta sem índice com a consulta utilizando GiN, a query com o índice GiN foi aproximadamente 1,69 vezes mais rápida, apresentando uma superioridade de cerca de 40,99%. Os resultados ainda indicam que a consulta com GiN apresentou um tempo de resposta 1,0358 vezes mais rápido em comparação ao GiST, cerca de 3,45% de superioridade.

Realizando a *query* “durateston”:

Código 4.8: *Query* do termo “durateston” sem índice invertido

```
SELECT COUNT(*)
    FROM tweets
    WHERE text LIKE '%durateston%'
LIMIT 100;
```

Código 4.9: *Query* do termo “durateston” com GiN

```
SELECT COUNT(*)
        FROM tweets
        WHERE fulltext @@ phraseto_tsquery('durateston')
LIMIT 100;
```

Código 4.10: *Query* do termo “durateston” com GiST

```
SELECT COUNT(*)
        FROM tweets
        WHERE to_tsvector('portuguese', text) @@ to_tsquery('
           ↪ durateston')
LIMIT 100;
```

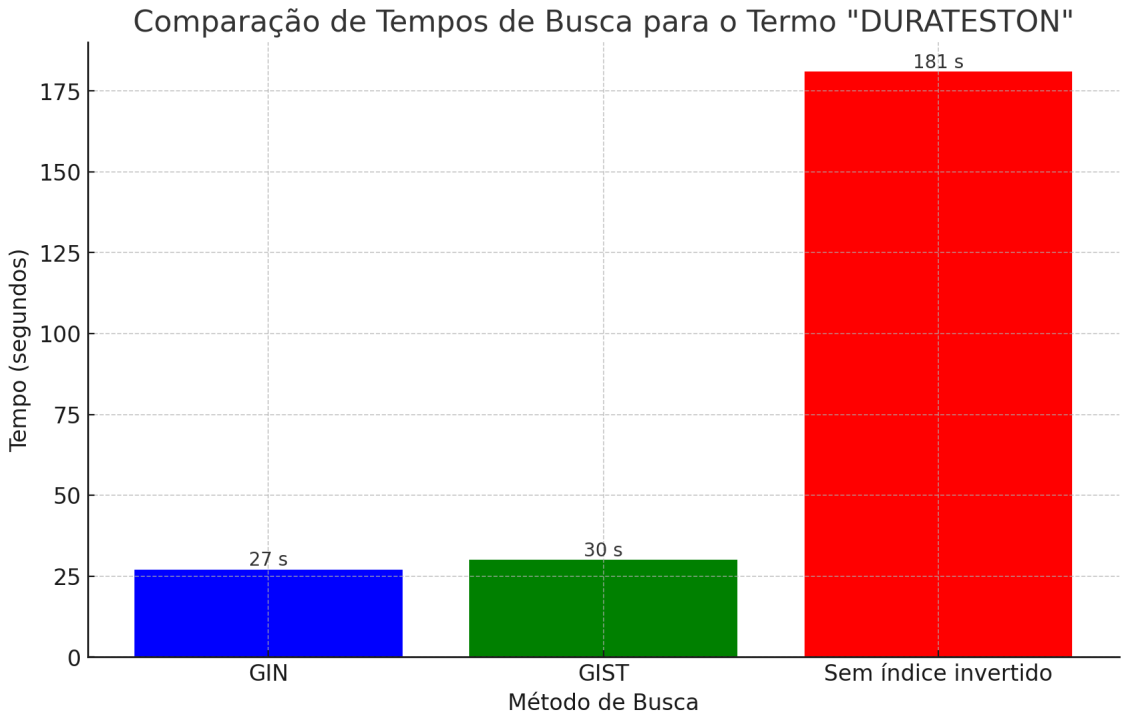

Figura 4.10: Gráfico comparativo de tempo relacionado ao termo “durateston”.

Seguindo o padrão de comparação, foi feito a busca da palavra “durateston”. Na Figura 4.10, sem utilizar índice, a *query* levou aproximadamente 3 minutos, 1 segundo e 331 milissegundos. Ao utilizar o índice GiN, a busca completa levou 27,740 segundos. Já com o GiST, demonstra o tempo de 30,235 segundos.

Ao analisar os resultados, a consulta com o índice GiN foi aproximadamente 6,54 vezes mais rápida, e teve uma superioridade de cerca de 84,7% em relação a consulta sem índice. Além disso, os resultados indicam que a consulta com GiN apresentou um tempo de resposta 1,090 vezes mais rápido em comparação ao GiST, cerca de 8.25% de superioridade.

Realizando a *query* “lebron”:

Código 4.11: *Query* do termo “lebron” sem índice invertido

```
SELECT COUNT(*)
    FROM tweets
    WHERE text LIKE '%lebron%'
LIMIT 100;
```

Código 4.12: *Query* do termo “lebron” utilizando GiN

```
SELECT COUNT(*)
        FROM tweets
        WHERE fulltext @@ phraseto_tsquery('lebron')
LIMIT 100;
```

Código 4.13: *Query* do termo “lebron” utilizando GiST

```
SELECT COUNT(*)
        FROM tweets
        WHERE to_tsvector('portuguese', text) @@ to_tsquery('lebron
            ↪ ')
LIMIT 100;
```

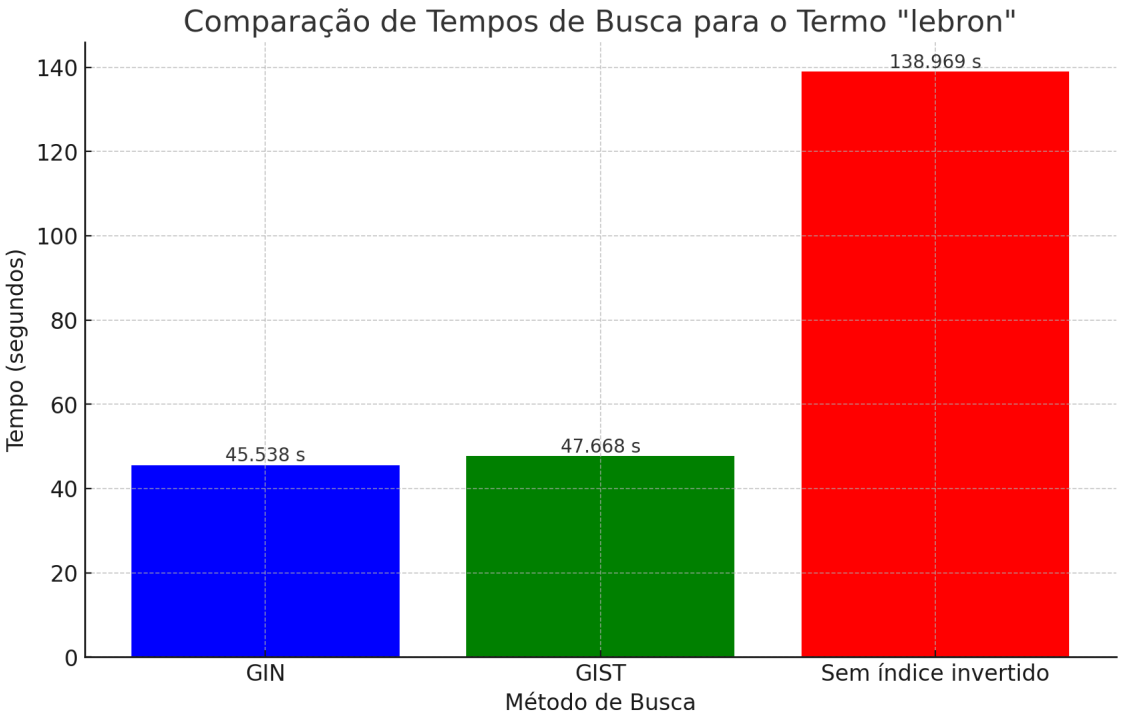

Figura 4.11: Gráfico comparativo de tempo relacionado ao termo “lebron”.

Por fim, há a comparação de desempenho de busca com o termo “lebron”. A Figura 4.11 mostra que a busca sem índice invertido leva cerca de 2 minutos, 18 segundos e 969 milissegundos para ser concluída. Enquanto utilizando o índice GiN, um tempo busca completa de 45,538 segundos, e com o GiST demandou aproximadamente o tempo de 47,668 segundos.

Desse modo, os resultados indicam que a consulta com GiN apresentou um tempo de resposta 3,05 vezes mais rápido que a consulta sem índice, obtendo uma superioridade de cerca de 67,23% Da mesma maneira, os resultados indicam que a consulta com GiN apresentou um tempo de resposta 1,047 vezes mais rápido em comparação ao GiST, cerca de 4,47% de superioridade.

Mediante o exposto, é notório a superioridade do GiN em relação ao GiST em determinados contextos, principalmente na primeira consulta. Esta diferença pode ser atribuída ao fato de que na primeira consulta, o índice GiST percorre toda a árvore para encontrar os dados correspondentes. Este processo envolve a leitura dos

nós da árvore, começando pela raiz e descendo até as folhas, comparando os valores conforme necessário.

Em contraste, na primeira consulta, o GiN indexa termos individuais e suas posições dentro dos documentos. Ele cria um índice invertido, onde cada termo aponta para os documentos que contêm esse termo. Isso permite que o GiN responda rapidamente às consultas, pois ele pode acessar diretamente os documentos relevantes através dos termos indexados.

Para consultas subsequentes, o GiST não necessariamente percorre toda a árvore novamente da mesma forma que na primeira consulta. Ele pode se beneficiar de várias técnicas de otimização, como o cache de páginas de índice em memória, que evita leituras repetidas do disco. Dessa forma, após a primeira consulta, o tempo de busca se assemelha com o tempo de busca do GiN. Entretanto, ainda de forma mais ineficiente.

Esta diferença é atribuída à capacidade do GiN de indexar elementos textuais individuais e ser mais particular ao *Full Text Search*. Nas consultas subsequentes, ele usa o índice invertido para encontrar os documentos que contêm os termos da consulta sem precisar reindexar ou percorrer toda a árvore novamente. Enquanto o GiST, ocm sua estrutura de árvore balanceada, requer mais tempo para processar consultas devido à necessidade de realizar uma busca mais abrangente na árvore, além de ser mais indicado para buscas complexas que envolvem atributos de similaridade ou proximidade.

No contexto geral, os resultados dos experimentos demonstraram uma melhoria significativa no tempo de resposta ao utilizar índices invertidos. Consultas realizadas sem o uso de índices apresentaram tempos de resposta consideravelmente mais altos, especialmente à medida que o volume de dados aumenta.

4.3 Limitações Encontradas

Em virtude de ser um grande sistema, diversas complexidades e desafios surgem, especialmente quando o objetivo é capturar, analisar e interpretar dados em grande

escala. Neste contexto, duas limitações críticas foram identificadas: a geração de IDs e limites de busca de informações (*Rate Limits*) impostos pela API do *Twitter*. Cada uma dessas limitações impõe restrições significativas ao sistema proposto, afetando diretamente a eficácia na obtenção e confiabilidade dos dados.

4.3.1 Geração de IDs

Na arquitetura de aplicações tradicionais, especificamente nas operações CRUD, um novo recurso é criado e armazenado em um banco de dados singular, sendo identificado por um identificador único. Este modelo, embora eficaz em estruturas unificadas, apresenta desafios significativos quando aplicado a sistemas distribuídos. Estes, exemplificados pela infraestrutura do *Twitter*, não operam sob uma única aplicação, mas sim por meio de várias aplicações trabalhando simultaneamente.

Neste cenário, é comum a existência de múltiplos data centers, potencialmente mais de um por país, cada qual hospedando *workers* que realizam diferentes serviços. Desta maneira, implica uma complexidade adicional no gerenciamento de identificadores únicos em várias instâncias. A separação dos bancos de dados por região ou país complica ainda mais a unificação dos dados, podendo resultar em conflitos de identificação.

Outro desafio técnico significativo é a capacidade de gerar IDs únicos sob demanda, na ordem de milhões por segundo. Uma solução implementada foi o balanceador de carga, que direciona solicitações para cada data center, atribuindo identificadores únicos com base nas informações já existentes. No início da operação do Twitter, utilizava-se um sistema de identificação baseado em IDs de 32 bits. Contudo, o crescimento acelerado da plataforma mudou para um sistema de 64 bits *unsigned*, desenvolvido pelo serviço *Snowflake*, capaz de gerar IDs de forma consistente.

Entretanto, a representação de inteiros de grande precisão em *JavaScript*, limitada a 53 bits, resultou em problemas de precisão, conhecidos como “munging”. Esse fenômeno ocorria quando um ID de 64 bits era recebido, mas apenas os primeiros 53 bits eram considerados, levando à perda de precisão e, consequentemente, à geração de recursos com identificadores imprecisos.

Essa limitação era evidente quando o sistema tentava inserir um novo registro no banco de dados e era impedido pela restrição de unicidade do ID, a qual indica a tentativa de inserção de um recurso já existente. A detecção de IDs afetados por este problema revelou que eles frequentemente terminavam em "0000", indicativo do uso completo dos 64 bits disponíveis.

A representação inadequada dos inteiros em *JavaScript* era evidenciada pela conversão de um número como 14728214603941869327 para uma *string*. Por exemplo, ao fazer essa conversão o resultado seria "14728214603941870000", mostrando claramente um erro de precisão. A solução definitiva para esses problemas de precisão veio com a introdução do BigInt na versão 10.4.0 do *Node.js*. Esse *wrapper* possibilita a representação exata de inteiros usando 64 bits, resolvendo assim as questões de precisão e representação de dados em ambientes distribuídos.

4.3.2 Twitter API e Rate Limit

Todavia, para captura dos dados, a modelagem foi construída em torno da API do *Twitter* que fornece acesso às informações. Para isso, foi utilizada a biblioteca *twitter-api-v2*, principalmente por obter o *Rate Limit*, por monitorar a frequência e o tempo das requisições e evitar que qualquer cliente que obtenha acesso à API envie quantidades excessivas de requisições resultando numa indisponibilidade abrupta no funcionamento do servidor.

Durante o período de desenvolvimento do projeto, foi realizada a solicitação de uma conta de desenvolvedor junto à plataforma *Twitter*. Com base na justificativa apresentada, que detalhava o propósito acadêmico e o escopo do monitoramento de termos relacionados ao doping, foi concedida a capacidade de realizar até 450 solicitações a cada 15 minutos através da API do *Twitter*, e caso excedesse, seriam negadas até renovar novamente tal espaço de tempo.

Ainda sobre o *Rate Limits*, por limitar a quantidade de *requests* da consulta das Interfaces de Programação de Aplicação (APIs), foi utilizado o *plugin-rate-limit* para auxiliar a controlar o número de requisições com o intuito de não ultrapassar este número limite.

Assim, devido à limitação da API e ao alto tráfego de dados online, torna-se impossível monitorar completamente a rede social. Portanto, foi utilizado o padrão *singleton* a fim de garantir a exclusividade de instâncias para cada classe e controlar que não existam várias instâncias simultâneas da mesma.

Esse padrão viabiliza a existência de apenas uma instância para cada classe, assegurando assim que os recursos sejam inicializados de maneira controlada por meio de suas respectivas classes. Essa abordagem não apenas garante a inicialização única dos recursos, mas também proporciona benefícios adicionais ao projeto, como a facilitação do compartilhamento de estados entre as instâncias inicializadas.

Capítulo 5

Conclusão

Este capítulo finaliza a monografia apresentando as conclusões alcançadas ao longo do desenvolvimento do estudo e os trabalhos futuros.

5.1 Considerações finais

O presente trabalho propôs o desenvolvimento de um sistema web para o monitoramento de termos relacionados ao *doping*, utilizando métodos de melhoria de eficiência de busca por meio de índices invertidos. O objetivo principal foi criar uma ferramenta que facilitasse a identificação e análise de discursos e menções ao *doping* em plataformas de mídia social, especificamente no *Twitter*.

Ao longo do desenvolvimento do sistema, diversas tecnologias foram empregadas, como *Node.js*, *AdonisJS*, *PostgreSQL*, entre outras, que garantiram a funcionalidade, desempenho, segurança e escalabilidade necessárias para o projeto. A implementação dos índices invertidos e a utilização de técnicas de cache mostraram-se essenciais para otimizar o desempenho das buscas, tornando-as mais rápidas e precisas.

Nos experimentos, foi realizada uma comparação de desempenho entre as ferramentas GiN e GiST, com um volume significativo de dados. Os resultados demonstraram que o índice GiN proporcionou uma melhor performance em consultas com uma grande quantidade de termos, enquanto o GiST mostrou-se mais eficiente em cenários

com menos variabilidade de dados. Isso reforça a importância de escolher o tipo de índice apropriado de acordo com as características específicas do conjunto de dados e das consultas esperadas.

Os resultados obtidos demonstraram que o sistema proposto é capaz de realizar buscas eficientes em grandes volumes de dados, proporcionando uma ferramenta capaz de auxiliar pesquisadores, profissionais da área esportiva e responsáveis por políticas públicas. A análise dos dados coletados no *Twitter* possibilitou uma compreensão mais aprofundada das percepções e comportamentos relacionados ao *doping*, contribuindo para o combate a essa prática no cenário esportivo.

5.2 Trabalhos Futuros

Com base no contexto do sistema proposto e na identificação de limitações identificadas, fica evidenciado que nenhum estudo se dá por definitivo. Nesse espírito, as seguintes proposições são oferecidas como sugestões construtivas para futuros trabalhos, visando não somente a ampliação e melhorias da plataforma, mas também o enriquecimento acerca do estudo do *doping*.

Uma direção interessante para melhorias futuras envolve a expansão da integração da plataforma atual com outras redes sociais, além do Twitter. Este avanço permitiria captar uma diversidade maior de usuários, fornecendo uma visão mais abrangente sobre o estudo e o monitoramento do uso de substâncias ilícitas. A extensão para múltiplas redes sociais não apenas ampliaria o escopo da pesquisa, como também faria a ferramenta ser mais conhecida com possibilidades de obter investimentos externos.

Adicionalmente, recomenda-se a realização de estudos e inserção de substâncias proibidas existentes em diferentes países e regiões, especialmente aquelas menos conhecidas ou específicas a certos contextos culturais. Esta abordagem permitiria uma análise mais profunda das questões culturais relacionadas à proibição ou legalização do uso de determinadas substâncias. Desse modo, contribuiria não apenas para um entendimento global do fenômeno do *doping*, mas também para a compreensão em

como as nuances sociais e culturais influenciam no uso de entorpecentes ao redor do mundo.

Ademais, recomenda-se a implantar a funcionalidade de exportação de relatórios de forma automatizada, utilizando técnicas de Inteligência Artificial (IA). Essa abordagem permitiria a geração dinâmica de relatórios detalhados e personalizados, baseados nas análises de dados coletados pelo sistema. A automação com IA poderia otimizar o processo de extração e organização das informações relevantes, proporcionando *insights* importantes de maneira mais eficiente e eficaz.

Por fim, sugere-se a implementação de um mecanismo de mapas, que permita aos usuários ampliar e examinar locais específicos na busca de termos e assim oferecer *insights* mais detalhados sobre o uso de drogas em diferentes regiões. Ao prover uma perspectiva macro sobre o tema, tal mecanismo enriqueceria significativamente a compreensão dos padrões de uso de substâncias proibidas, bem como das políticas públicas e intervenções necessárias para combater o *doping* de maneira eficaz.

Referências

AGENCY, W. A.-D. World anti-doping code 2021. WORLD ANTI-DOPING AGENCY, p. 21–28, 2021.

BARBOSA, R.; MATOS, P. M.; COSTA, M. E. A glance into the body: Yesterday's and today's body| um olhar sobre o corpo: O corpo ontem e hoje. 2011.

BELING, F. 10 perfis mais seguidos do instagram no mundo. Oficina da Net, 2023. Disponível em: <https://www.oficinadanet.com.br/post/19182-10-perfis-mais-seguidos-no-instagram-no-mundo>. Acesso em: 15/04/2023.

BORGES, J. V. et al. Esteroides anabolizantes: uma análise documental sobre o uso dessas substâncias por atletas profissionais e amadores. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, v. 7, n. 8, p. 501–522, 2021.

CARTIGNY, E. et al. Typologies of dual career in sport: A cluster analysis of identity and self-efficacy. *Journal of Sports Sciences*, Taylor & Francis, p. 1–11, 2020. Disponível em: <https://doi.org/10.1080/02640414.2020.1835238>.

COMMITTEE, I. O. Olympic values. 2021 International Olympic Committee, 2021. Disponível em: <https://olympics.com/ioc/olympic-values>. Acesso em: 13/07/2023.

FRANCK, K. M. et al. Diagrama entidade-relacionamento: uma ferramenta para modelagem de dados conceituais em engenharia de software. *Research, Society and Development*, v. 10, n. 8, p. 1–12, 2021.

GROUP, T. P. G. D. *PostgreSQL 16.2 Documentation*. 2024. Disponível em: <https://www.postgresql.org/docs/current/index.html>. Acesso em: 04 mar 2024.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. 3rd. ed. [S.l.]: Elsevier, 2011. ISBN 9780123814791.

HOSTMíDIA. *O que são cron jobs? Vantagens e desvantagens*. 2021. Disponível em: <https://www.hostmidia.com.br/blog/o-que-sao-cron-jobs/>. Acesso em: 17 jan 2024.

IBM. *O que é modelagem de dados?* 2020. Disponível em: <https://www.ibm.com/br-pt/topics/data-modeling>. Acesso em: 03 jan 2024.

JO, T. *Text Mining: Concepts, Implementation, and Big Data Challenge*. Cham, Switzerland: Springer International Publishing, 2019. v. 45. (Studies in Big Data, v. 45). ISBN 978-3-319-91814-3. Disponível em: <https://doi.org/10.1007/978-3-319-91815-0>. Acesso em: 12 abr 2024.

JORDAN, S. E. et al. Using twitter for public health surveillance from monitoring and prediction to public response. *Data*, MDPI, v. 4, n. 1, p. 1–20, 2018.

KEMP, S. Digital 2022: Brazil. DataReportal – Global Digital Insights, 2022. Disponível em: <https://datareportal.com/reports/digital-2022-brazil>. Acesso em: 25 jan 2024.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. 1. ed. Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo: Cambridge University Press, 2008. 482 p. ISBN 978-0-521-86571-5.

MASALA, D. et al. Enjoy the Sport - Schools against doping and drug dependence: a health education intervention in secondary schools. *Igiene e Sanità Pubblica*, v. 75, p. 271–282, 2019.

PIONTKOSKI, G. Comparativo de desempenho de consultas sql entre banco de dados em memória ram e em ssd. Universidade Tecnológica Federal do Paraná, 2017.

SRIKANTH, M. et al. Dynamic social media monitoring for fast-evolving online discussions. KDD '21: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining, p. 3576—-3584, 2021. Disponível em: <https://doi.org/10.1145/3447548.3467171>.

STAVRAKANTONAKIS, I. et al. An approach for evaluation of social media monitoring tools. In: *Common Value Management.: 1st International Workshop Common Value Management CVM 2012*. [S.l.]: Fraunhofer Verlag, 2012. p. 52–64.

STEFANO, N. D. The idea of beauty and its biases: Critical notes on the aesthetics of plastic surgery. *Plastic and Reconstructive Surgery-Global Open*, Lippincott Williams Wilkins, v. 5, n. 10, p. e1523, 2017. Disponível em: <https://journals.lww.com/prsgo/Fulltext/2017/10000/The_Idea_of_Beauty_and_Its_Biases__Critical_Notes.11.aspx>.

STRATE, J. *Expert Performance Indexing in SQL Server 2019: Toward Faster Results and Lower Maintenance*. Hugo, MN, USA: Apress, 2019. ISBN 978-1-4842-5463-9.

THELWALL, M. Social web text analytics with mozdeh. *Mozdeh*, p. 1–35, 2018.

VINHAS, L. Fundamentos de bancos de dados. *Earth System Science Centre*, 2016.

YESALIS, C. E.; BAHRKE, M. S. History of doping in sport. International Sports Studies, v. 24, p. 42–76, 2002. Disponível em: <https://www.doping.nl/media/kb/6495/Yesalis%20et%20al%202002.pdf>.

ZIA, A. et al. Artificial intelligence-based medical data mining. *Journal of Personalized Medicine*, MDPI, v. 12, n. 9, p. 1359, 2022.